특정 클래스에 적합한 Region 검출기와 Dominant Feature 를 이용한 클래스 모델링

이희진 O, 홍기상

포항공과대학교 전자전기공학과

huijin@postech.ac.kr, hongks@postech.ac.kr

요 약

본 논문에서는 다양한 region 검출기들과 feature 들의 구성을 사용하여 클래스를 대표하는 의미 있는 부분들을 정의한다. 계산의 효율성을 위해서 feature 의 dominant 한 성분들로만 구성된 dominant feature 들을 제안하고, 해당 클래스의 물체들을 나타내기 위해서 어떤 구성(region 검출기와 feature)이 가장 유용한지 판단하는 방법을 제안한다. Caltech 101 의 다양한 클래스들에 대해서 제안된 방식이 클래스 인식의 성능 향상에 얼만큼 기여할 수 있는지 실험하였고, 각 클래스마다 적합한 구성을 선택한 후 얻은 클래스 인식 성능이 가장 좋은 것을 확인하였다.

1. 서론

비전 분야에서 특정 클래스(자동차, 사람, 얼굴 등)의 물체들을 인식 하는 것은 데이터 베이스로부터 영상을 검색하거나 입력 영상에서 물체의 위치를 찾아내는 과정에서 중요한 부분이다. 해당 클래스의 물체 여부를 판단하기 위해서 크게 물체의 전체 구조 정보를 이용하는 방식과 물체의 부분 정보들을 이용하는 방식(part 기반 방식)으로 나눌 수 있는데, 물체의 전체 구조 정보를 이용하여 물체를 모델링 하는 것은 대체로 쉽지만, 물체의 가려짐이나 같은 클래스내의 모양 변화와 같은 문제들을 해결하기에는 어려움이 있고 따라서, 물체의 부분 정보들을 이용하여 이런 문제들을 보완한다[1 - 4].

Part 기반의 물체 인식에서 가장 중요한 것은 물체의 모델링을 위한 부분들을 정의하는 방법이다. 이를 위해, 물체의 부분들을 미리 정의하고 각 부분들에 대한 검출기를 학습시켜 joint likelihood 함수로 물체의 여부를 판단[1, 2] 하는 것이 일반적이지만, 최근 들어, 물체의 부분들을 미리 정의하는 것 대신에 의미 있다고 여겨지는 부분들을 자동으로 선택하고 나타내기 위한 방법들이 제안되고 있다[3, 4]. 이런 방법들은 다양한 사이즈와 비율의 부분들을 미리 후보로 가지고 있고, 이들을 사전에 정의된 feature 로 나타낸 후, 이 중에서 의미 있는 것들을 Adaboost 분류기를 통해서 검출함으로써, 물체의 부분들을 정의 한다.

하지만, 이와 같은 방법들은 의미 있는 부분들을 선출하기 위해서 미리 가지고 있어야 하는 후보들의 수가 너무 많고 따라서 학습시간이 오래 걸린다는 단점이 있다. 또한, 이는 특정 클래스에 따라서 feature 를 정의하는 것이 아니기 때문에, 정해진 것들만으로 클래스들의 중요한 특성을 잡기에는 명백히 한계가 있다.

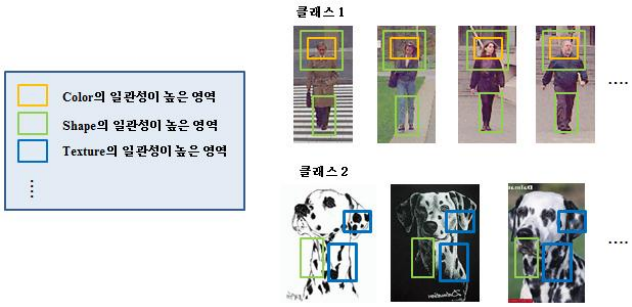

그림 1. 클래스에 따른 유용한 구성들

따라서, 이런 한계들을 보완하기 위해서 다양한 사이즈와 비율의 후보 부분들을 모두 사용하는 것 대신에 물체의 corner 부분이나 blob 부분 또는 정보가 많은 부분들을 검출 할 수 있는 다양한 region 검출기들을 사용하면, 훨씬 의미 있게 줄여진 후보들을 얻을 수 있다. [5]에서는 이와 같은 region 검출기들이 물체의 클래스 인식에 유용하다고 말하고, 그 중에서 어떤 것이 많은 클래스들을 인식하기 위해서 좋은지를 비교하여 나타내고 있다. 하지만, 이 역시도 특정 클래스의 물체들을 인식하기 위해 어떤 region 검출기와 feature 가 잘 동작하는지에 관한 분석은 나타내지 않는다.

본 논문에서는 후보 부분들을 검출하기 위해서 [5]와 같은 다양한 region 검출기들을 사용하고 검출된 부분들의 특성을 나타내기 위해 다양한 dominant feature 들을 제안한다. 또한, 이 중에서 특정 클래스

의 물체들을 인식하기 위해서는 어떤 구성이 가장 유용한지를 판단하고<그림 1>, 이와 같은 방식이 실제로 클래스 인식의 성능 향상에 얼만큼 기여할 수 있는지를 보여준다.

2. 제안된 방법

2.1 절과 2.2 절에서는 사용된 region 검출기들과 feature 들을 간단하게 설명하고 2.3 절에서는 이를 이용하여 특정 클래스의 모델을 나타내는 방법과 region 검출기와 feature 의 가장 적합한 구성을 찾는 방법을 제안한다.

2.1 Region 검출기

본 논문에서는 다음과 같이 4 개의 region 검출기들을 사용하였다.

DOG[6]: 스케일-공간상의 difference-of-Gaussian 국부 극대점에서 region 들이 얻어지고, 이는 물체의 blob 과 같은 특징을 찾아내는데 유용하다.

Harris-Laplace[7]: 공간상의 scale-adapted Harris 함수와 스케일상의 Laplace-of-Gaussian 의 국부 극대점에서 region 들이 얻어진다. 이는 물체의 corner 와 같은 특징을 찾아내는데 유용하다.

Hessian-Laplace[8]: 공간상의 Hessian determinant 와, 스케일상의 Laplace-of-Gaussian 의 국부 극대점에서 region 들이 얻어진다.

Salient[9]: 스케일-공간상 엔트로피의 국부 극대점에서 region 들이 얻어지고, 영상의 한 위치에서의 엔트로피는 다양한 사이즈를 가지는 영역내의 intensity 히스토그램들부터 계산된다.

이와 같은 region 검출기들로부터 얻어진 부분들은 검출된 스케일에 따라서 다양한 사이즈를 가지는데, 이는 feature 들을 적용시키기 위해서 동일한 사이즈로 정규화 한다.

2.2 Feature

후보로 검출된 부분들에 대해서 다음과 같이 color, gradient, appearance, shape 의 특성들을 나타내기 위한 feature 들을 제안한다. 각 feature 들은 gradient 의 특성을 뽑아내기 위해 사용된 기존의 Dominant Orientation Templates(DOT)[10] 방식과 유사하게 해당 물체 부분의 dominant 한 성분들만을 가지는 8 bits 의 binary feature 로 구성된다.

Dominant-color: HSV 색 공간의 dominant Hue(H) 값들을 기반으로 feature 를 구성한다. H 값의 범위[0, 360]는 7 만큼의 수로 양자화 되었고, 8 bits 에서 처음 7 bits 는 그 region 에서 가장 많은 수를 차지하는 H 값들이 해당하는 범위(bit)에 1 을 할당하고 그렇지 않은 bit 에 0 을 할당하기 위해서 사용된다. 마지막 bit 은 HSV 색 공간의 특성상 Saturation(S)나 Intensity(I)의 값이 너무 낮으면 H 값을 구분이 어려우므로[11] 이 경우에 1 을 할당하기 위해 사용된다.

Dominant-gradient: 이 특성을 나타내기 위해서는 기존의 DOT 알고리즘을 그대로 사용한다. 이는 gradient 의 dominant 방향성분들을 기반으로 feature 를 구성한다. 방향성분의 범위[0, 180]는 7 만큼의 수로 양자화 되었고, 8 bits 에서 처음 7 bits 는 gradient 의 크기성분을 기준으로 가장 큰 크기성분을 가지는 방향성분이 속하는 범위(bit)에 1 을 할당하고 그렇지 않은 bit 에 0 을 할당하기 위해서 사용된다. 또한 마지막 bit 은 크기성분이 너무 작은 경우에 균일한 영역일 수 있으므로 이 경우에 1 을 할당하기 위해서 사용된다.

Dominant-appearance: 해당 region 의 appearance 특성을 잡기 위해, dominant intensity 값들을 기반으로 feature 를 구성한다. Intensity 값의 범위[0, 256]는 7 만큼의 수로 양자화 되었고, 8 bits 에서 처음 7 bits 는 region 에서 가장 많은 수를 차지하는 intensity 값들이 해당하는 범위(bit)에 1 을 할당하고 그렇지 않은 bit 에 0 을 할당하기 위해서 사용된다. 또한, 마지막 bit 은 해당 영역에서 최대 intensity 값과 최소 intensity 값의 차이가 너무 작은 경우에는 균일한 영역일 수 있으므로 이 경우에 1 을 할당하기 위해서 사용된다.

Dominant-shape: 해당 region 의 shape 특성을 나타내기 위해 기존의 Shape Context[12]를 7-bin 으로 간략화 하여 사용한다<그림 2>. 8 bits 에서 처음 7 bits 는 각 bin 의 엣지점들의 수가 가장 많은 수를 차지하는 범위(bit)에 1 을 할당하고 그렇지 않은 bit 에 0 을 할당하기 위해서 사용되고, 마지막 bit 은 가장 많은 수의 엣지점들을 가지고 있는 bin 에서의 그 수가 너무 작은 경우에는 균일한 영역일 수 있으므로 이 경우에 1 을 할당하기 위해서 사용된다.

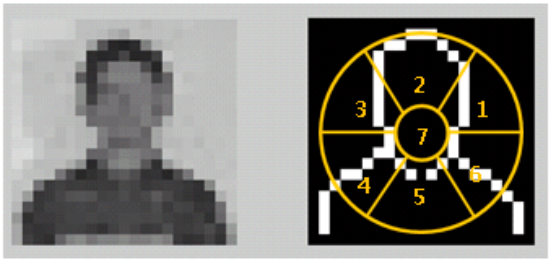

그림 2. 간략화된 Shape Context

2.3 클래스 대표 모델

이번 절에서는 특정 클래스를 대표하는 모델을 구성하기 위한 방법 제안한다.

각 클래스는 그 클래스 물체들의 학습 영상들을 가지고 있고, region 검출기와 feature 의 구성에 따라 대응하는 클래스 모델들을 생성 할 수 있다(총 16 개의 클래스 모델들). 각 클래스 모델의 생성은 다음과 같다.

1. Region 검출기의 인덱스를 m 이라 할 때, m 번째 검출기를 사용하여 해당 클래스의 모든 학습 영상들에 대해서 region 들을 검출한다.

2. Feature 의 인덱스를 n 이라 할 때, 검출된 region 들 각각에 대해서 n 번째 feature 를 적용하여 8 bits 의 binary feature 를 구성한다.

3. 학습 영상을 격자모양으로 나눠서 얻은 작은 부분들을 spatial word(SW)라고 하고 g 개의 SW 들이 있다면<그림 3>과 같이 g x 8 사이즈의 voting map 이 구성된다.

4. 구성된 voting map 에 과정 1 에서 얻어진 region 들의 center 위치를 기준으로 해당하는 SW 에 8 bits 의 binary feature 를 voting 하면 그 SW 에 점차적으로 히스토그램이 구성된다<그림 3>. 즉, voting map 의 각 SW 는 그것에 voting 한 region 들의 평균 center 위치와 사이즈, 그리고 대응하는 feature 들의 히스토그램으로 구성된다.

5. 해당 클래스에 대해서 feature 들의 일관성이 높은 물체의 부분들을 선출하기 위해서 각 SW 마다 히스토그램을 기반으로 Shannon 엔트로피(1)를 계산하고, 엔트로피가 낮은 k 개의 SW 들을 선출한다.

$$H(X) = -\sum_{h=1}^{8} p(x_h)\log(p(x_h)) \qquad (1)$$

6. 선출된 k 개의 SW 들의 평균 center 위치와 사이즈로 그 클래스를 나타내기 위한 물체의 부분들을 정의하고, 각 부분들은 대응하는 SW 의 히스토그램에서 최대값을 가지는 bin 과 최대값의 80%이내에 들어오는 bin 들에 1 을 그 외의 모든 bin 에 0 을 할당함으로써 다시 binary feature 로 표현된다.

7. 결론적으로 한 클래스의 모델은 k x 8 bits 의 binary feature 로 구성된다.

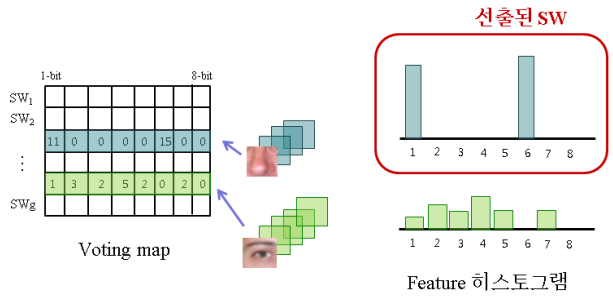

그림 3. Voting map 의 구성

위와 같은 방법으로 region 검출기들과 feature 들의 구성에 따라 대응하는 클래스 모델들이 생성되고, 이 중에서 해당 클래스에 가장 적합한 구성을 선택하기 위한 과정은 다음과 같다.

1. 클래스 모델의 인덱스를 d 라고 할 때, d 번째 클래스 모델의 binary feature T_d와 해당 클래스의 물체를 포함하는 학습 영상들에서 뽑아진 binary feature P_j 들과의 해밍 거리의 평균 P_d을 구한다.

$$p_d = \frac{1}{T_p}\sum_{j=1}^{T_p} d_h(T_d, P_j) \qquad (2)$$

여기서, d_h 는 해밍 거리 이고 T_p 는 클래스 물체를 포함하는 학습 영상의 수이다. 이때, P_j는 d 번째 클래스 모델에서 binary feature 를 얻기 위해 선출되었던 동일한 region 들에서 얻어진다.

2. 또한, 과정 1 에서와 같이 d 번째 클래스 모델의 binary feature T_d와 해당 클래스의 물체를 포함하지 않는 학습 영상들에서 뽑아진 binary feature N_j 들과의 해밍 거리의 평균 N_d을 구한다.

$$N_d = \frac{1}{T_N}\sum_{j=1}^{T_N} d_h(T_d, N_j) \qquad (3)$$

3. 최종적으로 해당 클래스에 가장 잘 맞는 클래스 모델(즉, region 검출기와 feature 의 구성)을 찾기 위해서는 다음과 같이 물체를 포함하는 영상과 포함하지 않는 영상과의 거리차이가 가장 큰 것을 선택한다.

$$\arg\max_{d \in \text{class model}} diff(P_d, N_d) \qquad (4)$$

여기서, $diff$는 두 값의 차이를 의미한다.

3. 실험 결과 및 분석

3.1 제안된 Feature 들

이 절에서는 특정 클래스에 속하는 부분 영상들에 대해 2.2 절에서 제안된 feature 들 중 어떤 것이 가장 의미 있게 동작하는지를 보여준다.

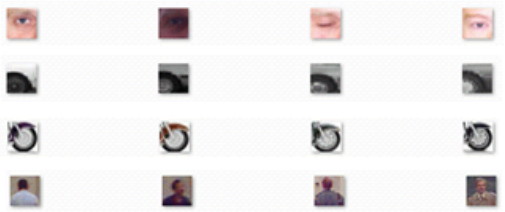

그림 4. 각 클래스의 부분 영상들

실험을 위해 준비된 클래스는 Caltech 101[13]의 face, car-side, motorbike 가 있고, MIT pedestrian[14]의 pedestrian 이 있다.

각 클래스의 부분 영상들은 <그림 4>와 같이 임의로 정해주었고, 각 클래스의 부분 영상들로부터 계산된 feature 의 엔트로피는 <표 1>과 같다. 특정 클래스에서 feature 의 엔트로피가 낮다는 것은 해당 클래스에 대해서 그 feature 의 일관성이 높다는 의미이므로 엔트로피가 낮을수록 더 적합한 feature 라고 판단할 수 있다. Face 클래스의 경우에는 color 의 일관성이 높게 나타나고 따라서 이 정보가 클래스를 나타내기에 가장 유용하다. 반면에 motorbike 경

우에는 color 가 다양하므로 이 정보 보다 appearance 의 정보가 잘 동작할 것이다. Car-side 의 경우에는 illumination 변화나 자동차의 color 의 변화(검은색, 흰색, 회색)에 의해서 appearance 보다 gradient 정보가 더 유용하고, pedestrian 의 경우에는 head-shoulder 의 정보를 잡을 수 있는 shape 의 정보가 가장 유용하다.

표 1. 제안된 feature 들의 엔트로피

	face	car-side	motorbike	pedestrian
color	**0.49**	-	0.79	0.62
gradient	0.65	**0.60**	0.77	0.81
appearance	0.59	0.65	**0.62**	0.72
shape	0.60	0.61	0.70	**0.54**

3.2 클래스 인식

이 절에서는 특정 클래스의 물체 인식에 어떤 region 검출기와 feature 의 구성이 적합한지를 확인하고, 이와 같은 방식이 전체적인 클래스 인식의 성능 향상에 얼만큼 기여하는지 보여주고자 한다. Caltech 101[13]으로부터 <그림 5> 와 같이 4 가지 클래스(airplane, car-side, face, motorbike)를 사용한다.

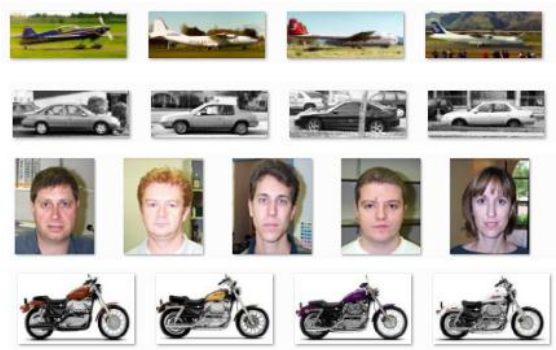

그림 4. 각 클래스의 일부 영상들

각 클래스로부터 학습을 위해 30 장의 영상들을 임의로 선출하고, 이들을 2.3 절에서 제안된 방법으로 해당 클래스를 대표하는 모델을 선출하기 위해서 사용한다. <그림 5>는 이 과정에서 선출된 region 검출기와 feature 의 구성 및 각 클래스를 대표하는 부분(녹색 원)들을 보여준다. 또한 이 과정을 통해서 해당하는 클래스의 k x 8 bits 의 binary feature 를 얻는다.

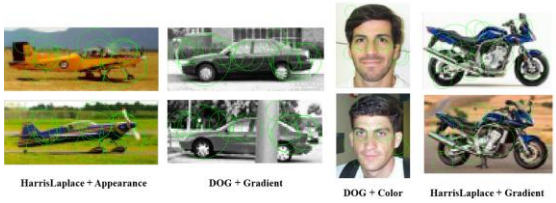

그림 5. 각 클래스에서 선출된 구성 및 영역들

각 클래스로부터 테스트를 위해 10 장의 영상들을 임의로 선출하고, 이 외에도 Caltech 101 의 background 클래스로부터 10 장의 영상들을 임의로 선출한다. 따라서 특정 클래스의 classification rate (%)를 측정하기 위해서 사용된 테스트 영상들은 해당 클래스를 포함 50 장으로 구성되게 되고, 이 중에서 학습 영상들로부터 얻어진 binary feature 를 이용해서 선출해낸 영상(binary feature 간의 해밍 거리가 작은 영상)들 중 정확하게 선출된 해당 클래스 영상의 비율이 classification rate(%)이 된다. 전체 클래스들에 대한 성능을 측정하기 위해서 각 클래스의 classification rate(%)의 평균을 구하고, 더 정확한 결과를 얻기 위해서 이 모든 과정을 5 번 반복한다. 이런 과정을 통해서 얻어진 전체적인 클래스 인식의 성능은 <표 2>와 같다. 각 클래스마다 가장 적합한 구성(region 검출기 + feature)을 찾아서 성능을 측정한 경우는 “Our” 이고, 모든 클래스에 대해서 하나의 구성만을 사용해서 성능을 얻은 경우 그 중에서 가장 성능이 좋은 두 가지 구성은 “DOG + Color”와 “Salient + Gradient”이다. 또한 region 검출 및 feature 선택의 모든 과정이 random 으로 정해진 경우는 “Random”이다.

표 2. Mean classification rate (%)

	Mean classification rate (%) ± std.
Our	**74.0 ± 1.22**
DOG+Color	62.0 ± 4.51
Salient+Gradient	59.5 ± 6.77
Random	37.5 ± 7.90

결과를 보면 알 수 있듯이, 특정 클래스마다 가장 적합한 구성을 선택한 후 얻은 클래스 인식 성능이 모든 클래스들에 대해서 하나의 구성만을 적용시켜 얻은 클래스 인식 성능들보다 훨씬 좋은 것을 확인 할 수 있고, 이는 특정 클래스마다 요구되는 region 검출기 및 feature 의 특성이 다르다는 것을 의미하고 가장 적합한 구성을 찾는 것이 중요하다는 것을 의미한다.

3.3 구성들간의 상보성

이 절에서는 MIT pedestrian[14]의 pedestrian 클래스 인식에 유용하다고 선출된 구성이 무엇인지를 확인하고, 선출 된 구성들간의 결합이 얼마나 성능을 증가 시킬 수 있는지 보여주고자 한다.

그림 6. 구글에서 수집된 클래스들의 영상들

Pedestrian 클래스 외에도 이 클래스와 혼돈 가능성이 있는 다양한 클래스들(pole, street light, traffic light, building)을 구글로부터 수집하였고 이는 <그림 6>과 같다.

Pedestrian 클래스에 적합한 구성을 선출하는 방식과 테스트 방식은 3.2 절과 동일하고, 이로부터 얻어진 classification rate(%)는 <표 3>과 같다.

표 3. Mean classification rate (%)

	Mean classification rate (%) ± std.
Best1	68.0 ± 4.00
Best2	58.0 ± 11.66
Best1+ Best2	**84.0 ± 7.48**
Random	32.0 ± 26.38

"Best1(Harris-Laplace + Color)"은 가장 적합하다고 선출된 구성이고, "Best2(Salient + Appearance)"는 그 다음으로 적합하다고 선출된 구성이다. 또한, "Best1 + Best2"는 이 두 구성을 <그림 7>과 같이 결합한 것을 의미하며, 결과를 보면 결합한 것의 성능이 결합하지 않은 것들보다 훨씬 좋아 지는 것을 확인할 수 있다. 이로 인해, 클래스 마다 적합한 구성들을 찾는 문제뿐만 아니라 이들을 잘 결합하는 것 또한 중요한 문제임을 알 수 있다.

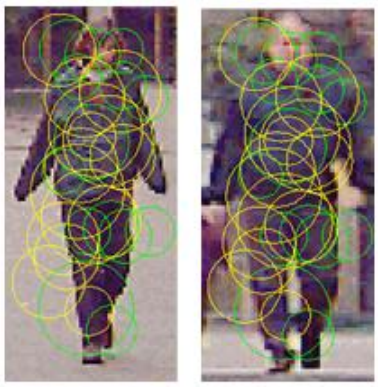

그림 7. Pedestrian 클래스에서 선출된 부분들
(녹색 원: Best1, 노란색 원: Best2)

4. 결론

본 논문에서는 다양한 region 검출기들과 feature 들의 구성을 사용하여 클래스를 대표하는 의미 있는 부분들을 정의하였다. 계산의 효율성을 위해서 feature 의 dominant 한 성분들로만 구성된 dominant feature 들을 제안하였고. 또한, 해당 클래스의 물체들을 인식하기 위해서 어떤 구성(region 검출기와 feature)이 가장 유용한지를 판단하기 위한 방법을 제안하였다. Caltech 101 의 다양한 클래스들에 대해서 클래스 인식 성능을 비교하였고, 각 클래스마다 적합한 구성을 선택한 후 얻은 클래스 인식 성능이 가장 좋은 것을 확인할 수 있었다. 또한 pedestrian 클래스에 대해서 선택된 구성들을 결합한 경우의 성능이 결합하지 않은 경우의 성능보다 훨씬 좋아지는 것을 확인 할 수 있었고, 이로 인해, 클래스마다 적합한 구성들을 찾는 문제뿐 아니라 이들을 잘 결합하는 것 또한 매우 중요한 문제임을 확인할 수 있었다. 따라서, 차후에 클래스의 대표 모델을 만들기 위해서 어떻게 이들을 잘 결합하고 구성할 것인지에 관한 방안을 연구할 예정이다.

참고문헌

[1] K. Mikolajczyk, A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors", ECCV, pp. 69-81, 2004.

[2] Zhe Lin, Daniel DeMenthon, "Hierarchical part-template matching for human detection and segmentation", ICCV, pp. 1-8, 2007.

[3] Zhe Lin, Larry S. Davis, "Multiple Instance Feature for Robust Part-based Object Detection", CVPR, pp. 405-412, 2009.

[4] Aharon Bar-Hillel, Eyal Krupka, and Chen Goldberg, "Part-Based Feature Synthesis for Human Detection", ECCV, pp. 127-142, 2010.

[5] K. Mikolajczyk, B. Leibe, and B. Schiele, "Local features for object class recognition", ICCV, pp. 1792-1799, 2005.

[6] D. Lowe, "Distinctive image features from scale-invariant keypoints", IJCV, pp. 91–110, 2004.

[7] K. Mikolajczyk, C. Schmid, "Scale & affine invariant interest point detectors", IJCV, pp. 63–86, 2004.

[8] K. Mikolajczyk, T. Tuytelaars and L. V. Gool, "A comparison of affine region detectors", IJCV, 2005.

[9] T. Kadir, M. Brady, "Scale, Saliency and Image Description", IJCV, pp. 83–105, 2001.

[10] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects", CVPR, 2010.

[11] S. Sural, G. Qian and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval", ICIP, pp. 589-592, 2002.

[12] S. Belongie, J. Malik and J. Puzicha, "Shape Context: A new descriptor for shape matching and object recognition", NIPS, pp. 831-837, 2000.

[13] L. Fei-Fei, R. Fergus and P. Perona, "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories", CVPR Workshop on Generative-Model Based Vision, 2004.

[14] A. Mohan, C. Papageorgiou and T. Poggio, "Example based object detection in images by components", PAMI, pp. 349-361, 2001.