CMDNet: 클릭 가능한 모바일 화면 객체 탐지를 위한 싱글 샷 아키텍처

조민석*, 한성수**, 정창성*
*고려대학교 전기전자공학과
**강원대학교 자유전공학부
jms0923@korea.ac.kr, sshan1@kangwon.ac.kr, csjeong@korea.ac.kr

CMDNet: Single Shot Architecture for Clickable Mobile Screen Object Detection

Min-Seok Jo*, Seong-Soo Han**, Chang-Sung Jeong*
*Dept. of Electrical and Engineering, Korea University
**Dept. of Division of Liberal Studies, Kangwon National University

요 약

모바일 디바이스 화면에 대하여 클릭 가능한 객체를 인식하기 위한 Object detection network architecture 를 제안한다. DSSD 를 Baseline 으로 SE block 이 추가된 Backbone network 와 SSD layer, FPN 구조를 사용한다. 기존의 1:1 비율의 네트워크의 Input resolution 을 모바일 화면과 유사한 1:2 비율로 변경하여 효율적으로 피처를 추출한다. 또한 해당 모델을 학습하기 위한 효율적인 데이터셋을 구축한다. 모바일 화면에서 클릭 가능한 객체를 기준으로 데이터를 수집하여 총 24,937 개의 Annotation data 를 Text, Image, Button, Region 등 8 개의 카테고리로 세분화하였다.

1. 서론

기술 발전의 속도가 나날이 증가함에 따라 신제품의 개발 및 출시가 매우 짧아졌다. 이에 따라 제품을 정확하고 빠르게 테스트하는 기술이 전자제품 시장에서의 경쟁력을 나타내고 있다. 또한, 제품의 종류와 숫자가 크게 증가해서 다양한 제품을 추가구현 없이 테스트할 수 있어야한다. 현재 모바일 기반의 제품을 테스트할 때는 Agent 를 기반으로 테스트한다. 그래서 같은 어플리케이션일지라도 디바이스 마다 다른 테스트 스크립트를 작성하여 개별로 테스트해야 한다. 이는 디바이스 의존도가 굉장히 높다. 이러한 문제는 제품에 Agent 를 설치를 하지 않고 Device-independent 하게 Test Case 들을 생성하고자 하는 요구로 이어진다. 따라서, 이미지나 영상을 활용해서 Test 를 진행하는 AI 기반의 Test Case 를 찾고 분류하는 특정 Domain 에 대한 기술이 필요하다. 본 논문에서는 이러한 문제를 해결하기 위한 Object detection 모델인 Clickable Mobile Screen Detection Network(CMDNet)를 소개한다. 해당 Architecture 는 모바일 스크린 이미지를 대상으로 설계하였다. 기존 모델의 Input resolution 비율인 1:1 을 모바일 스크린 이미지 비율과 유사한 Width, Height 1:2 의 비율로 변경하여 이미지의 변형에서 오는 피처의 손실과 변이를 최소화하였다. 넓은 범위와 초고해상도를 가진 모바일 스크린 이미지를 커버하기 위해 다양한 해상도에서 인식이 가능하도록 구성하였다. 그 과정에서 해상도, 시간, 정확도면에서 효율적인 트레이드 오프 관계를 도출하였다. CMDNet 은 모바일 스크린 이미지에 대해 10.2 FPS 가 나왔으며, 86.5 Mean Average precision(mAP)를 보였다.

2. 관련 연구

SSD[1]는 기존 Yolo[2]보다 향상된 속도와 mAP 를 얻었지만 작은 물체에 대한 성능은 다소 떨어진다. 이를 향상하기 위한 모델이 DSSD[3]이다. 기존의 SSD 의 피처 추출에 사용되었던 VGG[4]를 Resnet[5]기반의 Residual-101 로 교체하여 모델의 속도를 향상시켰다. 또한 Deconvolution 연산을 추가함으로써, 속도를 상대적으로 유지하면서 작은 객체들에 대하여 탐지 성능을 높였다.

2.1 Deconvolution Module

Deconvolution module 에서는 크기에 불변한 High-context 정보들을 효율적으로 활용하기 위해, 비대칭

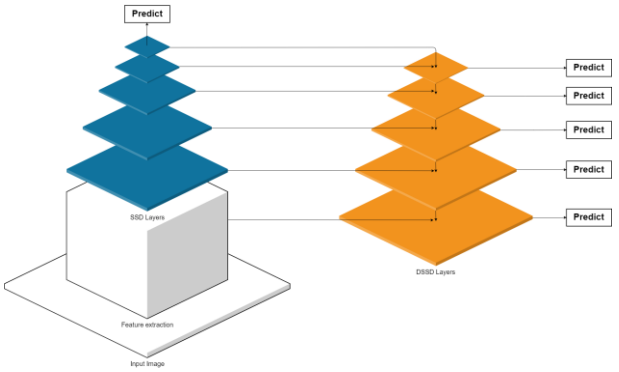
의 Hourglass[6]구조를 활용하였다. 또한, 대칭의 구조를 갖게 되면 추론시간이 두배로 늘어나기 때문에 비대칭 구조를 채택했다. Deconvolution module 의 두개의 인풋을 합치는 연산에서 Element-wise product 연산을 사용하였다.

2.2 Prediction Module

Prediction module 의 경우 Residual block 에서 Skip connection 에 한번의 Convolution layer 를 추가함으로써 기존 Residual block 보다 향상된 성능의 block 을 도출하였다. 테스트 시에는, 배치 정규화 과정을 제거함으로써 기존 SSD 대비 테스트 시간을 1.2 에서 1.5 배 줄였다.

3. Clickable Mobile Detection Network(CMDNet)

3.1 아키텍처



(그림 1) CMDNet Architecture

CMDNet 의 전체적인 구조는 (그림 1)과 같다. SSD 와 FPN[7]구조를 기본으로 한다. 크게, Backbone network 와 직접적인 결과를 도출하는 Header, Backbone 과 Header 를 잇는 Neck 으로 나눌 수 있다. Backbone network 는 SENet[8]을 사용하여 기존 DSSD 의 Resnet 대비 오버헤드의 증가 없이 중요한 피처를 추출하였다. Backbone network 를 통해 추출한 피처를 SSD layers 를 통해 Header 로 연결한다. 이때 FPN 형태를 이용하여 스테이지 별로 추출한 피처를 Deconvolution module 을 통해 동일한 채널수로 맞춰준다. Header 는 Prediction module 을 사용하여 Localization 과 Classification 을 수행한다.

기존 대부분의 Network architecture 는 Width 와 Height 의 비율이 1:1 을 크게 벗어나지 않는다. 하지만 모바일 화면의 경우 대부분의 1:2 의 비율을 가지고 있으며 해상도가 굉장히 높다. 해당 문제를 해결하기 위해 모델의 인풋 해상도를 1:2 의 비율로 정의하였다.

3.2 구현 세부사항

Ground Truth(GT) box 와 Intersection Over Union(IoU) 값이 임계 값(e.g. 0.5) 이상인 Predicted anchor box 만 학습에 사용하였다. Loss 함수는 Regression loss 와 Classification loss 두개의 Loss 함수를 합쳐서 사용했다. Regression loss 는 Smooth L1 을 사용하였으며, Classification loss 는 Cross entropy 를 사용했다.

캡쳐 된 모바일 스크린 이미지는 카메라로 찍은 데이터와는 달리 데이터셋 내부의 객체 하나하나가 흔들리거나 휘는 등 크게 변이될 가능성이 작다. 그래서 Random expansion augmentation trick 은 사용하지 않았다. 대신에 Horizontal flip 과 Vertical flip 을 적용하고 색상, 채도, 명도를 랜덤하게 변경하여 데이터셋의 일반성을 강화했다.

4. 실험

모바일 화면 데이터셋을 구성하여 4 장에서 제안한 모델을 실험하였다. 예측한 객체의 좌표와 GT box 를 기준으로 mAP 을 지표로 사용하였다. Baseline 으로 DSSD 모델을 사용하여 CMDNet 과 비교하였다. 모델의 속도를 비교하기 위해 FPS 를 사용하였다.

4.1 데이터셋

모바일 디바이스의 화면을 대상으로 데이터셋을 구축하였다. 다양한 해상도를 가진 디바이스들로 데이터 수집이 이루어졌으며 다수의 어플리케이션 화면을 대상으로 수집하였다. 총 1,261 장의 이미지에 대하여 24,937 개의 Annotation data 를 수집하였다. 전체 데이터 셋 중에서 371 장의 7,045 개의 Annotation data 는 Validation set 으로, 나머지 890 장의 17,892 개의 Annotation data 는 Training set 으로 나누었다. Class 는 클릭 가능한 객체를 기준으로 선정하였다. Text, Image, Button, Region, Status bar, Navigation bar, Edit text 의 7 개 Class 를 두었다. 데이터셋 규격은 VOC[9]규격을 따랐다. 각 Class 별 Annotation data 는 <표 1>과 같다.

<표 1> 클래스별 Annotation data

Class	Number of Annotation data
Text	8,462
Image	7705
Button	2,165
Region	3,563
Status bar	1,228
Navigation bar	721
Edit text	1,093

Text Class 가 8,462 개로 가장 많았으며, Navigation bar 가 721 개로 가장 적었다. 데이터셋의 예시는 (그림 2)와 같다.

4.2 트레이닝

구축한 데이터 셋 중에서 Training set 을 학습하였다. Multi-scale training 은 사용하지 않았다. 클래스 불균형 문제로 인한 False negative 문제를 해결하기 위해 Hard negative mining 을 사용하였다. GTX 2080Ti GPU 를 사용하여 Loss 가 0.1 이하로 떨어질 때까지 학습하였다. 0.001 의 Initial learning rate 에 0.9 의 Momentum 을 사용하였다. 또한, Overfitting 을 피하기 위해 Weight decay 는 0.9 로 설정하였다.

(그림 2) Dataset GT

4.3 실험 결과

실험 결과는 <표 2>와 같다. Baseline 인 DSSD-512 는 Resnet-101 을 Backbone 으로 사용하였고 512 x 512 의 1:1 비율의 Input resolution 을 가지고 있다. mAP 는 43.0 이며 FPS 는 12.9 이다. CMDNet-512 는 512 x 1024 의 1:2 비율의 이미지를 Input resolution 으로 가진다. Backbone network 로 SENet-101 을 사용하였다. mAP 는 68.4 로 DSSD-512 와 비교하여 59%상승하였다. FPS 는 10.2 로 초당 2 장정도 느린 속도를 보였다. CMDNet-1080 은 1080 x 1920 의 Input resolution 을 가지고 있다. CMDNet-512 와 마찬가지로 SENet-101 을 Backbone network 로 가진다. mAP 는 86.5 로 DSSD-512 에 비하여 101%, CMDNet-512 에 비하여 26.4% 증가하였다. FPS 는 DSSD-512 에 비하여 초당 3.4 하락하였으며, CMDNet-512 에 비하여 0.7 떨어졌다. Inference 결과는 (그림 3)과 같다.

(그림 3) Inference results

<표 2> Baseline 모델과의 비교

Method	Backbone	mAP	FPS	Input resolution
DSSD 512	Resnet-101	43.0 (Base)	12.9 (Base)	~ 512 x 512
CMDNet 512	SENet-101	68.4 (+59.0%)	10.2 (-2.7)	512 x 1024
CMDNet 1080	SENet-101	86.5 (+101.1%)	9.5 (-3.4)	1080 x 1920

5. 결론

본 논문에서는 모바일 디바이스의 화면에 대하여 클릭가능한 객체를 찾기 위한 Object detection network architecture 를 제안한다. 기존의 Backbone 인 Resnet 을 SENet 으로 교체하고, SSD layers 를 통해 FPN 구조를 쌓아 Header 로 연결하였다. 또한 해당 모델을 학습하기 위한 모바일 화면 데이터셋을 구축하였다. 데이터셋은 8 개의 클래스로 구분하였으며, 24,937 개의 Annotation data 를 수집하였다. 제안한 모델과 구축한 데이터셋을 이용하여 실험을 진행하였고 기존 DSSD 모델에 비하여 향상된 성능을 입증하였다.

이 논문은 2021 년도 4 단계 BK21 사업에 의하여 지원되었음.

참고문헌

[1] LIU, Wei, et al. Ssd: Single shot multibox detector. In: European conference on computer vision. Springer, Cham, 2016. pp. 21-37.

[2] REDMON, Joseph, et al. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas, 2016. pp. 779-788.

[3] FU, Cheng-Yang, et al. Dssd: Deconvolutional single shot detector. arXiv:1701.06659, 2017.

[4] SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, 2015. pp. 7-9.

[5] HE, Kaiming, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. pp. 770-778.

[6] NEWELL, Alejandro; YANG, Kaiyu; DENG, Jia. Stacked hourglass networks for human pose estimation. In: European conference on computer vision. Springer, Cham, 2016. pp. 483-499.

[7] LIN, Tsung-Yi, et al. Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. pp. 2117-2125.

[8] HU, Jie; SHEN, Li; SUN, Gang. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. pp. 7132-7141.

[9] EVERINGHAM, Mark, et al. The pascal visual object classes (voc) challenge. International journal of computer vision, 88.2: pp. 303-338, 2010.