Unsupervised Transformer : Spatial Transformer

한민규[1],
[1]전기전자공학부, 연세대학교

Abstract

이 논문의 Motivation은 CNN의 자체적인 한계에서 시작한다. CNN은 Image recognition에서 매우 강력한 model이지만, 입력 데이터에 대해 spatially invariant하지 못하다. 만약 이미지에서 찾으려고 하는 object가 위치가 바뀌거나, 회전하는 등의 deformation이 있다면 CNN model이 분류하지 못한다는 것이다. 이에 따라 Spatial Transfomer라는 module을 소개한다. 이 모듈은 translation, scaling, rotation 등의 distored 이미지를 transformation하여 CNN의 spatial invariance에 도움을 줄 수 있다.

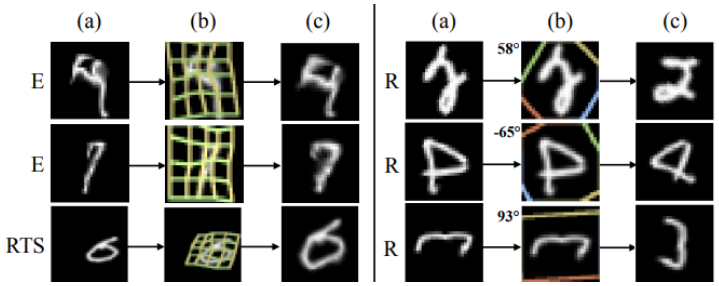

Figure 1: Spatial transformer가 변환시킨 distroed MNIST 이미지

Figure1은 spatial transformer의 역할을 보여주는 사진이다. Figure1-(a)는 회전, 이동, 크기 변화 ,elastic distortion 등이 적용되어 기존의 CNN이 분류에 실패한 이미지이다. spatial transformer는 Figure-(b)의 transformation을 적용하여 (c)의 이미지를 생상하였고, 분류에 성공하였다. 앞으로, 이전의 CNN의 spatial invariance를 위한 실험과 연구 그리고 Spatial transformer에 대해 더 자세히 알아본다.

1 Representation?

딥러닝 모델의 spatial invariance에 대해 더 잘 이해하기 위해 representation에 대해 알아볼 필요가 있다. representation이란 데이터를 인코딩하거나 묘사하기 위해 데이터를 바라보는 방법을 의미한다. 예를 들어 컬러 이미지를 바라보는 두 가지 방법으로 RGB format과 HSV format이 있다. 이미지의 체도를 낮추는 문제는 HSV, 파란색 픽셀은 선택하는 문제는 RGB가 더 선호된다. 이는 같은 입력 데이터에 대한 다른 representation이다. 모델의 학습에서 represetation을 찾는 과정은 좌표변환, 이동, projection, non-linear operation 등이 될 수 있다. 결국 더 좋은 representation을 찾는 것은 모델의 성능을 올리는 것이다.

2 Works for spatial invariance

image를 추론할 수 있는 능력은 딥러닝 모델의 바람직한 특성 중 하나라고 볼 수 있다. 즉, 부분적인 texture나 shape의 일그러짐과 image 내의 찾으려는 object의 pose를 풀어내는 것이다. 이와 관련된 기술과 연구에 대해 살펴본다.

2.1 Max-pooling

이미지 데이터의 특징으로 인접 픽셀들 간의 유사도가 매우 높아, 이미지는 픽셀 수준이 아니라 '특정 속성을 갖는 선택 영역'으로 표현될 수 있다는 아이디어에서 Pooling layer가 설계되었으며, 이 중 Max-pooling은 선택 영역에서 가장 큰 값을 해당 영역의 대표값으로 설정하는 것이다. Max-pooling layer를 도입한 CNN 모델은 이미지의 크기를 줄여 학습 속도를 빠르게하고 오버피팅을 줄일 수 있다. 또한, 네트워크가 features의 약간의 위치 변화에 대해 spatially invariant할 수 있도록 하였다.

This is an extended abstract. The full paper : 1506.02025

2.2 affine transform

[2], [4]에서 affine transform을 적용한 generative model로 additional input은 network에 학습시켜, 차별적인 features를 학습할 수 있도록하는 연구가 있었다. affine transform은 점, 직선, 평면을 보존하는 linear mapping으로, 이미지에 아핀 변환을 적용한 경우 평행한 선들은 모두 평행함이 보존된다. 따라서 generative model에 아핀 변환을 적용하여 카메라 각도나 기하학적 왜곡이 된 이미지를 보완함으로써 network를 효과적으로 학습할 수 있는 연구가 진행되었다.

2.3 others

Cohen과 Welling 저의 [1]에서는 original images와 transformed images의 representations 사이 선형 관계를 추정하여 architecture를 만들었고, 이는 feature maps가 더 invariant한 결과를 얻었다. 또한 T-CNN은 filter bank를 이용하였다. filter bank는 입력신호를 여러 구성요소로 분리하는 배열이며, texture 분석에 널리 사용되어 왔다. 이를 통해 filter bank를 통해 texture feature를 추출하는 강력한 tool로 CNN의 증가하는 depth에 의한 복잡함을 학습시킬 수 있었다. DasNet(Deep Attention Selective Network) [3]의 아이디어는 "the power of sequential processig"로 network가 반복적으로 convolution filter의 일부에 집중하도록 함으로써 분류의 성능을 향상 시켰다.

3 STN

Spatial Transformer가 적용된 CNN인 STN(Spatial Transformer Network)는 위의 Related Works의 아이디어로 한계점을 극복한 model이다. Max-pooling은 receptive field가 고정되어 있고, local하여 작은 spatial에 대해서만 invariant할 수 있었다. 또한 max-pooling이 적용된 CNN의 또 다른 한계점은 데이터 공간 배치의 variantions를 다루기 위한 mechanism이 미리 정의되어 있다는 것이다. affine transformation은 차별적인 feature를 학습할 수 있다는 장점이 있지만 대신 transformation supervision이 필요하다. 하지만, Spatial transformer는 역동적인 메카니즘으로, 각각의 이미지에 대해 적절한 transformation을 생산한다. 이 변환은 전체 feature map에서 수행되며 scaling, rotation, translation 등 여러 변환을 포함할 수 있다. 이러한 기능은 별도의 supervision없이 기존의 모델 구조에 spatial transformer layer를 삽입하고 동일하게 학습을 진행할 수 있다. 즉 ST layer는 전체 네트워크에 포함되 어 end to end learning이 가능한 것이다. 이제 STN에 대해 자세히 알아 본다.

3.1 What is STN

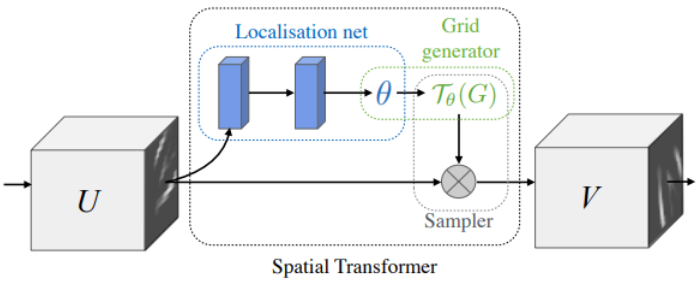

Figure 2: Spatial Transformer

Figure2는 Spatial Transformer의 구조이다. U와 V는 각각 input, output feature map이다. spatial transformer는 localisation network, grid generator, sampler로 이루어진다. 먼저, localisation network는 input에 맞는 transformation을 적용할 수 있는 spatial parameter θ를 뽑아내는 기능을 한다. grid

generator는 출력 이미지의 각 픽셀에 대응하는 입력 이미지 내의 좌표 그리드를 생성한다. 이는 아래에서 조금 더 자세히 설명한다. sampler는 spatial parameter를 적용하여 input feature map이 sampler를 통과하면 output feature map이 생성된다.

3.2 Workflow in ST layer

1. Input feature map U가 localisation net에 들어가 transformation parameter θ를 뽑아낸다. 이때 localisation network function은 $f_{loc}()$로, 적용되는 수식은 $\theta = f_{loc}(U)$이다. localisation net은 convolution network이든 Fully-connected network이든 마지막에 regression layer만 적용하면 된다.
2. 앞서 뽑아낸 transformation parameter θ는 grid generator에 들어가서 input feature map에서 샘플링 지점의 위치가 지정된 sampling grid를 생성한다.
3. Sampler에는 input feature map U와 sampling grid $T_\theta(G)$가 입력으로 들어가 Output feature map V를 뽑아낼 수 있다.

3.3 Sampling grid

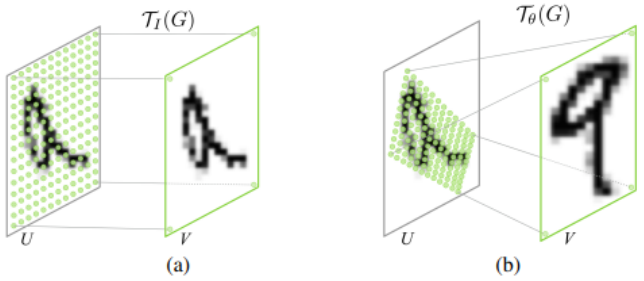

Figure 3: (a) : Identity transformation을 적용한 Sampling grid
(b) : Affine transformation을 적용한 Sampling grid

Figure3에서는 sampling grid가 적용된 것과 그렇지 않은 것의 차이를 명확하게 보여준다. 위의 작업은 다음과 같다. grid generator는 θ에 따라서 input feature map U상에서 sampling할 포인트를 정해주는 sampling grid $T_\theta(G)$를 생성한다. input feature map의 각 점은 $G_i = (x_i^s, y_i^s)$, output feature map의 각 점은 $G_i = (x_i^t, y_i^t)$로 표현되며 transformation T_θ로 각각 매핑된다. Figure3에서는 spatial transform의 확실한 설명을 위해 affine transform을 적용했고, 식은 아래와 같다.

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathtt{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

parameter를 많이 가지고 있는 transformation matrix일수록 complexity는 높아지지만 input image에 더 많고, 다양한 transformation을 가할 수 있다. 아래 식은 constraint가 많은 transformation으로 cropping, translation, isotropic scaling이 적용된다.

$$\mathtt{A}_\theta = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix}$$

더 복잡한 3D spatial transformation은 다음에 나오는 식과 같다. 이는 위 transformation 뿐만 아니라, rotation과 skew가 가능하다. 3D의 MNIST input에 대해 3D transformation이 적용되고, 연속되는 layer를 통해 2D projection이 적용된 과정을 Figure4에서 확인할 수 있다. 이 또한 역시 전체 네트워크의 end-to-end training의 적용이 가능하다.

$$\begin{pmatrix} x_i^s \\ y_i^s \\ z_i^s \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} & \theta_{14} \\ \theta_{21} & \theta_{22} & \theta_{23} & \theta_{24} \\ \theta_{31} & \theta_{32} & \theta_{33} & \theta_{34} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ z_i^t \\ 1 \end{pmatrix}$$

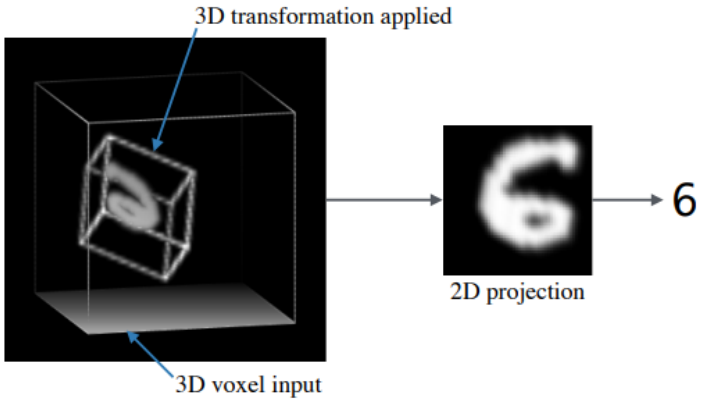

Figure 4: 3D spatial transformer, 2D projection

4 Conclusion

Model	
Cimpoi '15 [5]	66.7
Zhang '14 [40]	74.9
Branson '14 [3]	75.7
Lin '15 [23]	80.9
Simon '15 [30]	81.0
CNN (ours) 224px	82.3
2×ST-CNN 224px	83.1
2×ST-CNN 448px	83.9
4×ST-CNN 448px	**84.1**

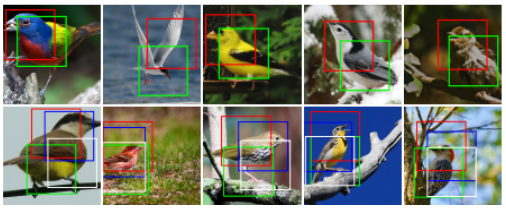

Figure 5: Applying Spatial transformer to CUB-200-2011 bird classification dataset

digit에 spatial transformation을 적용하는 것 뿐만 아니라, Figure5에서 새 이미지에서도 역시 ST layer가 적용되는 것과 성능이 향상되는 것을 확인할 수 있다. STN에서 ST layer는 여러 번 사용될 수 있으며 이 또한 성능이 향상될 수 있는 것을 보여준다. 이전 실험인 VGG와 ResNet에서 VGG는 Max pooling이 사용되지만 ResNet은 처음에 Max-pooling 한 번, Avg-pooling 한 번 적용된다. stride를 적용하여 이미지 사이즈를 축소시키는 것 보다 pooling을 하는 것이 좋지 않을까 하는 의문이 있었지만, 이번 STN의 paper를 읽으며 pooling layer의 한계점을 알 수 있었다.

Spatial transformer는 supervison없이 end-to-end로 전체 network가 학습이 진행되며, transformer 역시 미리정의되지 않고 학습 할 수 있다. 따라서, CNN의 spatial invariance를 위한 연구와 pooling, affine transform이 적용된 generative model 등에서 발견된 한계점을 보완하였다고 할 수 있다.

[1] Taco S Cohen and Max Welling. Transformation properties of learned visual representations. *arXiv preprint arXiv:1412.7659*, 2014.

[2] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International conference on artificial neural networks*, pages 44–51. Springer, 2011.

[3] Marijn F Stollenga, Jonathan Masci, Faustino Gomez, and Jürgen Schmidhuber. Deep networks with internal selective attention through feedback connections. *Advances in neural information processing systems*, 27, 2014.

[4] Tijmen Tieleman. *Optimizing neural networks that generate images*. University of Toronto (Canada), 2014.