Cake classification

Cristian Andreoli

Jun 2023

Contents

Abstract

This paper explores the application of machine learning for cake classification through image recognition. The study utilizes two main approaches: one using hand-crafted image features processed by a classification model, and the other employing deep learning techniques with convolutional neural networks (CNNs). The dataset consists of 15 categories of cakes, with 100 training images and 20 test images per category, all resized to 224 × 224 pixels. Low-level features and neural features extracted using the PVMLNet CNN architecture are employed for classification. The experiments involve training classifiers, analyzing feature combinations, identifying confused class pairs, and evaluating misclassified test images. Additionally, alternative neural features and fine-tuning techniques are explored.

1 Lab activity

1.1 Preliminaries

The data set contains 120 images for each of 15 kinds of cake. For each class 100 images are in the training set and 20 form the test set. All the images have been resized to 224 × 224 pixels. For feature extraction we will use low-level features. We will also use neural features, computed py using the 'PVMLNet' CNN.

1.1.1 PVMLNet

PVMLNet is a convolutional neural network (CNN) architecture specifically designed for image classification tasks. It is based on the popular AlexNet architecture but offers a simplified version with slight modifications. PVMLNet excels at processing color images of size 224 × 224 pixels, making it well-suited for a variety of computer vision applications

1.2 Low-level features

I utilized the color histogram as a feature for cake classification. Based on your results, here is a summary of the feature extraction and classifier training:
Feature Extraction:

- The color histogram feature extraction method was applied to the dataset.
- The shape of the extracted features for the test set is (300, 192), indicating that each image is represented by a feature vector of length 192.
- Similarly, for the training set, the shape of the extracted features is (1500, 192), representing the 192-dimensional feature vectors for each of the 1500 training images.

Classifier Training:

- A multilayer perceptron (MLP) classifier, implemented with the pvml.MLP class, was used for training.
- The input layer of the MLP has the same dimensionality as the extracted features, which is 192 in this case.
- The training process involved running the MLP for 5000 epochs, with a batch size of 50 and a learning rate (lr) of 0.0001.
- The reported results show the accuracy achieved on the training and test sets at epoch 4900:
  - Training Accuracy: 22.73%
  - Test Accuracy: 20.67%

  These results indicate that the MLP classifier, trained on the color histogram features, achieved modest accuracy in classifying the cakes. However, the accuracies on both the training and test sets are relatively low, suggesting that the classifier may not generalize well to unseen data.

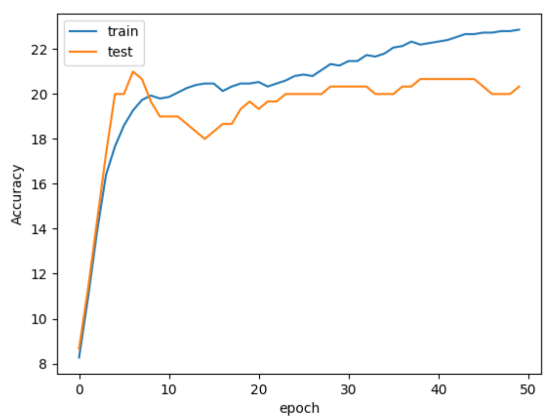


Figure 1: Training of the MLP with color histogram feature.

1.3 Neural features

The pretrained PVMLNet was utilized to extract features from the activations of the last hidden layer. These features were then used to train a perceptron without hidden layers.
The MLP classifier achieved a perfect training accuracy of 100.0%, indicating that it successfully learned the training

data. However, the test accuracy of 79.67% suggests that the model may not generalize well to unseen test examples. There is a discrepancy between the performance on the training and test sets, indicating a possible issue of overfitting.

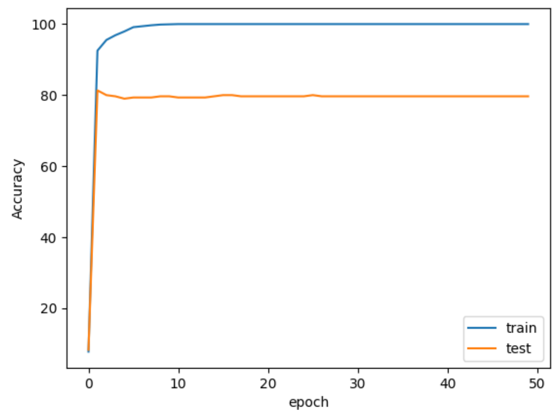


Figure 2: Training of the MLP without hidden layers.

1.4 Transfer learning

A new network was constructed by replacing the last layer of PVMLNet with the weights of the trained perceptron. This approach allows us to leverage the learned representations from PVMLNet and fine-tune the network specifically for the cake classification task. By replacing the last layer, which is typically responsible for the classification task, with the trained perceptron's weights, we can adapt the network to our specific needs.

2 Assignment

2.1 Combining features

You experimented with the following features for cake classification:

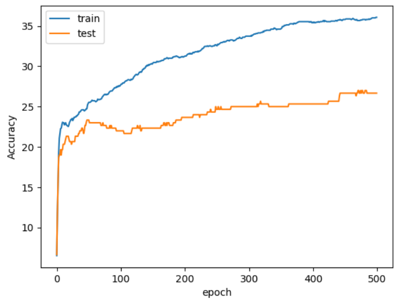
- features1: Color histogram
- features2: Edge direction histogram
- features3: Co-occurrence matrix

Learning rate and batch size are the same.

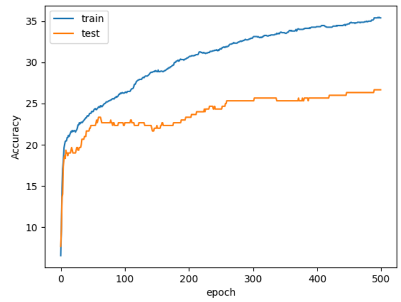
Features	Epoch	Training Accuracy	Test Accuracy
features1 + features2 + features3	5000	24.27%	22.33%
features1 + features2	5000	23.53%	23.33%
features1 + features3	5000	23.67%	21.67%
features2	5000	14.2%	9.6%
features1 + features2 + features3	50000	36.07%	26.67%
features1 + features2	50000	35.4%	26.67%
features1 + features3	50000	35.13%	26.33%
features2	50000	15.8%	11.6%

Table 1: Results of combining different features for cake classification.

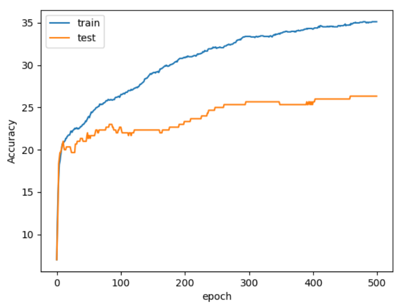
These results demonstrate the impact of combining different features on the classification performance. It is evident that certain feature combinations lead to better accuracies compared to others.



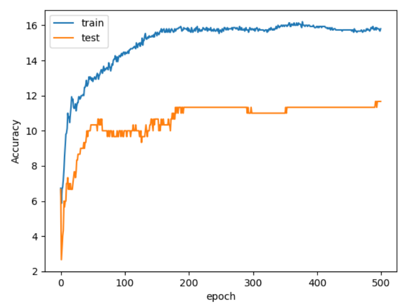
“F1+F2+F3”



“F1+F2”



“F1+F3”



“F2”

2.2 Analysis

To identify the pairs of classes that are more likely to be confused with neural features, I analyzed the classification results and examined the confusion matrix.

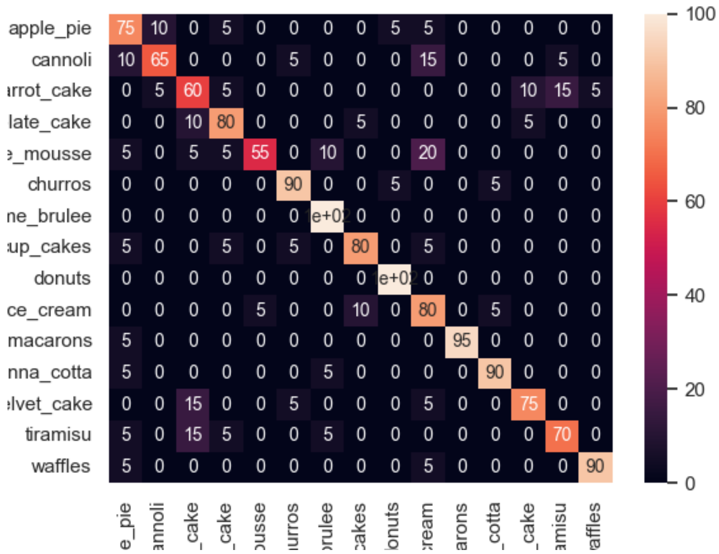


Figure 3: Confusion matrix.

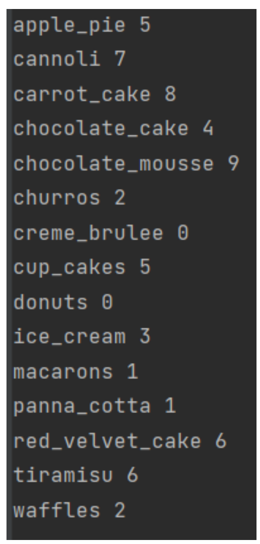

```
apple_pie 5
cannoli 7
carrot_cake 8
chocolate_cake 4
chocolate_mousse 9
churros 2
creme_brulee 0
cup_cakes 5
donuts 0
ice_cream 3
macarons 1
panna_cotta 1
red_velvet_cake 6
tiramisu 6
waffles 2
```

Figure 4: Number of wrong classification for class.

Figure 5: A mouss that was miss classified as an icecream. Probably because it seem a chocolate icecream

2.3 Neural features

I experimented with neural features computed from different hidden layers, you obtained the following results:
These results indicate the performance of the classifier when using neural features from different hidden layers. It

Hidden Layer	**Epochs**	**Training Accuracy**	**Test Accuracy**
activations[-5]	5000	98.53%	87.33%
activations[-6]	5000	99.93%	90.33%
activations[-3]	5000	100.00%	80.33%

Table 2: Results of using neural features from different hidden layers.

appears that activations[-6] achieved the highest training and test accuracies, followed by activations[-5]. However, activations[-3] achieved a perfect training accuracy but a lower test accuracy, suggesting possible overfitting on the training data.

2.4 Report

"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."