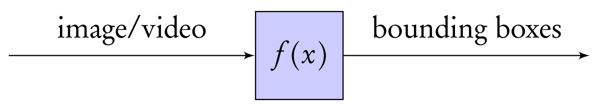
MB-LBP Face Detection with Gentle Adaboost

James Batten Valentin Iovene
batten_j@epita.fr toogy@lrde.epita.fr

October 5, 2016

Face detection is the task of localizing faces in images. One must distinguish between face *detection* and face *recognition*. The latter is the task of recognizing someone from an image of her face.

Face detection can be seen as a function f taking an image as input and returning *bounding boxes* around the faces contained in this image.



Face detection

— is used by *Facebook* to detect faces in images you share and make it easier for you to tag your friends,
— can be used to **track** faces in videos of people moving around (e.g. *Snapchat*'s face swap feature requires face detection),
— and is a prerequisite for any real-life applications of face recognition *(e.g. Law Enforcement localizing a suspect in a big city using video surveillance systems).*

The detection process needs to be fast enough to keep up with the demand (millions of users or streamed images).

There has been extensive research in this area but one of the major breakthrough happened in 2001 when *Paul Viola* and *Michael Jones* published their object detection framework [2] (which was named the "Viola-Jones framework").

Even though it was invented more than 15 years ago *(time of writing)*, *OpenCV*'s face detection implementation (source) is still based on it.

This article explains how the *Multi-Block Local Binary Patterns (MB-LBP)* visual descriptors (which will soon be introduced) can be used in lieu of the *Haar-like features* used by the original *Viola-Jones* framework for face detection. This approach was demonstrated by Zhang et al. in 2007. [1]

Contents

1 Principle

A face detection algorithm looks at *features* of an image to determine whether it is a face or not. In the Image Processing domain, they are called *visual descriptors*.

1.1 MB-LBP visual descriptors

MB-LBP of different sizes and at different locations are considered

```
std::vector<mblbp_feature> mblbp_all_features()
{
  std::vector<mblbp_feature> features;

  for(int block_w = min_block_size; block_w <= max_block_size; block_w += 3)
    for(int block_h = min_block_size; block_h <= max_block_size; block_h += 3)
      for(int x = 0; x <= initial_window_w - block_w; ++x)
        for(int y = 0; y <= initial_window_h - block_h; ++y)
          features.push_back(mblbp_feature(x, y, block_w, block_h));

  return features;
}
```

1.2 Feature selection using Gentle Adaboost

2 Detection procedure

2.1 Sliding window

The sliding window model is conceptually simple: **independently** classify all image patches as being object or non-object. Sliding window classification is the dominant paradigm in object detection and for one object category in particular – faces – it is one of the most noticeable successes of computer vision.

The following function generates a `std::vector` containing all potential windows in an image.

```
std::vector<window> get_potential_windows(const int img_w, const int img_h)
{
  std::vector<window> potential_windows;

  double w = initial_window_w; // current window width
  double h = initial_window_h; // current window height
  double scale = 1.0; // current scale
  double shift = scale * shift_delta; // current window shift (in pixels)

  while(w <= max_window_w && h <= max_window_h)
  {
    for(double x = 0.0; x < static_cast<double>(img_w) - w; x += shift)
    {
      for(double y = 0.0; y < static_cast<double>(img_h) - h; y += shift)
        potential_windows.push_back(window(x, y, w, h, scale));
```

```
    }

    // update values for next loop
    scale = scale * scaling_factor;
    shift = scale * shift_delta;
    w = w * scaling_factor;
    h = h * scaling_factor;
  }

  return potential_windows;
}
```

2.2 Cascade of classifiers

3 Classifier training process

3.1 Data

3.1.1 Dataset

3.1.2 Preprocessing

3.2 Gentle Adaboost

References

[1] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li. *Learning Multi-scale Block Local Binary Patterns for Face Recognition*, pages 828–837. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[2] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518 vol.1, 2001.