# Face Rec Notes

See <https://nikolak.com/pyqt-threading-tutorial/> for how to do a threaded GUI

Or this may be better

<https://mayaposch.wordpress.com/2011/11/01/how-to-really-truly-use-qthreads-the-full-explanation/>

## Narrative

1. Open database.
2. Read list of all known faces from database
3. read or calculate their face encoding signature into a known face list.
4. Read list of images for tagging from database.
5. For each image that needs tagging, open the image, rotate it and scan it for faces locations.
   1. For each face location create a face encoding signature and save the face location and face encoding to the faces table
      1. For each reference face (is\_reference\_face = 1) in the face\_table compare with this known face
      2. Add distance between faces to a face list

## Face Table

1. face\_id as INTEGER UNIQUE – unique ID of this face PRIMARY KEY
2. media\_id as integer - id of source image in media\_table ), FOREIGN KEY REFERENCES media\_table(id) )
3. face location coordinates as INTEGER NOT NULL
   1. face\_loc\_top
   2. face\_loc\_ right
   3. face\_loc\_bottom
   4. face\_loc\_ left
4. face\_encoding as binary BLOB NOT NULL (array of 128 floats)
5. is\_reference\_face as INTEGER – TRUE if this is a confirmed reference face to match against other faces to confirm it’s the same person (it should be assumed that all reference faces are also confirmed faces by a human but not all human confirmed faces are reference faces to match against because there can be many faces images for an individual but it’s generally faster to only match against one or a few of them for identification purposes)
6. is\_confirmed\_face as INTEGER – TRUE if this is a face match that has been confirmed by a human rather than an automated guess at a face match
7. tag\_id as string if known or NULL if not known (tag\_id of person identified in image at face location)

### Use Cases

1. Find all untagged faces for tagging
2. Find all known faces (i.e. those with tag names) that have NOT been confirmed so they can be displayed and confirmed
3. Find all known faces (i.e. those with tag names) that are “reference faces” to match other faces against

CREATE TABLE face\_table (

face\_id INTEGER UNIQUE,

media\_id INTEGER NOT NULL,

scaled\_image\_max\_height INTEGER NOT NULL,

face\_loc\_top INTEGER NOT NULL,

face\_loc\_right INTEGER NOT NULL,

face\_loc\_bottom INTEGER NOT NULL,

face\_loc\_left INTEGER NOT NULL,

face\_encoding\_blob BLOB NOT NULL,

is\_reference\_face INTEGER NOT NULL,

is\_confirmed\_face INTEGER NOT NULL,

tag\_id INTEGER,

PRIMARY KEY (face\_id),

FOREIGN KEY (media\_id) REFERENCES media\_table(id),

FOREIGN KEY (tag\_id) REFERENCES tag\_table(id) );

CREATE INDEX face\_table\_face\_to\_media\_index ON face\_table (face\_id, media\_id);

CREATE INDEX face\_table\_face\_tag\_index ON face\_table(tag\_id, face\_id);

CREATE INDEX face\_table\_reference\_face\_index ON face\_table(is\_reference\_face, face\_id);

# Quick Start

1. Setup simple gui
   1. start and stop button and a space for an image and progress bar
   2. Disable the stop and enable the start button
2. When user clicks start
   1. Search catalog for images to check
   2. Initialise progress bar with count of images to check
   3. Start a new thread to search for faces, passing it the database rows with images to search
   4. For all images marked for tagging in the catalog not already in face\_table
      1. Find faces in the image
      2. For each face
         1. Calculate face encoding
         2. Save the face encoding to the face\_table
      3. Image tagging complete
         1. Send message to gui thread (with drawn on image?) with tagged faces telling it an extra face has been tagged hence to update the progress bar
         2. Check that there has been no signal from the user that we should cancel work and if so, close andexit the thread

## Threading

1. When the user clicks start, initiate threads that populate a face\_data\_queue and then go into a loop
   1. If face\_data\_queue is empty:  
       sleep(20) # sleep 20 ms  
      QTCore.QEventLoop.processEvents()  
      else:  
       add item on top of queue to database and display list

## Face Clustering Chinese Whispers

<https://blog.csdn.net/LIYUAN123ZHOUHUI/article/details/70312716>

<http://alexloveless.co.uk/data/chinese-whispers-graph-clustering-in-python/>