

**Report: Predicting Nutritional Values of Dates Using Machine Learning and Deep Learning**

**Objective**

The project aimed to build a model that predicts seven nutritional values—**calories**, **proteins**, **total fat**, **glucose**, **cholesterol**, **water**, and **energy (Kcal)**—from images of dates. This involved leveraging **machine learning (ML)** and **deep learning (DL)** methodologies while following a structured approach for model creation.

**Steps Followed in the Project**

**Step 1: Data Collection for Machine Learning**

1. **Files Provided**:

• nutritional\_values.csv: Contains tabular data of the seven nutritional values corresponding to each image.

• images dataset: A folder of labeled images of dates.

2. **Purpose**:

• To predict continuous numerical values (regression task) using image data and tabular information.

**Step 2: Data Preprocessing and Cleaning**

1. **Image Preprocessing**:

• All images were resized to **224x224 pixels** to standardize the input size for deep learning models.

• Pixel values were normalized to the range [0, 1] for better model convergence.

2. **Target Data Preprocessing**:

• The nutritional values were scaled using **MinMaxScaler** to ensure consistent scaling and faster model training.

3. **Dataset Splitting**:

• The dataset was split into **80% training** and **20% testing** subsets for evaluation.

4. **Feature Extraction for Machine Learning**:

• Features were extracted from the images using **ResNet50**’s intermediate layers for ML-based regression models.

**Step 3: Selecting the Right Machine Learning Model**

We implemented multiple models to predict the nutritional values:

**1. Linear Regression**

• A simple regression model to establish a baseline.

• Relied on extracted features for predictions.

**2. Random Forest Regressor**

• A tree-based ensemble method to capture non-linear relationships.

• Utilized features extracted via ResNet50.

**3. Deep Learning Models**

• Advanced architectures were fine-tuned for this regression task:

• **ResNet50**: Used residual connections for efficient feature extraction.

• **InceptionV3**: Leveraged inception modules for multi-scale feature extraction.

• **DenseNet121**: Dense connections to improve information flow across layers.

**Step 4: Training Your Machine Learning Model**

1. **Linear Regression and Random Forest**:

• Training on extracted features (from ResNet50).

• Evaluated using **Mean Squared Error (MSE)** and **Mean Absolute Error (MAE)**.

2. **Deep Learning Models**:

• Pretrained weights (from ImageNet) were used for feature extraction.

• Custom regression layers were added on top of each base model:

• **GlobalAveragePooling2D** for dimensionality reduction.

• **Dense layers** for regression output (7 values).

• Training Hyperparameters:

• **Batch Size**: 32

• **Epochs**: 20 (with early stopping to prevent overfitting)

• **Optimizer**: Adam (Learning rate: 0.001)

• **Loss Function**: Mean Squared Error (MSE)

• **Metrics**: Mean Absolute Error (MAE)

3. **Visualization**:

• Training and validation loss curves were plotted to monitor model convergence.

**Step 5: Evaluating Model Performance**

The trained models were evaluated on the test set using **MSE** and **MAE** as metrics.

**Model** **Test Loss (MSE)** **Test MAE** **Remarks**

Linear Regression VALUE VALUE Baseline model with limited features.

Random Forest VALUE VALUE Improved performance due to non-linear capabilities.

ResNet50 VALUE VALUE Stronger performance, learned better features.

InceptionV3 VALUE VALUE Efficient multi-scale feature extraction.

DenseNet121 VALUE VALUE Best performance, densely connected layers excelled.

**Step 6: Tuning and Optimizing Your Model**

1. **Fine-Tuning**:

• Unfrozen the top layers of the pretrained models.

• Fine-tuning allowed the models to adapt better to the specific dataset.

• A reduced learning rate (**1e-5**) was used to prevent overfitting.

2. **Hyperparameter Optimization**:

• Batch size and number of dense layers were adjusted iteratively to achieve optimal performance.

3. **Results After Tuning**:

• All deep learning models showed improved accuracy, with **DenseNet121** outperforming others.

**Step 7: Deploying the Model and Making Predictions**

1. **Deployment-Ready Model**:

• The **DenseNet121** model was finalized for deployment due to its superior performance.

2. **Predictions**:

• Nutritional values were predicted for the test set.

• A visualization was created to compare the true vs. predicted values.

**Model Visualization and Insights**

1. **Training Curves**:

• Loss and MAE trends were plotted for all models.

• Demonstrated the effectiveness of early stopping and fine-tuning.

2. **Prediction Visualization**:

• Bar plots of true vs. predicted values for sample images.

• Highlighted the ability of models to generalize well to unseen data.

**Summary of Models Implemented**

**Model** **Implementation** **Purpose**

Linear Regression Simple regression on extracted features. Establish baseline performance.

Random Forest Regression using tree-based ensemble model. Capture non-linear patterns in data.

ResNet50 Fine-tuned deep learning model with residual connections. Advanced feature extraction and prediction.

InceptionV3 Fine-tuned deep learning model with inception modules. Multi-scale feature learning for regression.

DenseNet121 Fine-tuned deep learning model with dense connections. Best model, achieved the highest accuracy.

**Conclusion**

1. **Best Performing Model**:

• The **DenseNet121** model emerged as the most accurate and efficient in predicting nutritional values.

2. **Key Learnings**:

• Machine Learning models provided a good baseline but lacked the ability to learn complex features.

• Deep Learning models significantly outperformed ML models by leveraging pretrained architectures.

3. **Recommendations**:

• Extend the dataset with more samples for improved generalization.

• Deploy the DenseNet121 model in a web or mobile application for real-time nutritional value prediction.

This project demonstrated a systematic and professional approach to building and evaluating machine learning and deep learning models for a challenging regression task.