MSc Vibot/Computer Vision/MAIA

Applied Mathematics
Homework

Désiré Sidibé
ddsidibe@u-bourgogne.fr

Face recognition using PCA

About data normalization

In the face recognition project, we have to apply a normalization procedure to ensure that all face images are, more or less, aligned, i.e. to account for scale, orientation and localtion varaiations.
Here we provide a, hopefully, clearer description of this procedure.

To start with, we are given a set of facial features F_i for every face image in our dataset. And, we are also given a set of predifined locations for these features F_p in a reduced 64×64 window. Here, both F_i and F_p are vectors of size 10 (since we have 2 positions for each of the 5 facial features).

The goal is to find a vector $\bar{F}$ that will correspond to the "average" location to which to map each F_i, and the algorithm works as follows:

1. Initialize $\bar{F}$ with the features from one image, for example $\bar{F} \leftarrow F_1$
2. Start iterating; for each iteration t
   (a) Use a function
   $$\texttt{[A,b] = FindTransformation(}\bar{F}\texttt{,}\ F_p\texttt{)}$$
   to find the best transformation, given by (A, b), that maps the features in $\bar{F}$ to those in F_p.
   (b) Apply this transformation (A, b) to $\bar{F}$ to get $\bar{F}'$, using a funtion
   $$\texttt{[}\bar{F}'\texttt{] = ApplyTransformation(}A\texttt{,}\ b\texttt{,}\ \bar{F}\texttt{)}$$
   Set $\bar{F} \leftarrow \bar{F}'$.
   (c) For every image, i.e. for each F_i do
   $$\texttt{[A,b] = FindTransformation(}F_i\texttt{,}\ \bar{F}\texttt{)}$$
   Apply this transformation (A, b) to F_i to get F_i' :
   $$\texttt{[}F_i'\texttt{] = ApplyTransformation(}A\texttt{,}\ b\texttt{,}\ F_i\texttt{)}$$
   (d) Take average all of F_i' and set it to $\bar{F}$:
   $$\bar{F}_t \leftarrow \frac{1}{N}\sum_{i=1}^{N} F_i'$$
   .
3. Go to setp 2, and repeat until convergence (for example, when $\|\bar{F}_t - \bar{F}_{t-1}\| \leq \epsilon$, for a threshold ϵ).

So, you need to write two functions, that are going to be used many times. You could code them directly in the algo, but writing separate functions makes it easier to understand the code and to debug.
The two functions are:

`[A,b] = FindTransformation(`F_1`, `F_2`)`

that find the best transformation given two features vectors F_1 and F_2 such that $F_2 = AF_1 + b$, and

`[`F_2`] = ApplyTransformation(`A`, `b`, `F_1`)`

that apply a transformation to F_1 such that $F_2 = AF_1 + b$.

NOTE
Once the algorithm above has converged, you get a final feature vector $\bar{F}$. Then you have to use this final $\bar{F}$ and apply it to each image.
So for every image I_i and its facial features F_i, you do the following

- Find the best transformation (A, b) to map F_i to $\bar{F}$:

  `[A,b] = FindTransformation(`F_i`, `$\bar{F}$`)`

- Apply this transformation (A, b) to the image I_i to get a reduced size 64×64 image

  For this step, it is better to use an inverse mapping strategy, i.e.

  1. create an empty 64×64 image I_i^{small}
  2. for every pixel (x, y) in I_i^{small}, apply the inverse transformation $(A, b)^{-1}$ and find the corresponding pixel (x', y') in the image I_i
  3. set $I_i^{small}(x, y) \leftarrow I_i(x', y')$

- Save this 64×64 image for next steps

VOILA ! C'EST FACILE !