Emoji-based feature extraction and image prediction

Assignment1 for COMP4423

Shouwen Zheng
21097982d

***Abstract*—With the use of social media, people are increasingly inclined to use Emoji to express their feelings. Some machine learning workers can collect these Emojis for analysis and use CV classification to perform sentiment analysis. At the same time, it is combined with NLP to determine the ironic meaning of the context to effectively avoid the occurrence of online violence.**

This project mainly uses given and expanded limited Emoji characters as a data set to perform a series of feature extraction work and uses basic classification models to predict invisible test sets, achieving in some prediction models (Train * Validation) Good prediction effect. At the same time, new fused expressions and faces are predicted to achieve excellent results. Not only that, through visualization methods, this work demonstrates the basic operations of image processing in detail.

Keywords—Computer vision, feature extraction, graph prediction, data visualizations

I. Introduction

Emoji detection and classification are not only beneficial to the iteration of new types of emoticons, such as the development of Emoji for platforms such as Google and Apple, but are also beneficial to the supervision of social media platforms. The development of large language models and AI Agent in recent years has given the classification and prediction of emoticons a more promising future. The development of automatic replies, sentiment analysis, suicide detection and other technologies has made CV, NLP and AI more closely integrated.

Due to the development of cloud computing and hardware, CNN-based feature extraction and model construction have become mainstream, and some even use ensemble learning methods, such as bagging and voting integration to further improve the accuracy of the model.

This project does not focus on deep learning such as CNN, but uses basic feature extraction and classification models to demonstrate the process of Emoji classification prediction.

The data set of this project comes from Noto Color Emoji Font, and the total number of training samples is 160 and 25 Lables. And use different feature extraction methods: HOG, gray_histogram, SIFT, LBP, Combined features. In the classification model, SVM, KNN, and Ensembled Model (Hard Voting) are used for model construction. Finally, a cross-validation accuracy of nearly 60% was achieved.

It is worth noting that the entire feature extraction and classification model construction process is visualized, so the entire logic is very clear.

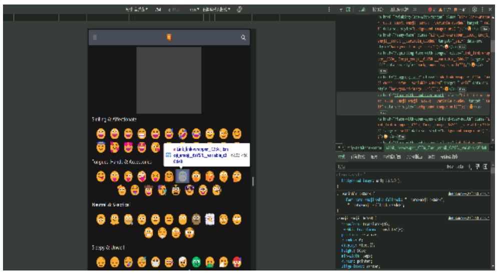
Language	Python
Package:	Opencv, cv2, Numpy, skimage, matplotlib, pandas, sklear

(**Table1**: Package and languages used in this project)

II. Method

A. Data collection

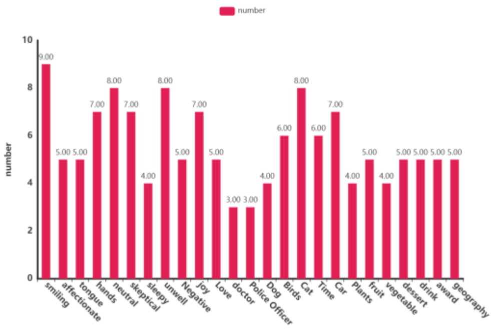
The data set comes from emojipedia.org. The author obtained Emoji char in batches through a web crawler, and through manual screening and processing, a data set with a sample number of 160 was obtained. The entire data set consists of an Emoji char list and an {Emoji:Lable} dictionary.



(**Figure1**: web crawler method)

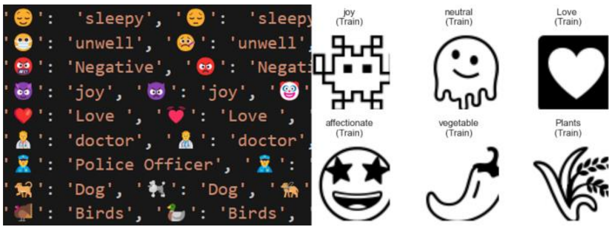
We pass the samples through the Noto Emoji Font font filter and normalize all samples into a consistent Font to increase the robustness and versatility of the data set.

Through the conversion function, Google Font Emoji is processed into an analyzable image format and processed in grayscale.



(**Figure2**: Sample Visualization)

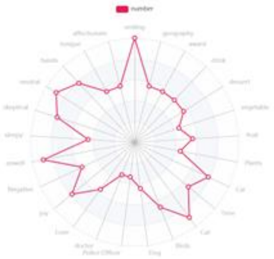
During the collection of the data set, the author found that the Emoji synthesized by Emoji Kitchen did not correspond to the Noto Emoji Font, but directly provided the image format. Therefore, the synthesized Emoji was not included in the construction of the data set to ensure the consistency of the provided format (Will be used in subsequent methods)



```
'😌': 'sleepy', '😔': 'sleepy
'😷': 'unwell', '🤕': 'unwell'
'🤬': 'Negative', '😡': 'Negati
'😈': 'joy', '👿': 'joy', '🤡'
'❤️': 'Love ', '💓': 'Love ',
'🧑‍⚕️': 'doctor', '🧑‍⚕️': 'doctor'
'👮': 'Police Officer', '👮':
'🐕': 'Dog', '🐩': 'Dog', '🦮'
'🦃': 'Birds', '🦆': 'Birds',
```

(**Figure3**: Emoji Processing)

Since the data of some labels in the collection are not balanced enough, we add corresponding weights in data preprocessing.



(**Figure4**: Label distribution)

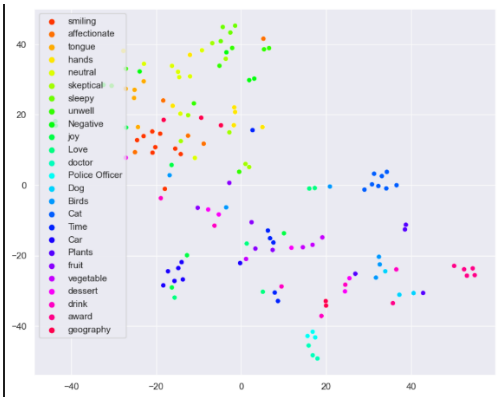
B. *Feature extraction*

In feature extraction, we use different models such as HOG, SIFT and LBP. In the next three parts, I will discuss the accuracy of each model in detail.

Since the image after NOTO Google filter does not have strong color features, a feature extraction method that can accurately detect Color is not used.

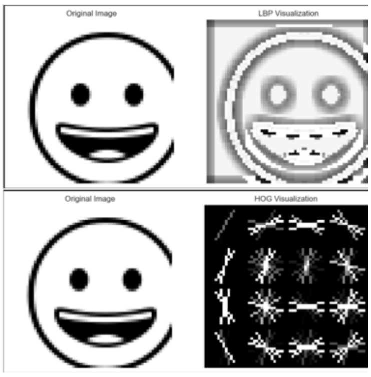
After comparison, we found that the extraction effect of SIFT is not very good. This may be because the texture features of Emoji are not particularly obvious compared with real pictures. Although the features extracted by other methods are obvious, they are limited by the small size of the data set. , unable to maintain the efficiency of the classification model.

So we use integrated feature extraction (combined_feat), call the LBP and HOG methods, and then couple the output parameters to obtain more obvious image feature vector list. Feed this feature vector list into the subsequent classification model to obtain more accurate training .



(**Figure5**: Feature Distribution of Combined Model)

At the same time, in order to show the clear steps of feature extraction, we visualize the training process.



(**Figure6**: LBP and HOG Visualization Example)

C. *Classification model building*

We cut the extracted feature vectors, label vectors, and original image vectors, and reorganize them in the manner of default training set:validation set = 0.8:0.2.

For classifier training, we tested the provided SVM model, KNN model and Ensembled model (Hard Voting)

During the SVM training process, we use GridSearchCV for automatic hyperparameter tuning to obtain better test accuracy:

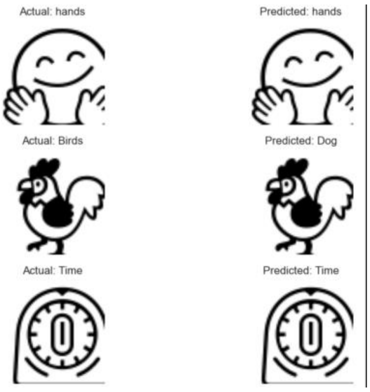
Best parameters: {'svc__C': 0.1, 'svc__kernel': 'linear'}

Best cross-validation score: 0.43

During the KNN training process, we manually debugged the hyperparameters and found that n_neighbors=3 has the best effect.

In order to obtain better general ability and robustness, we have added a new model, citing the Embleated model. In it, we trained a voting classifier including KNN, SVM, and RF (random forest). The voting method is hard voting. .At the same time, we also tested Bagging Modal, but due to poor performance, we deleted this model.

The following is a visualization of the process (taking the KNN model as an example):



(**Figure7**: KNN Cross-validation)

D. *External data testing*

For invisible test sets, we divide them into two categories:

- Mixed emoticon pack produced by Emoji Kitchen;
- Real facial expression pack

Since the test set is all pictures, we save the picture information using CSV and then test the test data on the model.

Since the KNN model achieved a good level in validation, we will use this model.

We will perform grayscale processing and resize on mixed Emoji and face images to meet the dual-channel needs of different feature extraction models. At the same time, we fine-tune, rotate and deform the images to enrich the data set.

III. Discussion and Conclusion

In this project, we tested different feature extraction methods, as well as various classification models.

	SVM	**KNN**	**Ensembled**
HOG	**41.38%**	**51.72%**	**41.38%**
SIFT	**10.34%**	**6.90%**	**13.79%**
LBP	**31.03%**	**10.34%**	**10.34%**
Combined	**41.38%**	**51.72%**	**10.34%**

(**Table2**:accuracy of each models)

Among them, the most accurate ones are based on the Combined_feature feature extraction model and the KNN classification model. One of the important reasons is that due to the limitation of the number of samples, using simple KNN will have a better fitting effect.

In our expectation, the Combined feature extraction method and Ensembled classifier should have better results, but they do not meet expectations. Even other methods combined with either of these two models will have better results. We guess it is because the sample space is small, and the complex combination model is more suitable for processing large data sets and many labels. In this project, there is a risk of over-fitting, so the performance is very poor.

We will expand the invisible test set test to include generated Emoji and face photos, and found that the accuracy is less than 20%. One important reason is the insufficient data set, and the other is that the real face data has not been trained, so the performance is not so good.

A. *Main work*

This project process focuses on the construction of feature extraction and classification models. At the same time, the tuning and processing of training data, verification data, and test data are also considered. We visualize the entire image processing process to make the logic clearer and show the interaction between different models.

B. *Challenges*

If the insufficient number of samples in this project and the limitations of the training set (NOTO Emoji) are used in the future, the model will not be perfect. At the same time, during training, the imbalance of data distribution for each label and the overfitting problem between the two integrated models also need to be taken into consideration.

When the author allocates the data set, he increases the weight of labels with insufficient training quantity to make up for the imbalance problem, but there are still many challenges waiting to be solved.

C. *Future work*

This project aims to demonstrate the entire computer image processing classification prediction process. So accuracy is secondary. If this project will be expanded into a complete Emoji classification prediction model in the future, on the one hand it will need to expand a large amount of labels and training data, and on the other hand it will be possible to introduce deep neural networks for training.

Due to the access to a large number of different types of data, there are also higher requirements for the generalization and robustness of the feature extraction model. Weighted translation processing of different features can be used to extract more obvious features and discard noise.

Emoji classification prediction is a very meaningful task in society. Currently, both Google and Microsoft have corresponding projects. In the future, this technology can be combined with NLP to play a greater role in social work on the Internet. This technology still has great potential, and I hope that despite the many challenges, this project can make great progress.