

# Image Classification Project in Python | Deep Learning Neural Network Model Project in Python

A PROJECT REPORT

Submitted by

Murodov Tupatilla

And

Ummatov Tovboy

Under the Guidance of

Tobis Idan

Project proposal

Problem statement

## **1. Preprocessing of Data (Fruits and Vegetables)**

Collecting information: We collect pictures of different fruits and vegetables and divide them into folders by class (for example, apples, bananas, etc.).

### **Preprocessing:**

Resize images to a fixed size.

Normalize pixel values (scale them between 0 and 1).

Use data augmentation methods (rotate, zoom, rotate) to improve generalization.

## **2. Reading the Image Dataset into TensorFlow**

Using libraries like `tf.keras.preprocessing.image` or `TensorFlow's ImageDataGenerator` to load images from directories and convert them into arrays suitable for model training

Split the data set into training, validation and test sets

### 3. Building a Deep Learning Neural Network (Sequential Model)

Defining a sequential model that is a linear collection of layers

Example layers for image classification:

**Convolutional layers** (for feature extraction).

**MaxPooling layers** (for down-sampling).

**Dense (fully connected) layers** (for classification).

**Dropout** and **Batch Normalization** for regularization.

The final layer would typically be a softmax output layer if you're classifying multiple categories.

### 4. Training the Model

Compile the model with a suitable optimizer (e.g. Adam), loss function (categorical\_crossentropy for multi-class classification) and metrics (e.g. accuracy)

Train the model on a pre-processed data set, adjusting hyperparameters such as learning rate, batch size and number of epochs

### 5. Evaluating the Model

After training, evaluate the model's performance on the validation and test datasets. Generate metrics such as accuracy, precision, recall, and confusion matrix to measure how well the model performs.

### 6. Deploying as a Web Application

Export the trained model (using `model.save()` in TensorFlow).

Use a web framework such as **Flask** or **Django** to create a simple interface where users can upload images.

When the image is loaded, preprocess it as during training, transfer it to the trained model and remove the planned class (fruit or vegetable).

#### Tools Used:

**Python** for scripting.

**TensorFlow** for building and training the deep learning model.

**Flask/Django** for deploying the model as a web application.