Final Instructions Document

📜 Instructions to Load and Run the ITQ-Based Video Retrieval Algorithm

🔷 Prerequisites

Before running the algorithm, ensure you have:

- Python Installed (Recommended: Python 3.8+)

- Required libraries installed (numpy, opencv-python, scikit-learn, moviepy)

- A sample video file placed in the "videos" folder

- The project folder organized as follows:

```
ITQ_Image_Hashing/
 ├── delete_images.py
 ├── image_loader.py
 ├── itq.py
 ├── itq_hash.py
 ├── itq_hashing.py
 ├── main.py
 ├── retrieval.py
 ├── video_loader.py
 ├── images/       (contains sample images)
 ├── videos/       (contains sample video(s), e.g., sample.mp4)
└── frames/       (will be created when extracting frames)
```

1□Setup and Load the Algorithm

Step 1: Open Command Prompt

Press Win + R, type "cmd", and hit Enter.

Step 2: Navigate to the Project Directory

~~~
```sh

cd C:\Users\Ryan\Desktop\ITQ_Image_Hashing
~~~

Step 3: (Optional) Create a Virtual Environment

```
python -m venv venv

venv\Scripts\activate
```

2□Install Dependencies

```
pip install numpy opencv-python scikit-learn moviepy
```

3□Run the Algorithm Step by Step

Step 1: Delete Old Frames (if any)

```
python delete_images.py
```

Expected Output:

✅ Deleted folder: frames

Step 2: Extract Frames from the Video

```
python video_loader.py --video videos/sample.mp4
```

Expected Output:

✅ Extracted 2300 frames from videos/sample.mp4

Step 3: Run the Main Program

```
python main.py
```

Expected Output:

✅ Running ITQ-based Video Retrieval...

✅ PCA Output Shape: (2300, 32)

✅ Best match found: frames/frame_1190.jpg (Hamming Distance: X)

The system processes all frames and finds the best matching frame for the query.

4□Troubleshooting Tips

- **Module Not Found Errors:** Ensure you have installed all dependencies.
- **Virtual Environment Issues:** Run "Set-ExecutionPolicy Unrestricted -Scope Process" if needed.
- **PCA Errors:** Adjust n_components in the code if necessary.

5□Running on Another Machine

1. Copy the entire ITQ_Image_Hashing folder to the new machine.
2. Open Command Prompt and navigate to the folder.
3. Create/activate a virtual environment and install dependencies.
4. Run the algorithm steps as above.

Summary:

1. Open CMD and navigate to the project folder.
2. Optionally create and activate a virtual environment.
3. Install dependencies.
4. Run delete_images.py, then video_loader.py, then main.py.
5. The output will display the best matching frame and its distance.

Your algorithm is now fully loaded and ready to run!

How to Run the Algorithm (CMD Commands)

1. **Navigate to the Project Directory:**
2. cd C:\Users\Ryan\Desktop\ITQ_Image_Hashing
3. **(Optional) Activate the Virtual Environment:**
4. venv\Scripts\activate
5. **Delete Old Frames:**
6. python delete_images.py
7. **Extract Frames from Video:**
8. python video_loader.py --video videos/sample.mp4 --output frames
9. **Run the Main Program:**
10. python main.py

11. **Expected Output:**
12. ✅ Extracted 2300 frames from videos/sample.mp4
13. ✅ PCA Output Shape: (2300, 32)
14. ✅ Best match found: frames/frame_1190.jpg (Hamming Distance: 8)

Explanation of the Fixes:

- In **itq.py**, the ITQ training is now done on a matrix with shape (num_samples, bits). The rotation matrix R is initialized as an orthogonal matrix with shape (bits, bits), ensuring that matrix multiplication data @ R is valid.
- In **itq_hash.py**, PCA is applied to simulate feature reduction to 32 dimensions before ITQ training.
- In **retrieval.py**, all frames are loaded from the frames folder, ITQ is trained on these frames, and then the system finds the best matching frame based on Hamming distance.
- **main.py** runs the complete pipeline: it assumes that frames have been extracted (or it could call the extraction step separately) and then performs retrieval.
- **video_loader.py** extracts frames from the sample video, and **delete_images.py** cleans up the frames folder.