**Pixel Link: Detecting Scene text via Instance Segmentation.**

**(Instance Segmentation 방식을 통한 text 영역 감지하기.)**

**-논문 정리 Pixel Link, Deng et al., AAAI2018-**

**201421553**

**소프트웨어학과**

**김 한**

**Abstract**

최신의 scene text detection algorithms은 대부분 deep learning 기반의 방법론인데 **bounding box regression에 기반한 학습 기반이며 text / non-text classification 과 location regression 이라는 두 종류의 예측(Prediction)을 수행**하게 된다. **regression은 bounding box를 찾는 중요한 역할을 수행하긴 하지만 반드시 필수적인 것이 아닌 것**이 text/not-text prediction이 text 위치정보 자체를 포함하고 있는 semantic segmentation 의 한 종류로써 여기지기 때문이다. (즉, text/non-text prediction 자체가 bounding box를 찾기도 한다는 말 같음. **= regression이 필수는 아니다**) 그러나 text instance는 image 내에서 text끼리 사이가 너무 가깝다는 특성 때문에 의미단위의 segmentation을 분리하는 것이 어렵다는 문제가 존재한다. 따라서 instance 분리 기법이 이 문제에서 요구 되어진다. 이 논문에선 **Pixel link라는 새로운 알고리즘(instance segmentation 기반)을 제안**한다. **먼저 Text instance는 같은 text instance 안의 pixel들 끼리 묶이면서 각각이 분리**된다. 그리고나서 Text bounding box가 기존의 Location Regression 없이 segmentation 결과만으로 직접적으로 추출된다. 실험으로 기존의 Regression-based 방법론과 비교하여, Pixel Link 방식이 여러 benchmark에서 훨씬 더 좋은 성능을 얻고 동시에 **training iteration과 적은 training data의 양을 요구**한다는 것을 보여 주고자 한다.

|  |  |
| --- | --- |
|  | **Bounding box regression : R-CNN을 사용 중 detection 정확도 향상 방식**으로 RCNN에 Object Detection에 적용된 regressor이다. 먼저 Regional Proposal을 위해 Selective Search를 진행하여 정보를 얻어낸다. 이 proposal의 feature를 계산하고 각 region의 classification 결과와 bounding box regression을 계산함으로써 box의 위치를 예측하거나 refine한다. 즉, **bounding box regression 이란 bound box의 parameter를 찾는 regression**으로 초기의 region proposal이 CNN이 예측한 결과와 맞지 않을 수 있기 때문에 **마지막 pooling layer의 feature 정보를 사용해 region proposal의 regression을 계산하는 방식**을 말한다. |

|  |  |
| --- | --- |
|  | **Semantic segmentation :** 모든 픽셀의 레이블을 예측하는 방식. 다른 말로 dense prediction 이라고도 불리며, 다음과 같이 하나의 object를 찾는 classification 을 발전시켜 위치까지 찾는 localization에서 복수개의 object를 찾는 object detection, 마지막으로 이를 densely하게 찾는 방식인 segmentation으로 발전된다. 여기서 semantic segmentation은 같은 class 내에 instance를 구별하지 않는다는 특징이 있으며, **각 instance를 구별하는 advanced 방식을 instance segmentation**이라 한다. |

**기존의 Image Processing 을 위한 CNN의 동작방식**

1. **Sliding Window를 이용**한 일종의 Brute Force한 방식으로 Image 전체를 Searching.

1-(1). 전체 Image를 모두 searching 해야 하므로 시간소비가 막대함.

1. Selective Search 알고리즘을 이용한 **Region Proposal**을 통해 2000개의 Object가 존재 할 만한 Region을 추천해줌.

2-(1). **추천된 Region + CNN 방식**이 바로 R-CNN 방식임. **따라서 추천 받은 region만 CNN이 탐색**을 하게 되므로 속도도 향상되고 정확도도 올라가는 효과가 생김. 그러나 CNN이 Positional 분산이

어느정도 존재하기 때문에 **정확히 박스 중앙에 들어오지 않아도 높은 스코어로 예측하**

**는 문제**가 발생.

1. Bounding Box Regression을 이용하여 **정확한 위치를 regression 해내고 Localization의 성능을 끌어올리는 효과**를 얻어냄.

* 이를 **Bounding Box Regression 방식**이라고 한다.

1. **Introduction**

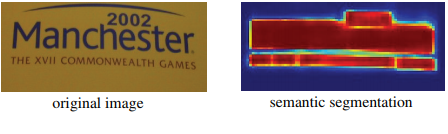
Reading text in the wild 와 robust reading은 오랫동안 흥미롭게 다뤄왔다. 그리고 이것은 보통 두가지의 sub-task로 분리되는데, **“text detection 과 text recognition**”으로 나뉜다.

**Detection**은 다른 말로 **localization**이라고도 하는데 이는 input으로 이미지를 받고, 이미지 안의 text의 위치를 output으로 내보내는 것을 주요 목표로 한다. Deep Learning 과 Object Detection의 큰 발전과 함께 훨씬 더 효율적인 알고리즘들이 제안되어 왔으며, CTPN, TextBoxes, SegLink, EAST과 이 예이다. 그리고 이러한 알고리즘들은 보통 두가지의 예측을 수행한다.

1. **Text/non-text classification**(text와 그렇지 않은 영역 분리). 이 **예측은 text bounding box 안에 존재하는 pixels의 확률을 취하는 방식**이다. 하지만 text/non-text classification은 보통 **Regression의 결과의 신뢰도(Confidence)로 활용**된다.
2. **Location regression**, text instance의 location 또는 instances의 segment/slice는 reference box로부터 offsets(상대적 위치)로서 예측된다. 또는 bounding box의 absolute location로서 예측된다.

**segLink와 같은 방식에선 추가적으로 segment들 사이에서의 linkage 또한 예측된다.** 이러한 **예측이 진행된 후에는, NMS 또는 segment를 취합하는 후처리가 최종 아웃풋인 bounding box를 얻기 위해 적용된다.**

.location regression은 오랫동안 object detection과 text detection에 사용되어져 왔다. 그리고 그 효율성이 증명되었다. Regression은 최근 기술에선 보통 text bounding box의 형성하는데 핵심 역할을 수행하고 있다. 그러나 위에서도 언급했지만 **text/non-text prediction 이 regression의 결과에 대한 신뢰도(confidence)로 사용되기도 하지만 그 자체로 이미 location 정보를 포함하는 segmentation score map으로써 사용**된다. 따라서 **regression은 필수적인 것이 아니다.**



그러나 그림 1에서 볼 수 있듯이, **text instance 끼리 매우 가깝게 붙어있다. (분리가 힘들다.)** 이러한 경우엔 의미 단위로 분리하는 것이 거의 불가능하다. 따라서 **instance level에 대한 segmentation**이 보다 더 필요하다. 이 문제를 풀기 위해서 PixelLink라는 방법론을 이 논문에 제안한다. 즉, **bounding box regression의 방식 대신에 instance segmentation의 결과로 곧바로 text location을 뽑아낸다**. 이를 위해 Pixel Link는 2가지 deep learning train을 진행하는데 첫번째로 **text-non/text prediction 과 link prediction**을 수행한다. 구체적으로 Text instance 내의 pixel은 positive 로 그렇지 않으면 negative로 labeling된다. 특히 이 link의 개념은 위에서 소개한 segLink에 의해 생각되어 졌는데 아주 큰 차이가 있다. 그것은 **모든 픽셀은 8방향의 이웃 픽셀을 가지고 있다는 것이다.** **따라서 주어진 픽셀과 그 픽셀의 이웃중의 하나라도 같은 instance에 속한다면 positive 로 labeling 하고 그렇지 않으면 negative로 labeling 하는 것이다**. 이렇게 예측된 positive link애 의해 Connected Components로 positive 픽셀들이 취합된다. 이러한 방식으로 instance segmentation이 완성되며, 각각의 Connected Component가 detect된 text가 되는 것이다. 다음으로 OpenCV의 minAreaRect를 이용하여 최종 detect된 bounding box를 얻을 수 있다.

CTPN : <https://arxiv.org/pdf/1609.03605.pdf>

TextBoxes : <http://jaynewho.com/post/6>

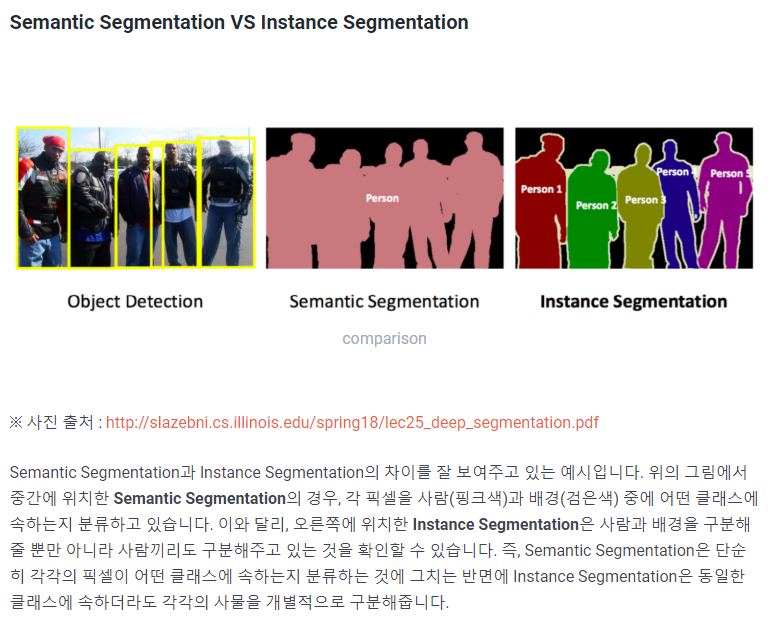
SegLink : <https://arxiv.org/pdf/1703.06520v3.pdf>

EAST : <https://brunch.co.kr/@kakao-it/318>

**2.Related Work**

* 1. **semantic & instance Segmentation(본 논문은 instance 방식 사용)**

segmentation task는 pixel-wise(픽셀별로) label을 이미지에 할당하는 것이다. 오직 Object 카테고리에 대해 고려될 때 이것을 semantic segmentation이라 부르고, FC Net을 사용한다. Instance Segmentation은 각 pixel에 대한 object 카테고리가 필요할 뿐만 아니라 instance 간의 차별 역시 필요로 하기 때문에 훨씬 더 어려운 문제이다. 또한 object instance 를 인식해야 한다는 점에서 semantic segmentation 이라기 보단 object detection에 보다 가깝다.



**FC net이란?** Fully Convolutional Network 의 약어로 2015년에 발표되었다. 별다른 어려운 기법을 사용하지 않았음에도 불구하고 semantic segmentation에서 워낙 뛰어난 성능을 보였다. Classification 문제를 해결하기 위해 통상적으로 **여러 단계의 convolution + pooling 으로 구성된 네트워크 뒤에 fully connected layer가 오는 형태가 일반적**이다. 그러나 이렇게 **Fully Connected Layer를 거치고 나면 위치나 공간에 관련된 정보는 모두 사라지게 된다.** Detection은 classification 문제 위치정보를 포함하는 문제를 말하는데 보통 bounding box라고 부르는 사각형 영역을 통해 대상의 위치정보까지 얻게 된다. 위에서 언급했던 R-CNN과 같은 것에서 class를 구분하는 SoftMax와 bounding box regressor의 구성을 통해 box의 위치정보까지 알아낼 수 있다. Semantic Segmentation은 단순히 bounding Box를 이용하여 원하는 대상을 나타내는 것이 아니라 **픽셀(pixel-wise) 단위로 labeling을 통해 예측을 수행하고 분리하는 것**을 말한다. 따라서 이를 위해 **FC Net은 마지막 뒤단계의 Fully Connected net을 1x1 의 convolution layer로 정의**함으로써 **위치나 공간의 정보를 잃어버리지 않음으로써 위에서 언급한 semantic segmentation의 일을 수행**할 수 있게 되었다.

* 1. **segmentation-based text detection**

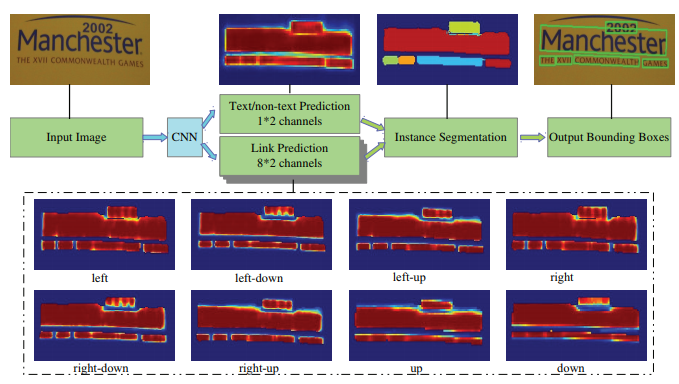
segmentation based 방식은 오랫동안 text detection에 채택되어져 왔다. Text/non-text, character classes, 그리고 character linking orientation 문제에 대해 예측함으로써 semantic segmentation 으로써 detection task를 수행했다. 여기선 단어나 라인 단위로 그룹핑되고 FCNet에 의해 saliency map(영상에서 눈에 띄는 영역) 으로부터 text block이 발견된다. 그리고 character candidates가 MSER**(가장 우수한 영역 검출 알고리즘 중 하나로 전처리로서 텍스트 후보영역을 찾는데 사용)**을 사용하여 추출된다. 단어나 라인은 hand-crafted rule(사람이 직접 추출한 규칙)을 사용하여 형성된다. Coarse(듬성듬성한 vs fine) 네트워크는 text region의 heat map을 대략적으로 생성하는데 사용되고나서 fine(정밀한) text network 에 의해 보다 정제된 detected region을 가지게 된다. 그러나 이 방법은 후처리에서 시간을 많이 잡아먹거나 만족스럽지 못한 결과를 가져온다.



* 1. **regression-based text detection (여긴 아직 정리 x)**

이 카테고리의 대부분의 방법론은 object detection의 개발의 이점을 취한다. CTPN은 RPN과 R-CNN으로 확장했다. Text slice를 예측하기 위해서. TextBoxes , SSD의 방식들은 일반적이지 않은 shape의 커널과 보다 넓은 aspect ratio 의 anchors를 채택했다. Scene text의 large-aspect-ratio 특징을 들어 맞추기 위해서. SegLink는 Linkage 예측을 사용하여 완전한 instance 묶어내는 text segment를 예측하기 위해 SSD를 채택했다. 또한 EAST는 NMS를 이용하여 처리되는 밀도 있는 예측을 수행한다. 이 논문에서는 textboxes, CTPN, SegLink, 그리고 EAST를 포함하는 최고의 성능을 냈던 공표 방식을 참고한다.(결과비교)

**Detecting Text via Instance Segmentation (Instance Segmentation을 통한 text Detect)**

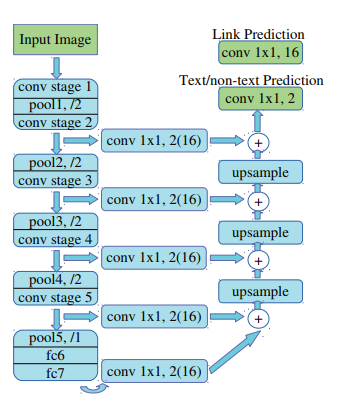


위의 그림과 같이 PixelLink는 instance segmentation을 통한 text를 detect를 수행한다. 예측된 positive pixels이 positive link에 의해 text instance로 결합된다. 그리고나서 Bounding box가 segmentation의 결과로 추출된다.

**즉, 정리를 해보면 본 논문은 instance segmentation을 진행한다. 그를 위해서 CNN으로 semantic segmentation이라는 text/non-text prediction을 진행하고 연달아 link prediction을 통해 가깝게 붙어있는 text 사이를 분리해내고 이를 합쳐서 instance segmentation을 완성하겠다는 말을 하고 있음.**

**Network Architecture**

다음의 SSD와 SegLink는 FCNet을 가지고 feature Extractor로 사용된다. 예를 들어 Fc6와 fc7은 conV로 전환된다. (Fully Convolutional Network의 사용)

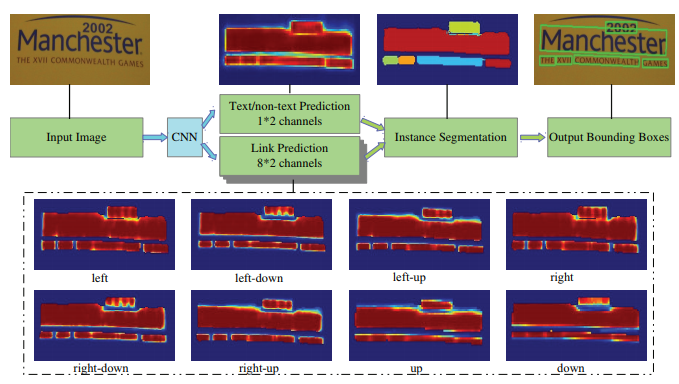


위의 그림처럼 전체 모델은 text/non-text 를 예측하는 부분과 link prediction이라는 2개의 파트로 나뉜다. 양쪽에 모두 SoftMax가 사용되고, 그들의 output은 1\*2=2 와 8\*2=16 채널을 가진다.

**SoftMax란?**

**Linking Pixels Together**

주어진 pixel과 link 에 대한 예측은, 두 다른 thresholds는 개별적으로 적용된다. 그리고나서Positive pixel은 positive links를 사용하여 그룹핑되면서 CC라는 detected text instance를 얻을 수 있다. 즉, instance segmentation을 할 수 있다. 또한 이 linking 과정은 disjoint 자료구조를 사용하여 구현된다.



그림에 대한 설명: PixelLink에 대한 네트워크 Architecture에 해당한다. 먼저 CNN 모델이 두종류의 pixel-wise prediction을 수행하기 위해 학습된다. 첫번째 원하는 예측이 text와 text가 아닌것을 구별하는 예측과 link prediction이 두번째이다. 이렇게 두개로 나뉜 예측이 된 후에, positive pixel들은 positive link에 의해 묶이게 됨으로써 instance segmentation을 수행하게 된다. minAreaRect가 다음으로 적용이 되는데 이는 instance segmentation의 결과를 가지고 bounding box를 뽑아내는 부분에 적용된다. Noise가 있는 prediction은 post-filter으로 효과적으로 제거될 수 있다. 이러한 Pixel Link 방식 중에서 text-non-text prediction을 통해 몇 단어가 구별이 되지 않더라도 link prediction을 통해 완전히 구분해낼 수 있다.

**오늘 내용을 정리를 해보면..**

**먼저 image를 input로 해서 CNN이 2가지를 학습하게 됨. 1.text와 text가 아닌 부분을 찾아내는 text-non-text prediction 을 위한 학습 2.단어와 같은 사이사이의 연결된 미세한 부분을 segmentation하는 Link Prediction을 위한 학습을 하게 됨. 이후에 이 두개의 positive pixel들이 positive link에 의해 묶이게 되는데 이 과정에서 disjoint 자료구조를 이용함으로써 결과적으로 이 논문이 말하는 instance segmetation의 결과를 뽑아 낼 수 있음. 그후에 noise를 좀 더 제거하기위해 post-filter를 적용하고 minAreaRect라는걸 이용해서 bounding Box를 찾게됨으로써 최종적으로 text localization(detection)을 수행하게 되는 것으로 보임.**

**사용하는 다른 최신기술 논문: text-non-text prediction을 위한 방법론으로 text slice를 찾기위한 CTPN + TextBoxes 이용. SSD라는 보다 다양한 Boxing shape를 찾는 방법이용. SegLink라는 SSD와 Linkage Prediction을 Instance 화하는 방법 사용. EAST라는 NMS를 이용한 보다 정밀한 예측방식 사용**

**3.3 Extraction of Bounding Boxes**

사실 detection task는 instance segmentation 후에 완성된다. 그러나 bounding box가 detection의 최종 결과물로 요구되어진다. IC13과 같은 대회에서의 Data를 통해. 따라서, 연결된 요소들의 bounding Box는 MinAreaRect 라는 openCV의 방식을 통해 추출되어진다. MinAreaReat 의 output은 oriented Rectangle 의 형태이며 이것은 쉽게 IC15에선 quadrangles 그리고 IC13 에선 rectangles 로 전환될 수 있다. PixelLink에서도 충분히 언급할한 것이 scene text 의 orientation에 제한이 없기 때문이다. 이부분이 regression 기반 방식과 pixelLink의 핵심차이라고 할 수 있다. 즉, bounding box 가 location regression 과 달리 instance segmentation 단계에서 얻어질 수 있다.

**3.4 Post Filtering after Segmentation**

PixelLink는 links를 통해 pixel를 그룹핑하는 작업을 하기 때문에 예측에 대해 noise가 생길 수 밖에 없다. 따라서 post filtering 단계가 필수적이라 할 수 있다. 여전히 이에 대한 효율적인 방법으로 detected 된 box의 기하학적 특징을 통해 filtering을 해내는 것이다. 예를들어 width , height, area, aspect ratio 등을 이용하는 것이다. IC15 training data에 대해서 구체적으로 선택되어진 필터링 기준에 대해서 계산된 training set에 대해서 계산된 것 중 99퍼센트로 일치하는 것이 threshold value로 선택되어 진다. 예를 들어 다시 10이 shorter side length위에서 threshold로 정해지는데 이는 99퍼의 text instance가 10 pixel보다 큰 shorter side를 가지기 때문이다.

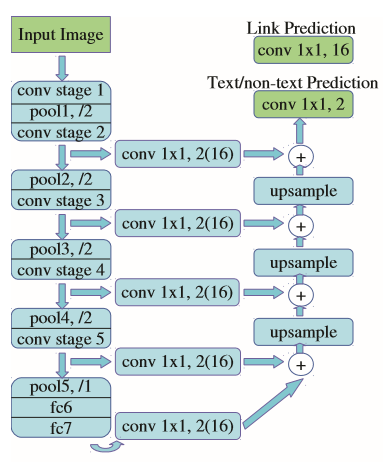
**4.optimization**

**4.1 ground truth calculation**

다음의 TextBlocks 라는 논문에 소개되고 있는 공식에선, text bounding box는 내부의 pixel들은 모두 positive로 labeling된다. 만일 중복된 것이 존재한다면 오직 중복되지 않는 pixel만 positive로 labeling이 되며 그렇지 않으면 negative로 labeling이 된다. 주어진 하나의 픽셀과 그것의 주위의 팔방향의 pixel에 대해서, 만일 그 픽셀들이 같은 instance에 속한다면 해당 그들 사이의 link는 모두 positive가 되고 그렇지 않으면 negative가 된다. Ground truth calculation은 conV3\_3 또는 conv2\_2와 같은 prediction later의 shape에서 resized된 input image에서 도출된다.

**4.2 loss Function**

Training의 loss는 link와 pixel에 대한 loss의 합으로 이루어져 있다. Link 의 loss는 오직 positive pixel로 계산이 되어지므로 pixel의 classification task가 link보단 훨씬 중요하다. 또한 람다의 값은 모든 실험에서 2.0으로 셋팅 되어 있다.



위의 그림은 pixelLink vgg16 2s 모델의 구조이다. Fc6,fc7은 ConV로 전환된다. Upsampling 기능은 bilinear interpolation을 통해 직접적으로 수행되어진다. 다른 stage 로부터의 feature map은 Upsampling 과 add operation의 cascading을 통해 융합된다. 모든 pool5,를 제외한 pooling layer는 stride로 2를 취하고 5만 1을 취한다. 따라서 fc7의 사이즈는 conv5\_3과 같고 업샘플링이 필요 없다. Conv layer는 각각 text-non text prediction 또는 link prediction에 대해서 1x1에 2개 또는 16개의 커널을 가진다.

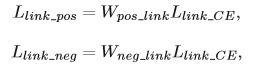
**Loss on pixels**

텍스트의 인스턴스 크기는 가변적이다. 예를 들면, 그림1에서 맨체스터라는 글씨는 다른 단어들을 다 합쳐도 더 크다. 따라서 loss를 계산할 때, 모두 같은 weight를 부여하는 것은 부적절한 방식이고 이는 성능에 영향을 줄 수 있다. 따라서 이 문제를 해결하기 위해서 instance-balanced cross entropy라는 새로운 weighted loss 방식을 소개한다. 구체적으로 말해서 n개의 text 인스턴스에 대해서 모든 인스턴스가 같은 weight를 부여함으로써 동등하게 여기는 방식이다. OHEM은 negative pixel을 선택하는 것을 적용한다. Negative pixel이 선택되면 다른 하나에 세팅하는 방식. R은 negative-positive ratio인데 이는 관행적으로 3으로 셋팅 한다. 이와 같은 방식의 메커니즘을 진행하면 weighted matrix를 구할 수 있는데 이를 W로 표시한다. 전체 pixel classification 에 대한 loss 식은 이러하다.



이는 text/non-text 영역의 예측을 하는 Cross Entropy Matrix이며 L(pixel\_CE) 라고 명명한다.

**Loss on Links**



Positive & negative Link는 다음과 같이 계산이 된다. 최종적으로는



다음과 같이 L(link)의 값이 계산된다.

**4.3 Data Augmentation(data 증식)**

SSD와 유사한 방식으로 data를 증식한다. 추가적인 random rotation step을 통해서. Input image는 처음 0.2의 확률로 rotated되며 angle은 0 파이/2 파이 2파이/3 으로 회전한다. 그리고나서 0.1~1의 범위내에서 crop된다. Aspect ration은 0.5에서 2까지 crop된다. 결국 균일하게 512x512로 resize되며, 이러한 augmentation 이후에 10픽셀 보다 작은 text instance는 무시된다. 20퍼센트 이하 역시 무시된다. 무시된 instance에 대한 loss를 계산하는 동안 0으로 셋팅 된다.

**5. experiments**

Data set으로 IC15 , IC13, TD500 을 이용하여 정확도를 계산하였다.

**5.2 Implementation Details**

pixelLink 모델은 SGD라는 **momentum = 0.9 weight decay = 5 x 10^-4 으로 최적화** 되었으며, fine tuning 이라는 imageNet pretrained model이 아니라 **VGG net은 Xavier 방법을 통해 random으로 초기화를 진행**한다. Learning rate는 **10^-3 + 100번의 iteration**이 진행되며 rest에 대해 10^-2으로 fixed 된다. 전체코드는 tensorflow와 파이썬으로 구현되었고 cython을 이용하여 compile 되었다. GPU(GTX titan X) 3개에 batch 사이즈 24로 실행되었다. Iteration마다 0.65초 전체는 7~8시간이 걸렸다.

**5.3 Detecting Oriented Text in IC15**training은 랜덤하게 초기화된 VGG16 모델로 시작한다. 4s 모델 같은 경우는 40K의 iterations을 반복한다. 2s 모델은 더 긴 60K의 iterations을 반복한다. Minimal shorter side length 와 area 는 post-filtering을 하는데 사용되며, 각각 10과 300으로 셋팅된다. 픽셀에 대한 thresholds와 link는 grid search로 찾게 되고 0.8 , 0.8로 셋팅된다. Input Image는 1280 x 768로 resize된다.