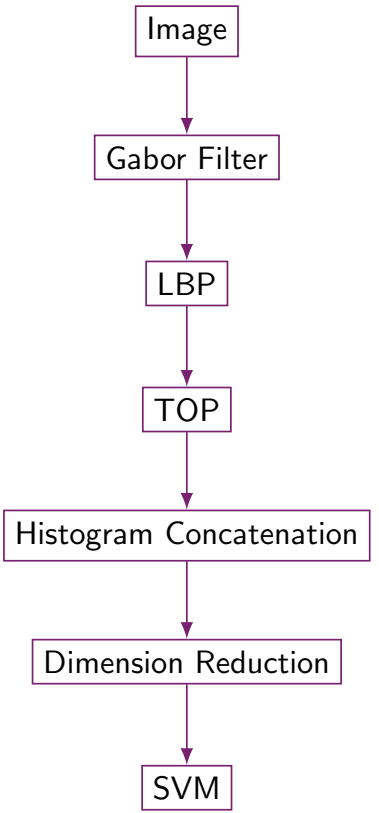
ESGI100: Crowd Emotion - Gabor Filter Selection and Computational Processing for Emotion Recognition

Erhan Coskun, Torran Elson, Daniel Jabryes, Sean Lim, James Mathews, Imad Muhi El-Ddin, Nikolai Nowaczyk, Rafał Prońko, Patrick Raanes, David Kofoed Wind, Matthew Woolway

ESGI100

April 11, 2014

Problem and Approaches



- Can we infer an optimum subset of 18 Gabor Filters for classification?
- Speed up using FFT
- Dimension reduction
- LBP: why 59 features rather than 256?
- Gridding

Finding the Optimum Subset of Gabor Filters

0,488713	0,531759	0,5301
0,483849	0,50977	0,45356
0,474587	0,467798	0,51598
0,494785	0,394204	0,50856
0,47241	0,459189	0,43952
0,488301	0,488352	0,5538
0,462362	0,50294	0,44445
0,532812	0,469046	0,47031
0,528018	0,510238	0,51682
0,459712	0,516778	0,46747
0,503215	0,496067	0,48502
0,475401	0,47505	0,55442
0,497885	0,5061	0,48933
0,596578	0,693722	0,60911
0,755597	0,573194	0,57738
0,516895	0,547131	0,52234
0,510153	0,543315	0,57855
0,640927	0,556124	0,58658

0,812087	0,790414	0,830341
0,7866	0,795169	0,843658
0,783895	0,689591	0,793548
0,854423	0,877927	0,896334
0,877915	0,875895	0,874561
0,820737	0,804938	0,754799
0,896141	0,86656	0,90626
0,899448	0,914527	0,889571
0,866132	0,88609	0,854865
0,856489	0,859606	0,84175
0,901605	0,824795	0,876945
0,808892	0,862207	0,806295
0,862764	0,85459	0,854361
0,84645	0,866322	0,843059
0,814439	0,859716	0,850162
0,827946	0,857035	0,857391
0,859431	0,849256	0,848155
0,752165	0,737825	0,806632

Figure 1 : 2AFC scores for the 18 different Gabor filters. Left table is AU1 (eyes) and right table is AU27 (mouth). Left is split by frequencies and right is split by angles. The columns corresponds to 3 runs (because of random training/test splits).

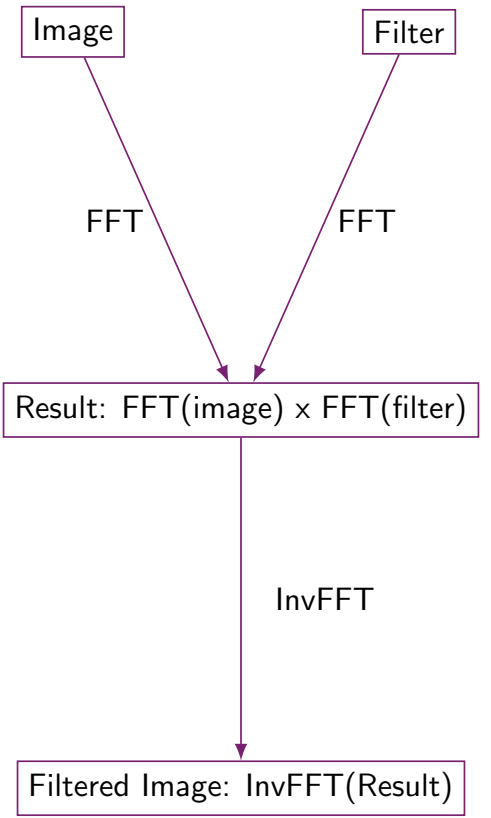
Traditional Gabor filter

- $g(x,y) = K \exp\{-\pi(a^2(x-x_0)_r^2) + b^2(y-y_0)_r^2)\}$
  $\cdot \exp\{j(2\pi(u_0 x + v_0 y) + P)\}$
- Gabor filter implementation on a spatial domain In MATLAB environment
- filter = gaussian.$*$sinuzoid; (no loop needed)

Picture Size (2^n)	CPU time
n=6	0.0625
n=7	0.9219
n=8	29.0625

Gabor filter using FFT

- Use the rule for the FFT of convolution:
- $FFT(image * filter) = FFT(image) \times FFT(filter)$



Gabor filter with FFT

- Typical results

picture size (2^n)	cputime
n=6	0.0156
n=7	0.0313
n=8	0.0625

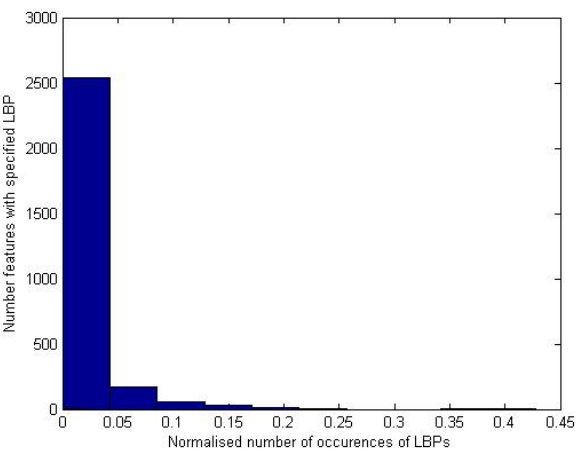
- If the image size is $\neq 2^n$ use FFTW

Separable methods (another alternative implementation)

- Gabor filter with orientation parallel to the image axes are separable mining that
- $G(x, y) = f(x)g(y)$
- This separation has been extended to work along to orientations $\theta = \frac{k\pi}{4}$ this aproach reduce the computation considerably.
- Traditional method 1.1 fps, Gabor with FFT 12.8 fps, Separable method 30 fps. See [Accurace and Efficient Computation of Gabor Features in Real-Time] for details.

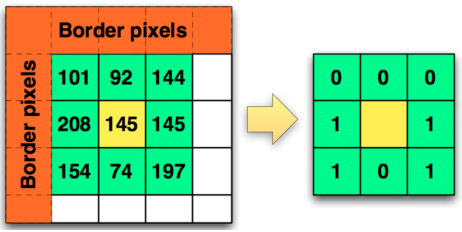
Dimension Reduction

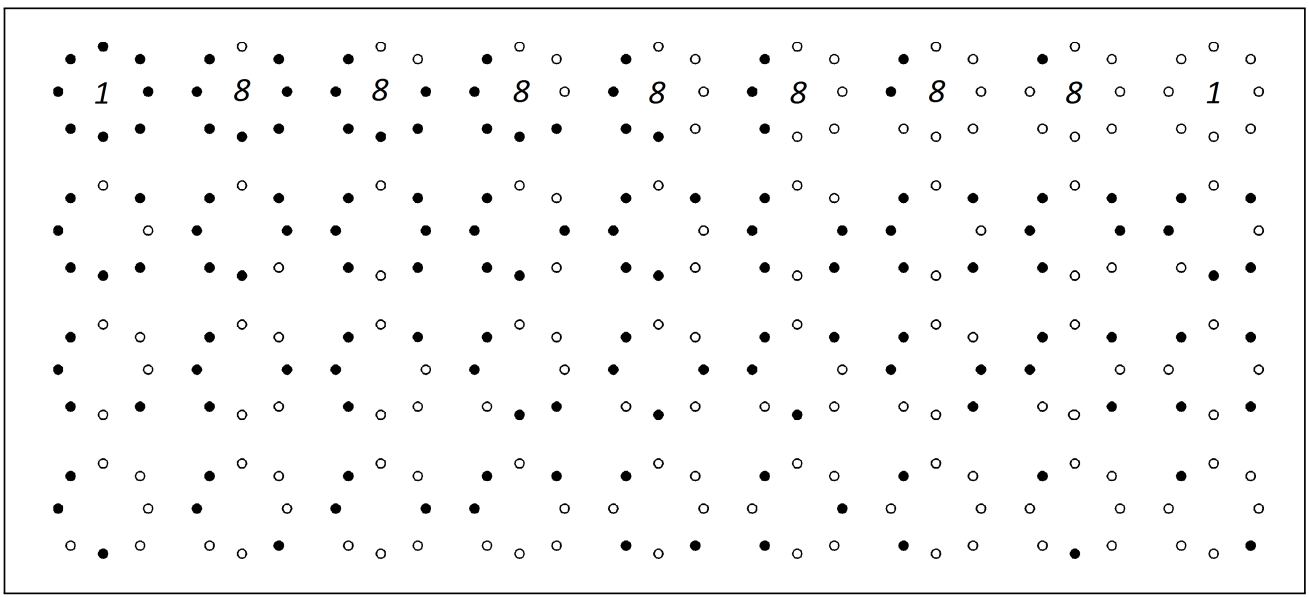
- Downsample by almost 70 % based on histogram below which shows all features per filter and take values > 0.1
- Reduce given Action Unit results for classifier training to 500 samples in order to balance the distribution of activated and non-activated results. This is because there were 100 times more non-activated results
- Classifier accuracy improved to 98% for example on AU1 by reducing features used to 442, from 50976



Local Binary Pattern (LBP): 59 Vs 256

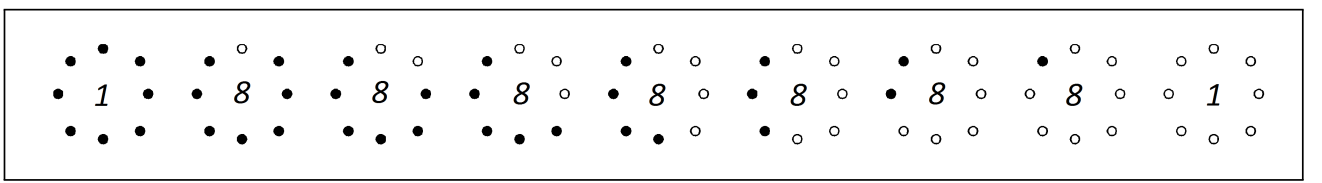
- For each pixel, we consider it's 8 neighbours.
- Assign 1's for values greater than pixel, else 0's.
- This gives 256 combinations.





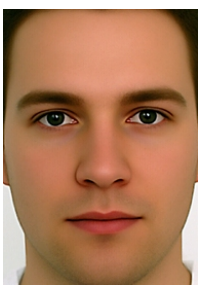
Local Binary Pattern (LBP): 59 Vs 256

- In facial images and texture images, experimental data has shown that roughly 90% of all LBP patterns come from the first row.
- Thus we only consider to use the 58 patterns below, with the 59th consisting of all other patterns.
- Does this 59th category tell us anything? (Currently used but slight speed up possible without it)



Grids

- Currently we take a rectangle of cropped face and compute binary patterns for every pixel.
- Some parts of this rectangle are clearly not useful for emotion recognition.
- Instead used ellipse stencil and keeps only 78% of the pixels to perform LBP on.



- Experimental data makes use of 4×4 grid for images. We can have different sized regions which would be useful for elliptic stencil.
- More regions increase accuracy but slows SVM.

Further Work

- Determine an optimal set of filters. For example using a greedy algorithm to search for optimal parameters.
- Implementing FFT
- Select appropriate filters for each AU using bottom up approach.
- Determine optimum grid size