Sneaker Image Classification Project: An End-to-End Deep Learning Assignment

A comprehensive machine learning project focused on classifying popular sneaker models from Nike and Adidas using deep learning techniques.

Problem Statement: Mastering Visual Classification of Iconic Sneakers

In the rapidly evolving world of e-commerce and fashion technology, the ability to accurately identify and categorize products from images is a crucial skill. This project challenges you to develop a comprehensive machine learning solution for classifying sneaker images. You are provided with a rich dataset of images featuring popular sneaker models from leading brands like Adidas and Nike. Your goal is to build an robust, end-to-end system that can automatically classify these sneakers, demonstrating proficiency in data exploration, model development, evaluation, and deployment. This assignment emphasizes practical application of deep learning techniques, aiming to equip you with the skills to tackle real-world visual recognition problems.

Dataset Overview: A Diverse Collection of Sneaker Imagery

The dataset, titled "Popular Sneakers Classification," comprises a substantial collection of thousands of sneaker images. These images are meticulously categorized into 10 distinct classes, primarily focusing on iconic models from Nike and Adidas. The diversity within the dataset is a key characteristic, with images varying significantly in:

- Resolution: Images are of different sizes, requiring careful preprocessing.
- Angle: Sneakers are captured from various viewpoints, enhancing the model's ability to generalize.
- Background: The images feature diverse backgrounds, promoting learning of intrinsic sneaker features rather than environmental cues.

This heterogeneity ensures that your developed model will be robust and perform well in varied real-world scenarios.

Key Class Labels

The 10 classes you will be working with are:

- adidas_nmd
- adidas_ultraboost
- adidas_yeezy
- nike_air_force_1
- nike_air_max
- nike_air_max_97
- nike_air_max_270
- nike_air_max_720
- nike_air_more_uptempo
- nike_air_vapormax

Each class is represented by a dedicated folder containing numerous images of the corresponding sneaker type.

The dataset is publicly available on Kaggle: https://www.kaggle.com/datasets/nikolasgegenava/sneakers-classification

Project Objectives & Deliverables

Your project should systematically address the following phases, with clear deliverables for each:

Dataset Exploration (EDA)

Analyze the dataset structure, class distribution, image properties, and identify potential issues.

Image Classification Model Development

Build, train and optimize a CNN model for sneaker classification.

Model Evaluation

Assess model performance through various metrics and visualizations.

Streamlit Web Application

Create an interactive interface for real-time sneaker classification.

Dataset Exploration (EDA)

Quantify images per class

Identify potential class imbalances in the dataset.

Check for missing or corrupt files

Implement robust checks and devise handling strategies.

Visualize class distribution

Create informative bar plots showing the distribution of images across classes.

Analyze image sizes

Visualize the distribution of image dimensions and aspect ratios to inform preprocessing.

Identify image formats

Determine and visualize the types of image formats present in the dataset.

Deliverable: A well-commented Jupyter Notebook (EDA.ipynb) presenting your visuals, detailed observations, and insights gained during the exploration phase.

Image Classification Model Development

Build, compile, and train a Convolutional Neural Network (CNN) capable of accurately classifying sneaker images.

Data Preparation

- Efficiently load images using TensorFlow's tf.data API or PyTorch's DataLoader
- Implement preprocessing: resizing, normalization, and data augmentation
- Split into training (80%) and validation (20%) sets

Model Architecture

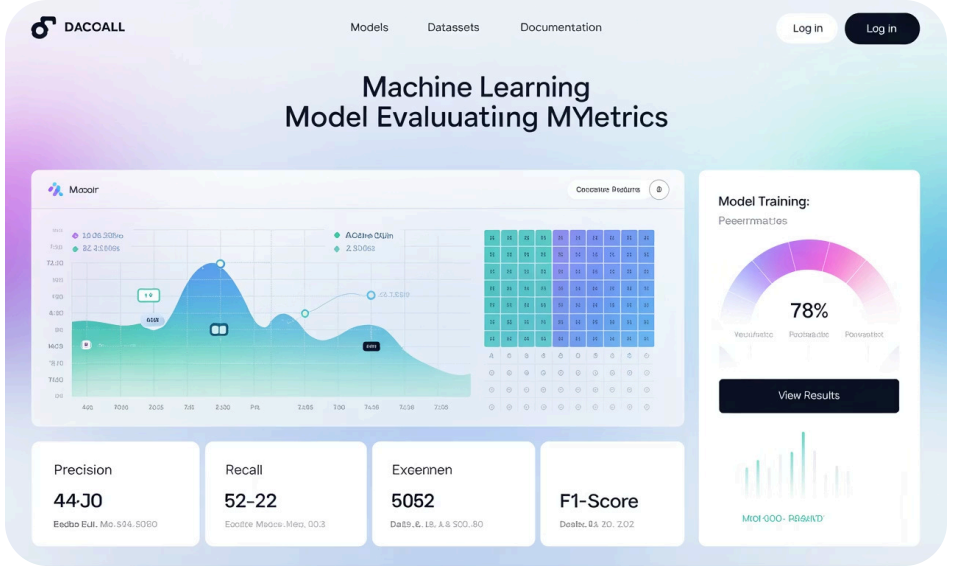
- Design a CNN architecture suitable for image classification
- Consider different layer types and activation functions
- Implement the model using TensorFlow/Keras or PyTorch

Model Training

- Compile with appropriate optimizer, loss function, and metrics
- Train the model while monitoring performance
- Save the trained model (.h5 for TensorFlow/Keras or .pth for PyTorch)

Deliverable: Jupyter Notebook (model_training.ipynb) containing the complete model building and training code, along with the saved model file (sneaker_classifier.h5 or sneaker_classifier.pth).

Model Evaluation

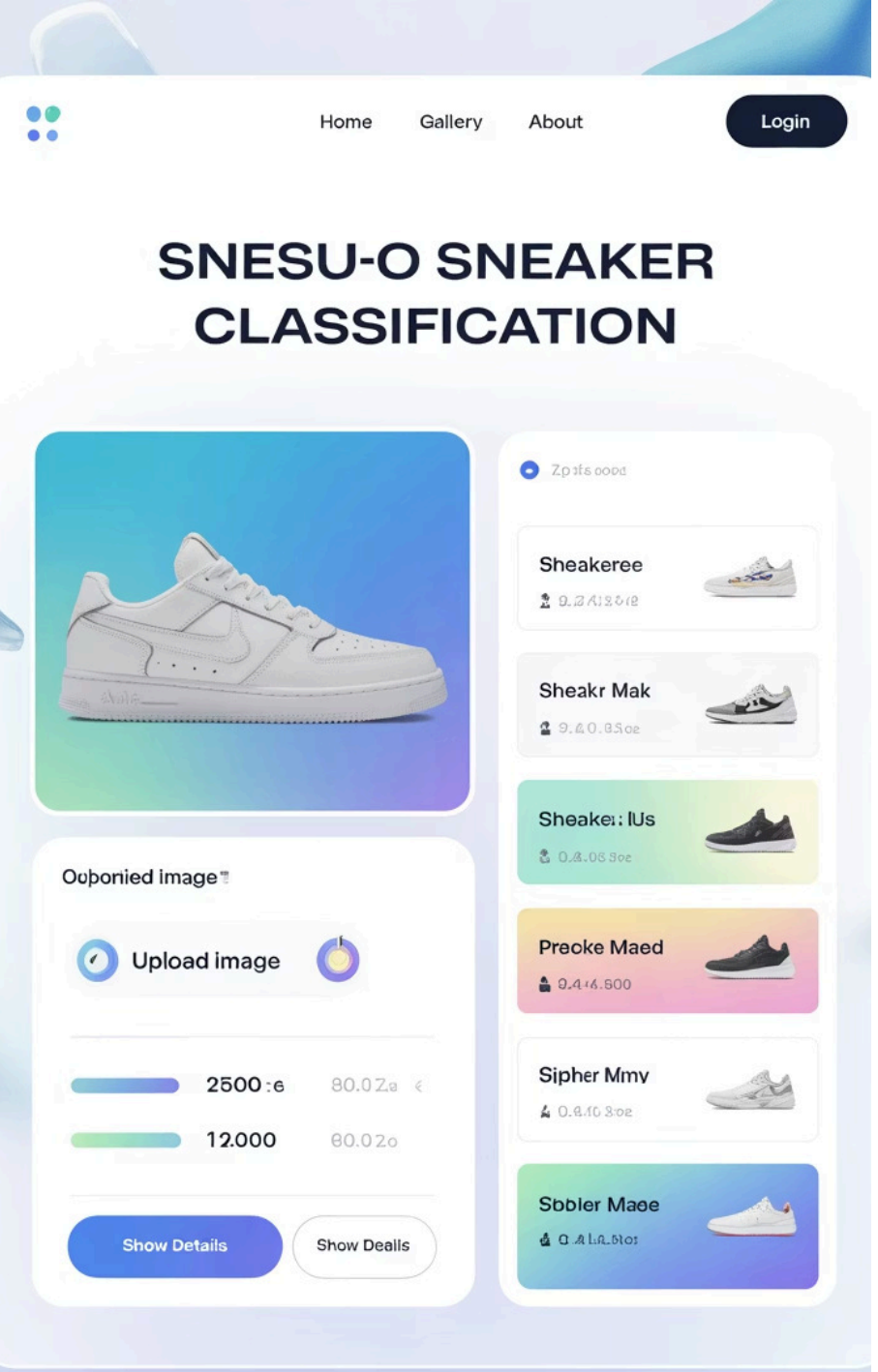


Rigorously evaluate the performance of your trained model and summarize its classification capabilities.

Key Activities:

- Generate and plot training and validation accuracy/loss curves
- Calculate and report overall accuracy on validation dataset
- *(Optional but Recommended):* Compute and visualize a confusion matrix
- *(Optional):* Calculate precision, recall, and F1-score for each class

Deliverable: A section within your model_training.ipynb (or a separate notebook model_evaluation.ipynb) showcasing the evaluation plots and a concise summary of the model's performance metrics.



Streamlit Web Application

Develop an interactive web application using Streamlit that allows users to upload sneaker images and receive real-time classification predictions.

User Interface

Build a clean interface where users can easily upload a single sneaker image for classification.

Model Integration

Incorporate your trained classification model to perform inference on the uploaded image.

Results Display

Clearly show the predicted class label (e.g., "Nike Air Max 97") and display probability scores for all 10 classes.

Deliverable: A single Python script (app.py) that runs the Streamlit web application, including all necessary imports and model loading logic.

Submission Checklist

To ensure a complete and successful submission, please include the following files in a well-organized directory:

Jupyter Notebooks

- EDA.ipynb (for dataset exploration and visuals)
- model_training.ipynb (for model building, training, and evaluation)

Model & Application Files

- sneaker_classifier.h5 (TensorFlow/Keras) OR sneaker_classifier.pth (PyTorch)
- app.py (Streamlit application)

Documentation

- README.md (Instructions on environment setup, running notebooks, and launching the Streamlit app)
- *(Optional but Recommended):* requirements.txt (List of Python libraries and versions)