**Deep learning-based Food Recipe Generator**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**Tata Lakshmi Durga Likhitha-AP19110010001**

**Valavala Mahesh-AP19110010388**

**Koliparthi Durgavamsi-AP19110010392**

**Godavarti Venkata Narasamma-AP19110010421**

**A picture containing text

Description automatically generated**

Under the guidance of

**Dr. Pandu Sowkuntla**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**May-2023**

**Certificate**

Date: 2-Jun-23

This is to certify that the work present in this Project entitled “**Deep learning-based Food Recipe Generator**” has been carried out by **Tata Lakshmi Durga Likhitha, Valavala Mahesh, Koliparthi Durgavamsi, Godavarti Venkata Narasamma** under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in **School of Engineering and Sciences**.

**Supervisor**

(Signature)

Dr. Pandu Sowkuntla

Assistant Professor,

Department of Computer Science Engineering.

# Acknowledgements

We are using this opportunity to express our gratitude to everyone who supported us throughout the successful completion of the project. We are thankful for their aspiring guidance, invaluably constructive criticism, and friendly advice during the project work. We are sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

We are grateful to my supervisor, Dr. **Pandu Sowkuntla**, for their exceptional guidance and mentorship, which have been invaluable in shaping the direction and success of this project.

We want to express our sincere gratitude to the head of the department **Dr. Jatindra** **Kumar Dash** and the course coordinator **Dr. Sambit Kumar Mishra** to facilitate an atmosphere of academic excellence and for providing the resources and support we needed for our final project.

We would also like to thank all the faculty, and staff at SRM University–AP for continuously encouraging me in exploring my best.

We want to express our gratitude for the assistance we received from friends and on this journey. Their encouragement, discussions, and brainstorming sessions have been incredibly beneficial in shaping our ideas and keeping us motivated during challenging times.

Your Sincerely,

Tata Lakshmi Durga Likhitha AP19110010001

Valavala Mahesh AP19110010388

Koliparthi Durgavamsi AP19110010392

Godavarti Venkata Narasamma AP19110010421

# Table of Contents

[Certificate i](#_Toc105799088)

[Acknowledgements ii](#_Toc105799089)

[Table of Contents i](#_Toc105799090)ii

[Abstract iv](#_Toc105799091)

[Abbreviations v](#_Toc105799095)

[List of Tables vi](#_Toc105799097)

[List of Figures vii](#_Toc105799097)

[List of Equations viii](#_Toc105799098)

[1. Introduction 1](#_Toc105799099)

[2. Preliminaries 2](#_Toc105799102)

[3. Related works 4](#_Toc105799106)

[3.1 Food101 mining discriminative components with RF](#_Toc105799104) 4

3.[2 Deep based ingredient recognition 4](#_Toc105799104)

[3.3 Chinese food net 5](#_Toc105799104)

[4. Existing work 6](#_Toc105799107)

[5. Proposed work 8](#_Toc105799108)

[5.1 Motivation 8](#_Toc105799104)

5.[2 Proposed recipe generator 9](#_Toc105799104)

[5.2.1 Activation Functions 12](#_Toc105799105)

5.2.[2 CNN 13](#_Toc105799105)

[5.2.3 AlexNet 14](#_Toc105799105)

[5.2.4 VGG 14](#_Toc105799105)

[5.3 Limitaitons for proposed work 14](#_Toc105799104)

[6. Experimental Analysis 15](#_Toc105799099)

6.[1 Experimental Setup 15](#_Toc105799104)

6.[2 Dataset Description 16](#_Toc105799104)

6.[3 Classification Results 17](#_Toc105799104)

6.[4 Results of Ingredients recognition and recipe generation 21](#_Toc105799104)

[7. Conclusion 25](#_Toc105799099)

[8. References 26](#_Toc105799099)

**Abstract**

Food identification is significant in choosing food and consumption, which is necessary for human well-being and health. It is therefore essential to computer vision and can also help with various food-related vision tasks, such as food categorization, recipe retrieval, and development. A common food classification job is to predict food names based on associated food images. New advances in deep learning improve food classification ability. Recipe generation is a more difficult challenge than food Classification and ingredient recognition since the evaluation of nutritious food depends not only on components but also on the size, shape, and colour of food owing to varied cooking methods.

Surprisingly, not much research work has been done on recipe generation. As a result of this research, we provide a deep learning-based recipe generation model that creates cooking instructions from food images. Our system predicts ingredients using a architecture which we had proposed as goes down, Modelling their relationships without enforcing any order, followed by the creation of culinary instructions while paying attention to the image and its predicted components. Extensive experimental investigation on various food photos was performed to evaluate the performance of the suggested model.

 On the large-scale Recipe1M dataset, a well-known dataset in the field of recipe instructions generation, we thoroughly evaluate the entire system and demonstrated that high-quality recipes have been generated by combining images, ingredients, and instructions. From the results, we observed that our system can produce recipes that, in the opinion of a human, are more compelling than those produced by retrieval-based methods.

# Abbreviations

CNN Convolutional Neural Network

RNN Recurrent Neural Network

LSTM Long Short-Term Memory

RF Random Forests

ResNet Residual Neural Networks

VGG Visual Geometry Group

Conv2d Convolution matrix with two dimensions

SVM Support vector machine

DFD Data flow diagram

ReLU Rectified Linear Activation function

pkl Pickle files

ckpt Check points

CSV Comma separated values

URL Uniform Resource Locator.

# List of Tables

Table 1. Hardware Requirements [15](#_Toc105799088)

Table 2. Software Requirements [15](#_Toc105799088)

# List of Figures

Figure 1. Data Flow Diagram [9](#_Toc105799088)

Figure 2. System Architecture [13](#_Toc105799088)

Figure 3. Comparison between Classification models [17](#_Toc105799088)

Figure 4. Plot for CNN [18](#_Toc105799088)

Figure 5. Plot for Alex [19](#_Toc105799088)

Figure 6. Plot for VGG [20](#_Toc105799088)

Figure 7. Homepage [21](#_Toc105799088)

Figure 8. Upload a food image [21](#_Toc105799088)

Figure 9. Uploaded a food image [22](#_Toc105799088)

Figure 10. Recipe 1 Output 1 [22](#_Toc105799088)

Figure 11. Recipe 1 Output 2 [23](#_Toc105799088)

Figure 12. Upload Image for model 2 [23](#_Toc105799088)

Figure 13. Recipe 2 Output [24](#_Toc105799088)

# List of Equations

[Equation 1 Activation Function for ReLu 12](#_Toc105799088)

[Equation 2 Loss Function for SoftMax 13](#_Toc105799088)

# Introduction

Food is not just subsistence; it shapes our culture and influences our well-being. The old adage goes, 'We are what we eat,' meaning that food directly affects our overall well-being. In today's digital age, social media is flooded with food-related posts, highlighting its significance in our society.

After the pandemic situation, many people are working from home. They often come across tempting food posts on social media but cannot go out to buy them. Consequently, they have started cooking these dishes at home. However, not everyone has the necessary skills and knowledge to prepare recipes at home, and guidance may not always be available. As a result, we believe there is a demand for food recipe generator systems that can deduce ingredients and cooking instructions from a prepared meal.

However, food recognition poses challenges due to its variability and complex composition. Overcoming these obstacles requires advanced computer vision and prior knowledge. Previous efforts have focused on categorization, but deeper analysis is needed to provide accurate food preparation descriptions.

# Preliminaries

Our proposed model has been implemented on CNN. In this section we discussed the fundamental concepts that used to understand the proposed model. Each CNN model that is used to train our proposed model has been discussed below.

**Autoencoders CNN (Keras and Torch)**

Autoencoders are artificial neural networks that can compress and recreate data. The encoder compresses the input, while the decoder attempts to reconstruct it from the compressed version. An autoencoder learns data grouping. In contrast, CNN refers to a sort of neural network that extracts features from data using the convolution operator (typically the 2D convolution when used for image processing applications).

**MiniVGG**

Mini VGG is a smaller version of the popular VGG (Visual Geometry Group) convolutional neural network architecture. It is CNN with 16 layers. VGG16 has a sequence of smaller filters. It has a tiny 3x3 receptive field with a 1-pixel stride. When using multiple smaller layers instead of smaller largelayer, the decision functions are improved and the network can converge more rapidly. This is because more non-linear activation layers are present. ReLu activation function is used with a convolutional stride of 1 pixel. In the pooling layers, the filters are used from 64 to 256 and 512 in the final layers. Due to the small convolutional filter of VGG, over-fitting doesn’t happen in the training phase. VGG is one of the best models to understand the spatial features of the images. resource-constrained situations.

**MiniGoogleNet**

Convolutional neural networks, such as MiniGoogleNet, a variation on the GoogleNet architecture, are small and effective for computer vision tasks. It is made up of a number of inception modules, which serve as the network's building elements. The parallel convolutional layers of various sizes, including 1x1, 3x3, and 5x5, included in these inception modules enable the network to record both local and global information. The network also incorporates auxiliary classifiers to promote intermediate feature learning and spatial downsampling, as well as max pooling and average pooling layers. With minimal computational complexity, this architecture has been shown to be useful in achieving high accuracy in image categorization and object identification applications.

**MiniAlexNet**

The original AlexNet architecture, a groundbreaking convolutional neural network for image classification tasks, has a more condensed version known as Mini AlexNet. Mini AlexNet attempts to keep the model's performance while reducing its size and computational complexity. In comparison to the original AlexNet, it accomplishes this by using fewer layers and filters. Mini AlexNet still features stacked convolutional layers, max pooling, and fully linked layers like its predecessor, while being smaller in scope. In situations where there are limited processing resources, Mini AlexNet can effectively extract features from photos and achieve competitive classification accuracy.

**CNN with 2 Fully Connected Layers**

The convolutional neural network (CNN) is a popular deep learning model for image categorization. Convolutional layers are followed by completely linked layers in this system. While fully connected layers link every neuron from the previous layer to the next, convolutional layers use filters to extract local characteristics from input images. The fully connected layers extract high-level features and convert them to class labels, enabling the network to forecast using the features that have been previously learned. These layers, which are usually at the conclusion of the CNN design, are essential for converting the extracted features into class probabilities or regression results.

**ResNet**

Residual Neural Network (ResNet) is a CNN model with 50 layers. This network needs an input image of size 224\*224. There are 64 different kernels with a kernel size of 7\*7 in a convolution and with a stride of size 2. Then there is a max pooling layer with stride size of 2. In the next layers, kernel dimension increases. Then comes average pooling and finally the softmax function in the final layer is used.

# Related works

From the Literature we observed that different approaches related to recipe generation have been proposed like recipe categorization, ingredient recognition etc. A brief description of all these studies has been discussed in this section.

## 3.1 Food-101–mining discriminative components with random forests.

The Authors Bossard et al. [1] explored the challenge of automatically recognizing food items in their study they suggested a new Random Forests-based technique to mine discriminative components and simultaneously transfer data between classes. To improve mining and representation capabilities, they evaluated patches aligned with image superpixels. The authors also introduced a comprehensive dataset with 101 food categories and 101 '000 images to test their rf component mining approach for food identification. Their model achieved an average accuracy of 50.76%, outperforming previous component-based classification algorithms on the challenging mit-Indoor dataset and other classification techniques except for CNN. Specifically, their solution improved accuracy by 11.88% and 8.13% compared to Improved Fisher Vectors and SVM classification, respectively.

## 3.2 Deep-based ingredient recognition for cooking recipe retrieval.

The Authors Jing-Jing Chen et al. [3] explored the problem of recognizing ingredients in food images and its relevance to cooking recipe retrieval in their study the authors argued that obtaining recipes that match the given food pictures can improve the computation of nutrition information, which is essential for various health-related applications. They highlighted the primary focus of current methods on classifying food categories based on the appearance of global dishes, with little attention paid to the composition of the ingredients.

The authors also noted the potential issue of zero-shot retrieval, which arises when retrieving recipes that contain unidentified dietary groups, rendering the current approach inadequate. In addition, they pointed out that achieving adequate performance in content-based retrieval without knowledge of food categories is equally challenging due to significant visual differences in food appearance and component composition. Since the number of ingredients is significantly lower than the number of food categories, the authors proposed that ingredient identification is ideal for zero-shot retrieval because it is more scalable than distinguishing each food category. However, they acknowledged that ingredient identification is significantly more challenging than food classification, raising questions about their use for retrieval.

The authors presented deep architectures for concurrent constituent identification and food classification learning to tackle these challenges, utilizing their mutual but ambiguous relationship. They applied the learned deep features and ingredient semantic labels cleverly to zero-shot recipe retrieval, demonstrating the practicality of their approach and illuminating the zero-shot issue in recovering culinary recipes. The authors evaluated their method on a large dataset of highly complex dishes photos and provided insights into fixing recognizable proof.

# 3.3 Chinesefoodnet: A large-scale image dataset for Chinese food recognition.

The Authors Xin Chen et al. [6] present in this study their work on "ChineseFoodNet," a large-scale food image dataset that enables automatic recognition of Chinese cuisine images. This novel and the challenging dataset include approximately 180,000 images of Chinese food divided into 208 categories, containing numerous examples of the same dish. The authors explain how they chose the culinary categories, obtained and cleaned the data, and labeled it for machine-learning purposes to make human labeling less expensive and time-consuming.

The authors gave a thorough analysis of different state-of-the-art deep convolutional neural networks (CNNs) using the ChineseFoodNet dataset in their paper. They also introduce "TastyNet," a novel two-step data fusion method that combines CNN prediction results with a voting procedure. The proposed approach accomplishes top-1 accuracy rates of 81.43 and 81.55 percent on the validation and test sets, respectively. This study contributes to the advancement of computer vision and machine learning techniques in the field of food image recognition and classification, specifically for Chinese cuisine.

# Existing work

Previous attempts at food comprehension have primarily focused on food and ingredient classification. On the other hand, a comprehensive visual food identification system ought to be able to differentiate not only the components that make it up but also how it was prepared. An embedding space-based image similarity score is used to obtain a recipe from a given dataset the image-to-recipe problem is also known as a retrieval problem. The quantity and variety of the dataset, in addition to the learned embedding's quality, are crucial factors in determining these systems' success. When there is no recipe in the static data that corresponds to the image query, it should not come as a surprise that these systems fail.

The "Inverse Cooking: Recipe Generation from Food Images"[11] paper introduces a novel approach that converts food images into corresponding cooking instructions using deep learning techniques. It leverages convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to capture visual features and generate coherent recipes.

"DeepRecipes: Investigating Massive Online Recipe Collections and Recovering Food Ingredient Amounts"[12] focuses on recovering accurate ingredient amounts from unstructured text data and provides valuable insights into ingredient recognition and quantity estimation.

"Learning Recipe Generation and Food Retrieval Structural Representations" [13] presents a deep learning-based approach that combines text-based representations with graph neural networks to capture the hierarchical relationships between ingredients and cooking steps.

"Sequential Learning for Image-Based Ingredient Recognition" [14] proposes a sequential learning framework for ingredient recognition in food images. To recognise ingredients in complex recipes, it employs convolutional neural networks (CNNs) and long short-term memory (LSTM) networks.

"Large Scale Visual Food Recognition"[15] proposes a deep learning-based approach for accurately classifying a wide range of food items from images. It utilizes convolutional neural networks (CNNs) and hierarchical fine-tuning to improve recognition performance.

From the aforementioned approaches we observed that all the approaches use deep learning models to either recognize ingredients or generate recipes but one of the drawbacks we found is that a comprehensive visual food identification system, on the other hand, should be able to distinguish not only the kind of dish or its components but also how the food was prepared.

# Proposed work

This section discusses the proposed deep learning-based recipe generation model. The motivation to propose the proposed model and its advantages and limitations are elaborated in this section.

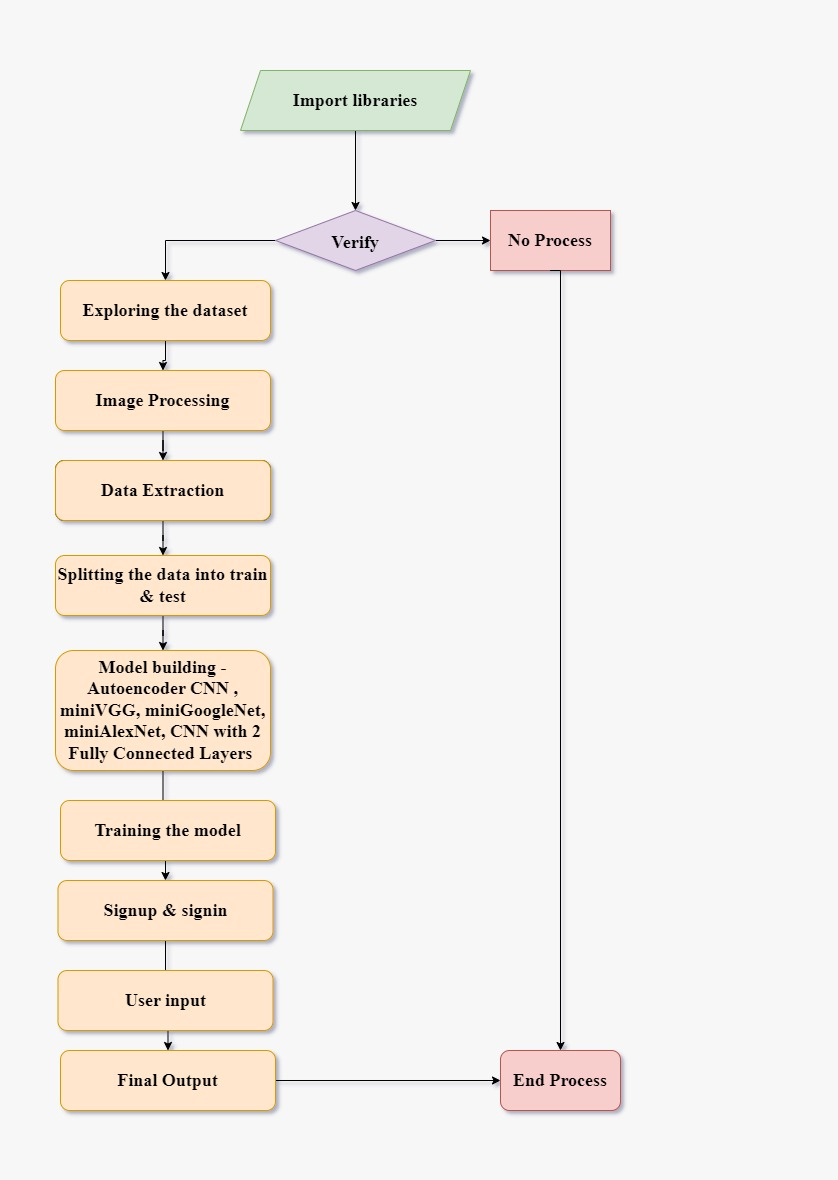
## 5.1 Motivation

In today's generation, it has become increasingly challenging to access and prepare traditional and healthy recipes. Even previous generations often lack knowledge of older recipes. The COVID-19 pandemic further highlighted the difficulty of obtaining food from outside sources, leading to a greater need for home cooking. However, many individuals struggle with the process of preparing meals due to a lack of cooking skills and knowledge of recipes. To address these challenges, we proposed a solution that leverages algorithms of machine learning to generate food recipes from food images. By allowing users to input images of the desired dish, our system can analyze the visual information and generate accurate recipes corresponding to the image. This innovative approach aims to simplify the cooking process and provide individuals with the necessary instructions to prepare a variety of dishes.

Computer vision has made great progress in recent years in tasks such as image classification and object detection. However, recognizing and understanding food presents unique challenges because of the complexity and variety of ingredients, as well as the deformations that occur during cooking. Cooked dishes further complicate the task with obscured ingredients and variations in color, shape, and texture. To reliably identify components, significant reasoning and prior knowledge are required for visual ingredient detection. To address these challenges, researchers are exploring innovative approaches that combine computer vision algorithms, deep learning models, and extensive food-related datasets. The goal is to develop robust food recipe generators that can not only identify ingredients but also provide detailed instructions for preparing the dish. This multidisciplinary approach shows great potential in making diverse and high-quality recipes more accessible while considering the intricacies of food preparation.

### 5.2 Proposed recipe generator

To solve the picture-to-recipe problem, a recipe is typically derived from a fixed dataset using the image similarity score in an embedding space. The number and diversity of the datasets, as well as the quality of the learned embedding, all have a substantial impact on the success of these systems. It should come as no surprise that these systems fail if the static dataset lacks a recipe that corresponds to the image query. To address retrieval systems' dataset limits, the image-to-recipe problem could be recast as a conditional generation problem. We employed two methods in this project. One is a pre-trained dataset, and the other one is we have trained the dataset using different models. We, therefore, developed a method in this work that converts an image into a instructions food recipe that generates title, ingredients, and cooking instructions.



**Figure 1: Data Flow Diagram**

The algorithm for the proposed work has been given below.

**Algorithm: Recipe Generator**

1. Start
2. Import the necessary libraries, and modules and initialize the **Flask** application.
3. Create an empty list **recipe\_list** to store the data and load the **core-data\_recipe.csv** dataset.
4. For each element i in the list of elements:
   1. get all values by using **.\_get\_value(i)** method
   2. define an object and call the **Recipe** file where in recipe file defines a class with methods to get and set various attributes of a recipe, such as a **recipe ID**, **name**, **ingredients**, and **cooking instructions**.
   3. Appending all the values to the list
      1. **recipe\_list.append(obj)**
5. Define the allowed file extensions as **set(['png', 'jpg', 'jpeg'])**  and a function to check if the uploaded file has a valid extension.
6. Implement a function to build CNN. If the model and weights files exist, load them; otherwise, build the model using the dataset.
   1. If **1model.json** exist:
      1. Read the model architecture from the JSON file using **json\_file.read()** and load the model weights using **load\_weights**.
      2. Load training history data from the pickle file using **pickle.load()**  and calculate **accuracy** and display the completion message with accuracy.

b. If **1model.json** doesn’t exist:

1. Load training data **X.txt.npy** and **Y.txt.npy** which are numpy files.
2. Reshape the data to **64\*64\*3**
3. Define input and output layers for the autoencoder model. and build the autoencoder, encoder, and decoder models.
4. Compile the autoencoder model and fit the autoencoder model to the training data.
5. Implement a function to predict the recipe from an uploaded image and that function saves the uploaded image.
   1. The uploaded image is read using the **cv2.imread** function and resized to (64, 64) using **cv2.resize**.
   2. The image which is uploaded is converted to RGB by removing all the noise in the image.
   3. The image is preprocessed by converting it to a numpy array using **np.array**, reshaping it using **np.reshape(1,64,64,3)**, normalizing the pixel values, and making a prediction using the classifier model i.e., CNN model from step-7.
   4. The predicted class is used to retrieve recipe information (name, ingredients, cooking instructions) from the obj object.
6. Run the application.
7. End

The above algorithm explains how it is running in the system. It accepts all the images which are in the format of JPG, JPEG, and PNG once the system is taking the image it preprocesses the image. We are creating one empty list called recipe and we have defined a class called recipe in that class there are methods that are used to retrieve the data from the CSV file here we are retrieving all the data which are required from the image and appending all the elements into recipe like recipe name, image, ingredient name, and process of that particular recipe.

After the image is taken, the image is sent to the CNN model we have trained the CNN model and converted it into JSON format so that it can be used in any system. if the JSON file is not available then the dataset is sent to CNN here Algorithm A1[6] **x.txt.npy** refers to images **and Y.txt.npy** refers to all the instructions of the recipe. in training the model we are resizing the image and converting the image from BGR to RGB before converting it to RGB we are converting the image to a grayscale image and removing all the noises from the image and converting the image into RGB. After completing all these we are sending the image to the function called predict in that the image again gets resized and a particular recipe is generated.

**5.2.1 Activation Functions**

ReLU (Rectified Linear Unit) is a popular activation function in neural networks. By producing the input value if it is positive and 0 otherwise, it generates nonlinearity. To extract picture characteristics, ReLU can be used in the image encoder. It helps in the acquisition and representation of key visual patterns and traits in food images. The image encoder's output can be subjected to ReLU activation to enable the generator to concentrate on important image features while ignoring unimportant or noise-related data. It is possible to create precise and relevant recipe names and cooking processes by combining the extracted picture features, which ReLU has improved, with additional inputs like ingredient embeddings.

Softmax is a neural network activation function used to convert a vector of real numbers into a probability distribution. The probability distribution across the word vocabulary can be generated and used in the decoder for cooking instructions. It enables the model to give different words probabilities that signify the possibility that each word will be the next predicted step in the formula. The generator can make confident predictions and choose the most likely phrases to generate cooking stages by using softmax. On the basis of the input image and other contextual data, the generator is then able to produce coherent and understandable recipe instructions. The softmax function makes sure that all possible word probabilities add up to 1, enabling the model to produce accurate and normalized predictions.

**Activation Function for RELU:**

**Mean Squared Error (MSE) Loss:**

1. Loss = (1/n) \* Σ(y\_true - y\_pred)^2  
   This loss function is typically used for regression tasks. It computes the mean squared difference between the true values (y\_true) and predicted values (y\_pred).

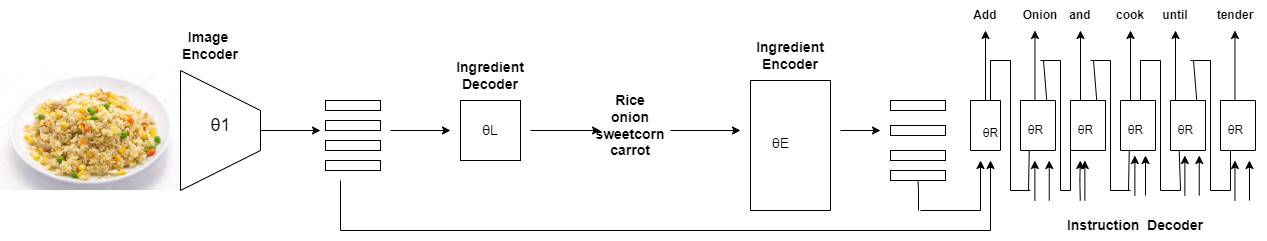
**Categorical Cross Entropy Loss:**

1. Loss = - Σ(y\_true \* log(y\_pred))  
   Categorical cross-entropy loss is used for multi-class classification problems. It calculates the difference between the true class labels (y\_true) and the predicted class probabilities (y\_pred).

**The Loss Function for SoftMax:**

Loss = - Σ(y\_true \* log(y\_pred))

Here You can refer to **Figure 1** That image is uploaded from the dataset and it will predict the recipe of the image. The process involves extracting image features using image processing i.e., image encoder, which is defined by parameters θ1. Using parameters L, the expected ingredients are obtained and encoded into ingredient embeddings eL using parameters e. The R-parameterized cooking instruction decoder generates a recipe title and a series of cooking processes. This is accomplished by paying close attention to image embeddings eI, ingredient embeddings eL, and previously predicted words (r0,..., rt1).



**Figure 2: System Architecture**

**5.2.2 CNN**

* **Figure 2** explains how the process is going to take place in our system.
* Here from in **Figure1** First we are importing all the libraries it checks whether all the packages are present or not if not present the process gets terminated.
* After verifying all the libraries, we need to insert the dataset, inserting the data set takes us forward to the next step for image processing.
* Once the image processing step is done it goes to data extraction with this data extraction it takes the relevant information from the image so that it is used for instructions generation.
* After this step now we are splitting the data for testing and training.
* For training the data the data is sent to a model called CNN
* After training the data with the help of web pages which we have designed the user needs to log in/signup and give the input to the model.
* From the trained model we get the output as name, ingredients, and process of that particular recipe

**5.2.3 AlexNet**

* You can refer to **Figure 2 and** here all the processes are the same as **5.2.2.**
* Once the preprocessing step is done the image is sent to the Alexnet model.
* Here in Alexnet, the model is trained in eight layers which we have five Convolution layers and three fully connected layers.
* with the help of this Alexnet we can extract more features than CNN.
* After training the model with the dataset we send the image for prediction and it generates the name, ingredients, and recipe process same as **5.2.2**

**5.2.4 VGG**

* You can refer to the **Figure 2 and** here all the processes are the same as **5.2.2.**
* Once the preprocessing step is done the image is sent to the VGG model.
* Here in VGG, the model is trained in sixteen layers which we have thirteen Convolution layers and five fully connected layers.
* with the help of VGG when we compare it with **5.2.2** and **5.2.3** it has more layers it helps in capturing intricate features from the image.

From the above proposed methods from **5.2.2**, **5.2.3, 5.2.4** uses advanced CNN models.

**5.3 Limitations for proposed work**

Existing computer vision systems must go beyond the obvious to create high-quality structured food preparation descriptions. In the coming days, we are going to add voice to the recipe which is generated and we are going to train these models by adding more recipes currently our dataset contains only 10,000 images we are going to increase the data set and train with different models and we are going to add some links also for the reference while generating the recipe (eg: video) and some set of instructions.

# Experimental Analysis

In this section the proposed work has been experimented by using deep learning models on the two different recipe generation datasets.

## 6.1 Experimental Setup

|  |  |
| --- | --- |
| **Category** | **Description** |
| Processor | Intel core i5 Processor |
| Internet | Ethernet connection (LAN)/wireless adapter (Wi-fi) |
| Hard Disk | 500GB |
| Memory (RAM) | Minimum 8GB; Recommended 64 GB or above |

### Table 1: Hardware Requirements

|  |  |
| --- | --- |
| **Category** | **Description** |
| Domain | Deep Learning |
| Programming Language | Python, HTML, CSS, Sqlite3 |
| Tools & Libraries | Numpy, Pandas, Mathplotlib, Keras, Tensorflow, OpenCV, Flask |
| IDE | Anaconda prompt, Jupyter Notebook |
| Operating System | Windows 11 |

**Table 2: Software Requirements**

**6.2 Dataset Description**

Throughout history, food has played an important part in human culture and society. With the increasing interconnectedness of our world, there has been a surge in the availability of vast amounts of food-related information. This has sparked a keen interest in comprehending and analyzing culinary trends, dietary habits, and recipe recommendations.

Our primary objective in analyzing the dataset is to uncover the intricate connections between ingredients and cooking techniques. This dataset, known as **core-data\_recipe**, serves as the foundation for training and evaluating the recipe generation system. It is organized in a tabular format, typically stored as CSV (Comma-Separated Values). The dataset encompasses several columns, including image file paths or URLs, ingredient lists, and step-by-step cooking instructions.

Each entry in the dataset contains essential information such as id of recipe, recipe name, image, ingredients,and cooking instructions. The cooking\_directions field includes estimated preparation time and corresponding temperature for each recipe. Additionally, the nutrition column provides information about the nutritional content derived from each recipe. Here, we have approximately 45k recipes.

The **"Images"** folder contains a vast collection of approximately 1125 subfolders, each representing a unique recipe, and serving the purpose of accurately training the model. Within each subfolder, there are approximately 6 images of the corresponding recipe. These subfolders contain recipes from various cuisines like South Indian, North Indian, and Chinese dishes, providing a wide range of options.

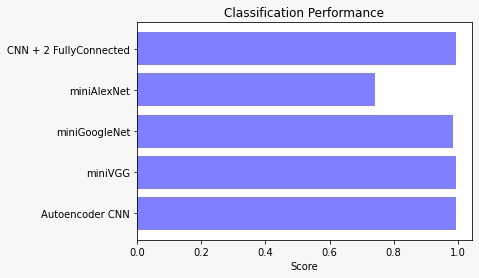
**Modelbest.ckpt:** This file is a trained model checkpoint that represents the best-performing model during the training process. It contains the learned weights and parameters of the deep learning model used for recipe generation. The "Modelbest.ckpt" file can be downloaded to use the pretrained model for recipe generation tasks.

**ingr\_vocab.pkl**: This file is related to the vocabulary of ingredients used in the recipe generation system. It is typically a pickle file that stores the mapping between ingredients and their corresponding numerical representations or embeddings. The "ingr\_vocab.pkl" file can be downloaded and loaded into the code to access the ingredient vocabulary and facilitate ingredient-related operations.

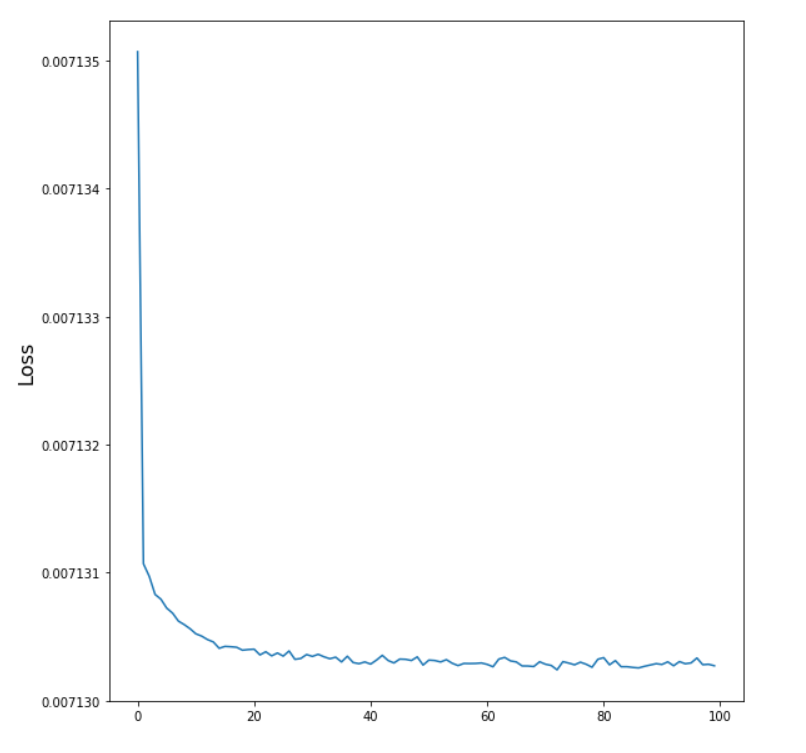
**instr\_vocab.pkl:** This file corresponds to the vocabulary of cooking instructions or recipe steps in the recipe generation system. Similar to the ingredient vocabulary file, it is usually a pickle file that contains the mappings between instructions and their numerical representations or embeddings. By downloading the "instr\_vocab.pkl" file and loading it into the code, users can access the instruction vocabulary and perform operations related to recipe steps.

**6.3 Classification Results**

As mentioned in **section 1.1**(i.e., preliminary section) We have trained our dataset using five algorithms. CNN, MiniAlex Net, Mini Google Net, MiniVGG, Autoencoder CNN. using all these five algorithms we have compared the accuracy among them and then we got the best accuracy for CNN and Autoencoder CNN so we used these algorithms to build our model and predict the recipe instructions.

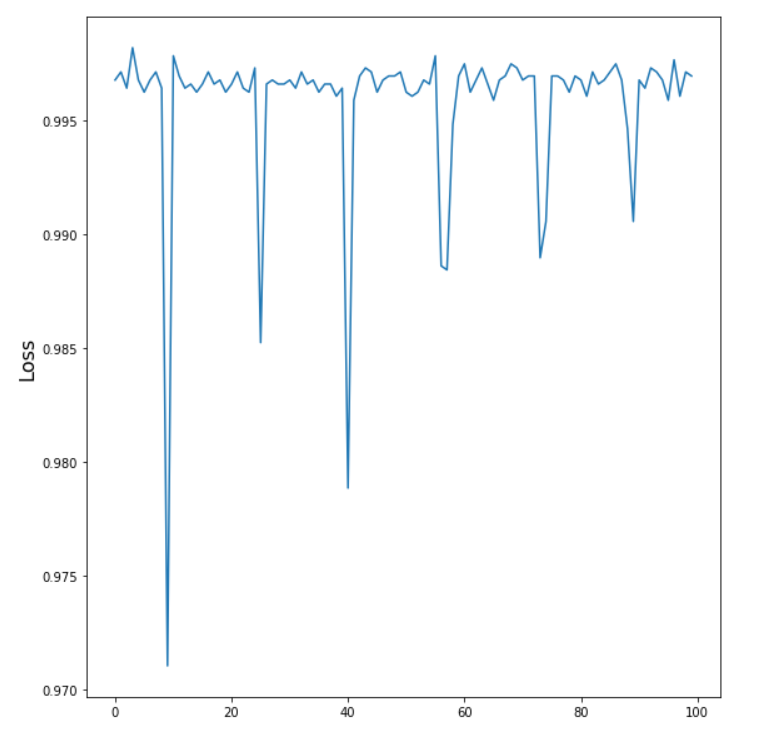


**Figure 3: Comparison Between Classification Models**



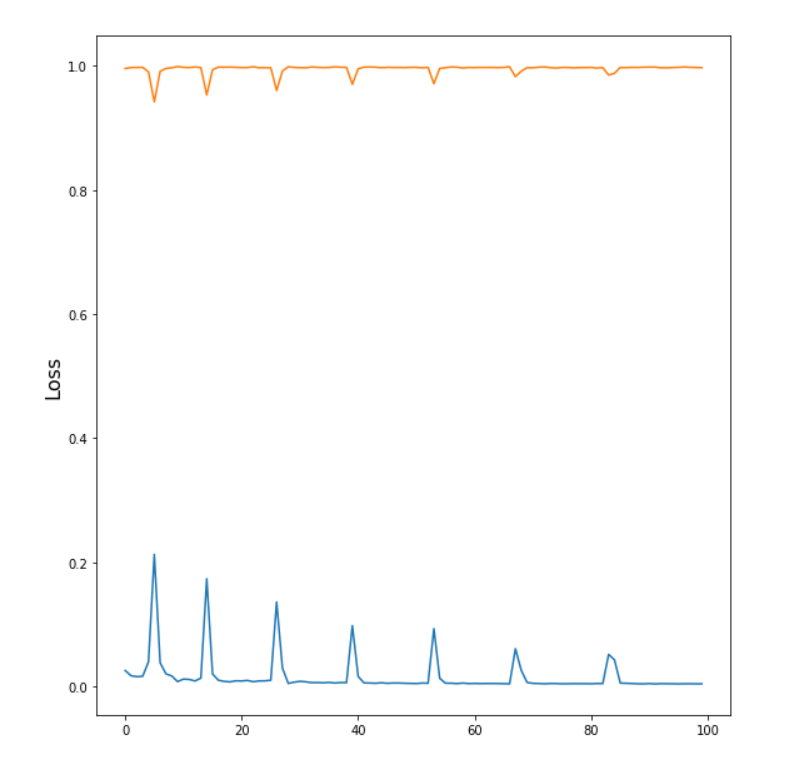
**Figure 4: Plot for CNN**

The provided **Figure 4** illustrates the graph of the CNN model performance after training the data over 100 epochs. The X-axis represents to accuracy, while the Y-axis represents loss. The model achieved an accuracy of **99.92%**.



**Figure 5: Plot for AlexNet**

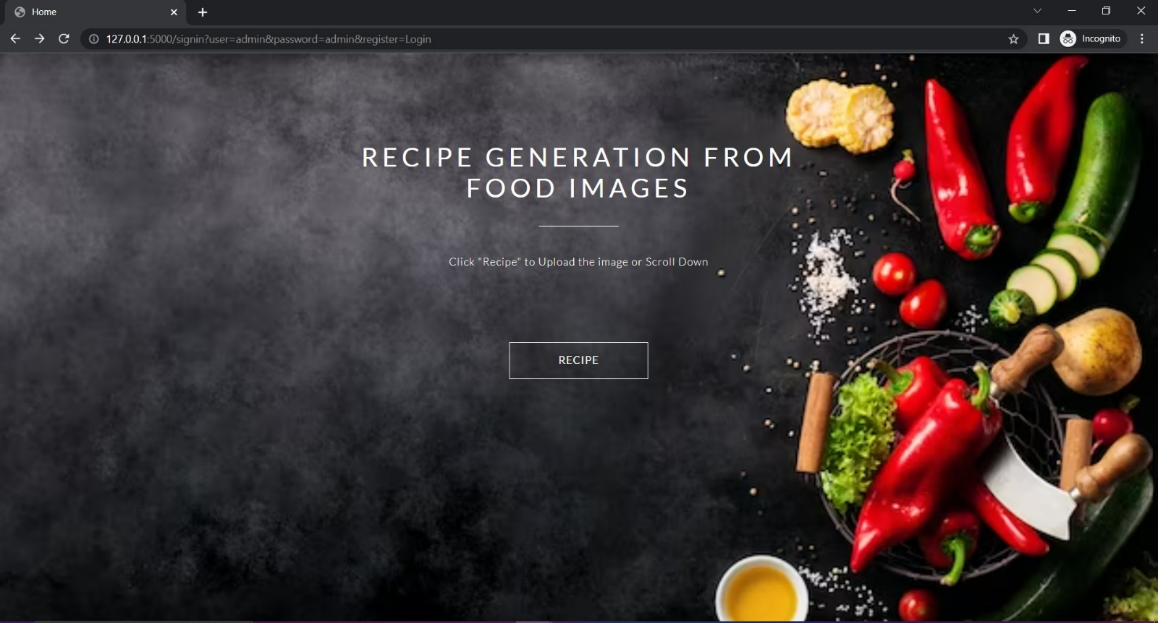
**Figure 5** illustrates the performance of the miniAlexNet model after training the data over 100 epochs. The graph represents accuracy on the X-axis and loss on the Y-axis. After 100 epochs, the model achieved an accuracy of **99.7%.**

****

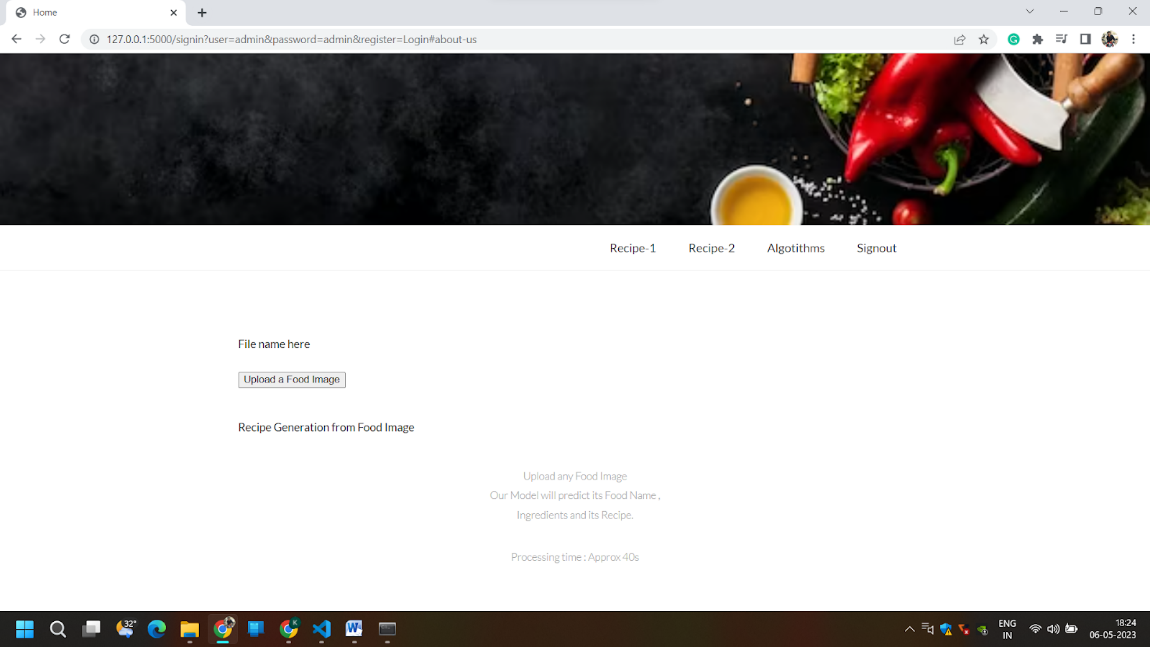
**Figure 6: Plot for VGG**

The graph in **Figure 6** shows the performance of the miniVGG model after training the model over 100 epochs. The Y-axis represents accuracy, and the y-axis represents loss. The accuracy achieved after 100 epochs is **99.66%**, with a corresponding loss of 0.43%.

**6.4 Results of Ingredients Recognition and Recipe Generation**

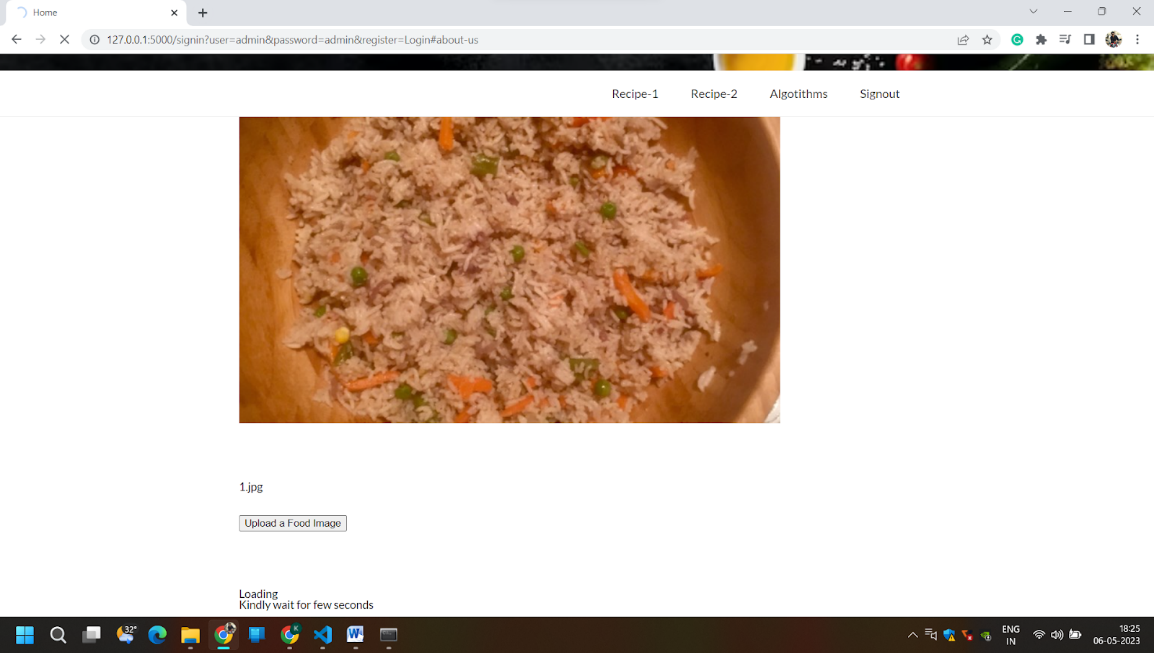
****

**Figure 7: Home Page**

****

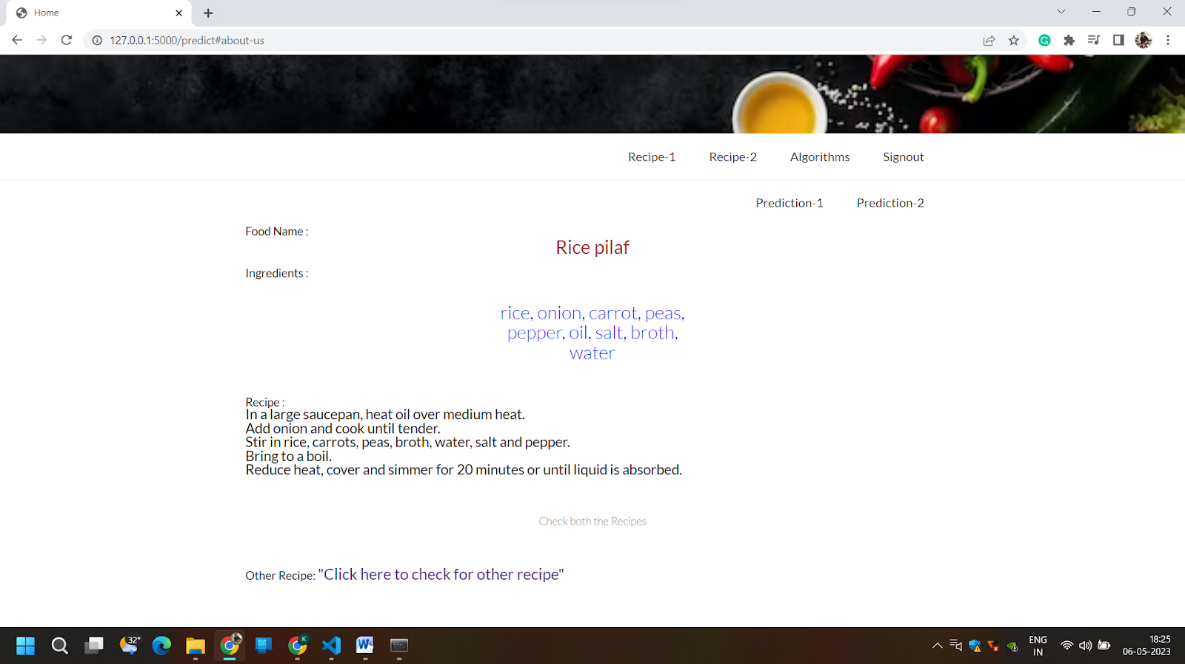
**Figure 8: Upload a Food Image**

As shown in **Figure 8** we need to insert any food image from our device for which we want to recognize the ingredients and generate the recipe.

****

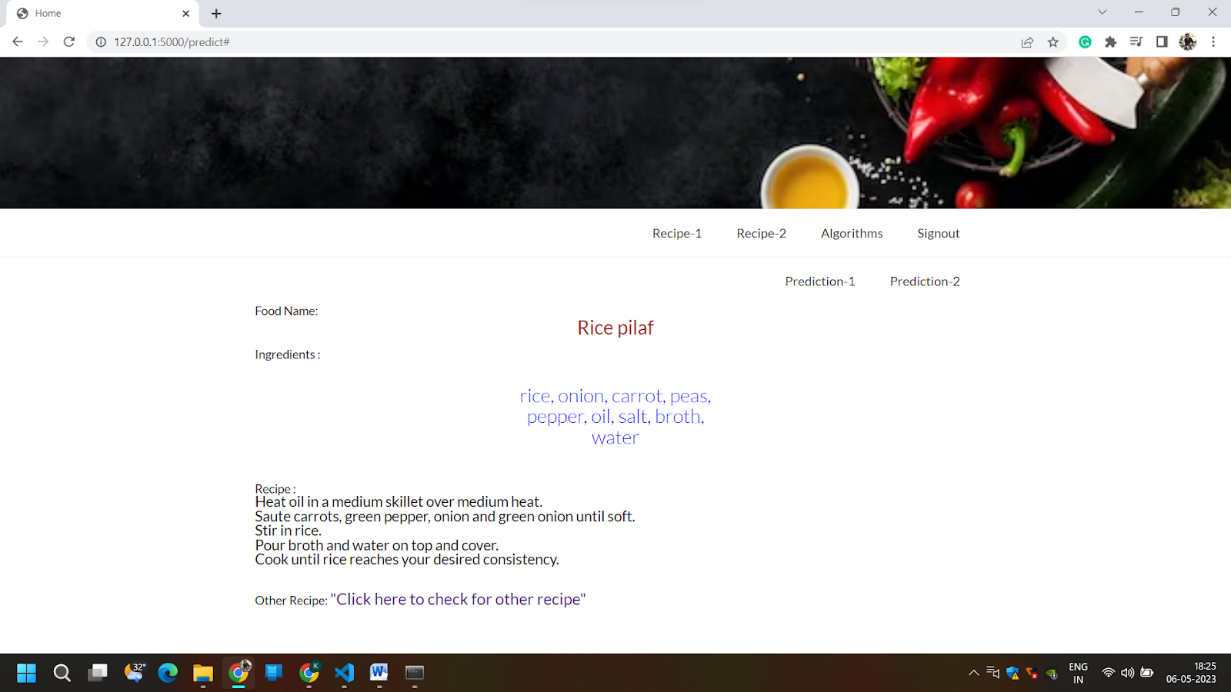
**Figure 9: Uploaded a Food Image**

Here we gave one of the food images which we downloaded from Google now the system is taking the food image and it is processing the food image to recognize the ingredients.

****

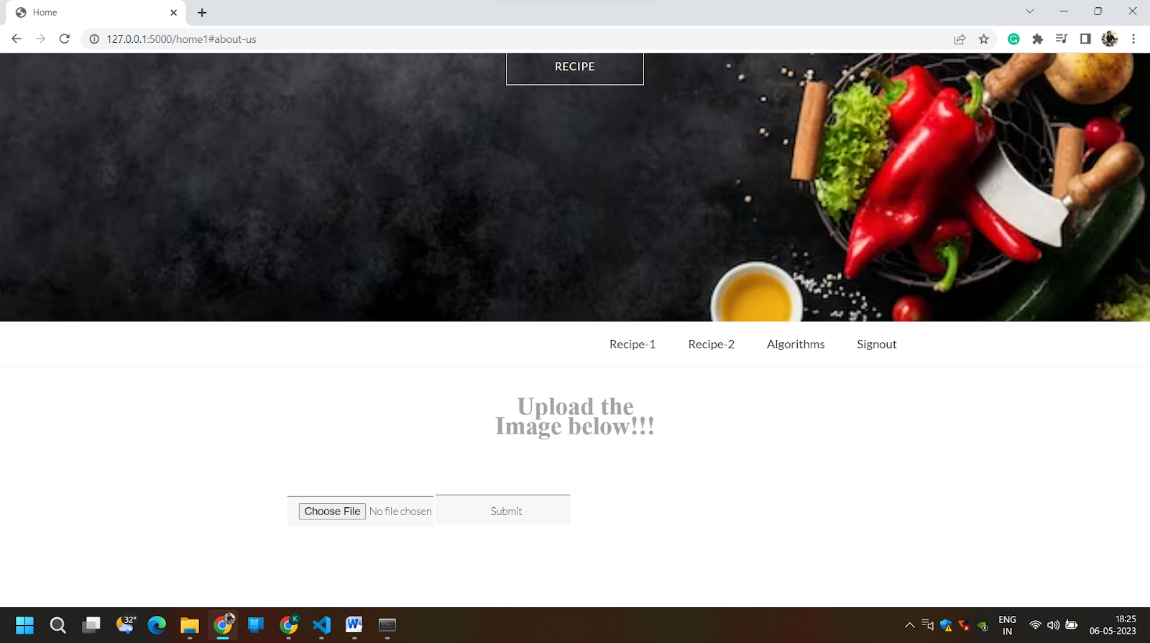
**Figure 10: Recipe 1 Output 1**

The **Figure 10** shows that It takes the food image as input and pre-processes the image. After Pre-processing the image, you can see the above **Figure 10** First it generated the Recipe name, required ingredients, and Making of that recipe which you can see in **Figure 10.**

****

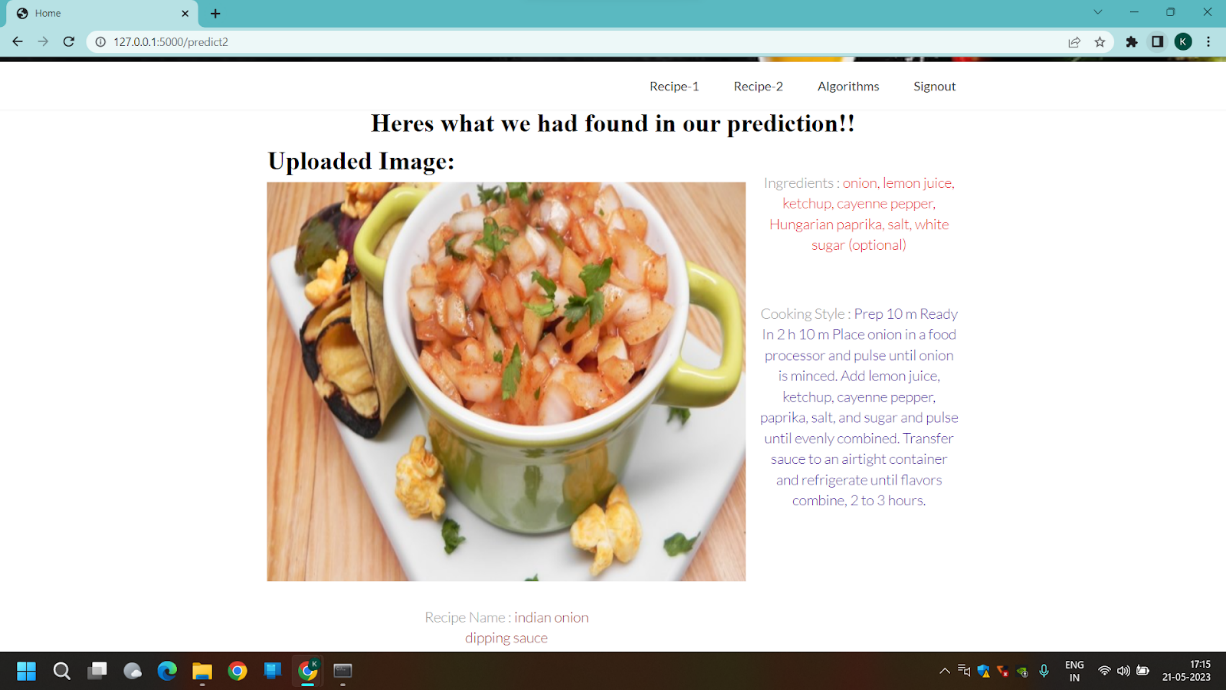
**Figure 11: Recipe 1 Output 2**

The above image **Figure 11** shows the second predicted output this gives the output the same as **Figure 10.**

****

**Figure 12: Upload Image for model-2**

This Recipe-2 Also takes the image from the user and this model is one which we have trained by giving all the data in Excel and trained by sending 10k images as a dataset.



**Figure 13: Recipe 2 Output**

This **Figure 13** Shows you the output of the model which we have trained. It gives you the output as the image which the user gave as the input, Recipe name, Ingredients, and cooking details.

# Conclusion

In this research, we proposed an image-to-recipe generating system which takes a food image as input and generates a recipe as output with a name of the food item, ingredients used for food, and cooking instructions. We have compared this with many other models and compared the accuracy with all models we have trained and chosen the best suitable algorithm.  We initially predicted ingredient sets from food images, demonstrating the importance of modelling dependencies. Then we looked at instruction generation using visuals and inferred ingredients, emphasising the significance of reasoning about each media at the same time. Finally, user research findings confirm the task's difficulty and show our system outperforms cutting-edge image-to-recipe retrieval systems.

 In the coming days, we are going to add voice to the recipe which is generated and we are going to train these models by adding more recipes currently our dataset contains only 10,000 images we are going to increase the data set and train with different models and we are going to add some links also for the reference while generating the recipe (eg: video) and some set of instructions.

# References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In ECCV, 2014.

[2] Micael Carvalho, R´emi Cad`ene, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings. In SIGIR, 2018.

 [3] Jing-Jing Chen and Chong-Wah Ngo. Deep-based ingredient recognition for cooking recipe retrieval. In ACM Multimedia. ACM, 2016.

 [4] Jing-Jing Chen, Chong-Wah Ngo, and Tat-Seng Chua. Cross-modal recipe retrieval with rich food attributes. In ACM Multimedia. ACM, 2017.

[5] Mei-Yun Chen, Yung-Hsiang Yang, Chia-Ju Ho, Shih-Han Wang, Shane-Ming Liu, Eugene Chang, Che-Hua Yeh, and Ming Ouhyoung. Automatic Chinese food identification and quantity estimation. In SIGGRAPH Asia 2012 Technical Briefs, 2012.

[6] Xin Chen, Hua Zhou, and Liang Diao. Chinesefoodnet: A large-scale image dataset for Chinese food recognition. CoRR, abs/1705.02743, 2017.

[7] Bo Dai, Dahua Lin, Raquel Urtasun, and Sanja Fidler. Towards diverse and natural image descriptions via a conditional gan. ICCV, 2017.

 [8] Krzysztof Dembczy´nski, Weiwei Cheng, and Eyke H¨ullermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In ICML, 2010.

[9] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In ACL, 2018.

[10] Claude Fischler. Food, self, and identity. Information (International Social Science Council), 1988.

 [11] A. Salvador, M. Drozdzal, X. Giro-i-Nieto and A. Romero, "Inverse Cooking: Recipe Generation From Food Images," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 10445-10454, doi: 10.1109/CVPR.2019.01070.

[12]Li, K., & Chen, Y. (2021). DeepRecipes: Exploring Massive Online Recipes and Recovering Food Ingredient Amounts.

Retrieved from <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9423993>

[13]H. Wang, G. Lin, S. C. H. Hoi and C. Miao, "Learning Structural Representations for Recipe Generation and Food Retrieval," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 3, pp. 3363-3377, 1 March 2023, doi: 10.1109/TPAMI.2022.3181294.

[14] M. Zhang, G. Tian, Y. Zhang and H. Liu, "Sequential Learning for Ingredient Recognition From Images," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 33, no. 5, pp. 2162-2175, May 2023, doi: 10.1109/TCSVT.2022.3218790.

[15] Hassannejad, H., Kiani, M., Salvador, A., & Eirinaki, M. (2021). Large Scale Visual Food Recognition. arXiv preprint arXiv:2103.16107.