Assessment Task 2: Project Implementation

31256 Image Processing and Pattern Recognition

Name: Naiyun Liang Student ID: 24570057

Name: Ivan-Qicong Liu Student ID: 14430375

Name: Ometh Liyanage Student ID: 14580500

Name: SOHYUN LEE Student ID: 25637733

Name: Cong Bao Nguyen Student ID: 14479051

Table of Contents

1. Introduction to the Problem

Face recognition technology has become an important component in real-world system. Especially, it has increased the efficiency of the security and verification area. The technology automatically identifies individuals based on their facial features, so it shows strength in terms of user convenience and time efficiency. However, face recognition technology still has several problems. The technology is vulnerable to differences in light and angle. It does not recognize well if there is a difference in the light or angle of the same person. These flaws can cause troublesome to users who rely on the technology daily for identification purposes. Furthermore, attacks that try to deceive the technology like using a photo in front of the camera may lead to serious social problems. Therefore, it is necessary to improve the adaptability and accuracy of face recognition technology.

Our project initially focused on the implementing an advanced face recognition system using the Scale Invariant Feature Transform (SIFT), which can strongly represent the features of human faces. Since SIFT has scale and rotational invariance and is robust to lightning variation, we decided to extract and use it. We thought that if we use the extracted features as input of a specific classification model, it would be able to implement the face recognition technique we are aiming for. However, during the project, we found that SIFT did not meet our accuracy requirements, so we tried to use Convolutional neural network (CNN), which is widely used for image classification. It has the advantage of effectively preserving the spatial characteristics of the image.

The dataset we chose is the Labeled Faces in the Wild (LFW), a widely used public dataset in the facial recognition field. It contains more than 13,000 images of about 5,000 people collected from the web. Since the dataset includes different lighting conditions and angles per one face image, it can help us to build the model that can robustly recognize faces.

To implement the advanced face recognition technology, we had various steps from data preprocessing and feature engineering to classification model construction. By evaluating and comparing the performances of each model, we gradually came up with ways to improve its accuracy and reliability in handling diverse conditions. Through this project, we hope to overcome the existing problems in current face recognition technology and create a more secure system.

2. Overview of the Project and Methodology

In order to approach to achieve the result there are some techniques and methods that were followed to ensure the quality of the system by data preprocessing and feature extraction. These techniques include Scale-Invariant Feature Transform (SIFT), SIFT with bag of visual world (BOVW), SIFT with BOVW and colour conversion, noise reduction (Gaussian Filter), scale-space pyramid & difference of gaussians (DoG), key point localization and orientation assignment (Based on gradient). In addition, these methods were chosen to the reliability, adaptability and accuracy of the face recognition system. Following are the reasoning and what kind of result we achieved by using each of them.

Scale-Invariant Feature Transform (SIFT) was considered in this project for its ability to detect and describe the key features of an image regardless of image rotation, scale and light changes. It can identify the features of the face such as eyes, nose and mouth even when the environment varies, which makes this technique is highly effective and suitable for face recognition tasks where it can be seen significantly change in lighting and orientation. At first, we converted each image to greyscale since the SIFT is performing better in high intensity values rather than color channels. The descriptors are aggregated by imputing with the mean, which was simple, but it was ineffective approach. In order to get the result, we split the dataset to 70% training and 30% testing using random indexing. However, the result we got from this approach was relatively low, which indicates that there might be issues with feature extraction or classification process.

The Bag of Visual Words (BOVW) approach with SIFT feature extraction for image classification was chosen because of it's inspired by text processing, where image is represented as a collection of "visual words". This technique allows images to be treated as similarly to documents. We followed the steps as previous we used for SIFT with few changes such as VLFleat library setup used to perform SIFT feature extraction and k – means clustering. For training and testing we used same split amount as before and we were able to increase the accuracy compared to above method, yet it was relatively low. The possible reasons for this were low amount of number of clusters, when the descriptors were aggregated into a histogram which led to the loss of spatial information and due to the variation of lighting the which makes challenging for classification.

For the above-mentioned approach, we considered of adding color-SIFT features for image classification. This variation takes advantage of color information by extracting SIFT

features by separating each color channel (Red, Green, Blue). We used the SVM classifier to train and the dataset was divided 70% for training and 30% testing, which the output result was increased compared to grayscale-only SIFT features. Although we did receive a higher result, its not sufficient for a face recognition model. This result made us explore more into other methodologies such as noise reduction (Gaussian Filter), scale-space pyramid & difference of gaussians (DoG), key point localization and orientation assignment (Based on gradient). However, the result we received from these methods were unreliable, which made us focus into Convolutional Neural Network (CNN) for an effective result.

CNN was also taken advantage of due to its accuracy and reliability over SIFT. Convolutional Neural Networks (CNN) visualizes and interprets information through raw images which can identify facial features such as edges and colours. Although we placed SIFT as our primary function in the first report, we received poor results in our initial testing which leads to our new discovery of CNN. SIFT seemed it would produce the highest accuracy on paper with its specialty being facial recognition, but the extended implementation of CNN gave us more accuracy on the wider aspect of facial recognition. Furthermore, on the addition of techniques used, we applied BOVW which is a pre-deep learning method that builds images to assign main points on an image.

For our training approaches, we are not only using the typical 80% training and 20% testing, but we also vary our ratio of approaches ranging from 100% training and 30% testing due to the lack of datasets we hold. This however, only applies to CNN as SIFT does not require training and testing for its parameters of accuracy.

The Dataset we use to execute the code is from a public domain named Labeled Faces in the Wild Home, which is specified for facial verification. The dataset contains more than 13000 images of faces and is labeled by name. We chose this dataset as its purpose suited our needs on providing images for one person. Although it is possible to find a similar dataset on a website like Kaggle, the amount of images on LFWH (Labeled Faces in the Wild Home) exceeds the usual amount you would usually find on Kaggle which ranges around 400 to 500. The dataset we chose also trains and thoroughly tests the code as there are similar faces for different people. The accuracy of the code can be thoroughly tested through this method as it can show its potency in finding the difference between people with similar facial structures.

3. Results

3.1. Dataset description and preparation

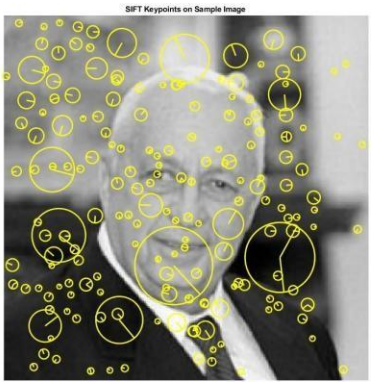
Dataset: In order to ensure that our results will not be affected by the use of the dataset, we use the Labeled Faces in the Wild (LFW) dataset, which is a public dataset and has been used and recognized by the academic and research communities for many years. This dataset consists of a main folder and many subfolders. Each subfolder represents a person. The name of the subfolder is the name of the person in the image in the subfolder, and this name will also be used as the label of each image.

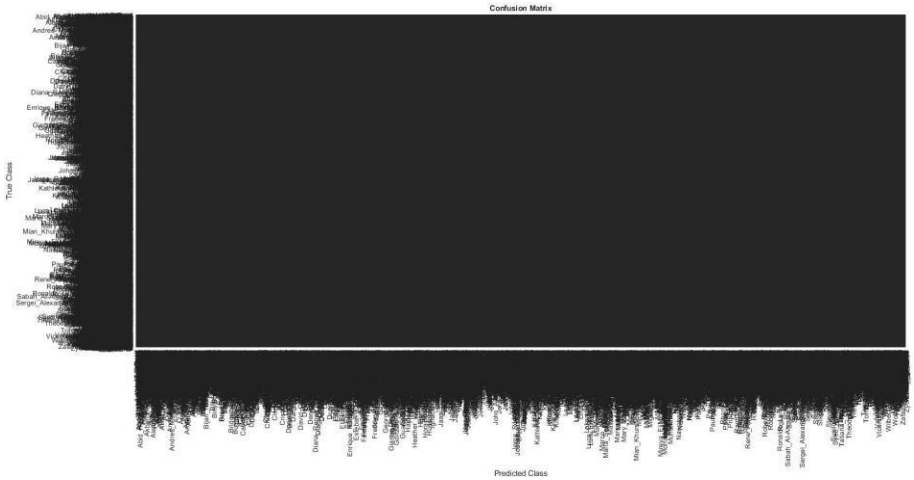
Filtering: The filtering preparation work was added in the middle of the entire project because at the beginning of the project, we simply used the dataset for testing, and the results were terrible because we ignored that if there are only a few or one picture for a person's face recognition training, such results must be unreliable, so we also performed different filtering for different test methods. For the SIFT method, whether it is used alone, combined with BOVW, or combined with the colour conversion method, we will filter out people with less than 50 pictures, and the retained people are those with at least 50 pictures or more. For the CNN method, we chose to perform different amounts of filtering, 20/30/40, to test the impact of different numbers of training sets on our research results.

3.2 SIFT

We first used SIFT (Scale Invariant Feature Transform) because SIFT can identify key information points with the same features and characteristics after length or scale and rotation. Our group believes that SIFT is particularly suitable for facial feature recognition. We first converted each image to grey to extract features. For each grey image, we computed vectors representing the local feature around each keypoint, also called the SIFT descriptors. Through this calculation, each image can provide essential features of the person. The next step is data segmentation. We chose a more common segmentation method, such as 70/30, that is, training the model on 70% of the data and retaining 30% for testing and evaluation. At the same time, we trained a multi-class support vector machine (SVM) classifier for the vector features extracted by SIFT. We chose it since SVM can process higher-dimensional spatial vectors and is very effective in face recognition tasks.

The trained SVM classifier has an accuracy of 32.68% on the test set of SIFT features. This result shows that the model can correctly identify about one-third of the images, which means that the model has only learned some valuable patterns but cannot be directly used for face recognition in real life.





Due to this low recognition accuracy, we reflected on the SIFT method. From the information points represented in the image, a large part of the information points are not shown on the face but mainly on the environment. From this point of view, if we can find a dataset containing only faces, we believe this method's accuracy can be improved. At the same time, we use the confusion matrix for visualization. The confusion matrix compares the accurate labels (actual individuals) with the predicted labels (predictions made by the classifier). The same person is identified as several others due to their background, clothes, and environment. It is difficult for the classifier to distinguish one person from another. This further proves the view that we need images containing only faces, which can reduce the influence of the surrounding environment and improve accuracy.

3.2.1 SIFT with Bag of Visual Words (BOVW)

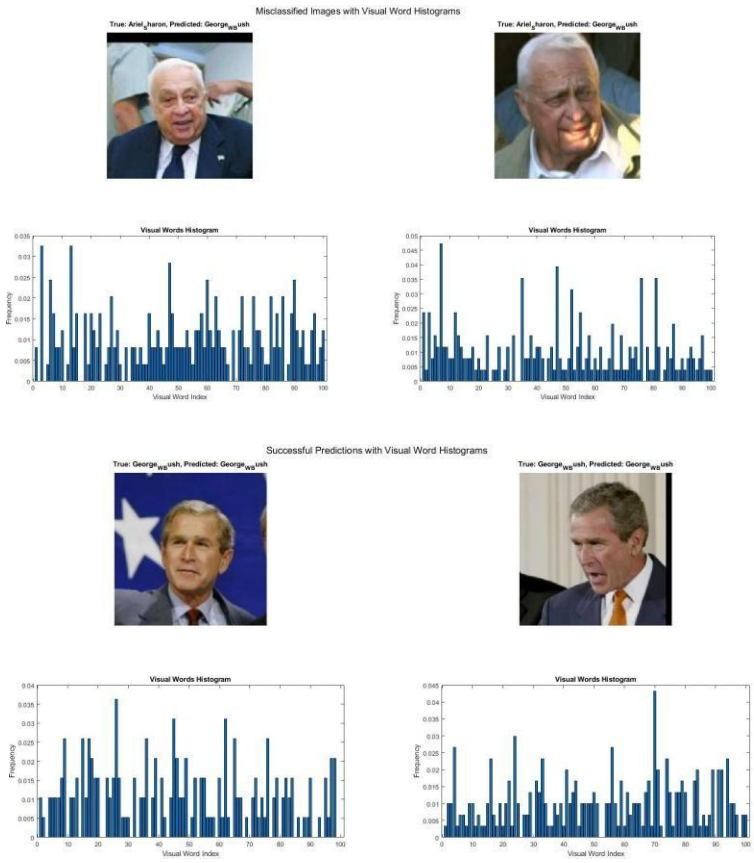
Based on SIFT, we also used a combination of Bag of visual words and SIFT. We used K-means clustering to divide all the different feature vectors extracted from the image into 100 visual words. Each visual word can represent a similar local feature vector, and these visual words can be regarded as image descriptors to describe the image. We visualized the first ten visual words. Each visual word is a 128-dimensional vector, which aligns with the SIFT descriptor. Each bar chart represents a visual word, and the role of the bar chart is to show different visual words/feature vectors (how they represent spatial features). For each image, we construct a histogram based on the frequency of each visual word. The role of the histogram is to show the distribution of visual words in the image because each image is composed of different visual words with different frequencies. This method also uses the SVM classifier. Although the methods differ, the accuracy is similar after the classifier learning. The trained Bag of Visual Words model achieved an accuracy of 33.66% on the test set, which got very similar results to the normal SIFT method.

And the following are some of the true predictions and wrong predictions:

3.2.2 SIFT with BOVW and color conversion

Based on SIFT and BOVW, we also want to determine whether different color channels can improve accuracy. Unlike standard grey images, we apply ordinary SIFT to each image's red, blue and green channels to enhance its effect. In addition to recognizing the vector features of SIFT descriptors, this method also captures the color information of each image. SIFT vector features are calculated for each channel, and the feature vectors of the three-color channels are combined into a new feature vector. This processing can better and more richly express each image, giving it color features and more details.

The accuracy rate increased from 33.66% to 35.63%. It can be seen that adding color information to the descriptor based on SIFT and BOVW can improve the model's ability to recognize different faces, but it is minimal. However, we think this is a meaningful attempt.



Here are some examples of true and wrong predictions, for the wrong predictions we can see that in the histogram there are less similarities presented, but in the true predictions, there are similar patterns presented in the histogram.

3.3 CNN

After testing SIFT, we found that simply using some methods could not achieve our expected results, so we decided to try a very effective method for facial recognition - CNN. We designed three convolution layers, a maximum pooling layer and batch normalization for CNN. We chose this architecture because CNN can capture different and more complex patterns from facial features, especially for edge features and some texture features. This method can better distinguish individuals with different features. Before training, we need to re-unify the image, change it to the same size (128×128) and add three color channels to ensure the capture of color information. At the same time, we also apply data enhancement, such as randomly selecting an angle from positive 15 degrees to negative 15 degrees for rotation and randomly moving along the x-axis and y-axis. These subtle changes help the model find those constant features and generalize better. Next is the architecture of CNN. The input layer is set to 128×128 size and three color channels. As we did with the image at the beginning, we must ensure that the image we process is in the same format as the input layer. Then, we set up three sets of convolutional layers. Each set first sets 3×3 filters for a total of 32/64/128. Then, add extra pixels to make the input and output sizes consistent. Then follow the batch normalization, Relu activation and max pooling to ensure these layers can extract practical features from the input image. Next is the fully connected layer. The first fully connected layer extracts the features captured from the image from the previous layer and introduces them into a dense layer with 256 neurons for combination.
Then, the Relu layer applies nonlinearity to the output of the fully connected layer. The second fully connected layer will be used to directly define the output, ensuring that the number of neurons is consistent with the number of labels. The SoftMax layer converts the scores of each class output into corresponding probabilities. Finally, a classification layer calculates the loss for each time. It allows the network to learn by comparing its predictions with the actual situation and the calculated loss during training and testing. Next, we set the training options, such as learning rate, number of pieces of training, number of epochs, etc. During the setting process, we ensure that the number of trainings is not too short to avoid incomplete training and large deviations in results. At the same time, we choose to display the results of each training course through icons for visualization and observation.

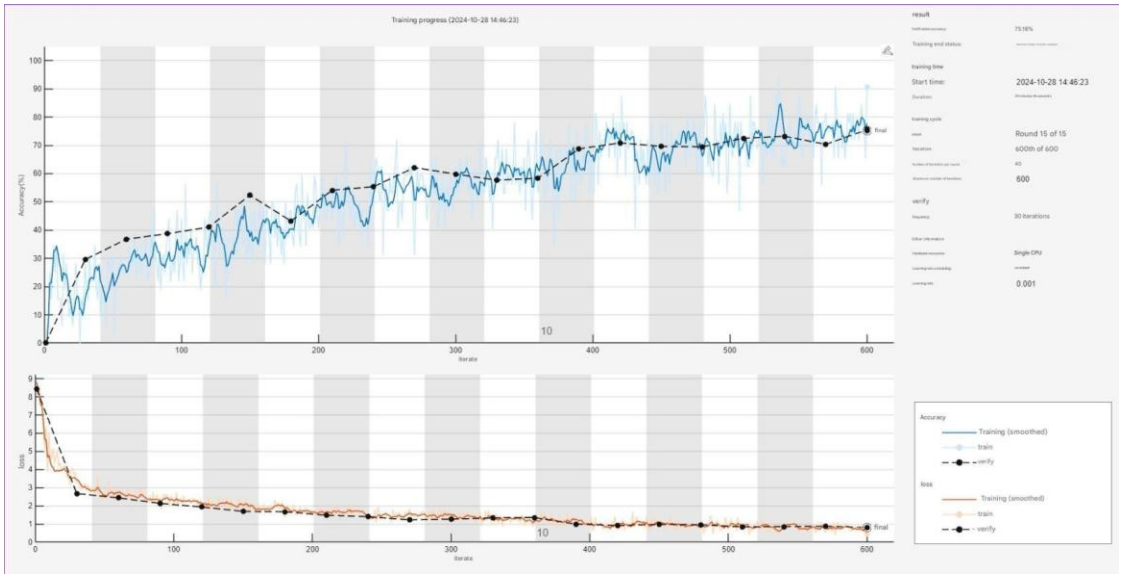
In this test, we try to use the number of person images as a controllable variable to test the accuracy of the model prediction when the minimum number of images that can be selected for each person is 20/30/40.

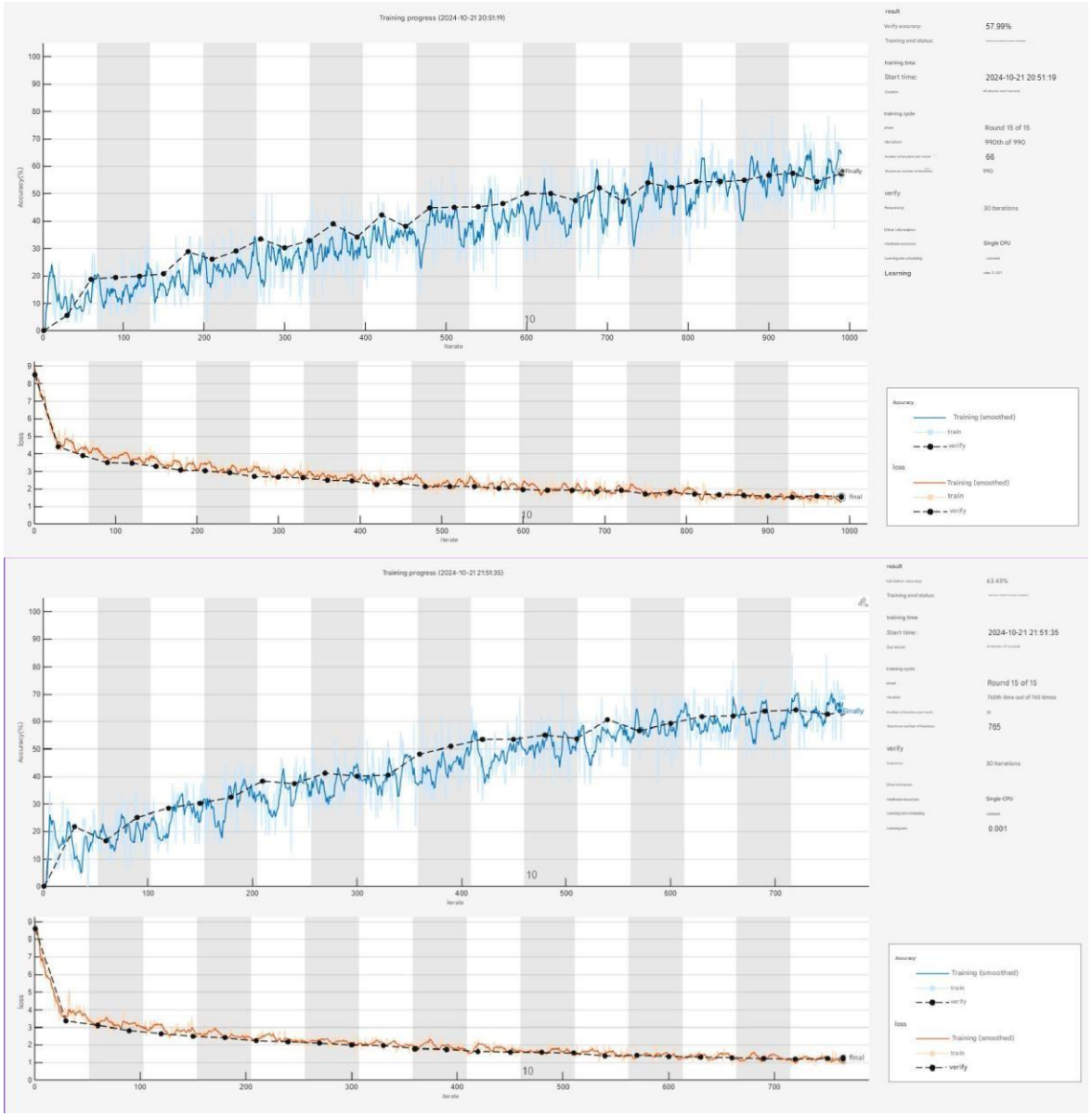
- 65.20% accuracy when individuals with at least 20 images were used.
- 67.14% accuracy when individuals with at least 30 images were used.
- 80.18% accuracy when individuals with at least 40 images were used.

From this result, it can be seen that with the increase in the number of images of each individual, the prediction accuracy has been significantly improved, so we can infer that if the number of individual images continues to increase, the accuracy may be further improved. This also means that the basis of face recognition requires many images for training. This result also shows that the designed CNN can effectively learn and apply facial features to new data. And from the images of the 3-training processes, we can see if we use individuals with at least 20 images, the prediction curve is a bit far from the training curve, and when we use individuals with at least 40 images, the prediction curve is getting closer with the training curve. And the loss curve presents a similar pattern among the 3 tests.

3.3.1 CNN with more layer, larger image size and more data augmentation

Based on CNN, we want to try to make some improvements to the original design and further improve the accuracy, so we increased the original input size to 224×224, tried to use larger image pixels to make the features more prominent and more accessible to identify, and added additional random rotation angles in the data augmentation options, from positive 20 degrees to negative 20 degrees. The random translation distance increased by 5 in each direction. We randomly selected a value from 0.9 to 1.1 to scale and performed random angles (-5°, +5°) shear transformation along the x-axis and y-axis. This shearing can change the image's shape so that the model can still capture or identify features when the image is slightly distorted, thereby reducing the requirements for the image. In terms of convolution layers, we also added a convolution layer, which used 3×3 filters. However, this time, the number increased to 256, and other aspects were consistent with other convolution layers.


Training progress (2024-10-28 14:46:23)
Accuracy(%)
Iteration
final
10
loss
Iteration
final
10
result
75.18%
Training end status:
training time
Start time:
2024-10-28 14:46:23
training cycle
Round 15 of 15
600th of 600
600
verify
30 iterations
Single CPU
0.001
Accuracy
Training (smoothed)
train
verify
loss
Training (smoothed)
train
verify



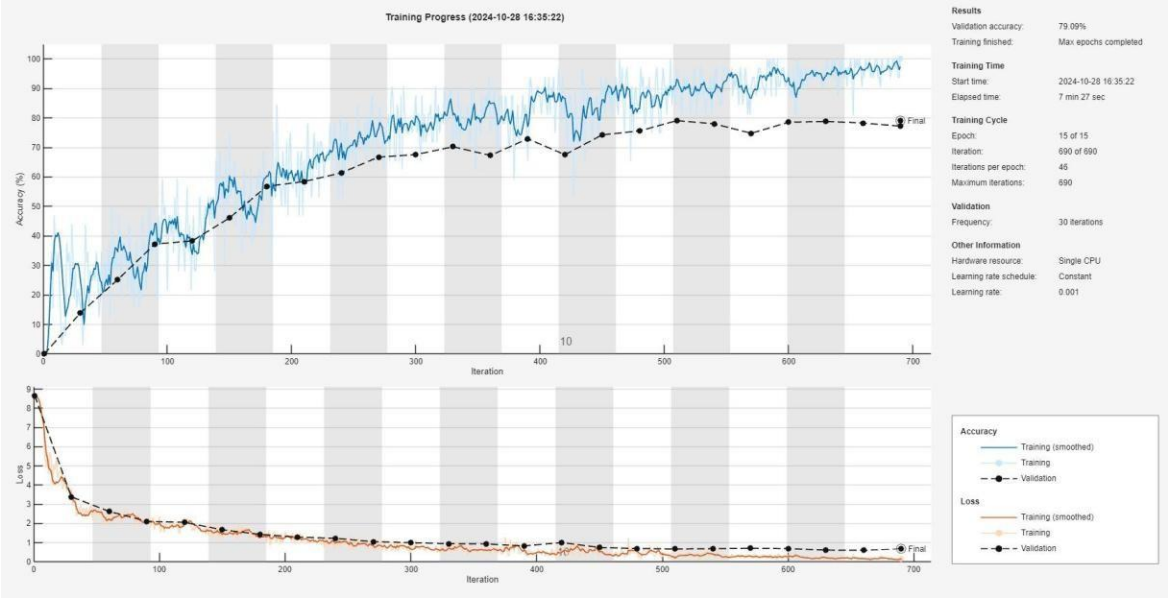
After the same tests as the original CNN, we found some problems. Although we tried to improve the structure, the original CNN could have been better. At the same time, the time to train the model almost doubled. From the results, our improvement was a failure.

From the 3 images we can see the prediction curve nearly coincides with the training curve, this shows that it is a good fit, the prediction will go higher and higher as the training accuracy gets higher. But when the training cycle keep the same, the CNN with more layer, larger image size and more data augmentation will perform worse than the original one, and it takes longer training time, but I trust when the training cycle gets higher, it will perform better than the original CNN.

3.3.2 CNN with different data splitting

Based on the original CNN, we tried different data splits. We initially used a 70/30 split, but we wanted to see what effect 80/20 and 90/10 splits would have on the results. Therefore, we only selected people with more than 40 images. In order to control a single variable, we only changed the data split and retrained. This time, the data was stable at around 80%. From this result, it can be seen that the data split did not affect the result.


Training Progress (2024-10-28 16:45:29)
Accuracy (%)
Iteration
Final
Loss
Iteration
Final
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation
Results
Validation accuracy: 82.89%
Training finished: Max epochs completed
Training Time
Start time: 2024-10-28 16:45:29
Elapsed time: 7 min 59 sec
Training Cycle
Epoch: 15 of 15
Iteration: 780 of 780
Iterations per epoch: 52
Maximum iterations: 780
Validation
Frequency: 30 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.001
Training Progress (2024-10-28 16:23:34)
Accuracy (%)
Iteration
Final
Loss
Iteration
Final
Accuracy
Training (smoothed)
Training
Validation
Loss
Training (smoothed)
Training
Validation
Results
Validation accuracy: 81.43%
Training finished: Max epochs completed
Training Time
Start time: 2024-10-28 16:23:34
Elapsed time: 10 min 21 sec
Training Cycle
Epoch: 15 of 15
Iteration: 600 of 600
Iterations per epoch: 40
Maximum iterations: 600
Validation
Frequency: 30 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.001



And through the images, we can find the 3 different data splits show nearly the same pattern among the three training.

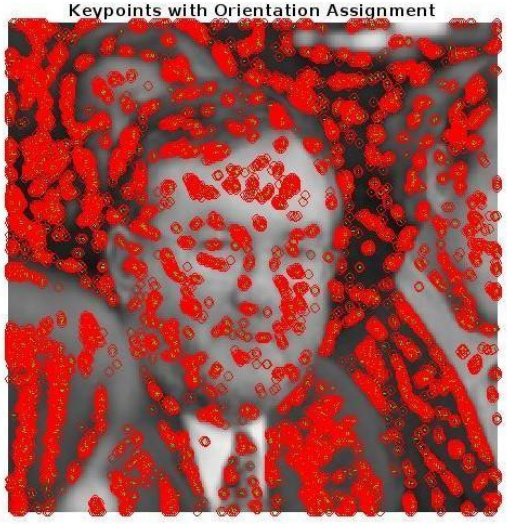
3.4 Some features failed to implement

At the beginning of this project, we combined the required technologies according to the technology selection decision of the previous evaluation report. We used these technologies to identify images and calculate recognition accuracy. However, during the implementation of the project, we found that these technologies could not be combined well. Many of these technologies were used commonly, but when we tried to combine them and use SIFT for training and testing, too many errors occurred, and the accuracy rate remained at 0% during the implementation process. This may be related to the selected data set because we did not filter the data set at the beginning of the project. However, we thought that the filtering problem would not lead to 0% accuracy, so after filtering, we reflected and only selected a small part of the technologies for implementation and tried to use technologies like deep learning for training and trial. Below are some images of some technologies we used in the early stage of training.

Difference of Gaussian

Detected Keypoints

Keypoints with Orientation Assignment



4. Discussion and Challenges

The face recognition project was a valuable learning experience, combining theoretical knowledge with practical challenges. The team's journey to achieving reliable face recognition was both iterative and challenging, from goal setting to implementation and evaluation. Achieving the target accuracy for our face recognition model was one of the most challenging and iterative aspects of this project. This process involved extensive experimentation and adjustments, which ultimately contributed to the model's improvement in accuracy.

4.1 Achievement of Initial Goals

Starting this project, our team had limited experience with advanced techniques in image processing, feature extraction, and pattern recognition. Techniques like Scale-Invariant Feature Transform (SIFT) and Difference of Gaussians (DoG) were challenging, as we needed to apply theoretical concepts into practical applications. In the initial steps, we faced challenges and spent a lot of time in research and practical testing to figure out how to use the theories we had studied.

Our initial goals focused on creating a reliable face recognition system that can handle varied lighting conditions, face orientations, and spoofing issues. With a limited understanding of these concepts, each increase in accuracy was a result of persistent testing, adjustment, and application of newly acquired knowledge. Through iterative testing and refinement, we achieved a substantial improvement in accuracy, from an initial 0%, then achieving improvements of 7%, 16%, 56%, 70% and finally reached 80%. This improvement reflects the effectiveness of the actual techniques implemented, particularly in managing data complexity and improving model stability.

4.2 Challenges Faced and Overcoming Strategies

From the outset, one of the biggest challenges was the dataset itself. The dataset we initially chose contained many people with very few images, often only one per person. This lack of images made it extremely difficult for the model to learn patterns and recognize individuals' faces accurately. In machine learning, especially with face recognition, having multiple images per person is essential for training the model to identify unique features under varying conditions. The lack of images per person posed a major issue to effectively training and testing. With only one image for each individual, the model has almost no basis for

learning the variability in facial features. Due to these limitations, our initial accuracy was exceptionally low, as the model was essentially overfitting on very limited data. These early setbacks highlighted the importance of quality data, and we had to apply filtering technique, keeping only people with enough images to stabilize the model.

Another challenge is the application of new techniques in a practical context, which involve multiple stages of trial and error. Each stage not only taught us more about how image and pattern recognition techniques operate, but also presented specific obstacles that were time-consuming to address. We learned that SIFT's performance can vary widely based on parameters like scale key point thresholds, which were challenging to understand and apply correctly without prior experience at first. Using DoG for anti-spoofing was a new approach for us, designed to improve security by distinguishing live faces apart from photos. This technique made the model more complex, so we had to carefully manage its impact on accuracy without making the model too complicated. Applying this gave us a better understanding of security in face recognition.

Achieving high accuracy in face recognition is also a big challenge, especially under varied lighting, face orientations conditions that can significantly interupt feature detection. Our initial accuracy rates, beginning at a low percentage, 7% and only improving to 16% after preliminary adjustments, show the need for more advanced techniques. To improve, we refined our preprocessing by adding more images per person, filtering and reducing noise, and shuffling the dataset for better accurate score. We then integrated CNN, which significantly enhanced accuracy by allowing the model to learn deeper patterns in facial features. Combined with optimized SIFT for feature extraction, adjusting key point thresholds and scale space, our model accuracy progressed increasingly and reached up to 80%. Each improvement required persistence and carful tuning, showing that building a reliable face recognition model relies heavily on patient adjustments to both data quality and algorithm configurations.

4.3 Future Work and Improvements

One major limitation of our current system is processing speed, which affects its suitability for real-time applications. To make the face recognition system suitable for real-time uses, faster processing is essential. Real-time performance is important because any delay can make the system less practical and secure. Currently, our system operates with MATLAB, which is effective for testing but not fast enough for real-world processing. Using GPU-based

processing could help improve the speed of recognition by allowing the model to handle many images at once. Additionally, using more optimized frameworks like TensorFlow or Pytorch would make the system faster and more efficient, as they are designed for handling the complex computations involved in deep learning models like CNNs. With these changes, our system could maintain its accuracy while meeting the demands of high-speed applications.

Furthermore, currently, we are using the LFW dataset, which provides a good dataset for face recognition training. However, expanding the dataset to include a wider range of lighting conditions, facial expressions, and facial obstructions, like glasses or masks, would greatly enhance the model's strength. While LFW includes valuable diversity, it has limitations in covering extreme lighting or highly varied facial angles that might appear in practical applications. To address these limitations, we could supplement the LFW dataset with additional images that capture a wider range of facial variations and environmental conditions. By enhancing the diversity of data, the model would be better equipped to handle unexpected conditions, achieving greater accuracy and reliability.

4.4 Key learning

The implementations of face recognition underscored several valuable lessons. One of the most significant lessons was recognizing how crucial high-quality data is to achieving accuracy in face recognition. Starting with the Labelled Faces in the Wild dataset, we quickly saw that while it offered a solid base, it had limitations in terms of lighting variations, facial angles, and obstructions. As a result, the model struggled in initial tests with only 7% accuracy. This showed us that a model is only as strong as the data it learns from. Moving forward, we understood that more diverse data is essential, and we realized that additional datasets could further enhance our model's adaptability to real-world scenarios.

The process of incrementally improving accuracy from 7% to 80% highlighted how machine learning models are well-developed through continuous adjustment rather than one-time setup. Achieving high accuracy requires carefully adjusting parameters, like key point thresholds in SIFT, re-running tests, analyzing results, and repeating the cycle. Each adjustment contributed to incremental improvements, teaching us the value of patience and precision in fine-tuning models. We learned that even small parameter changes could significantly impact results, and that achieving optimal performance requires a detailed, iterative approach.

Early on, we underestimated the role of preprocessing in enhancing model performance. Filtering out low-quality images, applying noise reduction, and shuffling the data were all necessary steps to ensure the model had clean, consistent data to learn from. These preprocessing steps helped the model focus on meaningful facial features without being distracted by irrelevant background noise or inconsistencies. This experience reinforced that spending time on preprocessing is not optional, but a crucial stage that directly influences model accuracy and stability.

Another important lesson was understanding the importance of accuracy and processing speed. While we achieved higher accuracy by incorporating complex techniques like CNNs and SIFT, we found that these methods increased computation time, making real-time processing challenging. This experience taught us that high accuracy alone is not enough in real-world applications, models must also operate efficiently, especially in security and authentication systems that require quick responses.

Throughout the project, we had to change approaches repeatedly as new challenges come. From dealing with data limitations to adjusting parameters and adding CNNs to enhance accuracy, we learned that machine learning projects are not straightforward. Adaptability is essential when unexpected obstacles arise, and having a flexible mindset allowed us to respond effectively to these challenges. Security was also a key focus, emphasizing the importance of spoofing detection techniques. By integrating the Difference of Gaussians, we improved the model's ability to detect faces. This experience underscores the importance of building secure models, knowing that problem-solving in machine learning often requires creative, iterative approaches.

In conclusion, the project was a comprehensive learning experience, showing that successful face recognition models require not only advanced algorithms but also careful attention to data quality, iterative testing, and a balance between accuracy and efficiency.