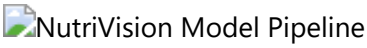
NutriVision: A Multi-Modal Food Nutrition Estimator and Meal Tracker



NutriVision is an end-to-end multi-modal system designed to estimate the nutritional content of meals from images and track daily calorie intake. The system leverages state-of-the-art computer vision and natural language processing models to analyze food images and provide detailed nutritional information. In addition, NutriVision offers a full-featured web application built with Streamlit, including user authentication, meal tracking, statistics, and an intuitive dashboard.

Table of Contents

Overview

NutriVision is designed to help users gain insights into their dietary habits by estimating the nutritional content (e.g., calories, fat, carbohydrates) of food items from images. The system supports multiple data sources:

- **Food Nutrition Dataset**: Aggregated nutritional information from multiple CSV files.
- **Fruits & Vegetables Nutritional Values**: Detailed nutritional data on produce.
- **Fast Food Nutrition Dataset**: Nutritional information from popular fast food items.
- **Food Image Datasets**: Food-101, Vegetable Image Dataset, and Fruit & Vegetable Image Recognition Dataset.

The web application enables users to:

- Register and log in with additional health information (height, weight, age, gender, and profile picture).
- Upload meal images to get nutritional inference.
- View a dashboard with daily calorie statistics and interactive graphs.
- Track their meal history, complete with images, captions, and predicted nutritional values.
- Manage their account details.

Features

- **Multi-Modal Inference**: Combines a BLIP-based caption generator with CLIP image and text encoders.
- **Custom Regression Heads**: Different prediction heads for Food Nutrition, Fruits & Vegetables, and Fast Food data.
- **User Authentication**: Registration and login with additional health tracking fields.
- **Meal Tracking**: Record meals with images, inferred nutritional details, and timestamps.
- **Dashboard & Analytics**: Interactive graphs displaying daily calorie intake and historical data.
- **Streamlit Web Application**: A professional, responsive UI with top navigation and a sky-blue/white theme.
- **Local SQLite Database**: Stores user credentials and meal records for persistent tracking.

Model Architecture & Training

Architecture

NutriVision's multi-modal model integrates the following components:

1. **BLIP Caption Generator**
    - **Model**: Salesforce BLIP Image Captioning Base
    - **Role**: Generates descriptive captions from input images (frozen during training).
2. **CLIP Encoder**
    - **Model**: OpenAI CLIP ViT-B/32
    - **Image Encoder**: Produces a 512-dimensional embedding from the input image.
    - **Text Encoder**: Produces a 512-dimensional embedding from the BLIP-generated caption (frozen).
    - **Fusion**: The image and text embeddings are concatenated into a 1024-dimensional feature vector.
3. **Multi-Head Regression**
    - **Heads**: Three separate regression heads for different nutrition sources:
        - **Food Nutrition Head**: Predicts values (e.g., calories, fat, carbohydrates) using a 3-dimensional output.
        - **Fruits & Vegetables Head**: Uses a 9-dimensional output.
        - **Fast Food Head**: Uses an 8-dimensional output.
    - **Training Loss**: Mean Squared Error (MSE) is used to train the regression outputs.

Training Methodology

- **Datasets**:
    - **Food Nutrition**: CSV files (FOOD-DATA-GROUP1.csv to FOOD-DATA-GROUP5.csv) are combined, grouped by food name (converted to lowercase), and averaged.
    - **Fruits & Vegetables**: Nutritional data is preprocessed using a helper function to clean and convert numerical fields.
    - **Fast Food**: CSV data is aggregated similarly.

- **Food Images**: Food-101 and additional image datasets (Vegetable Image Dataset and Fruit & Vegetable Image Recognition) are used to map images to nutrition mappings.

- **Training Loop**:
  The model is trained using a custom PyTorch training loop that:
  - Loads images from a unified multi-source dataset.
  - Performs inference to generate captions and embeddings.
  - Routes the fused embeddings through the appropriate regression head based on the nutrition source.
  - Computes the MSE loss and updates model weights.

- **Checkpointing**:
  The trained model is saved as `nutrivision_multihand.pt` for later inference in the web application.

Tech Stack

- **Programming Language**: Python 3.8+
- **Deep Learning Framework**: PyTorch
- **Pretrained Models**:
  - BLIP (Salesforce) for image captioning
  - CLIP (OpenAI) for image and text embeddings
- **Web Framework**: Streamlit
- **Database**: SQLite (for user and meal data storage)
- **Visualization**: Plotly (for interactive graphs)
- **Additional Libraries**: Transformers, Pillow, tqdm

Streamlit Deployment

The web application is built using Streamlit and offers the following functionalities:

- **User Authentication**: Secure login and registration forms.
- **Meal Upload & Analysis**: Users upload meal images, which are processed using the NutriVision model.
- **Dashboard**: Visual metrics and interactive graphs (e.g., daily calorie intake over time).
- **Meal History**: Displays a detailed history of meals with images, captions, and nutritional details.
- **Account Management**: Shows user profile details and health information.

Deployment Instructions

1. **Local Deployment**:
   - Install dependencies:

```
pip install streamlit torch torchvision transformers Pillow plotly
tqdm
```

- Place the model checkpoint file `nutrivision_multihand.pt` in the root directory.
- Run the app:

```
streamlit run app.py
```

Database Schema

`users` Table

- **id**: INTEGER, Primary Key
- **username**: TEXT, Unique
- **password**: TEXT
- **height**: REAL (in centimeters)
- **weight**: REAL (in kilograms)
- **age**: INTEGER
- **gender**: TEXT
- **profile_pic**: TEXT (file path to profile picture)

`meals` Table

- **id**: INTEGER, Primary Key
- **user_id**: INTEGER, Foreign Key to `users.id`
- **meal_time**: TIMESTAMP
- **source**: TEXT (nutrition category: food_nutrition, fv, fastfood)
- **caption**: TEXT (generated caption from BLIP)
- **predicted**: TEXT (JSON string of predicted nutritional values)
- **calories**: REAL (predicted calories)
- **meal_image**: TEXT (file path to the uploaded meal image)

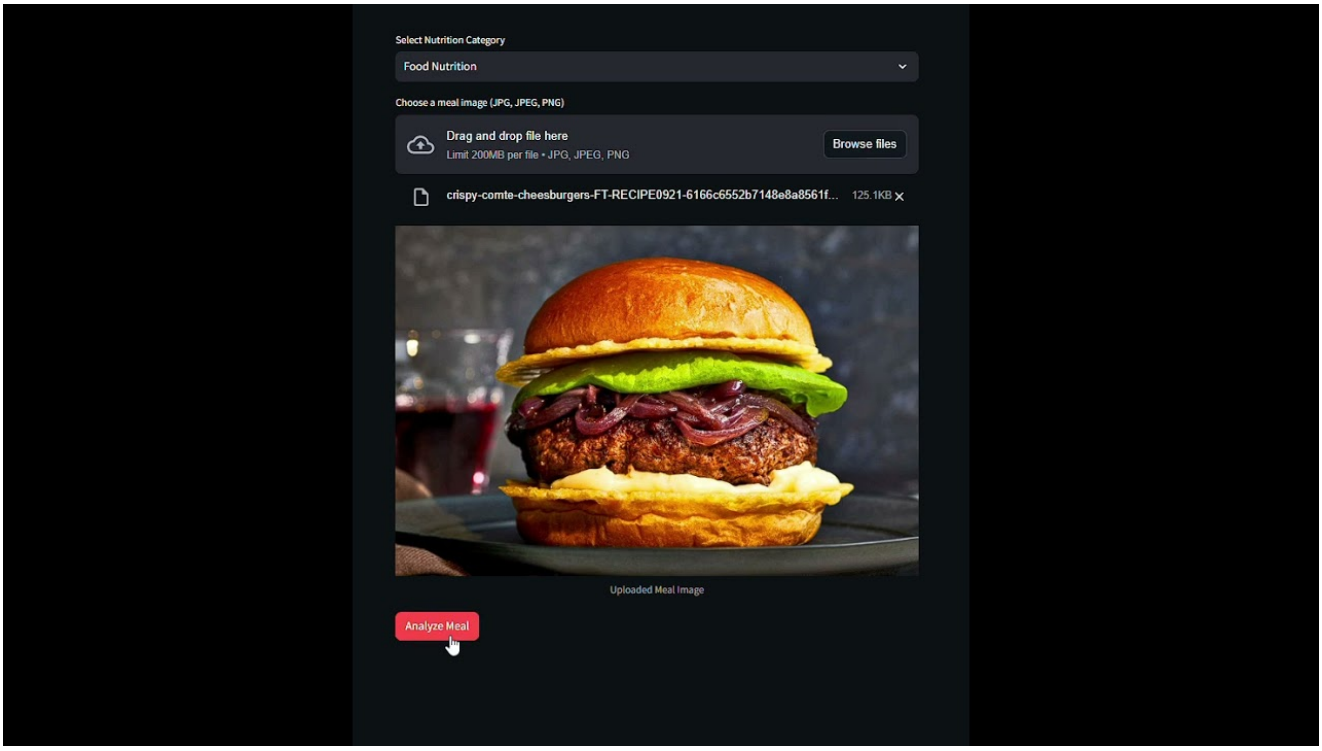
Setup & Installation

1. **Clone the Repository**

```
git clone https://github.com/yourusername/nutrivision.git
cd nutrivision
```

Watch the Working Video

For a quick overview of NutriVision's features and functionalities, watch the



Working Site

You can access the live NutriVision web application at the following link: NutriVision Live