# Face Recognition with the SIFT Algorithm

[1] Minho Kim

Visual Computing Lab
mhkim98@seoultech.ac.kr

## 1 Introduction

### 1.1 Motivation

Face recognition is a pivotal task in computer vision that involves identifying or verifying individuals based on their facial features. It has wide-ranging applications, from security and surveillance systems to personal device authentication and social media tagging. The challenge lies in achieving high accuracy under varying conditions such as lighting, pose, expression, and occlusions. Recently, machine learning algorithms have been rapidly advancing for this task [2][7][8].

In line with the purpose of this project, I aimed to solve the face recognition problem using classical image processing algorithms. Among the various algorithms, I chose the Scale-Invariant Feature Transform (SIFT) algorithm [9] to extract robust features that can withstand various conditions, such as changes in facial pose. Similar attempts have been made in past research [3][4][10][11].

### 1.2 Project Summary

This project focused on developing and evaluating a face recognition model using classical image processing techniques. The key achievements are as follows:

1. Model Development: A face recognition model was developed utilizing the SIFT algorithm, chosen for its robustness in extracting features under various conditions such as changes in facial pose.

2. Performance Evaluation: Extensive experiments were conducted on facial image datasets, demonstrating that the SIFT-based face recognition model outperformed the well-known machine learning algorithm Support Vector Machine (SVM).

3. Face Retrieval Demo: A prototype of a face retrieval program was created, showcasing the practical application of the SIFT-based face recognition model.

## 2 Details of the approach

### 2.1 Utilization of the SIFT Algorithm

The Scale-Invariant Feature Transform (SIFT) algorithm is a widely used computer vision algorithm for detecting and describing local features in images. The

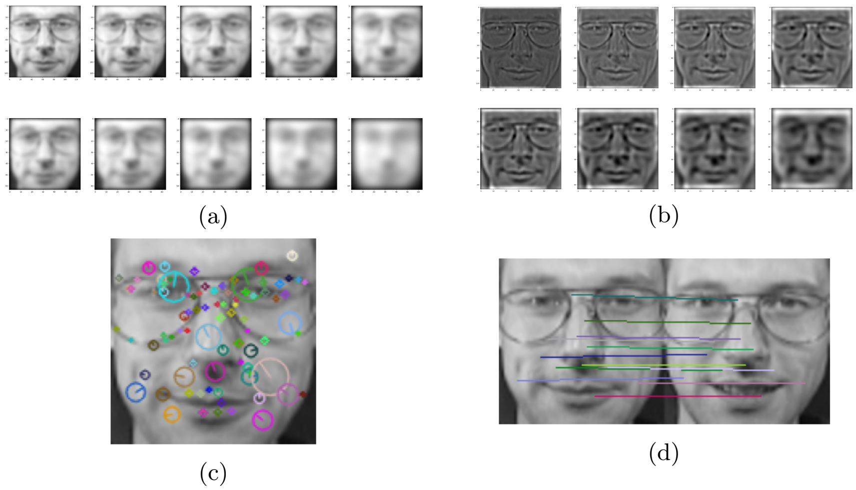



**Fig. 1.** the intermediate images during the computation process of SIFT: (a) represents the gaussian image pyramid, (b) shows the difference of gaussian (DoG), (c) illustrates the result of keypoint descriptors, and (d) displays the keypoint matching results between two different images of the same person.

algorithm is known for its robustness to changes in scale, rotation, and illumination, making it particularly useful in diverse and challenging conditions.

The SIFT algorithm consists of the following steps: 1. Scale-Space Extrema Detection 2. Keypoint Localization 3. Orientation Assignment 4. Keypoint Descriptor 5. Keypoint Matching. To obtain images in Figure 1 without using open-source libraries like OpenCV and instead implementing code from scratch using Python [5][6].

Through these steps, we can obtain keypoints and their corresponding descriptors (feature vectors of 128 dimensions). By finding matching keypoints between two images, we can compute how similar the faces in the two images are.

### 2.2 Classification Strategy

To classify an test image into a specific class during inference, we need to determine which class the image belongs to. For this purpose, I designed a classifier that uses the training image set as the reference, or "gallery" for the image retrieval task. The classifier works by comparing the test image with the gallery images and determining the class of the test image based on the class of the gallery image that has the highest number of matching keypoints with the test image.

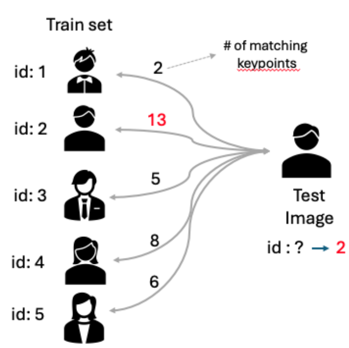
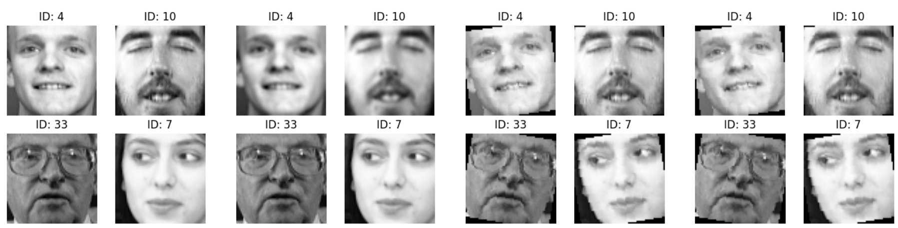



**Fig. 2.** classification strategy


**Fig. 3.** Olivetti Dataset




(a) (b) (c) (d)

**Fig. 4.** Various modified test datasets: (a) Original test set, (b) Test set with Gaussian blur applied,, (c) Test set with slight rotation, (d) Test set with both Gaussian blur and rotation applied.

# 3 Results

## 3.1 Experimental Results

**Olivetti Dataset** For this project, I utilized the easily accessible Olivetti Dataset [1] from Kaggle. This Dataset comprises 400 grayscale images, each of size $64 \times 64$ pixels. There are 40 face classes, with 10 images per class. Figure 3 provides an example of the Olivetti dataset. Additionally, for experimentation purposes, I split 30% of the data into a test dataset. Furthermore, as depicted in Figure 4, various transformations were applied to create multiple test sets.

**Quantitative Results** In Section 2, through the SIFT-based image matching method and classification strategy, I designed a classifier to determine the class to which a test image belongs. To compare performance, I utilized the well-known machine learning algorithm SVM (Support Vector Machine). The performance measurement results can be found in Table 1. I performed optimization on the parameters of the SIFT algorithm, including sigma value for Gaussian filter, the number of Octave Layers for the Gaussian pyramid, edge threshold, and

| | testset1 | testset2 | testset3 | testset4 |
|---|---|---|---|---|
| SVM | 91.6 | 90.8 | 55.0 | 45.8 |
| SIFT | 96.7 | **96.7** | **94.2** | **94.2** |
| SIFT† | **97.5** | 95.0 | 72.5 | 60.8 |

**Table 1.** The rank 1 accuracy performance table. The dagger symbol (†) indicates additional optimization of SIFT algorithm parameters.

contrast threshold. The default values for each parameter are 1.6, 3, 10, and 0.04, respectively. However, through optimization, I found that the best performance for the Olivetti dataset (testset1) in this project was achieved when these values were set to 0.6, 1, 12, and 0.02, respectively. The dagger symbol (†) in Table 1 denotes the results after optimizing the parameters of SIFT algorithm on testset1.

### 3.2 Face Retrieval Demo

I visualized the process of matching images from the training set with a query image (test image) using the classifier I previously created and ranked them accordingly. The demo video can be watched at the following link:

YouTube Link : https://www.youtube.com/watch?v=eHQQ2tH2jKM

## 4 Discussion and Conclusions

In this project, I designed a classifier for face recognition using the classical image processing algorithm technique known as the SIFT algorithm. Through experimentation, I demonstrated that the SIFT-based classifier outperforms machine learning algorithms on specific dataset. Particularly interesting is the fact that while machine learning algorithms struggle with test sets 2, 3, and 4, which involve scale and rotation transformations, the SIFT algorithm shows promising performance. More experiments are needed, but this suggests that the features extracted by the SIFT algorithm are robust to scale and rotation. However, it was observed that the process of comparing both the training dataset and test images during inference is inefficient in terms of inference speed and memory usage, indicating the need for improvement.

## References

1. Olivetti dataset https://www.kaggle.com/datasets/imrandude/olivetti
2. Alansari, M., Hay, O.A., Javed, S., Shoufan, A., Zweiri, Y., Werghi, N.: Ghostfacenets: Lightweight face recognition model from cheap operations. IEEE Access **11**, 35429–35446 (2023)
3. Geng, C., Jiang, X.: Face recognition using sift features. In: 2009 16th IEEE International Conference on Image Processing (ICIP). pp. 3313–3316 (2009). https://doi.org/10.1109/ICIP.2009.5413956

4. Geng, C., Jiang, X.: Sift features for face recognition. In: 2009 2nd IEEE International Conference on Computer Science and Information Technology. pp. 598–602 (2009). https://doi.org/10.1109/ICCSIT.2009.5234877
5. Islam, R.: Implementing sift in python: A complete guide (part 1) (2020), https://medium.com/@russmislam/implementing-sift-in-python-a-complete-guide-part-1-306a99b50aa5
6. Islam, R.: Implementing sift in python: A complete guide (part 2) (2020), https://medium.com/@russmislam/implementing-sift-in-python-a-complete-guide-part-2-c4350274be2b
7. Kim, I., Han, S., Park, S.J., Baek, J.W., Shin, J., Han, J.J., Choi, C.: Discface: Minimum discrepancy learning for deep face recognition. In: Proceedings of the Asian conference on computer vision (2020)
8. Kim, M., Jain, A.K., Liu, X.: Adaface: Quality adaptive margin for face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18750–18759 (2022)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**, 91–110 (2004)
10. Luo, J., Ma, Y., Takikawa, E., Lao, S., Kawade, M., Lu, B.L.: Person-specific sift features for face recognition. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07. vol. 2, pp. II–593–II–596 (2007). https://doi.org/10.1109/ICASSP.2007.366305
11. Majumdar, A., Ward, R.K.: Discriminative sift features for face recognition. In: 2009 Canadian Conference on Electrical and Computer Engineering. pp. 27–30 (2009). https://doi.org/10.1109/CCECE.2009.5090085