A coarse-to-fine capsule network for fine-grained image categorization

CTF-CapsNet (Coarse-to-Fine Capsule Network):

CapsNet arranged in RPN (Region Proposal Networks)

CTF-CapsNet is driven by:

three focal margin losses between label prediction and ground truth, and

three regeneration losses between original input images/ feature maps and reconstructed images

이미지 분류는

- superordinate-level (동물, 식물, 건물 등 큰 분류)
- basic-level (동물 중에 개, 고양이, 말, 소 등)
- subordinate-level (개 중에 치와와, 퍼그, 허스키 등 소분류)
  - fine-grained가 여기 속함.
  - 보통 attention을 object-level representation보단 local part-level representation으로 바꿈
  - CNN 기반 전략이 발전했지만 그래도 언급되는 두 가지 결함:
    - 여러 개체 간의 공간 계층 구조에 대한 무지
    - 회전 불변성의 부족 (이미지를 돌렸을 때도 같은 이미지라고 구분하는 능력 부족)
  - Fine-grained categorization을 위한 많은 CNN 기반 연구는 일반적으로 스펙트럼 및 공간 영역 사이에서 미묘한 시각적 신호에 집중.
  - 그러나 작은 패치 사이의 공간적 관련성과 논리성은 고려되지 않음. 공간 관련성과 논리성에 포함되는 feature, 각도, 크기, 위치, 컨텍스트, 피라미드, 또는 공간 지각 기반 울트라메트릭 (초거리 공간)이 빠질 수 있다는 뜻.
  - 이를 해결하기 위해 나온 게 CapsNet (후술)
  - 이 논문에선 특색 있는 지역들을 region proposal network (RPN)을 사용해 배치하는 coarse-to-fine CapsNet (CTF-CapsNet)을 사용한다.

Related works:

- Fine-grained visual categorization:
  - Surface learning-based strategies

    초창기 이미지 분류에 사용하던 기법들: Convex radius-margin-based support vector machine이나 fast multi-label low-rank-linearized SVM. 좋은 시작 기법이었으나 개선의 여지가 아직 많고 깊은층의 feature를 학습하기 어려워함.

  - Annotation-based strategies

    사람이 직접 찍어주는 Human-in-the-loop가 여기 해당. 성능이 오르긴 했지만 새로운 타타겟 적용할 때 cost가 너무 높은 단점이 있다.

  - Template-based based strategies

    요즘 연구들은 미리 학습된 알고리즘이나 필터와 이미지의 중심을 위주로 적용한다. Destruction and construction (잘게 쪼개서 feature를 추출하고 다시 합치는 방식)이나 Selective Sparse Sampling Network (S3N) 등으로 attention을 집중하는 방법을 사용했었다. 또한 Kullback-Leibler divergence으로 다양성을 중요시하거나 Part-Stacked CNN 등을 쓰는 방법도 있다.

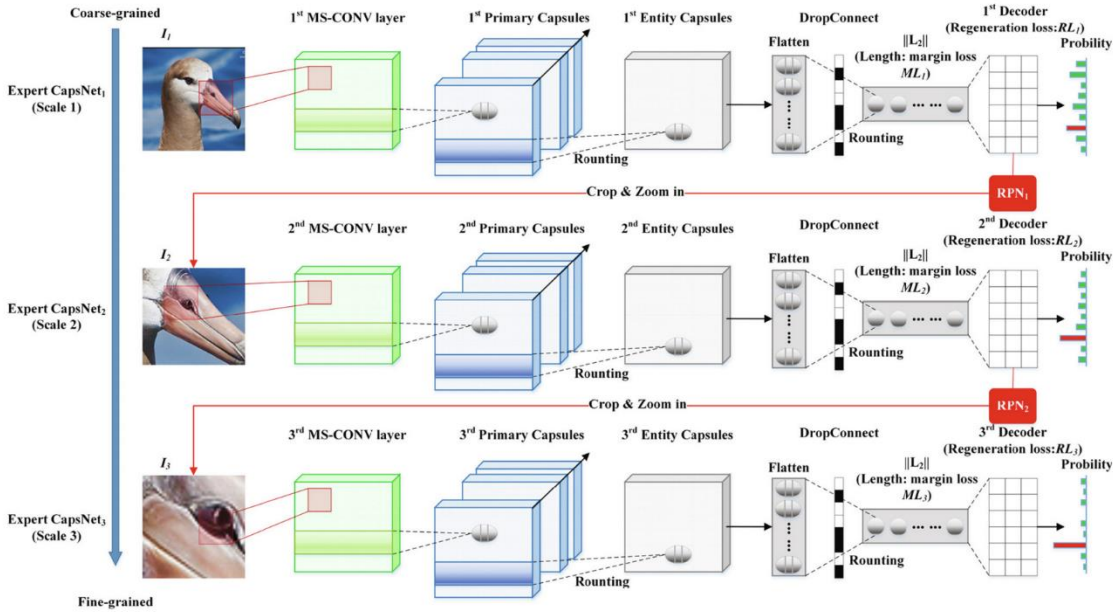
  - Codebook-based strategies

    Hierarchical Visual Codebook을 사용하는 방법도 있었다.

- Capsule network (CapsNet)
  - 앞서 말했듯이 CNN은 물체의 공간적인 부분을 잘 잡아내지 못한다 (물체의 위치에 따라 캡슐이 동적 routing의 연결 정도를 증가/감소하기 때문이다. 또한 Max-pooling 같은 기법은 많은 정보를 무시하기 때문에 정확도가 떨어질 수 있다.
  - 여기서 capsule이란 part-whole topology를 학습하기 위해 나타내는 벡터이다. 모든 capsule은 2가지로 구성되어 있는데, pose matrix와 activation probability distribution이다.
  - Capsule feature 벡터의 길이는 본체가 현재 국소적으로 수용하는 필드에 나타날 확률이라고 할 수 있다.
  - CapsNet은 feature를 잘 발굴하고 다양하게 발굴한다.

Coarse-to-fine capsule network:

- Pipeline을 세가지 단계의 multi-dimensional CapsNet으로 만들었다.
  - 계층적 feature를 점점 전문화된 방식으로 인코딩 해주고
  - 파악하기 어려운 perceptions도 비슷하게 쌓을 수 있다
- 또한 하나의 object-level description과 두개의 component-level description으로 구성되어 있다. Object-level에서는 전역적인 레이아웃을 표현하고, 나머지 component-level에서는 추천 패치를 통해 도메인별 시각적 구별점을 밝혀낸다.



- Overall pipeline: Expert CapsNet
  - Standard multi-scale convolutional layer (MS-CONV, 초록색)

    픽셀 강도를 도메인별 기능 설명의 활동으로 바꿔준다. 이 레이어의 결과값은 캡슐의 입력 값으로만 사용된다. 추가적으로, 더욱 복잡한 기능을 학습하게 하기 위해 16채널과 4 필터를 위에 적용해준다. 이론상 primary capsules (파란색)에 도달하기 전에 원하는 만큼 convolutional layer를 쌓을 수 있다. 각 레이어 별로 하위 샘플링 레이어는 버리고, 대신 차원을 줄이기 위해 stride는 1보다 크거나 같게 해줬다. 라우팅은 없다.

  - Multi-scale capsule encoding layer (reshaped and squashed output of the last convolutional

layer) consisting of primary capsules

첫번째 캡슐 인코딩 층으로, 역 그래픽 관점으로 봤을 때 렌더링 작업을 반전시키는데 활용된다. 가장 낮은 단계의 다차원 개체이며 CNN의 기능 탐지 스칼라 값을 벡터 값으로 변환해준다. 여기서 활성화된 캡슐은 그 다음 층인 개체 캡슐층과 transformation matrix를 통해 예측한다.

- Entity capsules layer (capsules denoting all the types of training samples, 회색)

  앞의 레이어 (PrimCaps)의 출력 벡터에 따라 entity capsule layer를 발전하고 전파시킨다. 동적 라우팅을 통해 가중치를 줄이거나 늘려서 CTF-CapsNet을 적응시킬 수 있다. 전 레이어와 현 레이어를 Routing by aggrement를 통해 학습시킬 수 있으며, 캡슐의 참여도를 극대화할 수 있다.

- DropConnect layer

  오버피팅을 방지하기 위해 적용하는 레이어. CNN과 다르게 몇몇 element를 삭제하는게 아니라 벡터 전체를 소멸시킨다. 그렇지 않으면 벡터의 방향이 완전 달라지고 치명적인 왜곡이 생길 수 있다.

- Decoder layer

  성공적으로 잘 예측한 캡슐을 따로 추출해서 다음 스케일의 CTF CapsNet에 적용해준다. 이때 캡슐의 결과값과 픽셀 강도의 차이를 RMSE (root mean square error)로 regeneration loss를 측정하며 훈련동안 이 차이가 줄어들 것이다. 이 디코딩 레이어 덕분에 CTF-CpasNet은 closed-loop autoencoder가 된다. 첫번째 스케일에서 성공적으로 5번의 레이어를 통과하면 첫번째를 통과한 사진들을 RPN(region proposal network)을 통해 크롭하고 다음 스케일에 넣어준다. 비슷하게 세번째 스케일까지 마치면 finest perception scale이 된다.

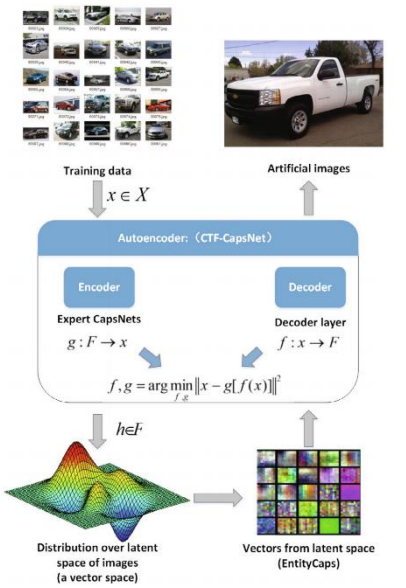


Fig. 3. The data stream in closed-loop autoencoder.

- Detailed structure of expert CapsNet:

  - 수학적인 부분

Algorithm 1 Dynamic Routing Algorithm in CTF-CapsNet

1: Procedure: DR ($\hat{u}_{j|i}$, n, m)
2: for capsule i in layer m and capsule j in layer $(m + 1)$:
$b_{ij} \leftarrow 0$
3: for n iterations do:
4: for capsule i in layer m:$c_j \leftarrow \text{softmax}(b_i)$
5: for capsule j in layer $(m + 1)$:$s_j \leftarrow \sum_i c_{ij}\hat{u}_{j|i}$
6: for capsule j in layer $(m + 1)$:
$v_j \leftarrow \text{squashed}(s_j)$
7: for capsule i in layer m and capsule j in layer $(m + 1)$:
$b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i}v_j$
Return v_j

Table 4
Detailed comparison statistics of model training on hand-crafted rice dataset.

Approach	Bounding-box	Part annotation	Training accuracy	Validation accuracy	Required epoch	Required time (h)
CTF-CapsNet (ours)			0.9401	0.9186	39	10.57
Part-alignment CNN [56]	√		0.853	0.7908	31	6.08
Part-stacked CNN [11]	√	√	0.8742	0.8561	38	8.32
SPDA-CNN [13]	√	√	0.9045	0.8874	42	13.03
Mask-CNN [14]		√	0.9256	0.9058	44	14.54
AutoBD [57]			0.9307	0.9013	37	7.68
Two-level Attention CNN [3]			0.9374	0.9092	53	20.5
RA-CNN [16]			0.9428	0.9173	54	23.92
MA-CNN [17]			0.9582	0.9318	47	18.27
DCL [39]			0.9326	0.9143	49	19.74
S3N [40]			0.9542	0.9364	37	7.53