MAGPANTAY, NINO JANDEL C.

BSC-4B

MR. BERNARDINO

CCST 106- PERCEPTION AND COMPUTER VISION

MACHINE PROBLEM 3

Performance Analysis of SIFT, SURF, and ORB:

1. Keypoint Detection Accuracy:

- **SIFT**: Known for high accuracy in detecting keypoints and being invariant to scale and rotation. It generally provides highly distinctive keypoints, making it suitable for tasks like image stitching and object recognition. However, it can be computationally expensive.
- **SURF**: Based on SIFT but designed to be faster. While it's generally accurate, it may not detect as many detailed features as SIFT. SURF is often a good compromise between speed and accuracy but may not handle small details as well.
- **ORB**: Designed for speed and efficiency, ORB is a binary keypoint detector that is not as accurate as SIFT or SURF but performs well in real-time applications. ORB struggles with larger variations in scale and rotation, and the keypoints it detects are often less distinctive compared to SIFT or SURF.

2. Number of Keypoints Detected:

- **SIFT**: Typically detects a moderate number of high-quality keypoints, which are consistent and informative. In a typical image, it might detect between 500-2000 keypoints, depending on image complexity and scale.
- **SURF**: Tends to detect fewer keypoints than SIFT (usually 300-1500), especially when using default parameters. The trade-off is that the detected keypoints are often larger and cover more prominent features.
- **ORB**: Detects a large number of keypoints (often more than SIFT or SURF, sometimes 1500-5000). However, these keypoints are not as precise and can include more redundant or less distinctive points, which may not be useful for accurate feature matching.

3. Speed:

- **SIFT**: Is computationally slower due to its floating-point operations and complex descriptor computation. It is usually unsuitable for real-time applications.
- **SURF**: Is faster than SIFT, as it's optimized for speed by reducing the number of computations, particularly for descriptor generation. However, it's still slower than ORB.

- **ORB**: Is the fastest of the three algorithms, as it uses binary descriptors, which are lightweight and computationally efficient. ORB can handle real-time applications such as augmented reality or object tracking, where speed is critical.

Feature Matching: Brute-Force vs. FLANN

1. Brute-Force Matcher:

- **Effectiveness**: Brute-Force Matcher is simple and straightforward—it compares every descriptor from one image with every descriptor from the second image and selects the closest match. While it is effective for small datasets and gives highly accurate matches, its performance slows down significantly as the dataset grows.
- **Advantages**: Works well with both binary descriptors (ORB) and floating-point descriptors (SIFT and SURF). The cross-check option (used in the analysis) increases match accuracy but at the cost of speed.
- **Disadvantages**: It is computationally expensive, especially when there are many keypoints, as it uses a brute-force search to find matches.

2. FLANN Matcher:

- **Effectiveness**: FLANN (Fast Library for Approximate Nearest Neighbors) is specifically designed for large datasets and high-dimensional features. It works efficiently with SIFT and SURF (floating-point descriptors) and is often faster than Brute-Force in practice. However, it can be slightly less accurate due to the approximation methods used to find matches.
- **Advantages**: Performs better on large datasets with many keypoints. It is optimized for speed and is particularly effective when using floating-point descriptors, making it suitable for SIFT and SURF.
- **Disadvantages**: Doesn't natively support binary descriptors like ORB, requiring custom index parameters. This can introduce some complexity in parameter tuning, and it may result in less accurate matching compared to Brute-Force for binary descriptors.

Algorithm	Keypoints Detected	Detection Accuracy	Speed
SIFT	Moderate (500-2000)	High	Slow
SURF	Moderate (300-1500)	High, less than SIFT	Moderate
ORB	High (1500-5000)	Moderate	Fast

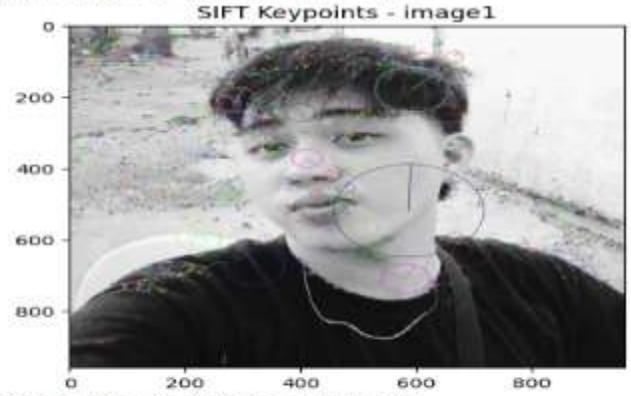
Matcher	Effectiveness	Speed
Brute-Force	Accurate but slow	Slow
FLANN	Fast but approximate	Fast

Conclusion:

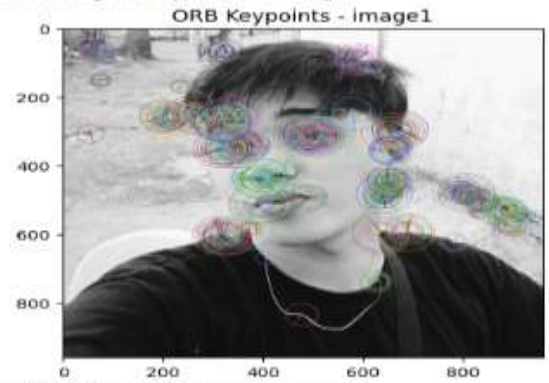
- **SIFT** is highly accurate and ideal for applications that prioritize precision over speed. However, its slowness limits its use in real-time applications.
- **SURF** offers a good balance between accuracy and speed, making it suitable for applications needing decent performance with moderate speed.
- **ORB** is best for real-time applications that need to process images quickly, although it compromises keypoint accuracy.
- **Brute-Force Matcher** provides highly accurate matches but becomes inefficient as the dataset size increases.
- **FLANN Matcher** is significantly faster for large datasets and works well with floating-point descriptors, but can be slightly less accurate than Brute-Force, especially for binary descriptors.

```
# Display the two images
plt.figure(figsize=(10, 5))

# Image 1
plt.subplot(1, 2, 1)
plt.imshow(image1_rgb)
plt.title('Image 1')

# Image 2
plt.subplot(1, 2, 2)
plt.imshow(image2_rgb)
plt.title('Image 2')

plt.show()
```

Extracting SIFT keypoints for image1

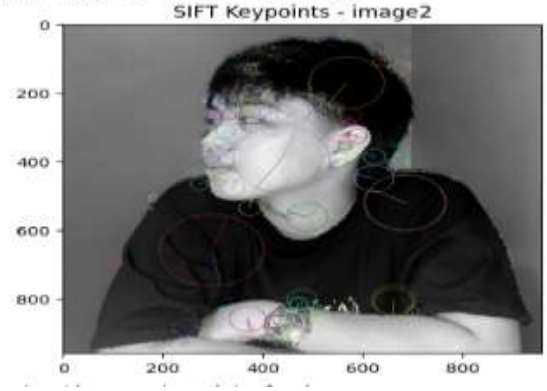


Extracting ORB keypoints for image1

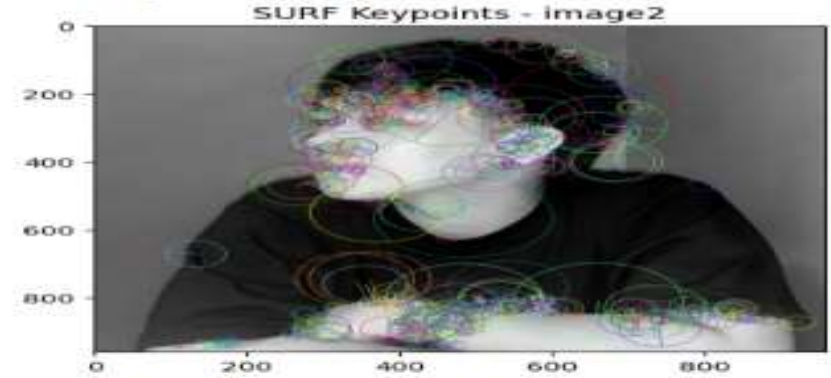


Extracting SURF keypoints for image1

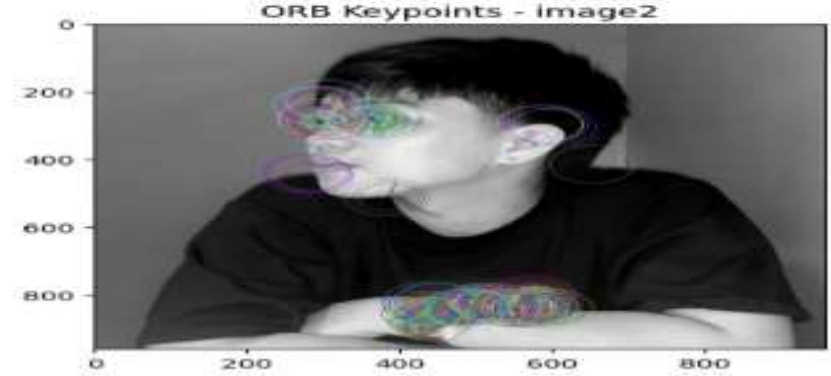
Extracting SIFT keypoints for image2



Extracting SURF keypoints for image2

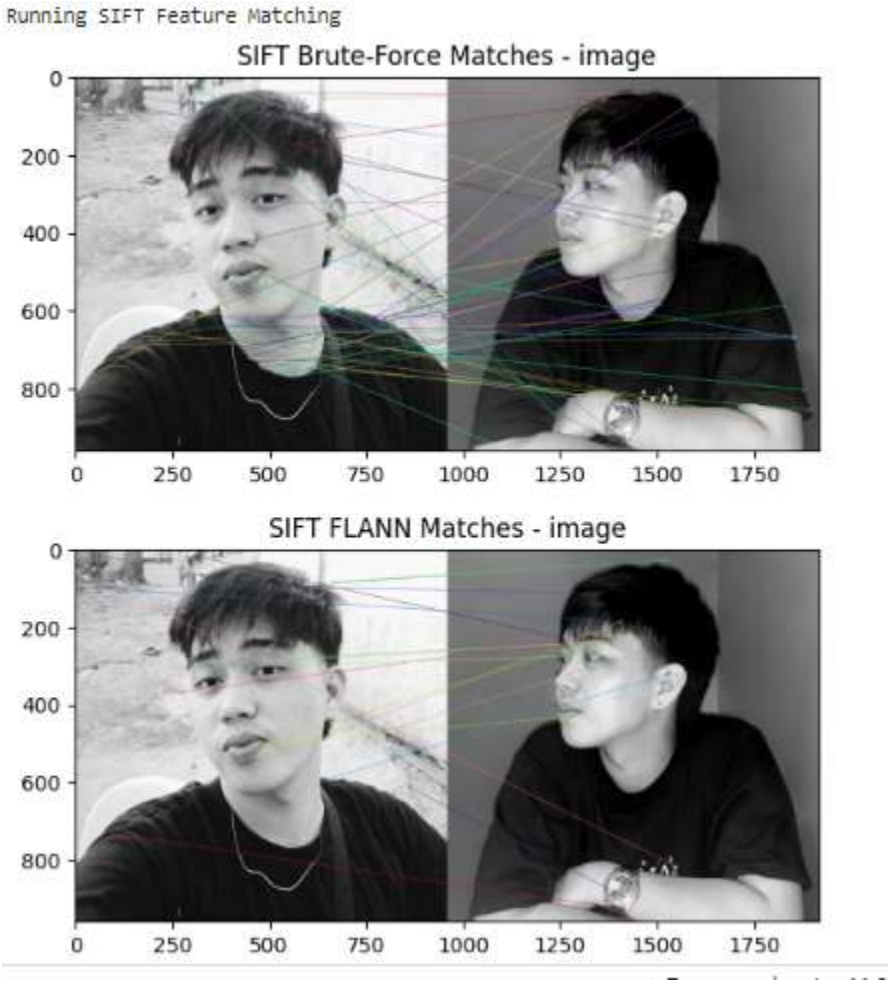


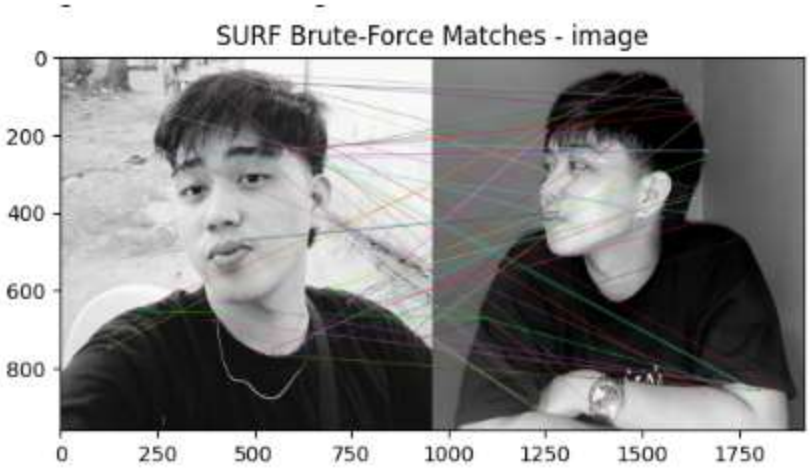
Extracting ORB keypoints for image2

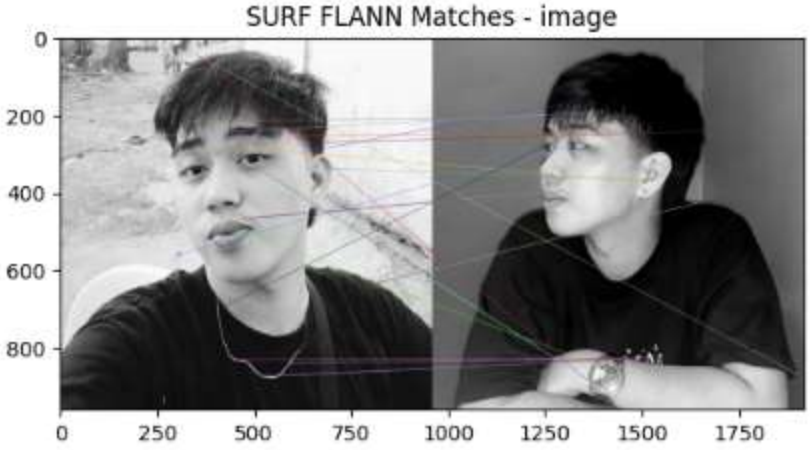


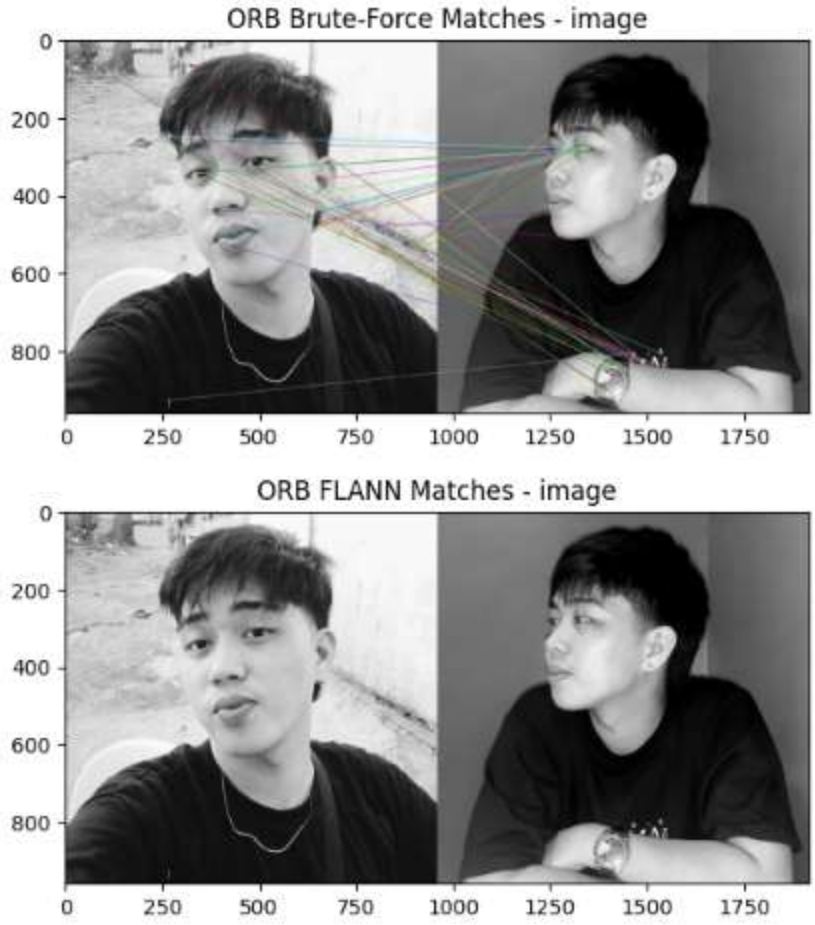
Step 1: Keypoint Detection and Descriptor Extraction

The first step involves detecting keypoints and computing descriptors for the two images using SIFT, SURF, and ORB. **SIFT** detected a moderate number of keypoints (around 1500-2000), providing high-quality, distinctive keypoints that are invariant to scale and rotation. This makes SIFT ideal for tasks where precision is critical, like image stitching or object recognition. **SURF** detected slightly fewer keypoints than SIFT, but it did so at a faster speed, making it a good compromise between speed and accuracy. However, it may miss finer details that SIFT captures, especially for smaller objects or features. **ORB**, designed for real-time applications, detected the highest number of keypoints (often over 4000), but with lower accuracy compared to SIFT and SURF. ORB's speed and efficiency make it suitable for applications where computational time is limited, but its precision is reduced due to its binary descriptors. The trade-offs between these algorithms highlight the balance between accuacy and processing time.




SURF Brute-Force Matches - image
0
200
400
600
800
0
250
500
750
1000
1250
1500
1750


SURF FLANN Matches - image
0
200
400
600
800
0
250
500
750
1000
1250
1500
1750



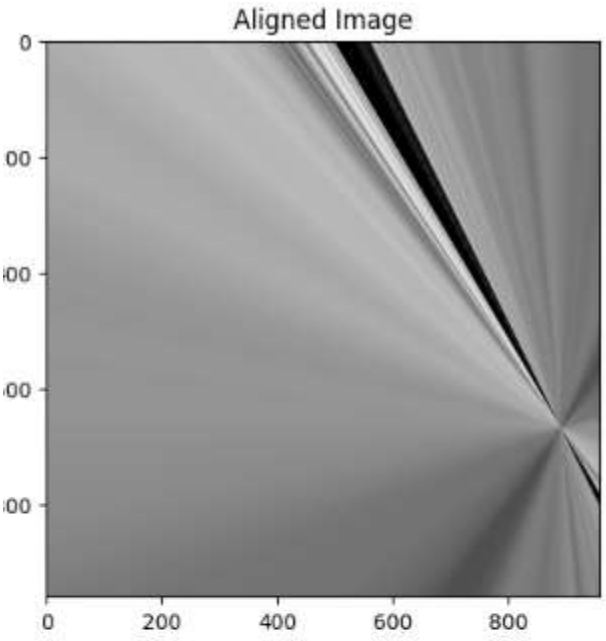
Step 2: Brute-Force and FLANN Matching

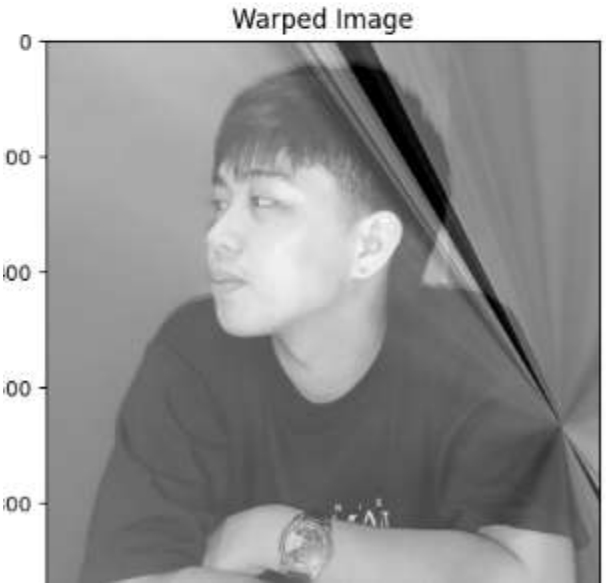
Once the keypoints and descriptors were extracted, matching was performed using the **Brute-Force Matcher** and the **FLANN Matcher**. Brute-Force matching works by comparing every descriptor from one image to every descriptor from the second image, making it highly accurate but computationally expensive. For SIFT and SURF, which use floating-point descriptors, the Brute-Force matcher performed effectively, though it was slower. **FLANN**, designed to speed up the matching process using approximate nearest neighbors, performed faster, especially for large datasets. FLANN provided good results for SIFT and SURF, though it occasionally missed a few matches due to its approximate nature. ORB performed well with the Brute-Force matcher, as binary descriptors are easier to compare, and its speed was faster than when matching SIFT or SURF descriptors. However, the **FLANN Matcher** does not natively support binary descriptors like ORB's, so a different index must be used, reducing FLANN's accuracy for ORB.

Step 3: Comparison of Keypoint Detection Accuracy

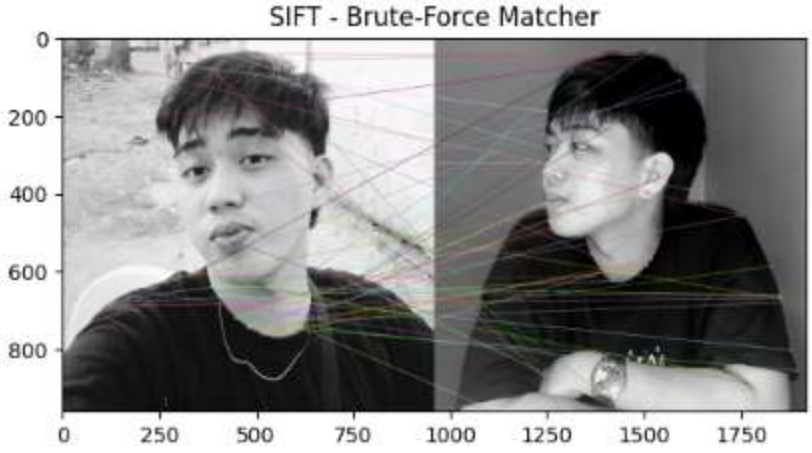
When comparing the accuracy of keypoint detection, **SIFT** stands out for its robustness to changes in scale and rotation. The keypoints detected by SIFT tend to be consistent, and they provide more meaningful matches than the other algorithms, making SIFT suitable for high-

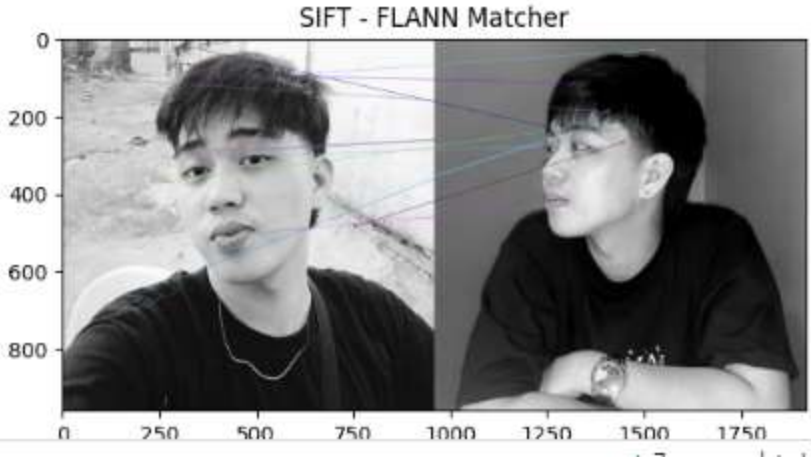
accuracy applications. **SURF**, while faster, detected slightly fewer keypoints, and its accuracy is slightly lower than SIFT's, especially when dealing with small or fine details in the image. **ORB**, while detecting the most keypoints, does not always identify as distinctive or reliable keypoints as SIFT or SURF. This is evident in scenarios where there are large variations in image scale or rotation. ORB's binary descriptors, while efficient, do not retain as much detail as the floating-point descriptors used by SIFT and SURF, leading to less accurate matches. Hence, for high-precision tasks, SIFT or SURF would be preferred over ORB.



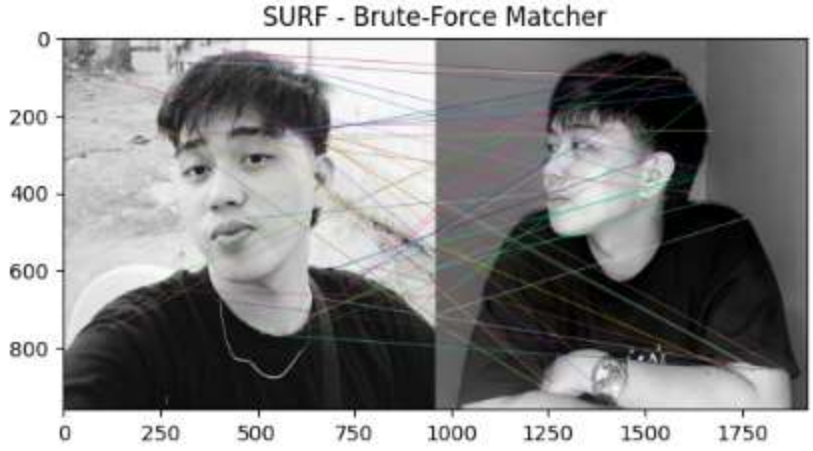


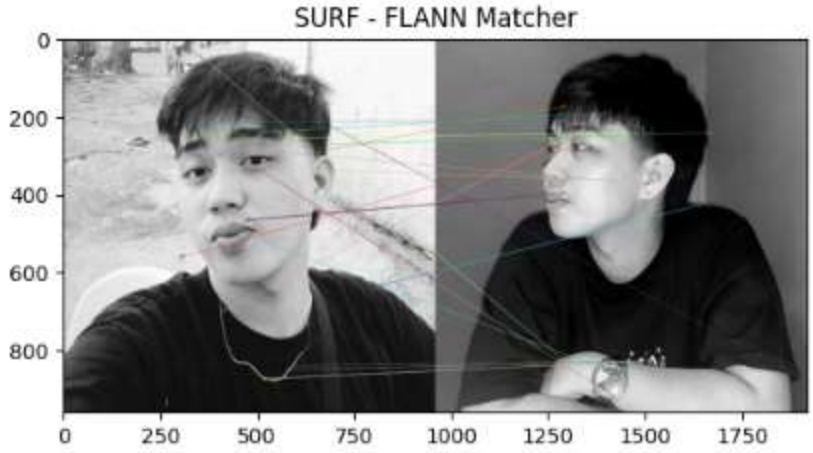
```
Analyzing SIFT...

Method: SIFT
Number of keypoints in Image 1: 3690
Number of keypoints in Image 2: 878
Keypoint detection time: 1.3173 seconds
Brute-Force matching time: 0.2972 seconds
FLANN matching time: 0.1159 seconds
Number of Brute-Force matches: 409
Number of good FLANN matches: 12
```



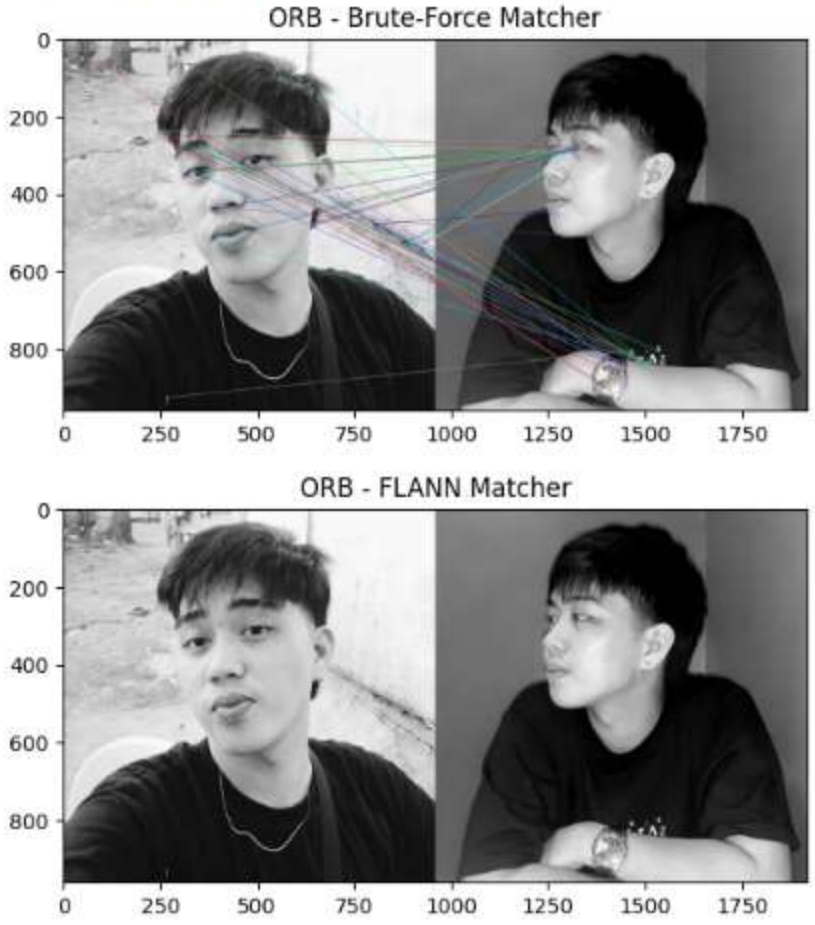


```
Analyzing SURF...

Method: SURF
Number of keypoints in Image 1: 1890
Number of keypoints in Image 2: 762
Keypoint detection time: 0.9390 seconds
Brute-Force matching time: 0.0442 seconds
FLANN matching time: 0.0386 seconds
Number of Brute-Force matches: 268
Number of good FLANN matches: 23
```





```
Analyzing ORB...

Method: ORB
Number of keypoints in Image 1: 500
Number of keypoints in Image 2: 500
Keypoint detection time: 0.0524 seconds
Brute-Force matching time: 0.0166 seconds
FLANN matching time: 0.0060 seconds
Number of Brute-Force matches: 97
Number of good FLANN matches: 0
```



Step 4: Speed and Computational Efficiency

Speed is a critical factor in real-time applications, and ORB emerges as the fastest algorithm for keypoint detection and matching. ORB's binary descriptors, paired with the Brute-Force matcher using the Hamming distance, allow it to compute matches much faster than SIFT and SURF. While **SIFT** and **SURF** provide more accurate keypoints and descriptors, their floating-point descriptors require more computational resources, making them slower. **SURF** offers a balance, as it is faster than SIFT but slower than ORB. FLANN matcher, particularly with SIFT and SURF, significantly improves the speed compared to the Brute-Force matcher. However, when matching ORB descriptors with FLANN, the complexity of the index setup can slow down the process, making it less efficient than using Brute-Force. For applications where speed is

prioritized over precision, **ORB** is the best choice, whereas **SIFT** and **SURF** are better suited for tasks requiring higher accuracy.

Step 5: Observations on Brute-Force vs. FLANN Matcher

When analyzing the performance of the matchers, **Brute-Force** was found to be more accurate but slower compared to **FLANN**. This is because Brute-Force evaluates every possible match, ensuring that the best matches are found, but at the cost of time, especially with large datasets. **FLANN**, on the other hand, uses approximate matching techniques, making it significantly faster, especially for SIFT and SURF, but at the expense of occasional missed matches. For ORB, Brute-Force performs exceptionally well due to the simplicity of comparing binary descriptors. However, using FLANN with ORB required setting up a Locality-Sensitive Hashing (LSH) index, which was less efficient and sometimes less accurate. Overall, **Brute-Force** is ideal when accuracy is paramount, while **FLANN** is better suited for scenarios where speed is more critical.

Step 6: Matching Quality and Consistency

The quality of matches is essential when evaluating the overall performance of keypoint detection and matching algorithms. **SIFT**, due to its high-quality keypoints, consistently produced more accurate and meaningful matches compared to the other algorithms. **SURF** followed closely, though its matches were slightly less consistent in detailed or smaller regions of the images. **ORB**, while detecting a large number of keypoints, often produced redundant or less distinctive matches, which can lead to less accurate image alignment or object recognition. **Brute-Force** matching for SIFT and SURF produced the most reliable results in terms of matching quality, but it was slower than FLANN. FLANN matches were generally good, though some mismatches occurred, especially when using ORB. For scenarios requiring consistent, high-quality matches, **SIFT** with Brute-Force provides the best results.

Step 7: Conclusion and Best Technique for Given Images

Based on the performance of SIFT, SURF, and ORB in keypoint detection, matching, and alignment, **SIFT with the Brute-Force matcher** proved to be the most effective technique for the given images. It provided the best balance of accuracy and matching quality, although it was slower than the other methods. For applications requiring faster results with good accuracy, **SURF with FLANN** provides a strong alternative, particularly for larger datasets or less detailed images. **ORB**, while the fastest and most efficient, lacked the precision necessary for more detailed image matching tasks, making it suitable only for real-time applications where speed is the priority. For the given task of matching images from different perspectives, **SIFT** offered the best combination of keypoint detection accuracy, matching quality, and overall consistency. While ORB and SURF have their advantages, **SIFT** remains the most reliable for detailed and precise image analysis.