### **Food Calorie Estimation using Deep Learning**

### **Abstract**

Calories are essential for maintaining a healthy body. Monitoring calorie intake helps ensure good health. This project introduces an efficient method to identify food items and estimate their calorie content using deep learning. A CNN-based model was developed with TensorFlow to classify and detect food in images, calculating calories based on average values for each food category.

### **Keywords**

Deep Learning, Food Classification, Food Detection, Convolutional Neural Network, Keras, TensorFlow, Hyperparameter Tuning, Pattern Recognition

### **Introduction**

#### **Motivation**

Maintaining a balanced diet is crucial as poor eating habits can lead to diseases. Many people struggle to estimate their calorie intake due to a lack of nutritional information or the tedious process of manually recording it. An automated system to measure calories in meals would greatly help. Recent advancements in deep learning and convolutional neural networks have made object detection much more accurate. In this project, we used a deep learning approach to recognize fruit images and analyze different model architectures for accurate dietary assessment.

### **Background**

Earlier methods for food classification often used machine learning algorithms like Random Forest and SVM. These techniques relied on handcrafted features, which were computationally expensive for large datasets. In comparison, our CNN model developed with Keras and TensorFlow achieved a 97% accuracy. We also experimented with hyperparameter tuning, adjusting factors like learning rate and the number of neurons, to see how they influenced model performance. Additionally, we explored the effects of regularization methods (like L1 and L2 norms), optimizers, and initializers on model training.

### **Algorithm**

This study used classification algorithms such as Random Forest and Support Vector Machines (SVM), along with Convolutional Neural Networks (CNN). For the training dataset, we labeled images using the LabelImage tool and applied transfer learning with the SSD\_MobileNet\_V1\_COCO model weights for eight food categories. TensorFlow's object detection API was used to fine-tune the model. The model identified multiple food items in a single image, counted each detected class, and calculated the total calorie count using predefined calorie values.

### **Methods**

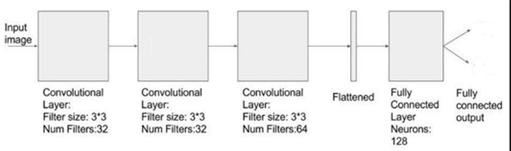
#### **Image Classifier**

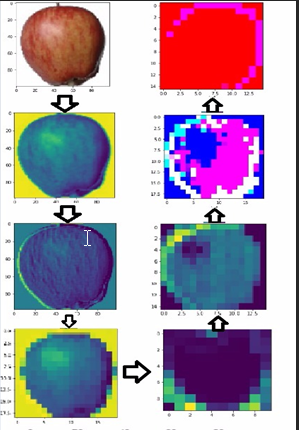
We used several classifiers, including Random Forest, SVM, and CNNs built with TensorFlow and Keras. The Fruit-360 dataset was used for training. The CNN model was built in steps:

1. **Sequential Layers**: A linear stack of layers was created, specifying input shapes and batch sizes.
2. **Convolutional Layers**: Filters performed convolution operations, with filter size being a key parameter.
3. **Pooling**: Max pooling was used to reduce the spatial size and computational load.
4. **Fully Connected Layers**: These layers connected the final outputs from earlier layers.

The CNN architecture used various initializers and activation functions (e.g., ReLU). Results showed that RMSProp optimizer and categorical cross-entropy loss gave the best performance.



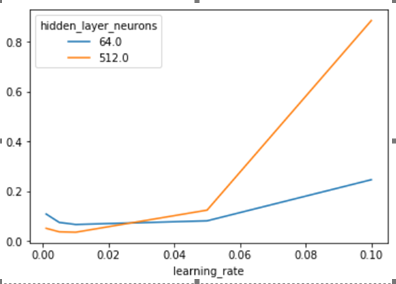




#### **Hyperparameter Tuning**

We adjusted learning rates and the number of neurons in hidden layers. Findings:

* A low learning rate led to slow learning, while a high rate caused instability.
* Increasing the number of neurons had less impact but made models more sensitive to the learning rate.



#### **Multi-Class Object Detection**

A dataset with 500 images across 8 categories was labeled using LabelImage. The labeled data was converted into a format compatible with TensorFlow's object detection API. The model was trained for 1200 epochs, achieving a low training loss and generating a frozen inference graph for final predictions.

For multiple food items in an image, the system calculated calorie totals by summing the values of detected food classes.

### **Results**

Using TensorBoard, we analyzed metrics like learning rate, classification loss, and total loss. Our final system successfully detected food items and calculated total calories for images.

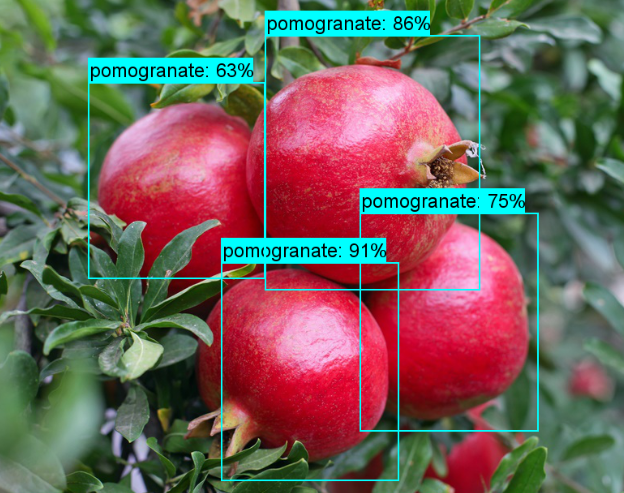
Examples:

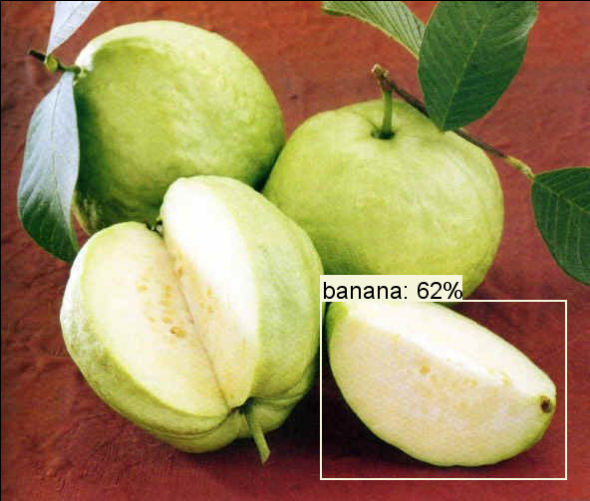
{'pomogranate': 4}

TOTAL CALORIES IN IMAGE:288

{'banana': 1}

TOTAL CALORIES IN IMAGE:105

****

****

### **Conclusion**

We developed a CNN-based system for food detection and calorie estimation using an 8-class dataset. Future improvements could include gathering more data, optimizing hyperparameters, and refining the network architecture to enhance accuracy.