Bui Ngoc Thanh Hung - Report

Exercise 1 - Hero Name Recognition

Introduction

This project aims to develop a program that can predict the class of images in a folder and generate an output file that has a similar format to the test.txt file in the dataset. The program should be able to run on any system by following the instructions provided in the README file.

Solution

The solution consists of several steps. First, we need to download and preprocess the data. Then, we will use a feature extraction technique called SIFT to extract features from the images. Finally, we will use a Flannbasematcher of the KDTREE algorithm to predict the class of each test image.

This approach is used for its simplicity and efficiency on such a small dataset, considering there are only three working days to complete the project without a sufficient amount of data. Moreover, the size of the test images is relatively small (images' height is typically around 100 pixels) making it difficult to apply more advanced methods, for example, zero-shot learning.

1. **Downloading and preprocessing data**

I download the template image of each hero from the provided URL. However, some of the downloaded hero figures are significantly different from how they appeared in the test dataset, hence I used some of the test images as the corrected templates and stored them in the template_correct folder.

2. **Feature Extraction using SIFT**

SIFT (Scale-Invariant Feature Transform) is a feature extraction technique that can be used to extract robust and distinctive features from images. SIFT works by detecting and describing key points in an image invariant to scale, orientation, and lighting changes. I used the SIFT algorithm provided by the OpenCV library to extract features from our preprocessed images.

3. **Match the template and image by a Flann-based matcher**

Then, we will use the Flann-based matching algorithm with KDTREE indexing and a specified number of trees, checks, and nearest neighbors to find corresponding features between two images. We will only keep the matches with a distance ratio less than a threshold value as good matches. The number of good matches will be returned as a measure of similarity between the two images.

4. **Predict using the greedy algorithm**

After matching, we have a list of the top-k templates with the highest similarity scores. Finally, the name of the template with the highest matching score is returned as the predicted class of the input image.

Results

	precision	recall	f1-score	support
Lulu	1.00	1.00	1.00	2
Lee_Sin	1.00	1.00	1.00	2
Annie	1.00	1.00	1.00	3
Evelynn	0.67	1.00	0.80	2
Dr._Mundo	1.00	1.00	1.00	3
Katarina	1.00	0.67	0.80	3
Miss_Fortune	1.00	1.00	1.00	2
Ezreal	1.00	1.00	1.00	2
Jinx	0.91	0.91	0.91	11
Graves	0.00	0.00	0.00	0
Ahri	1.00	1.00	1.00	2
KaiSa	0.11	0.50	0.18	2
Amumu	1.00	0.67	0.80	3
Blitzcrank	1.00	0.50	0.67	2
Garen	1.00	0.50	0.67	4
Camille	1.00	1.00	1.00	2
Jhin	1.00	0.50	0.67	2
Master_Yi	1.00	1.00	1.00	1
Akali	1.00	1.00	1.00	9
Jarvan_IV	1.00	0.60	0.75	5
Fiora	1.00	1.00	1.00	3
Darius	1.00	0.60	0.75	10
Kennen	1.00	1.00	1.00	1
Leona	1.00	1.00	1.00	1
Fizz	1.00	1.00	1.00	1
Lux	1.00	1.00	1.00	2
Jax	1.00	1.00	1.00	1
Ashe	1.00	1.00	1.00	3
Corki	1.00	1.00	1.00	2
Diana	1.00	1.00	1.00	1
Alistar	1.00	1.00	1.00	1
Janna	1.00	1.00	1.00	2
Gragas	1.00	1.00	1.00	1
micro avg	0.89	0.85	0.87	91
macro avg	0.93	0.86	0.88	91
weighted avg	0.96	0.85	0.88	91

The proposed solution achieved nearly 88% f1-score on the test dataset. As mentioned, the performance can be further improved with enough data.

Remaining Problems

The main challenge in this project is to extract robust and distinctive features from images that can be used to train a machine learning model. SIFT is a good feature extraction technique, but it can be slow and memory-intensive, especially if we have a large number of images. Another challenge is to select the best machine learning model that can generalize well to unseen images. We can overcome these challenges by using more efficient feature extraction techniques, such as SURF or ORB, and by using more advanced machine learning algorithms, such as deep learning with enough amount of data.

Improvements

To improve the accuracy, we can use more advanced feature extraction techniques and machine learning algorithms. We can also use data augmentation techniques to generate more training images and reduce overfitting. Another improvement would be to use transfer learning to fine-tune a pre-trained deep learning model on our dataset. We can also use ensemble methods to combine multiple models and improve the overall performance. Finally, we can optimize the hyperparameters of our model using a grid search or a random search.

Ablation study

Here is some approaches that did not work:

1. Zero-shot image classification using HuggingFace pipeline

   The code uses the Hugging Face pipeline to perform zero-shot image classification on a merged image. The image is created by appending a template image to a test image, both of which are cropped before being merged. The resulting merged image is then converted into an image object and passed into the pipeline with predefined candidate labels. The pipeline returns the scores for each label, indicating the probability that the image belongs to that class.

2. Metric Learning using CLIP and Pix2Struct model

   This approach uses a pretrained Pix2Struct or CLIP model to embed images of a template and a target image. It then calculates the cosine similarity between the two embeddings to determine how similar they are.

3. Few-shot learning using Resnet models

   This approach follows the paper titled "Few-shot Image Classification: Just Use a Library of Pre-trained Feature Extractors and a Simple Classifier". The idea is similar to metric learning but CNN was used instead of transformer.

For the detailed implementation of each approach, see the ablation folder