SNDT WOMEN'S UNIVERSITY
Estb:1916

FOOD RECOGNITION AND INGREDIENT DECODER

By
Ms. Poonam Badhe - 04
Ms. Kaveri Chavan - 13
Ms. Vijaya Govekar - 22
Guide: Ms. Prajakta Gotarne

TABLE OF CONTENTS

INTRODUCTION

- Food is a necessity in everyone's life.
- Obesity is increasing at an alarming rate, endangering the lives of many people. Controlling calorie intake is essential for preventing obesity and a variety of other diseases.
- In addition, analyzing food images and estimating calorie intake can assist people in adhering to a healthy food diet. It can also help regular people keep up with their daily diet.
- Food recognition and calorie estimation have grown in popularity in today's world as people become more health-conscious about their diets.
- This can be accomplished in two steps. The first step is to successfully recognize a food image, and then the calorie can be estimated in the second step.

Problem Statement

- People are becoming more aware of the impact of food consumption on health. They are becoming more conscious of the importance of leading a healthy lifestyle.
- Detection of food can assist people in adhering to a healthy diet and awareness of what they are consuming.
- Calorie Estimation

Literature Survey

Sr no.	Year	Title	Purpose/ Context	Advantage	Limitations
1	Aug 2021 International Research Journal of Engineering and Technology (IRJET)	Indian Food Image Recognition with MobileNetV2 Authors- Priya N , Preetam Kumari2, Poorvika N, Sanjana R, Dr. Hema Jagadish	A custom-built CNN model and a transfer learning based MobileNetV2 model for the purpose of food recognition and classification is proposed. A calorie estimation algorithm based on image features and nutritional information is proposed.	The dataset consists of 12 classes of Indian food images, with 100 images per class.	After experimentation, it was found that the MobileNetV2 model outperformed the custom CNN model with an accuracy of 79.45%.

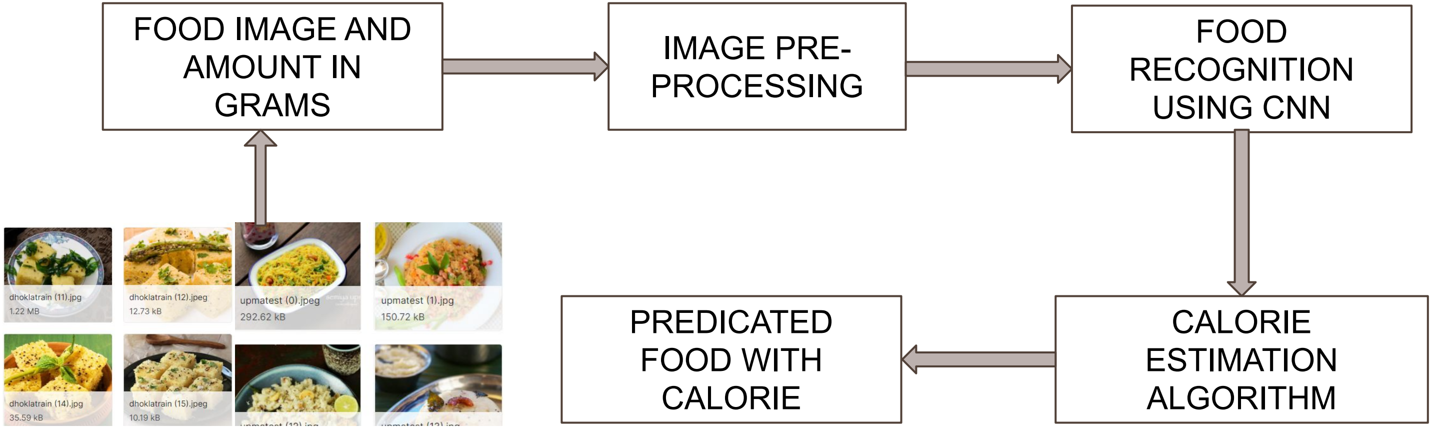
Literature Survey

Sr no.	Year	Title	Purpose/ Context	Advantage	Limitations
2	Dec 2021 The Maharaja Sayajirao University of Baroda	Overview of Deep Learning in Food Image Classification for Dietary Assessment System Author- Bhoomi Shah and Hetal Bhavsar	This paper defines the role of deep learning techniques based on a convolutional neural network for food object recognition. Studied the challenges in food recognition.	Different networks available and its comparison, various deep learning frameworks, the segmentation and classification methods, the performance achieved in terms of Top-1 and Top-5 accuracy, and the comparison of all the CNN architecture with its advantages and disadvantages were studied.	There are variety of food domains that has not been touched yet as not much work done on recognition of liquid food items, and also, there is very less work done till now for European food and Indian cuisine.

PROPOSED SYSTEM:

01	Input	• Users can upload the image of food as input
02	Image Preprocessing	• Image preprocessing
03	Recognition	• Recognition of food will be done by using CNN
04	Calorie Estimation	• The calories range of each food class is obtained from nutrition website and then used along with the HSV method to estimate the calories of the food image
05	Display Output	• The system will predict the name of the dish and the amount of calories

SYSTEM ARCHITECTURE:



Input

IMAGE PREPROCESSING:

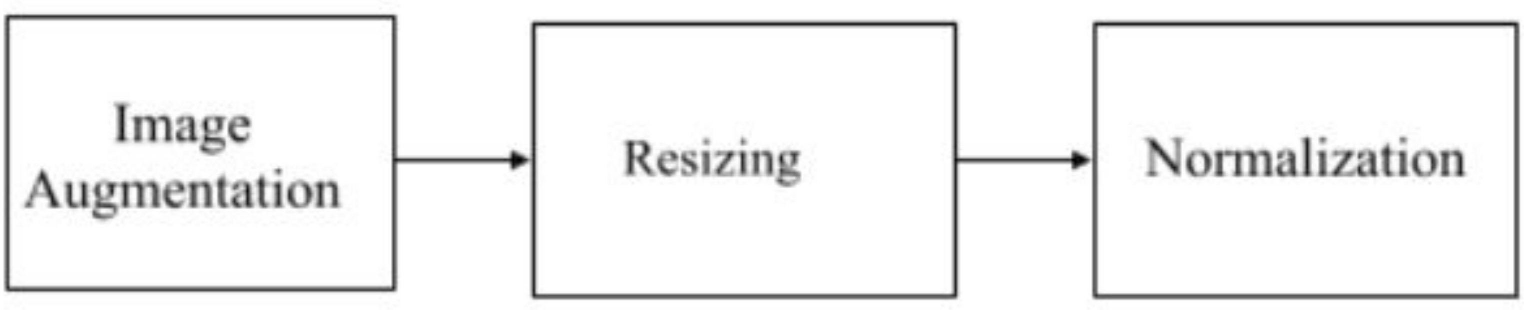
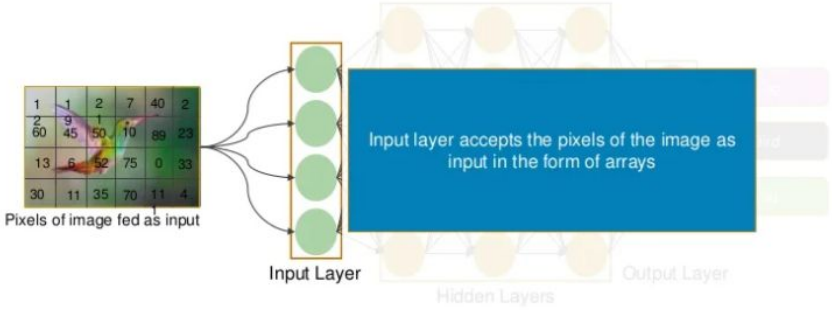


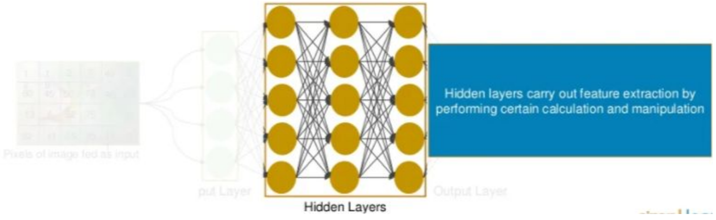
Image augmentation artificially creates training images through different ways of processing or combination of multiple processing, such as random rotation, shifts, shear and flips, etc.

Resizes the image to the specified width and height.

Image normalization ensures optimal comparisons across data acquisition methods and texture instances.

HOW IMAGE RECOGNITION WORKS USING CNN?





ART OF LEARNING STRATEGY MODELS OF CNN:

- LeNet
- AlexNet
- ResNet
- GoogleNet
- MobileNetV1
- MobileNetV2
- Inception v2
- VGG 19
- VGG 16

And many more…

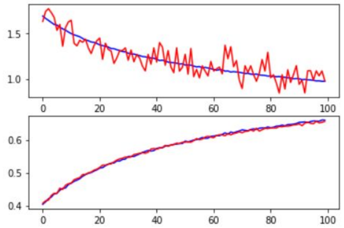
Year	CNN	Developed by	Place	Top-5 error rate	No. of parameters
1998	LeNet(8)	Yann LeCun et al			60 thousand
2012	AlexNet(7)	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	1st	15.3%	60 million
2013	ZFNet()	Matthew Zeiler and Rob Fergus	1st	14.8%	
2014	GoogLeNet(19)	Google	1st	6.67%	4 million
2014	VGG Net(16)	Simonyan, Zisserman	2nd	7.3%	138 million
2015	ResNet(152)	Kaiming He	1st	3.6%	

MobileNet VS ResNet:

The two architectures are compared by training two models in CIFAR-10 classification and then has been evaluated and compared with performance and accuracy.

MobileNet:

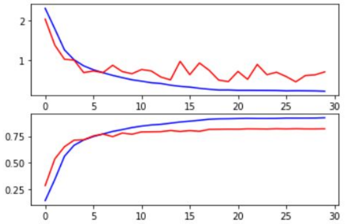
- MobileNets are small, low-latency, low-power models parameterized to meet the resource constraints of a variety of use cases.
- They can be built upon for classification, detection, embedding and segmentation similar to how other popular large scale models.
- MobileNet has accuracy 65% in 100 epochs.



Accuracy Score of MobileNet = 0.654

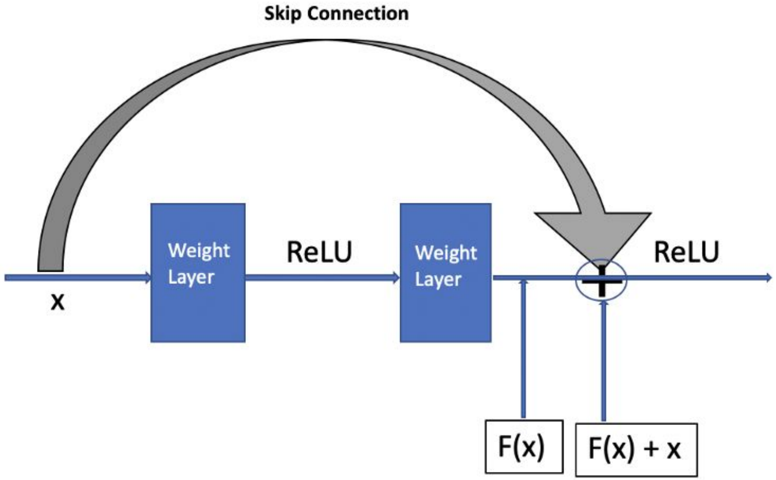
ResNet:

- The pre-trained network can classify images into 1000 object categories, such as a keyboard, mouse, pencil, and many animals.
- As a result, the network has learned rich feature representations for a wide range of images.
- The ResNet-50 has accuracy 81% in 30 epochs



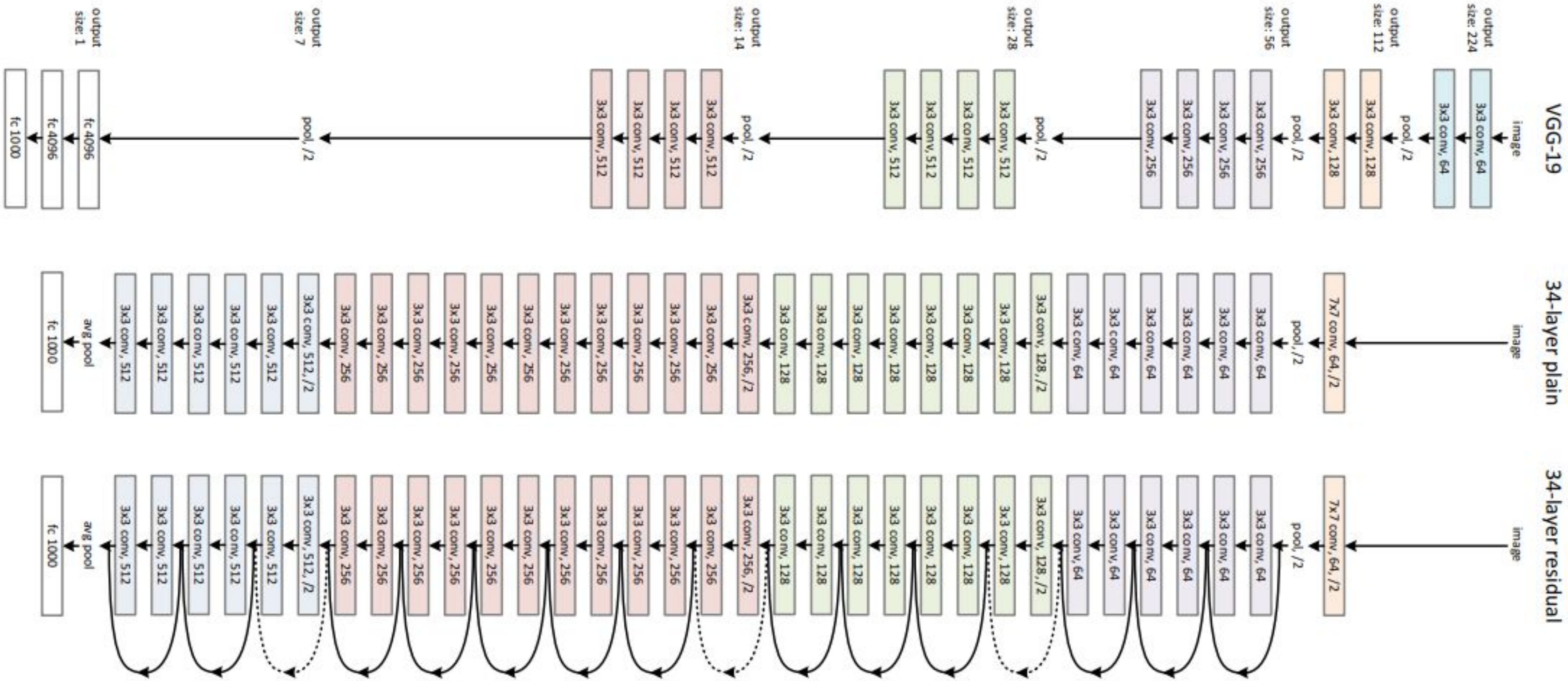
Accuracy Score of ResNet50 = 0.8132

ResNet :



- Consists of Residual blocks that include shortcut skip connections to jump over some layers.
- The skip connection skips training from a few layers and connects directly to the output.
- Easier to optimize , and gain accuracy that is considerably increased.

ResNet Architecture:



CALORIES ESTIMATION ALGORITHM:

- The image sequences are captured with BGR colour space by default. In BGR(Blue,Green,Red) Red occupies the least significant area, green the second and blue the third.
- The BGR colour space is converted to HSV (Hue Saturation Value) space. This is achieved by a function of the OpenCV module called cvtColor( ) using COLOR_BGR2HSV.
- The algorithm uses the concept of indexed images which is a direct mapping of pixel values to colormap values.
- The calories range of each food class is obtained from nutrition websites < https://www.nutritionix.com/ > and then used along with the HSV method to estimate the calories of the food image.
- The formula used for calorie calculation and its code will be shown in phase-2.

HARDWARE/ SOFTWARE REQUIREMENTS

SOFTWARE

Google Colab, Tensorflow, Keras, Pandas, Python 3

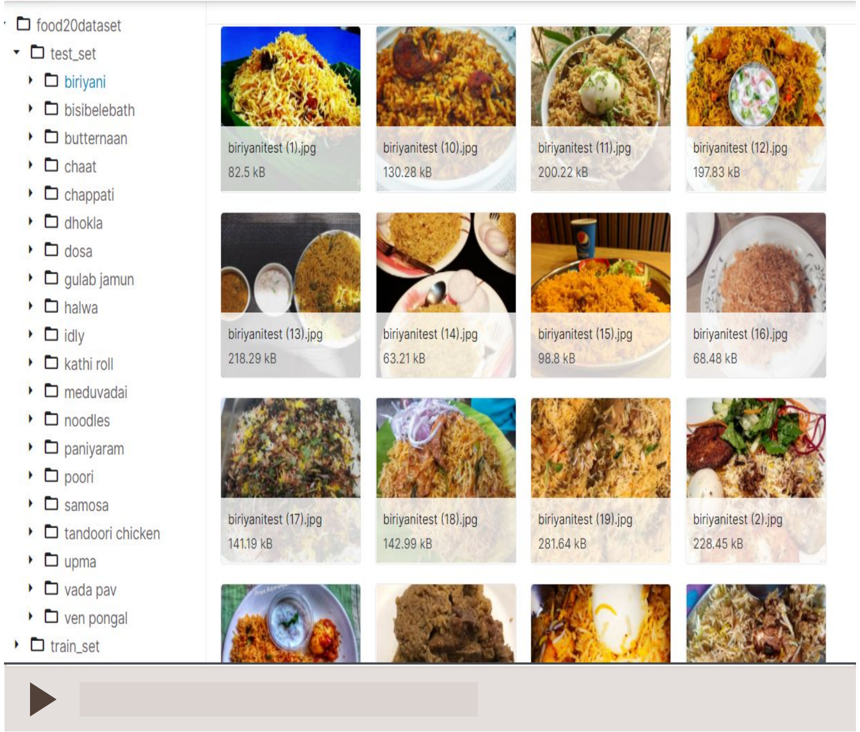
HARDWARE

Graphical User Interface (GUI), Windows 10

IMPLEMENTATION Phase -1

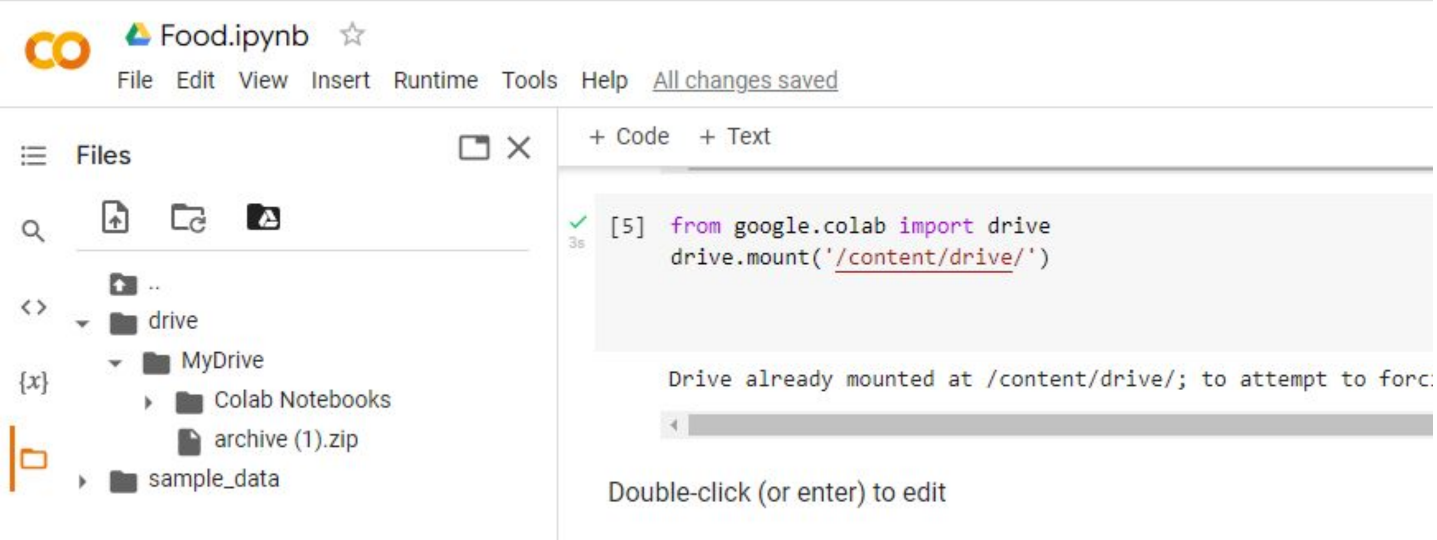
Collection of Dataset

- From original food-20 dataset with 20 indian food categories. A subset of 6 food categories (chapati , dosa, biryani, gulab_jamun, halwa, Idly) is used.
- Two main subfolders: test and train
- The acquired images have resolutions in the range of a minimum of 200 x 150 to 5760x3840 pixels per image.
- The data has been collected from real-world images and is subject to distortions and improper illuminations of certain regions.



IMPLEMENTATION Phase-1

Collection of Dataset



- A Dataset of 6 food categories () is imported into google colab using google drive. We will start the training of dataset in phase 2.

IMPLEMENTATION
Phase - 2

Import Libraries

```
[ ] import tensorflow as tf
    from tensorflow import keras
```

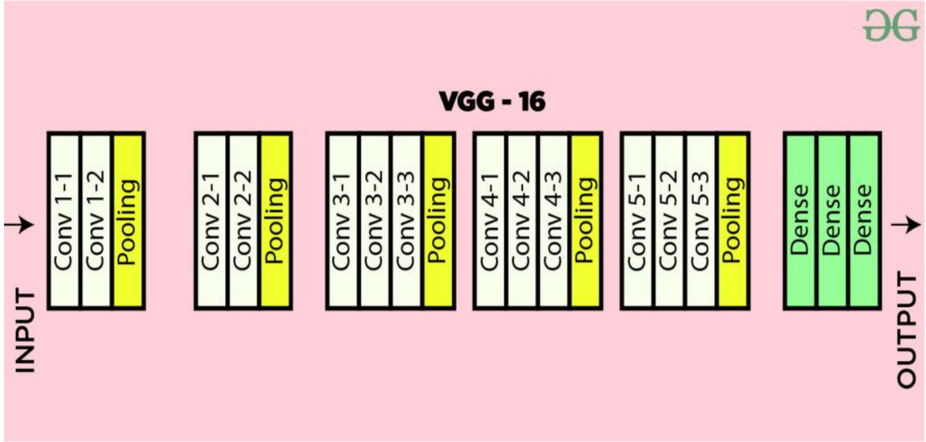
```
from keras.layers import Input, Lambda, Dense, Flatten
from keras.models import Model
from keras.applications.vgg16 import VGG16
from keras.applications.vgg16 import preprocess_input
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
import numpy as np
from glob import glob
import matplotlib.pyplot as plt
```

Import Dataset And Resize Images

```
[ ] # re-size all the images to this
    IMAGE_SIZE = [224, 224]
```

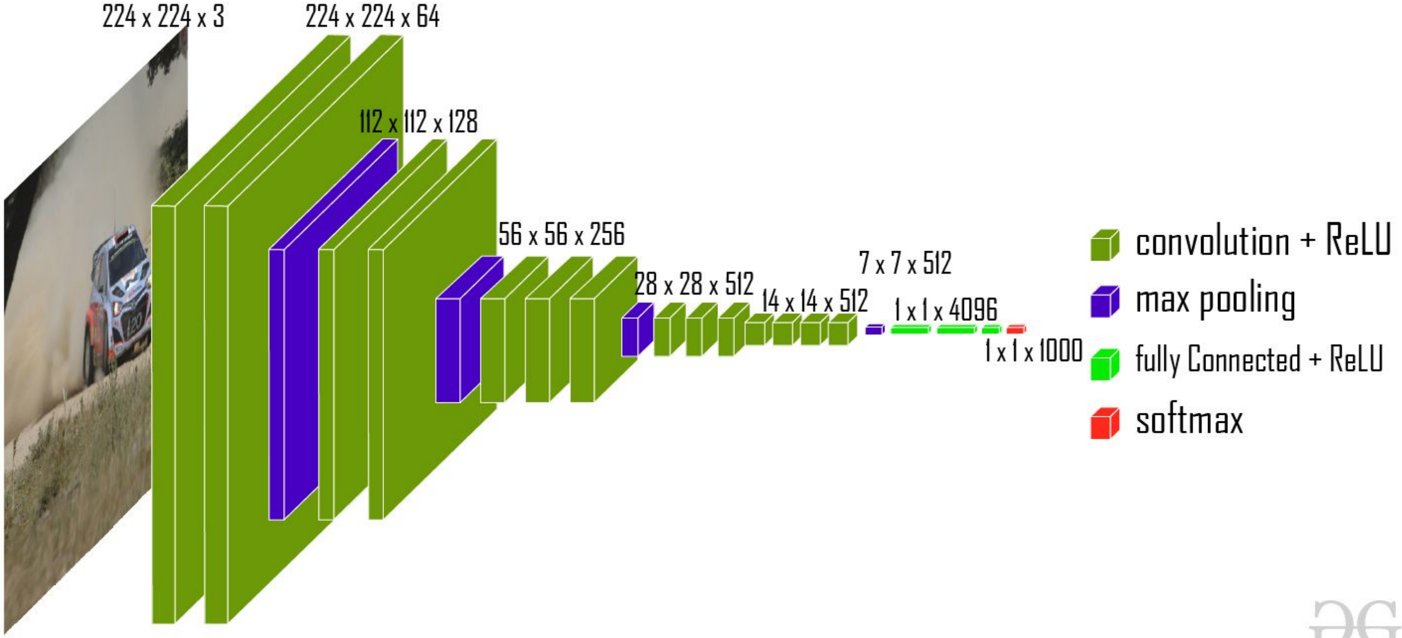
```
[ ] train = '/content/drive/MyDrive/food20dataset/train_set.csv'
    test= '/content/drive/MyDrive/food20dataset/test_set.csv'
```

Vgg 16



- VGG stands for Visual Geometry Group; it is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers.
- The "deep" refers to the number of layers with VGG-16 consisting of 16 convolutional layers.
- The VGG architecture is the basis of ground-breaking object recognition models.
- Developed as a deep neural network, the VGGNet also surpasses baselines on many tasks and datasets beyond ImageNet. Moreover, it is now still one of the most popular image recognition architectures.

Vgg 16 Architecture:



This model achieves *92.7% top-5* test accuracy on ImageNet dataset which contains *14* million images belonging to 1000 classes.

Features of Vgg 16 :

- It is also called the OxfordNet model, named after the Visual Geometry Group from Oxford.
- Number 16 refers that it has a total of 16 layers that has some weights.
- It Only has Conv and pooling layers in it.
- always use a 3 x 3 Kernel for convolution.
- 2x2 size of the max pool.
- has a total of about 138 million parameters.
- Trained on ImageNet data
- It has an accuracy of 92.7%.
- it has one more version of it Vgg 19, a total of 19 layers with weights

IMPLEMENTATION
Phase - 2

Adding preprocessing layer:

```
[ ] # add preprocessing layer to the front of VGG
    vgg = VGG16(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)

    Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
    58892288/58889256 [==============================] - 0s 0us/step
    58900480/58889256 [==============================] - 0s 0us/step
```

```
[ ] # don't train existing weights
    for layer in vgg.layers:
      layer.trainable = False
```

```
[ ]  # useful for getting number of classes
    folders = glob('/content/drive/MyDrive/food20dataset/train_set/*')
```

```
[ ] # our layers - you can add more if you want
    x = Flatten()(vgg.output)
```

```
[ ] # x = Dense(1000, activation='relu')(x)
    prediction = Dense(len(folders), activation='softmax')(x)
```

IMPLEMENTATION
Phase - 2

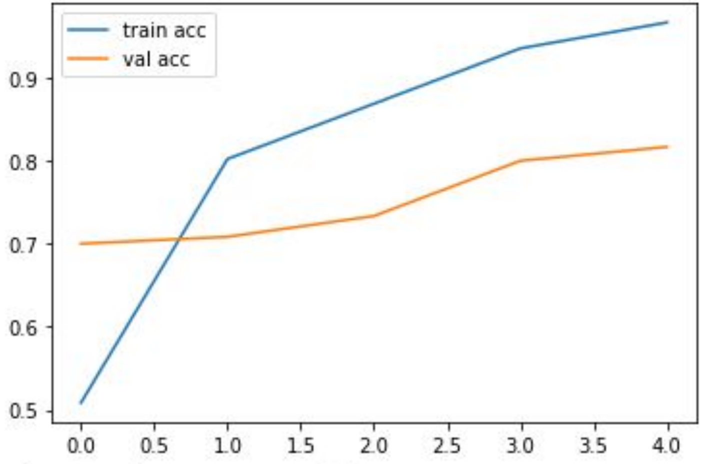
Augmentation

augumentation

```
[ ] from keras.preprocessing.image import ImageDataGenerator

    train_datagen = ImageDataGenerator(rescale = 1./255,
                                       shear_range = 0.2,
                                       zoom_range = 0.2,
                                       horizontal_flip = True)

    test_datagen = ImageDataGenerator(rescale = 1./255)

    training_set = train_datagen.flow_from_directory('/content/drive/MyDrive/food20dataset/train_set',
                                                     target_size = (224, 224),
                                                     batch_size = 32,
                                                     class_mode = 'categorical')

    test_set = test_datagen.flow_from_directory('/content/drive/MyDrive/food20dataset/test_set',
                                                target_size = (224, 224),
                                                batch_size = 32,
                                                class_mode = 'categorical')

    Found 480 images belonging to 6 classes.
    Found 120 images belonging to 6 classes.
```

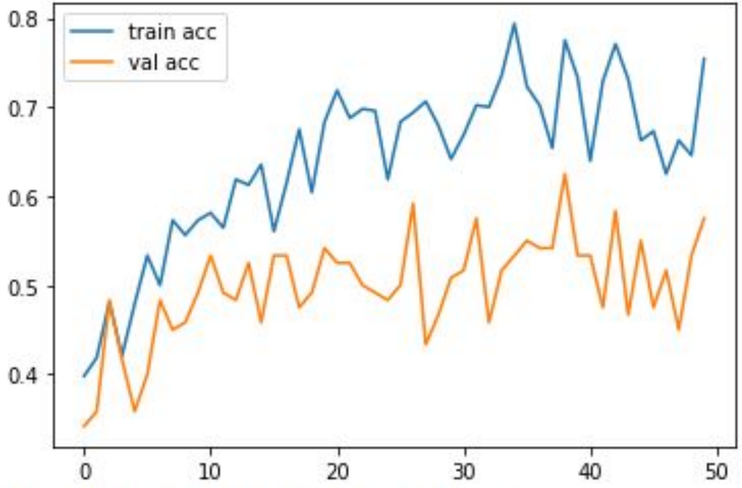
IMPLEMENTATION Phase - 2

Vgg16 VS ResNet:



<Figure size 432x288 with 0 Axes>

accuracy: 0.9667 val_accuracy: 0.8167

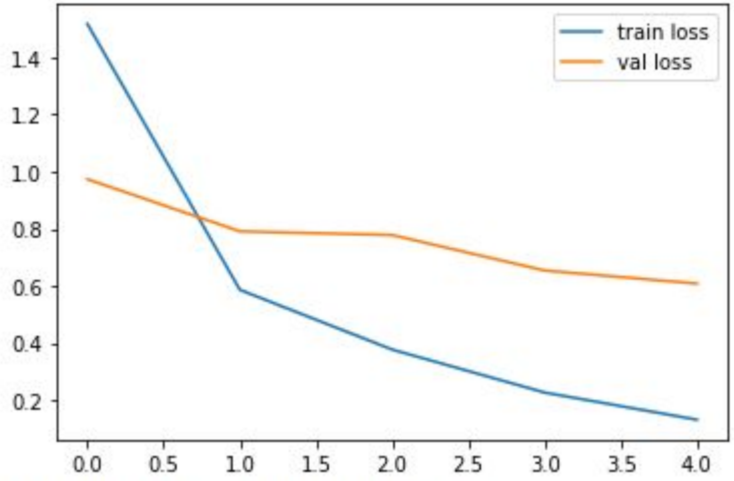


<Figure size 432x288 with 0 Axes>

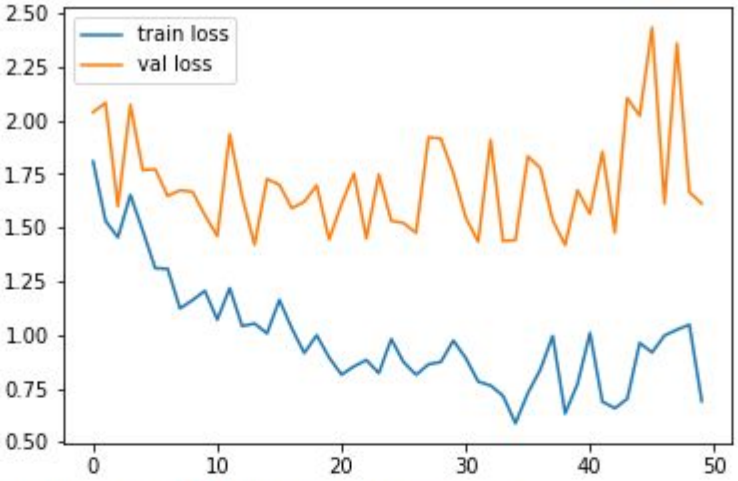
accuracy: 0.7750n val_accuracy: 0.6250

IMPLEMENTATION Phase - 2

Vgg16 VS ResNet:



<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>

loss: 0.1316   val_loss: 0.6082

loss: 0.6337 - val_loss: 1.4191

Predictions: (Output)

```
img = keras.preprocessing.image.load_img(
    "/content/drive/MyDrive/food20dataset/test_set/halwa/halwatest (14).jpg", target_size=IMAGE_SIZE
)
img_array = keras.preprocessing.image.img_to_array(img)
img_array = tf.expand_dims(img_array, 0)  # Create batch axis

predictions = model.predict(img_array)
score = predictions[0]
print(score)
```

```
[0. 0. 0. 0. 1. 0.]
```

Input Image from Test set
Convert image to array
Give this image array as input tp trained model
Model will predict scores on based on predictions.

Predictions: (Output)

```
#saved ingredients of dishes
Biriyani =[ 'rice','mint leaves','salt','refined oil','green cardamom','clove
Chapati = ['whole wheat flour','salt']
Dosa = ['Rice','Urad Dal','Chana Dal','Methi seeds','Water']
Gulab_Jamun =['Maida (All purpose flours)','Sugar','Water']
Halwa = ['Semolina','Ghee','Almonds','Sugar','Cardamom Powder']
Idly = ['Rice','Moong-Dal']
ingredients=[]
ingredients.append(Biriyani)
ingredients.append(Chapati)
ingredients.append(Dosa)
ingredients.append(Gulab_Jamun)
ingredients.append(Halwa)
ingredients.append(Idly)
```

Made list of list containing ingredients of the 6 classes

Predictions: (Output)

```
class_names=['Biriyani','Chapati','Dosa','Gulab_Jamun','Halwa','Idly']

output_class=class_names[np.argmax(score)]
print("The predicted class is: ", output_class)

print("The ingredients are:",ingredients[np.argmax(score)] )
#for i in range(len(output_class)):
  #print(output_class[i])
```

```
The predicted class is:  Halwa
The ingredients are: ['Semolina', 'Ghee', 'Almonds', 'Sugar', 'Cardamom Powder']
```

Successfully predicted the name of the dish and its ingredients.

Future Scope

- We would like to add weekly calorie estimation process to further enhance the health quotient of the user.
- Calories estimation can be improved using volume estimation and other extracted features. A more sophisticated tool for image classification can be developed using more than 6 classes.

REFERENCES:

Indian Food Image Recognition with MobileNetV2
Priya N V1, Preetam Kumari2, Poorvika N3, Sanjana R4, Dr. Hema Jagadish5

Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

Kagaya, Hokuto, Kiyoharu Aizawa, and Makoto Ogawa. "Food detection and recognition using convolutional neural network." In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1085-1088. 2014.

Kaggle,food20 dataset(indian food) | Kaggle

REFERENCES:

Shah, Bhoomi, and Hetal Bhavsar. "Overview of Deep Learning in Food Image Classification for Dietary Assessment System." In *Intelligent Systems, Technologies and Applications*, pp. 265-285. Springer, Singapore, 2021.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

O'Shea, Keiron, and Ryan Nash. "An introduction to convolutional neural networks." *arXiv preprint arXiv:1511.08458* (2015).

https://github.com/krishnaik06/Transfer-Learning

THANKS!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**