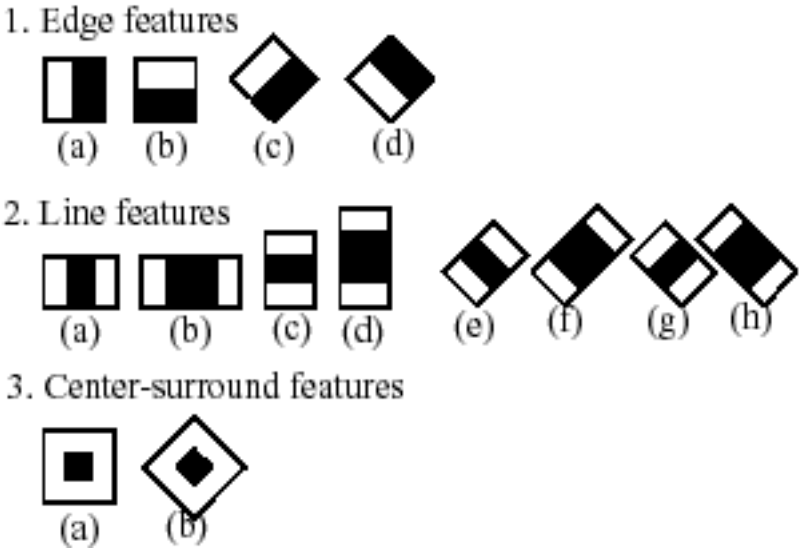
Instrukcja	**Zaawansowane przetwarzanie obrazów**
4	**Temat**: Wykrywanie twarzy Przygotował: mgr inż. Tomasz Michno

1 Wstęp

Biblioteka OpenCV pozwala na wykrywanie twarzy z użyciem tzw. kaskad Haar'a (ang. Haar Cascades). Jest to jeden z najbardziej efektywnych sposobów detekcji twarzy. Jego działanie opiera się o porównywanie obrazu z cechami, które wyglądają jak na obrazku poniżej:



źródło obrazu: http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html

Wszystkie cechy zgrupowane są w formie tzw. kaskady - porównywanie odbywa się etapami, w celu przyśpieszenia algorytmu. Na początku sprawdzane jest tylko kilka najważniejszych cech. Następnie, gdy test się powiedzie, algorytm sprawdza kolejną grupę (kaskadę), zazwyczaj nieco liczniejszą, potem następną itd. aż do wyczerpania wszystkich cech. Jeżeli testy dla wszystkich cech będą poprawne, algorytm uznaje sprawdzane miejsce w obrazie jako twarz.

Dane dla algorytmu znajdują się w plikach XML (w katalogu OpenCV powinien znajdować się folder haarcascades/). OpenCV dostarcza programistom pliki m.in. do rozpoznawania:

- twarzy: np. haarcascade_frontalface_default.xml
- oczu: np. haarcascade_eye.xml
- uśmiechu np. haarcascade_smile.xml
- człowieka np. haarcascade_body.xml

Możliwe jest również własnoręczne przygotowanie plików.

Ładowanie pliku kaskad odbywa się za pomocą funkcji cvLoad:

```
1 CvHaarClassifierCascade* cascade=(CvHaarClassifierCascade*)cvLoad
    ("../haarcascades/haarcascade_frontalface_alt.xml");
```

Poniżej znajduje się kod wyświetlający w miejscu znalezionej twarzy prostokąt.

```
1  CvMemStorage* storage = cvCreateMemStorage(0);
2
3  CvSeq* faces = cvHaarDetectObjects(frame, cascade, storage, 1.1,
       3, CV_HAAR_DO_CANNY_PRUNING, cvSize (100, 100));
4
5  for(int i = 0; i<(faces ? faces->total:0); i++)
6      {
7
8           CvRect *r=(CvRect*)cvGetSeqElem(faces,i);
9           cvRectangle(frame,
10                      cvPoint(r->x, r->y),
11                      cvPoint(r->x + r->width, r->y + r->height),
12                      CV_RGB(255,0,0), 1, 8, 0);
13     }
```

Linia nr 1 odpowiada za utworzenie w pamięci bufora na dane tymczasowe dla algorytmu. Następnie w funkcji cvHaarDetectObjects() wyszukiwane są twarze w obrazie. Funkcja ta ma następujące parametry:

```
1 CvSeq* cvHaarDetectObjects(const CvArr* image,
2 CvHaarClassifierCascade* cascade,
3  CvMemStorage* storage,
4  double scale_factor=1.1,
5  int min_neighbors=3,
6  int flags=0,
7  CvSize min_size=cvSize(0,0),
8  CvSize max_size=cvSize(0,0) )
```

image - obrazek na którym ma zostać znaleziona twarz
cascade - załadowany plik z kaskadami (używając funkcji cvLoad)
storage - obiekt CvMemStorage służący do przechowywania tymczasowych danych dla algorytmu
scaleFactor - ile razy zmniejszyć obrazek wejściowy (przyśpiesza działanie algorytmu)
minNeighbors - ile minimalnie cech powinno występować w swoim pobliżu, aby szukać twarzy w danym miejscu
flags - parametr obecnie nieużywany (przyjmuje wartość 0)
min_size, max_size - minimalny i maksymalny rozmiar twarzy do odszukania w obrazie
Jako swój wynik funkcja zwraca listę prostokątów, w których znaleziono twarze.
Linie 5-13 pokazują, jak z listy typu CvSeq odczytać znalezione prostokąty.

2 Zadanie

Napisz program, który będzie rozpoznawał używając kamery internetowej twarz. Następnie w miejscu znalezionej twarzy powinien być nakładany obrazek - np. maska karnawałowa.