**How to Add More Images & Update Your Facial Recognition Project**

You can expand your training set anytime by following **this repeatable workflow**. Here’s a simple, step-by-step method for updating your files and retraining the model when you add more images, so your project always stays up-to-date.

**1. Add New Images to Your Images Folder**

* **Put your new images** (e.g., named 26.jpg, 27.jpg, etc.) into  
  D:\FacialRecognition Project\images
* Make sure each filename is unique and consistent (like numeric names or descriptive person labels).

**2. Extract Landmarks for All Images**

* **Rerun your facial landmark extraction script** (facial\_landmark\_extraction.py)  
  This will process *all* images in the folder and update landmarks.csv with the latest landmarks.

shell

cd "D:\FacialRecognition Project"

python facial\_landmark\_extraction.py

* Each image’s landmarks are added as a new row in landmarks.csv.

**3. Update Label List in train\_knn.py**

* Update the names list so it matches the number of total training images:
  + For 27 images, use:

python

names = [f"Person{i}" **for** i **in** range(1, 28)]

* + For 35 images, use:

python

names = [f"Person{i}" **for** i **in** range(1, 36)]

* Make sure the length of the list matches the number of rows in landmarks.csv.  
  You can check the row count using:

python

**import** pandas **as** pd

df = pd.read\_csv('landmarks.csv')

**print**(len(df))

**4. Relabel & Retrain**

* **Run your updated labeling/training script:**  
  (train\_knn.py updates landmarks\_labeled.csv, retrains KNN model, and saves knn\_face\_model.pkl)

shell

python train\_knn.py

* If you wish, you can adjust names to more descriptive text (e.g., actual names instead of "PersonNN").

**5. Test Recognition**

* You can now use predict\_face.py as before.  
  No changes needed here unless you want to recognize one of the new images.

**Summary Table**

| **Step** | **What to Do** | **Where/What File** |
| --- | --- | --- |
| Add new images | Place images in images/ folder | D:\FacialRecognition Project\images |
| Extract landmarks | Rerun facial\_landmark\_extraction.py | Updates landmarks.csv |
| Update label list | Make names list match new total image count in train\_knn.py | D:\FacialRecognition Project\train\_knn.py |
| Relabel and retrain | Rerun train\_knn.py | Updates model and labeled CSV |
| Test new recognition | Use predict\_face.py as before | D:\FacialRecognition Project\predict\_face.py |

**Tips for Easy Expansion**

* Always keep your images and CSV files organized.
* To avoid mistakes, double-check the number of images and update label lists accordingly.
* Consider automating label assignment based on filenames if you start using actual names (ask for help if you want that feature).
* Every time you add/remove images from the folder, repeat steps 2–4.

**If you want even easier automation (e.g., automatically generate labels from image filenames instead of "PersonNN"), just ask and I’ll provide ready-made code for you!**

1. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/75162262/4139a27f-fdfd-4b0f-8939-14f623e64c0c/facial_landmark_extraction.py>
2. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/75162262/e38a2ed3-3007-457a-b737-49103cc927d2/get_landmarks.py>
3. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/75162262/b4b44081-2c3b-4fb0-b196-103a15ceda82/landmarks.csv>
4. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/75162262/02460c59-c684-4216-9031-dff4531bd111/landmarks_labeled.csv>
5. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/75162262/d266ada0-5437-433d-a2f1-42ed9107781c/predict_face.py>
6. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/75162262/696ae8b7-4f48-4deb-9627-9f1965b728ae/train_knn.py>
7. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/75162262/27e256c2-3408-4209-8d67-adcced0240ca/trimming.py>