Syracuse University

FOOD PREDICTION SYSTEM

Laxman Kumar SUID: 419347253
Rima Shah SUID: 625658814

CSE 731 – ARTIFICIAL NEURAL NETWORKS

PROF. C. MOHAN

DECEMBER 10, 2020

Table of Contents

1. Introduction

- Nowadays people are getting conscious about healthy living, nutritious food and diet. That is why food dish prediction is getting popular all over the world to monitor dietary data of people.
- We have executed an approach to recognize a food dish from its image taken using transfer learning and Convolutional Neural Networks (CNN).
- Using the Food 101 image dataset to build the food recognition system, we performed transfer learning on a pre-trained Densenet121 model and then implemented the CNN model using Keras.
- The model was built using various CNN layers, max pooling layers and drop out layers to obtain the best results.

2. Problem Statement

- The goal of this project is to build a system that accepts an image of a food dish as input and can generate the possible name of the dish and if possible, its ingredients.
- As the problem statement states, the goal is to predict the name of the dish based on the image.
- Currently, we are only focusing on 101 different food classes due to computation limitations.

3. Data Description

- We are using the Food 101 dataset, which is a subset of the Recipe 1M+ dataset.
- As the name suggests it contains a total of 101 categories of food with a total of 101,000 images.
- With every class containing 1000 images, there are 750 training images and 250 testing images.
- A sample set of 5 GB has been used for this project.
- Figure1. displays a sequence of random images selected from various classes.

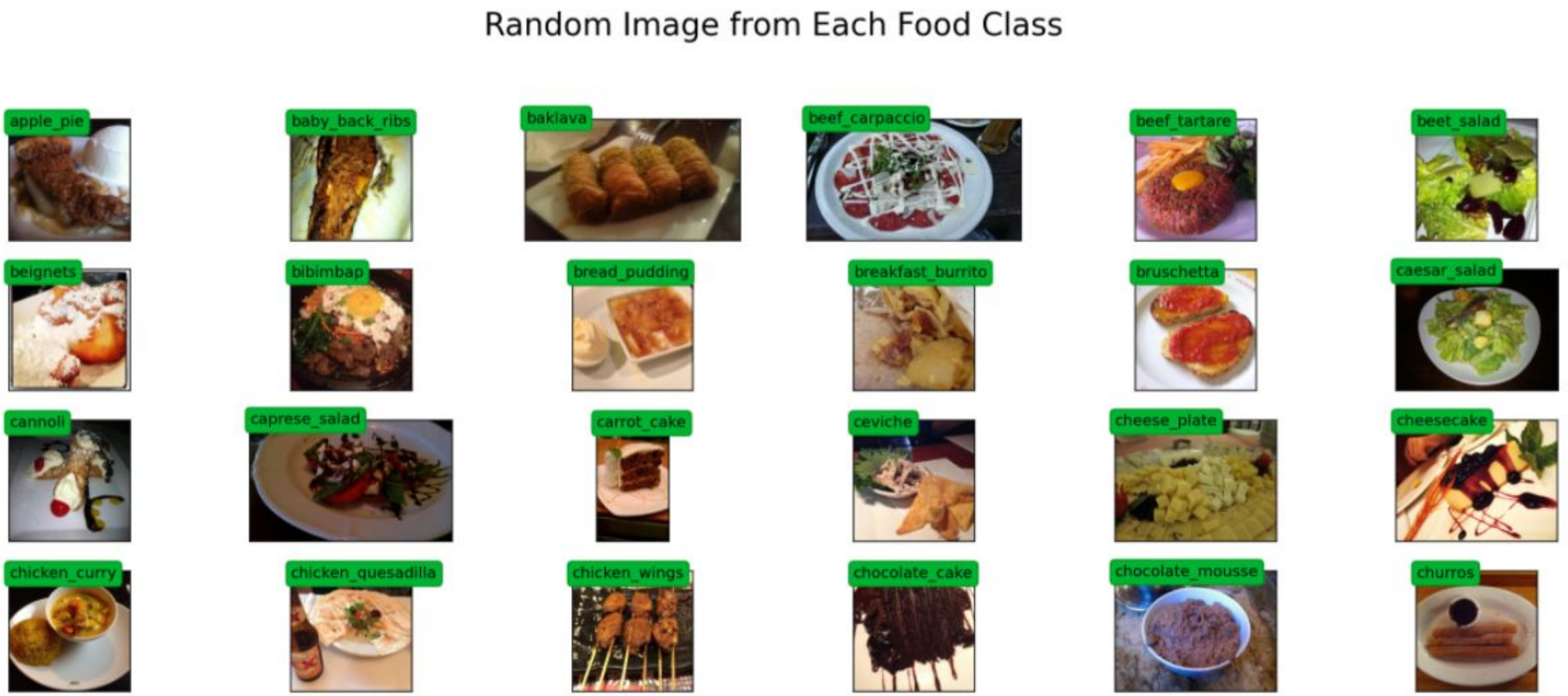


Fig. 1 Overview of Images

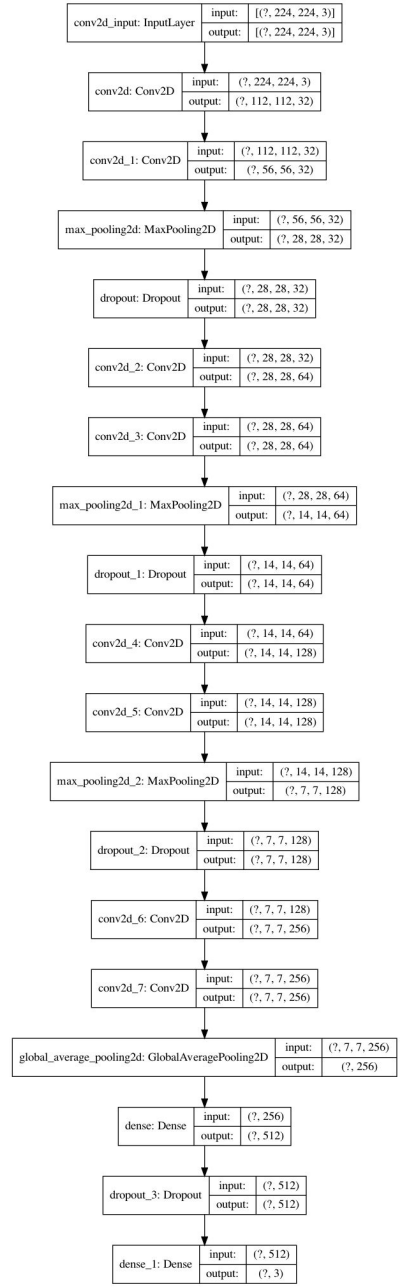
4. Data Preprocessing

- Firstly, we separated all the images into train and test folder based on the information from text file.
- Since all the images were not of similar sizes, we have also rescaled the images to a size of 224*224 pixels for efficient image processing.
- To train our model for obtaining maximum accuracy, we have performed data augmentation on our images.
- Here we have implemented resizing, adding hue and saturation, flipping the image horizontally and rotating the images.

Fig. 2. Images after performing data augmentation

5. Modeling

- Transfer Learning - ResNet101, DenseNet121 and AlexNet
- Our CNN architecture contains 9 hidden layers, 4 dropout layers and 4 MaxPooling layers
- We used keras and tensorflow for building the model
- We also tried PyTorch but didn't implemented fully
- We ran this model for 3 classes, 5 classes and 7 classes



5. Modeling

- We ran the model on randomly chosen 3 classes. We repeated the experiment for 4 times to verify the accuracy on randomly chosen classes.
- We then ran the model on 5 classes and 7 classes and recorded the metrics.
- We then tried running model on 11 classes but it didn’t worked well and we closed our experiment here.

6. Result

Model	Train_Acc	Val_Acc	Test_Acc
DenseNet121	68.12	64.10	-
3Class – v1	86.27	81.07	75
3Class – v2	84.76	78.40	85.9
3Class – v3	82.18	74.53	76.6
3Class – v4	93.87	88.80	95.3
5Class – v1	82.12	74.74	79.7
5Class – v2	82.90	78.26	67.2
5Class – v3	83.16	74.61	71.9
7Class – v1	59.68	48.44	42.2
7Class – v2	76.30	73.96	75.0
7Class – v3	70.05	58.20	65.6

7. Future Scope

- Build a model for number of classes greater than 7
- Running model for 200 epochs
- Using the original dataset instead of a sample
- Trying other variation of transfer learning models
- Using PyTorch for all models

8. References

1. http://pic2recipe.csail.mit.edu/
2. https://arxiv.org/abs/1810.06553
3. https://keras.io/api/layers/convolution_layers/
4. https://pytorch.org/hub/pytorch_vision_densenet/