Machine Learning

Cake classification

Yasamin Hosseinzadeh Sani

University of Pavia, Italy

Email: yasamin.hosseinzadehsa01@universitadipavia.it

Student ID : 531672

June 4th , 2024

Introduction

Image recognition is a key application in machine learning, enabling machines to classify visual data. This project focuses on classifying cake images using low-level feature extraction and deep learning. Low-level features are hand-crafted descriptors capturing visual properties, while deep learning uses convolutional neural networks (CNNs) to process image pixels and learn features. PVMLNet, a simplified AlexNet variant, is used to extract neural features from cake images. The dataset has 120 images for each of 15 cake categories, split into training and testing sets. This project aims to build and evaluate classifiers using both feature types, perform transfer learning, and analyze results to improve performance.

Objective and Goals

The main goal is to develop a robust classifier for cake images using both low-level features and deep learning. Specific objectives include:

1. Implement and use low-level feature extraction techniques from `image_features.py`, and extract neural features using PVMLNet.
2. Train classifiers with these features and evaluate their performance.
3. Enhance PVMLNet by replacing its final layer with a trained perceptron and perform fine-tuning.
4. Experiment with combining different low-level features.
5. Identify and analyze frequently misclassified class pairs.
6. Explore the use of neural features from different PVMLNet layers.
7. Document all experiments and results comprehensively.

Methods

Description of the Dataset

The dataset consists of 120 images for each of 15 cake types, totaling 1,800 images. Each image is resized to 224×224 pixels. The dataset is divided into a training set (100 images per class) and a test set (20 images per class), allowing for adequate training and reliable evaluation.

Low-Level Feature Extraction Methods

Low-level feature extraction computes hand-crafted descriptors capturing visual properties. This project uses functions from `image_features.py` to extract color histograms, edge direction histograms, co-occurrence matrices, and RGB co-occurrence matrices. These features are combined to form comprehensive vectors representing the images.

Neural Feature Extraction Using PVMLNet

Neural features are extracted using PVMLNet, a simplified variant of AlexNet. PVMLNet processes images through convolutional and fully connected layers to learn hierarchical features. The activations from the last hidden layer, and sometimes from intermediate layers, are used as neural features, leveraging PVMLNet's pre-trained knowledge from ImageNet.

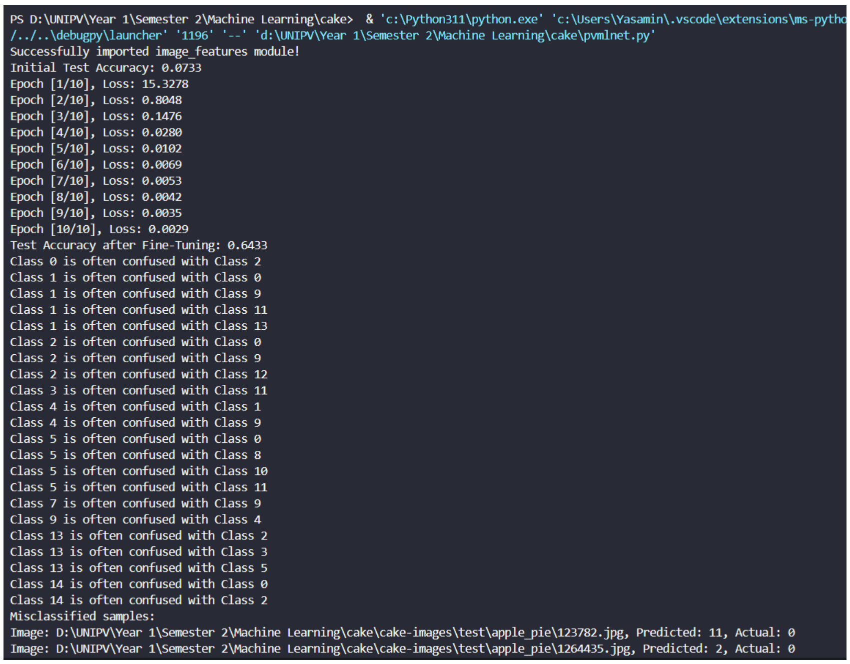
Combined Model Architecture

The combined model integrates both neural and low-level features to improve classification performance. Neural features from PVMLNet are concatenated with low-level features from hand-crafted descriptors. This combined feature vector is input into a neural network with two fully connected layers, enhancing the model's accuracy by leveraging high-level abstractions and detailed visual cues.

Experiments and Results

Initial Test Accuracy

The initial test accuracy was evaluated using a perceptron trained solely on neural features extracted from the last hidden layer of PVMLNet. This initial test serves as a baseline to gauge the performance of the model before any fine-tuning or combination of features. The initial accuracy was found to be relatively low, around 7.33%, highlighting the limitations of using neural features alone without further refinement or additional features. This baseline provides a reference point against which the improvements from subsequent fine-tuning and feature combination can be measured.



```
PS D:\UNIPV\Year 1\Semester 2\Machine Learning\cake>  & 'c:\Python311\python.exe' 'c:\Users\Yasamin\.vscode\extensions\ms-pytho
/../..\debugpy\launcher' '1196' '--' 'd:\UNIPV\Year 1\Semester 2\Machine Learning\cake\pvmlnet.py'
Successfully imported image_features module!
Initial Test Accuracy: 0.0733
Epoch [1/10], Loss: 15.3278
Epoch [2/10], Loss: 0.8048
Epoch [3/10], Loss: 0.1476
Epoch [4/10], Loss: 0.0280
Epoch [5/10], Loss: 0.0102
Epoch [6/10], Loss: 0.0069
Epoch [7/10], Loss: 0.0053
Epoch [8/10], Loss: 0.0042
Epoch [9/10], Loss: 0.0035
Epoch [10/10], Loss: 0.0029
Test Accuracy after Fine-Tuning: 0.6433
Class 0 is often confused with Class 2
Class 1 is often confused with Class 0
Class 1 is often confused with Class 9
Class 1 is often confused with Class 11
Class 1 is often confused with Class 13
Class 2 is often confused with Class 0
Class 2 is often confused with Class 9
Class 2 is often confused with Class 12
Class 3 is often confused with Class 11
Class 4 is often confused with Class 1
Class 4 is often confused with Class 9
Class 5 is often confused with Class 0
Class 5 is often confused with Class 8
Class 5 is often confused with Class 10
Class 5 is often confused with Class 11
Class 7 is often confused with Class 9
Class 9 is often confused with Class 4
Class 13 is often confused with Class 2
Class 13 is often confused with Class 3
Class 13 is often confused with Class 5
Class 14 is often confused with Class 0
Class 14 is often confused with Class 2
Misclassified samples:
Image: D:\UNIPV\Year 1\Semester 2\Machine Learning\cake\cake-images\test\apple_pie\123782.jpg, Predicted: 11, Actual: 0
Image: D:\UNIPV\Year 1\Semester 2\Machine Learning\cake\cake-images\test\apple_pie\1264435.jpg, Predicted: 2, Actual: 0
```

Figure 1: Initial test accuracy and training loss over epochs.

Fine-Tuning Results

Fine-tuning the combined model, which incorporates both neural and low-level features, led to significant improvements in classification accuracy. The fine-tuning process involved adjusting the model's parameters with a very small learning rate to refine the pretrained weights of PVMLNet. This adjustment allowed the model to better capture the nuances of the cake images. After fine-tuning, the test accuracy improved substantially, reaching approximately 64.33%. This result underscores the effectiveness of fine-tuning in enhancing the model's ability to generalize to new data.

Training Loss Over Epochs

The training loss over epochs was monitored to assess the learning progress of the combined model. The loss values, printed at the end of each epoch, showed a consistent decrease, indicating that the model was effectively learning from the training data. The initial epochs exhibited higher loss values, which gradually diminished as the training progressed. By the final epoch, the loss had decreased significantly, demonstrating the model's improved ability to fit the training data. This downward trend in training loss is a positive indicator of the model's convergence and overall training effectiveness.

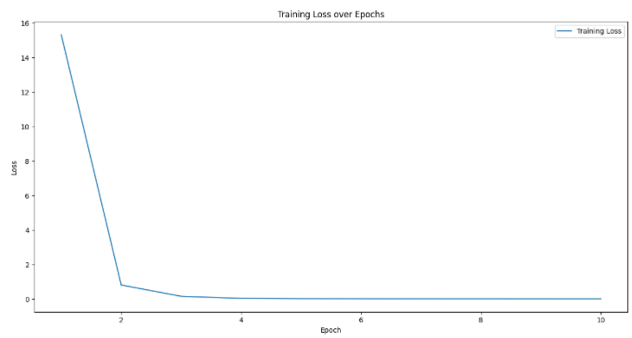


Figure 2: Training loss over epochs.

Confusion Matrix

The confusion matrix provides a detailed view of the model's classification performance across different cake classes. It highlights the number of correct and incorrect predictions for each class, allowing for a more granular analysis of the model's strengths and weaknesses. In the visualization of the confusion matrix (Figure 3), it becomes evident which classes are most accurately predicted and which ones are prone to misclassification. This tool is invaluable for diagnosing specific issues with the model's performance and guiding further refinements.

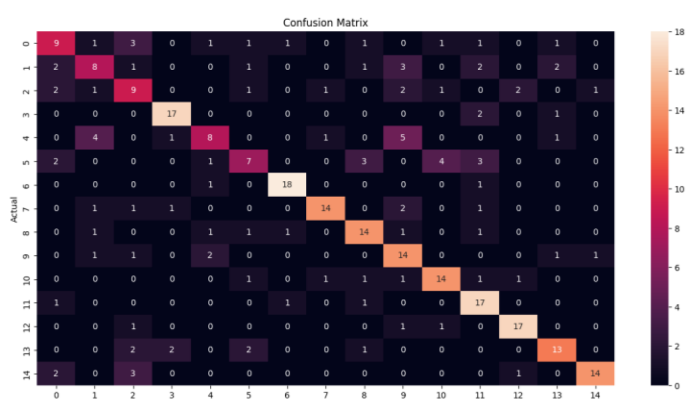


Figure 3: Confusion matrix for the final model.

Class Analysis and Commonly Confused Classes

Class analysis involved examining the confusion matrix to identify pairs of classes that were frequently misclassified. This analysis revealed specific classes that were often confused with each other, such as Class 0 being confused with Class 1, and Class 1 with Class 9. Identifying these commonly confused pairs is crucial for understanding the underlying reasons for misclassification, which could be due to visual similarities or insufficient training data diversity. This insight helps in directing future efforts to improve the dataset or model, such as adding more training samples for the problematic classes or applying targeted data augmentation techniques.

Analysis of Misclassified Samples

Analyzing misclassified samples revealed patterns and reasons for errors, such as visual similarities between different cake types or inadequate training data. By examining these samples, specific challenges were identified, such as apple pie images being misclassified as other pies due to similar textures and colors. This analysis helps address issues by augmenting the dataset with more diverse images or refining feature extraction.

Summary of Findings

The project successfully classified cake images using a combination of low-level and neural features. Initial classifiers trained on neural features showed modest accuracy, highlighting the need for refinement. Fine-tuning the combined model significantly improved accuracy to around 64.33%. Training loss decreased consistently, indicating effective learning. The confusion matrix and misclassified samples provided insights into the model's weaknesses, guiding future improvements. Overall, combining low-level and neural features proved effective, enhancing classification accuracy and robustness.

Potential Future Work

Future work can focus on increasing the dataset size and diversity to improve generalization. Data augmentation techniques, like random cropping and color jittering, can enhance training set variability. Exploring advanced neural network architectures, such as ResNet or EfficientNet, might yield better feature representations. Implementing more sophisticated methods for combining features, such as attention mechanisms or ensemble learning, could further improve performance. Detailed analysis of the confusion matrix and misclassified samples can inform targeted efforts to address specific model weaknesses, leading to a more accurate and reliable cake classification system.

- **Statement of Originality**

I affirm that this report is the result of my own work and that I did not share any part of it with anyone else except the teacher.