DJS240003
Dhruv Shelke

CS 6375 – Machine Learning – Project 3 Report

Mushroom Classification Using CNN and SVM with Custom Feature Extraction

Introduction

This project implements a deep learning-based mushroom classification system, aiming to classify mushroom images into 9 distinct classes: *Agaricus, Amanita, Boletus, Cortinarius, Entoloma, Hygrocybe, Lactarius, Russula, and Suillus.* Unlike traditional transfer learning using pre-trained models, this project explores a custom convolutional neural network (CNN) architecture along with an SVM classifier built on top of features extracted from the CNN.

Dataset

- Source: Kaggle mushroom image dataset
- Structure: Each mushroom class is represented by a folder containing corresponding images
- Preprocessing:
    - Images are resized to 160x160
    - Normalized pixel values to [0, 1] range
    - Labels encoded as integers (0–8)

Approach

1. CNN-Based Classifier (Softmax Model)

- **Architecture**:
    - Input → Conv2D → BatchNorm → LeakyReLU → MaxPooling
    - Stacked Conv Blocks with Dropout & L2 Regularization
    - Flatten → Dense → Softmax
- **Training Setup**:
    - Loss: categorical_crossentropy
    - Optimizer: Adam
    - Metrics: Accuracy
    - Regularization: Dropout, L2
    - Callbacks: EarlyStopping, ReduceLROnPlateau
    - Augmentation via ImageDataGenerator for improved generalization
- **Output**: model1_softmax.h5

2. SVM Classifier (Feature Extraction)

- After training the CNN model, the penultimate dense layer (before Softmax) is used to extract features from images.
- Features are collected for the training dataset and used to train an **SVM with a linear kernel** using scikit-learn.
- The model is saved as svm_model.joblib.

Testing Pipeline

Implemented in proj3_test.py:

- Supports both --model_type cnn and --model_type svm
- Accepts a test CSV (mushrooms_test.csv) with image_path and label columns
- Evaluates performance using:
    - .evaluate() for CNN
    - predict() and accuracy_score for SVM

Testing Command Lines:

1) python proj3_test.py --model_type cnn --model model1_softmax.h5 --test_csv mushrooms_test.csv
2) python proj3_test.py --model_type svm --model model1_softmax.h5 --svm_model svm_model.joblib --test_csv mushrooms_test.csv

Results

Model Type	**Test Accuracy**
CNN (Softmax)	~53.88%
SVM + CNN Feat	~66.67%

Conclusion

This project demonstrates how a custom CNN can serve as both a classifier and a feature extractor for downstream tasks like SVM-based classification. While the CNN provided an end-to-end solution, the SVM pipeline showed promise in terms of interpretability and modularity. The architecture and methodology were chosen to **avoid using pre-trained models**, instead focusing on original model design and training from scratch.