THE UNIVERSITY OF DANANG

VNUK – INSTITUTE FOR RESEARCH AND EXECUTIVE EDUCATION

Challenge 1.

Vietnamese Food Classification & Web Deployment

This report is submitted to the *Information Communication and Technology* for the *Artificial Intelligence* Module

Kien Hoang Mai Duc – 23020015

CSB35037: Artificial Intelligence

Term 1 – AY 2025-2026

Dr. Tran The Vu

October 29, 2025

Table of Contents

Executive Summary

This project details the development and deployment of a Vietnamese food recognition and classification system using Deep Learning and Transfer Learning. The system was designed to meet two core assignment requirements: robust model training (60%) and full web deployment (40%). The model, built upon the **MobileNetV2** architecture using the PyTorch framework, was successfully trained to classify three specific Vietnamese dishes, achieving a quantified validation accuracy of **95.44%**. The application was deployed via **Streamlit** and secured the bonus criterion by integrating the **Google Gemini API** to provide rich, contextual descriptions and ingredient details for the identified meal, moving beyond simple classification to deliver a comprehensive information tool.

Keywords: Vietnamese Food Classification, Deep Learning, Transfer Learning, MobileNetV2, PyTorch, Streamlit, Google Gemini API, Image Recognition, Model Deployment

I. System Requirements and Architectural Overview

The project design was driven by the weighted criteria of the assignment, focusing on separating the model's intelligence (training) from its accessibility (deployment).

1.1. System Requirements and Criteria Mapping

No.	Criteria	Weights (%)	Implementation Detail	Status
1	Train the food model	60%	MobileNetV2 fine-tuned on Pho, Banh Mi, and Banh Cuon.	Completed
2	Deploy the model	40%	Streamlit web application deployment (`main.py`).	Completed
Bonus	Recognize Ingredients	-	Integration of Google Gemini API for contextual enrichment.	Achieved

1.2. Architectural Overview

The system employs a serverless, hybrid vision-and-generation architecture. The core classification is handled locally by the loaded PyTorch model, and the subsequent descriptive analysis is outsourced to the powerful Gemini API.

Data Flow:

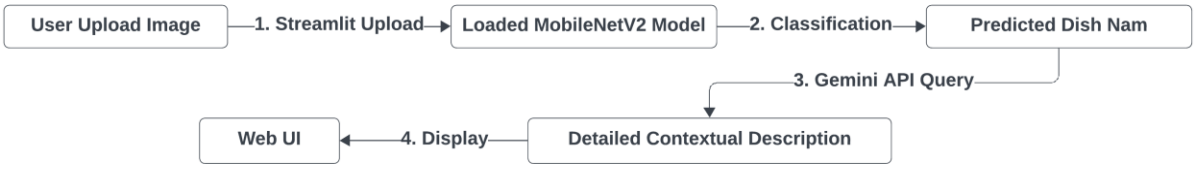

Figure 1. Data Flow

II. Deep Learning Model Training (60%)

The training phase was meticulously documented in the `vietfood-classifier.ipynb` notebook, detailing the steps from dataset preparation to final performance metrics.

2.1. Dataset Acquisition and Preparation

The dataset used for training was sourced from the **30VNFOODS** from Kaggle. This initial dataset covers 30 distinct Vietnamese dishes.

Selected Subset: To ensure high-quality training focus and meet the minimum requirement, the dataset was filtered to include only three major classes:

1. Pho (Vietnamese Beef Noodle Soup)
2. Banh Mi (Vietnamese Sandwich)
3. Banh Cuon (Vietnamese Steamed Rice Rolls)

Data Augmentation: The training process utilized extensive data augmentation techniques (as detailed in the training notebook) to improve generalization and prevent overfitting. These included:

- Random Resizing to (224, 224) pixels.
- Random Rotation up to 20^{o}.
- Random Horizontal Flipping
- Normalization using ImageNet statistics.

2.2. Methodology: Transfer Learning and Fine-Tuning

***2.2.1. Framework*:** The training was conducted using **PyTorch**, leveraging its high-level API for efficient deep learning implementation.

***2.2.2.Transfer Learning*: MobileNetV2** was chosen as the base architecture. This model is ideal for deployment due to its small size and efficient execution.

1. The pre-trained weights from the ImageNet benchmark were loaded.
2. All convolutional layers of MobileNetV2 were initially frozen (weights were not updated).

3. The original classification head was replaced with a new custom head suitable for a 3-class problem.
4. Fine-Tuning: After initial training, the final layers of the MobileNetV2 block were unfrozen and the model was trained with a very low learning rate. This process adjusted the pre-trained features to be highly specific to the visual characteristics of Vietnamese cuisine.

2.3. Model Structure

The final classification layers of the fine-tuned MobileNetV2 model architecture are described as follows:

```
self.classifier = nn.Sequential(
    nn.AdaptiveAvgPool2d((1, 1)),
    nn.Flatten(),
    nn.Linear(base_model.last_channel, 128),
    nn.ReLU(),
    nn.Dropout(0.3),
    nn.Linear(128, 3),
    nn.Softmax(dim=1)
)
```

2.4. Evaluation and Performance

The model's performance was evaluated on a dedicated test set of 658 samples, achieving high metrics across all categories.

Overall Accuracy: The final validation accuracy was recorded at **0.9544**.

Classification Report (F1-Score Analysis):

Dish Class	Precision	Recall	F1-Score	Support
Banh Cuon	0.927	0.9474	0.9371	228
Banh Mi	0.9844	0.9403	0.9618	268
Pho	0.9467	0.9877	0.9668	162
Weighted Avg	0.9552	0.9544	0.9545	658

The F1-scores, particularly the high scores for Banh Mi and Pho, demonstrate the model's excellent capacity for balanced and reliable classification across the target dishes.

III. Deployment and Application System (40% Criterion)

The deployment phase on translating the trained PyTorch model into a simple, accessible, and interactive web application.

3.1. Application Framework and Features

The application is built using **Streamlit** (requiring the `streamlit` package), which allows the entire user interface, logic, and model interaction to reside within a single Python script (`main.py`).

Key Features Implemented

- **Image Uploader:** Allow users to select images of food for classification.
- **Real-time Prediction**: Instantaneous output of the predicted dish name and confidence score.
- **Contextual Enrichment (Gemini API):** An API call is executed using the `google-genai` libraries to fetch a rich description (ingredients, history) based on the model's prediction, fulfilling the bonus requirements.

3.2. Public Deployment (Streamlit Cloud)

The application code (`main.py` and dependencies) was hosted on GitHub repository and automatically deployed using **Streamlit Cloud**. This provides a persistent, easily accessible live environment, eliminating the need for local setup to test the core functionality.

***Live Application URL*:** https://vnfood-hmdk.streamlit.app/

3.2. Application Usage and Operation Flow

The application offers two primary modes of operation: local development and public access.

A. Public Access (Deployed App)

Users can simply navigate to the provided URL to upload an image and receive instant classification and contextual information from the Gemini API.

B. Local Development and Reproduction

The development environment is reproducible using the following steps:

1. **Install Requirements:** The dependencies (including `torch`, `streamlit`, `google-genai`, etc.) must be installed from the provided requirements.txt file.

   ```
   pip install -r requirements.txt
   ```

2. **API Key Setup:** The Google Gemini API is loaded via `credentials.py` or environment variables via `python-dotenv` for the contextual description feature. The key must be set as an environment variable (`GEMINI_API_KEY`) or within a `.streamlit/secrets.toml` file.

3. **Run Locally:** The application is run from the terminal using the following command:

   ```
   streamlit run main.py
   ```

4. **Access the Web Interface:** The terminal will output a local network address (e.g., `http://localhost:8501`). The user opens this link in a web browser to access the "🍜 Vietnamese Food Recognizer" interface.

3.3. Web Application Interface

The web interface provides a clean, centered design with custom CSS for improved aesthetic appeal.

Figure 2. Web Interface

IV. Conclusion and Future Deployment

The project successfully delivered a robust and efficient food recognition system, validated by a 95.44% accuracy on the test set. By combining the efficiency of a fine-tuned MobileNetV2 with the contextual power of the Gemini API, the system delivers both a core computer vision solution and an enriched user experience.

Future work should focus on the following enhancements:

- **Dataset Scaling:** Expanding the dataset to include a wider range of Vietnamese cuisine categories to increase the model's overall utility and breadth of knowledge.
- **Interpretability (Grad-CAM):** Integrating visualization techniques like Grad-CAM to provide visual proof of the model's decision-making process, highlighting the features in the image that led to the classification.
- **Batch Processing:** Implementing a feature to allow users to classify multiple images in a single session.
- **User Feedback Loop:** Implementing a mechanism within the Streamlit app to collect user feedback on predictions to continuously improve model accuracy in a production environment.