Lab Project Submission

submitted for

Machine Learning (UML501)

Calorie-Click: Total Nutritional Calculator

submitted by

Aryan Garg **102103768**

Samarth Paliwal **102103775**

3 COE 27

submitted to

Dr. Ashutosh Aggarwal

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala

1. Introduction

1.1. Introduction

The culinary diversity of Indian cuisine presents a captivating challenge for automated food recognition systems. As technology continues to evolve, the application of deep learning in computer vision offers a promising solution to accurately identify and categorize Indian dishes based on their visual attributes.

1.2. Objectives

The primary objective of this project is to develop an efficient and accurate system for the automated classification of Indian food items. Leveraging the capabilities of deep learning, we aim to create a model that can recognize and categorize a diverse range of dishes, from popular staples to regional specialties.

2. Need of the Project

2.1. Dietary Analysis and Personalization

For individuals, the system serves as a valuable tool for dietary analysis. It can assist those with dietary restrictions or preferences by providing instant information about the food they are consuming and the nutritional information of the meals they consume.

3. Methods Used

3.1. Deep Learning - Scratch

At first, we experimented by training complex model architectures, like VGG from scratch. This usually led the model to be underfit because of the large number of trainable parameters. Smaller models were also not as effective.

3.2. Deep Learning – Transfer Learning

The most accurate model was trained using a pre-trained ResNet50 model—a deep convolutional neural network renowned for its image classification tasks. Transfer learning is implemented to leverage the knowledge acquired by the model from a large dataset, enhancing its capacity to understand the images of Indian food.

3.3. Random Forest Classifier

The Random Forest Classifier is an ensemble learning method that belongs to the family of decision tree-based models. Ensemble learning involves combining multiple individual models to create a stronger, more robust model. Random Forests are particularly powerful and versatile for both classification and regression tasks.

3.4. Gaussian Naïve Bayes

Gaussian Naive Bayes is a probabilistic machine learning algorithm that belongs to the family of Naive Bayes classifiers. It is specifically designed for datasets where the features are continuous and assumed to follow a Gaussian (normal) distribution. This classifier is known for its simplicity and efficiency, making it suitable for various classification tasks.

3.5. Decision Tree Classifier

Decision Trees, uses tree-like structures break down complex decisions into a series of simpler, manageable choices, offering a transparent and interpretable solution for classification problems.

3.6. KNN – 3, 11, 21 Points

KNN is a simple and intuitive machine learning algorithm used for both classification and regression tasks. The key idea is to predict the class or value of a new data point based on the majority class or average value of its k nearest neighbours in the feature space.

3.6.1. Implications of Different Values of k:

3.6.1.1. Small k (e.g., k = 3):

- Low bias, high variance.
- The model is sensitive to noise and outliers.
- Captures local patterns well but may not generalize to the overall structure of the data.
- Prone to overfitting.

3.6.1.2. Moderate k (e.g., k = 11):

- A balance between bias and variance.
- Less sensitive to noise than smaller k.
- Captures both local and some global patterns.
- Generally, a good starting point.

3.6.1.3. Large k (e.g., k = 21):

- High bias, low variance.
- More robust to noise and outliers.
- Captures global patterns but may miss local details.
- Prone to underfitting.

3.7. Voting Classifier

A Voting Classifier is an ensemble machine learning model that combines the predictions from multiple individual models (classifiers) to make a final prediction. It operates based on the principle of "majority voting," where each classifier in the ensemble "votes" for a class, and the class with most votes becomes the final predicted class.

4. Dataset Description

4.1. Web Scraping

Using Google Images, we used requests and BeautifulSoup4 to extract images. Each class had approximately 630 images, which was then manually filtered to remove anomalies and cropped to focus on the food item. Left with approximate 6500 Images. The Dataset was saved on Kaggle.

4.2. Data Augmentation

All images were resized to (224x224) and normalised with the mean and standard deviation of the dataset. Apart from that, to increase the size of the dataset, a separate copy of images were also saved with Horizontal Flip and Random Affine between 0 and 15 degrees.

5. Experimental Results

5.1. CNN Based Models

5.1.1. From Scratch Model

This model consists of multiple convolutional layers with ReLU activation and max-pooling operations, followed by fully connected layers for classification. The network starts with 3 input channels and gradually increases the number of filters in convolutional layers, capturing hierarchical features. The final fully connected layers reduce the feature dimensionality and output the classification results for 20/14 classes.

5.1.2. Transfer Learning:

The model modifies the classification head of a pre-trained ResNet model, which was trained on ImageNet. It replaces the existing fully connected layer with a new sequence of layers, including a linear layer with 256 output features, ReLU activation, dropout for regularization, another linear layer for the final output of 20/14 classes, and Log Softmax activation.

5.2. Traditional ML Models

Using traditional Machine Learning models, we were able to extract the images into a set of rows and columns, where each column represented a pixel in the image. The total column count was 224x224x3 (1,50,528). We were able to prove that our CNN based approach was more effective in the final output.

5.3. Comparative Analysis

Model Name	Accuracy % (No Normalisation and Augmentation)	Accuracy (With Normalisation & Augmentation)
Scratch Model	45.21	-
ResNet Transfer Learning	80.46	-
Random Forest Classifier	29.44	29.86
Gaussian Naïve Bayes	19.16	18.85
Decision Tree Classifier	18.85	17.46
KNN (3)	14.21	15.06
KNN (11)	16.15	15.37
KNN (21)	14.68	16.11
Voting Classifier (Ensemble)	19.93	19.35

6. Contributions

Aryan Garg: Worked on Dataset Scraping, ResNet as well as Scratch Architecture. Developed API to receive images, make prediction, and send result back to UI.

Samarth Paliwal: Worked on Dataset Scraping and Traditional Machine Learning Models. Also Developed Front and Backend in MERN Stack.