Preprocessing for Detecting Small Signs from Large Image

1. Introduction

Object detection은 현재 많은 분야에서 그 활용도가 높아지고 있다. 현재까지 이러한 object detection을 수행하는데 있어 CNN(Complex Neural Network)을 이용하여 왔다. CNN을 이용한 연구결과로 R-CNN, Fast-CNN, Faster-CNN, YOLO(You Look Only Once) 그리고 SSD(Single Shot Multibox Detector)와 같은 매우 효율적인 object detection system이 개발되었다.

그러나 큰 Input image에 대해 작은 object를 찾고자 할 경우 앞서 소개한 system들이 잘 찾아내지 못하였다. 더구나 input image의 사이즈가 크면 network에 넣고 수행함에 있어 memory부족이 발생하였다.

앞서 말한 memory문제를 해결하기 위해서는 network를 이전보다 간단하게 하여 성능에 저하가 발생할 수 있지만 뉴런 수를 줄이는 방법과 원래의 input image를 down sampling하여 사용하는 방법이 있다. 이 두 방법으로 필요한 memory를 줄일 수는 있지만 small object를 찾아내야 한다는 문제에는 답을 주지 못한다.

이를 해결하기 위해 아래와 같은 방법을 하려고 한다.

1) Input image로부터 학습을 통해 small object가 있을 확률이 높은 region을 먼저 찾는다.

2) 1)에서 구한 자료를 mask처럼 사용하여 Input image에 masking하여 small object가 있다고 생각하는 부분 외의 부분은 무시한다.

3) 2)에서 masking한 image를 maxpool을 통해 size를 줄이고 이를 YOLO와 같은 기존 network를 이용하여 찾고자 하는 object를 찾아낸다.

이 논문에서는 1)에 해당하는 작업을 수행하려고 한다.

2. Process

1) Data Set

앞으로 사용할 데이터 셋은 TT100K(Tsinghua-Tencent 100K)라는 교통표지판 detection & classification을 위해 만들어 놓은 데이터셋으로 아래의 <Figure 1>에 있는 여러 종류의 교통표지판을 포함한 2048x2048 크기의 <Figure 2>와 같은 야외 사진들로 이루어져 있다.

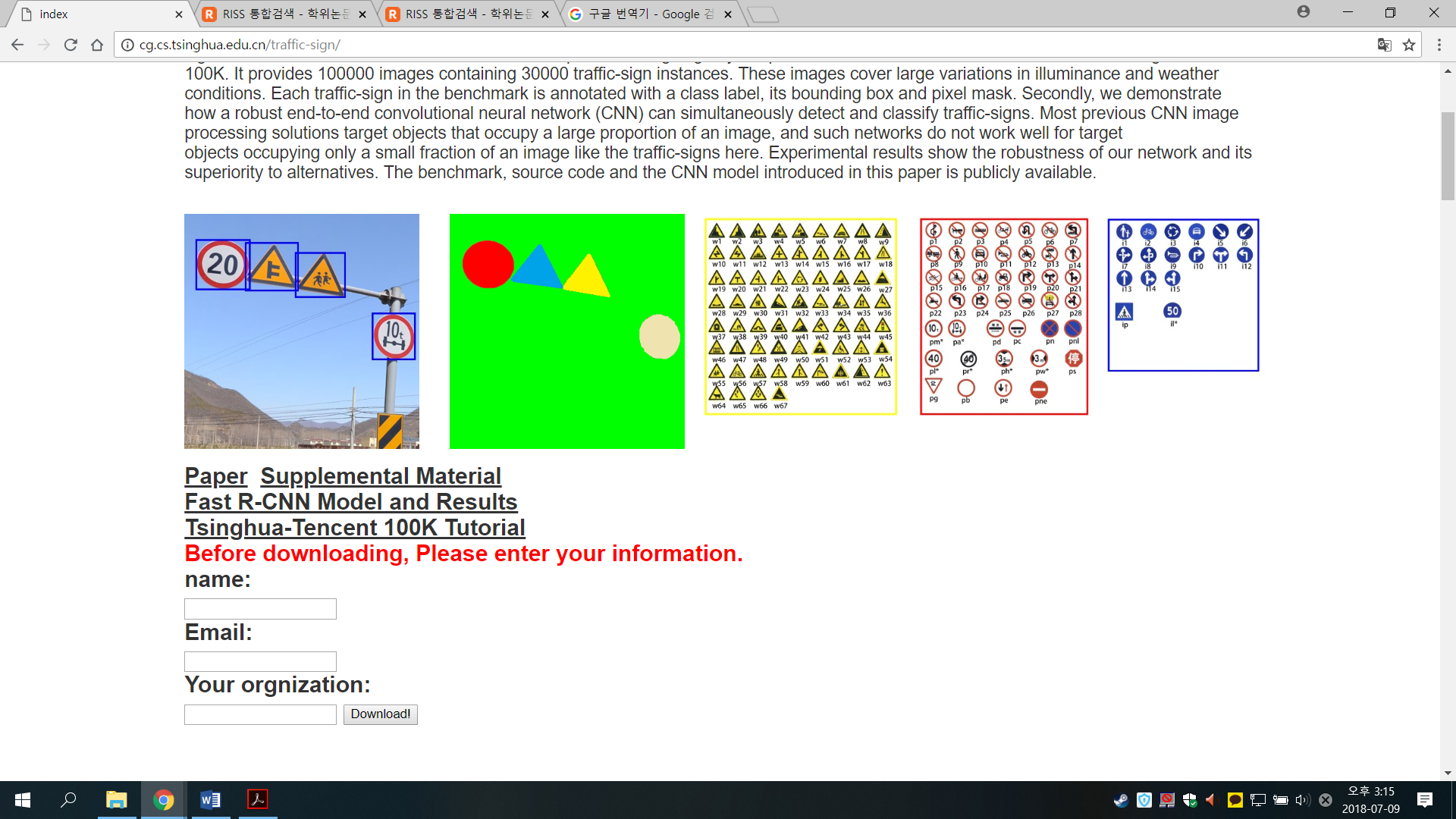


Figure 1. TT100K의 traffic sign 종류

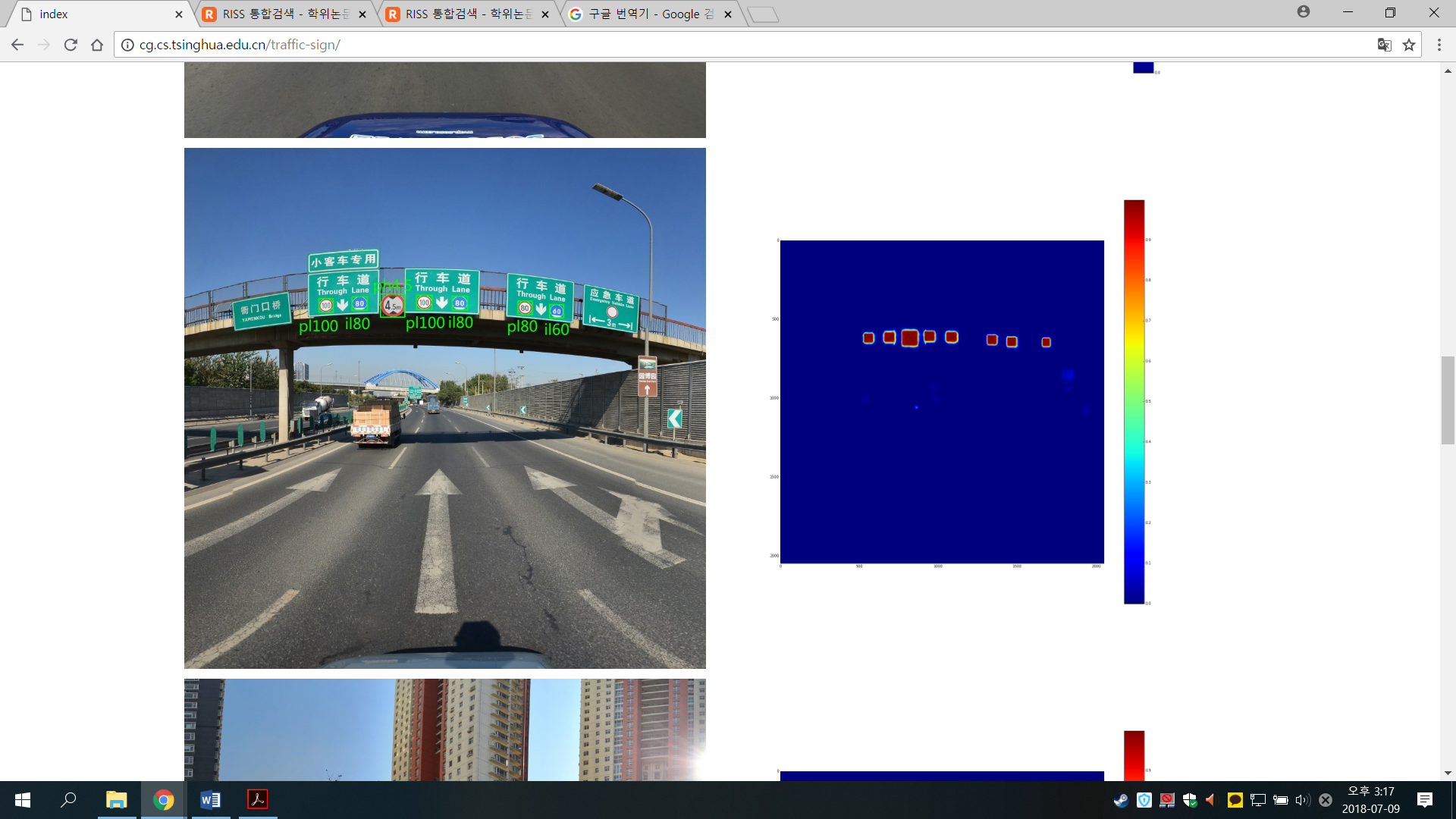


Figure 2. TT100K의 data 예시

Training Set: 6105

Test Set: 3071

2) Network

Network로는 Alexnet으로부터 변형시켜 사용하였다. Alexnet은 ILSVRC(Image Large Scale Visual Recognition Challenge)-2012에서 우승하면서 많은 각광을 받았고 Alexnet만큼 검증이 많이 이루어진 모델은 드물기 때문에 기존의 Alexnet로부터 2048x2048의 input size로부터 원하는 output size를 얻기위해 전체적으로 변형을 가하였다.

먼저 input size는 512x512x3이고 output은 32x32x1로 이미지에 찾고자 하는 object가 그 위치에 있을 확률로써 각 grid가 -1에서 1 사이 값으로 갖는다.

Alexnet은 224x224x3의 input size을 사용했기 때문에 <Figure 3>, <Figure 4>, <Figure 5>와 같이 network를 변형하였다.

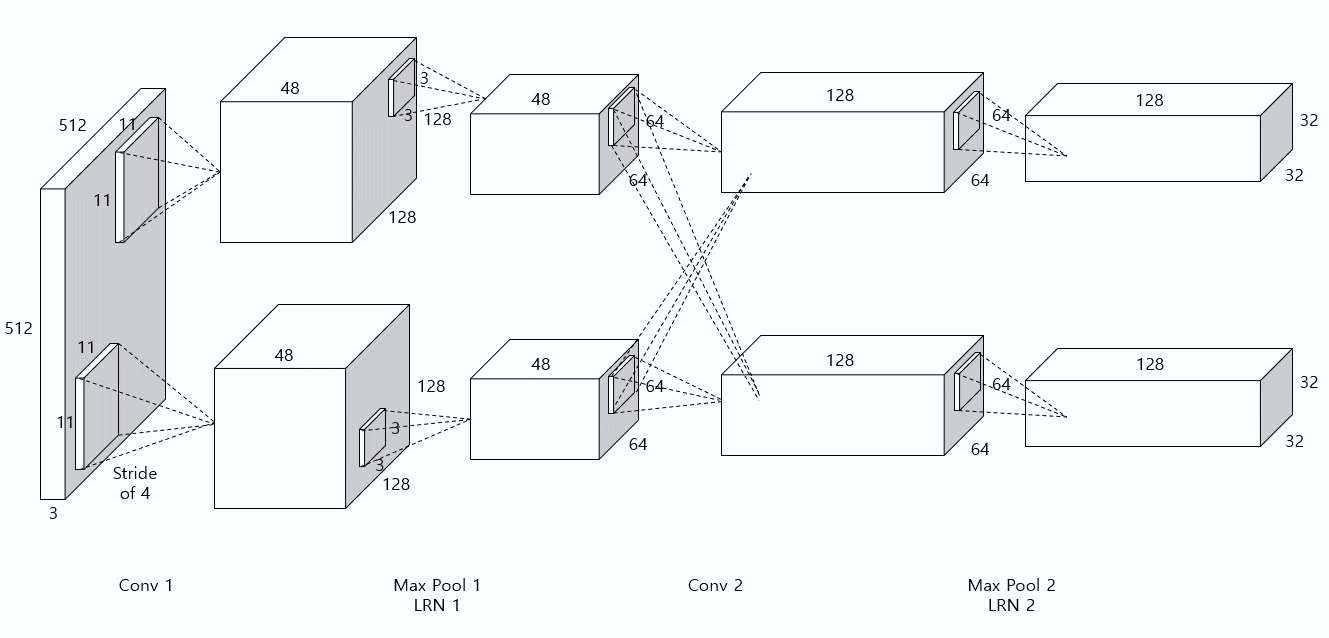


Figure . Alexnet

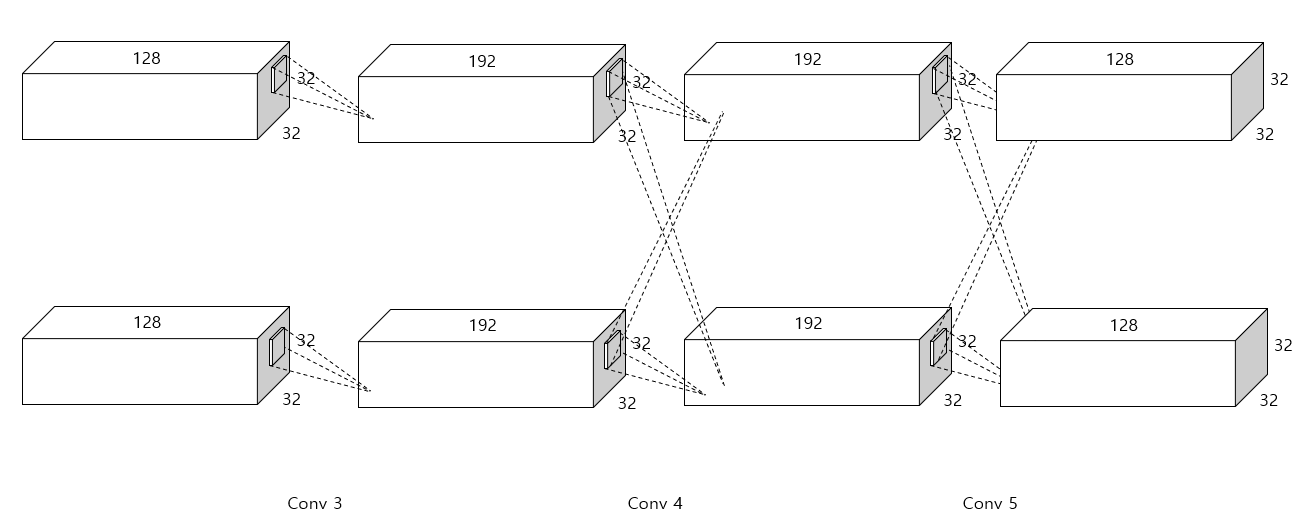


Figure . Alexnet

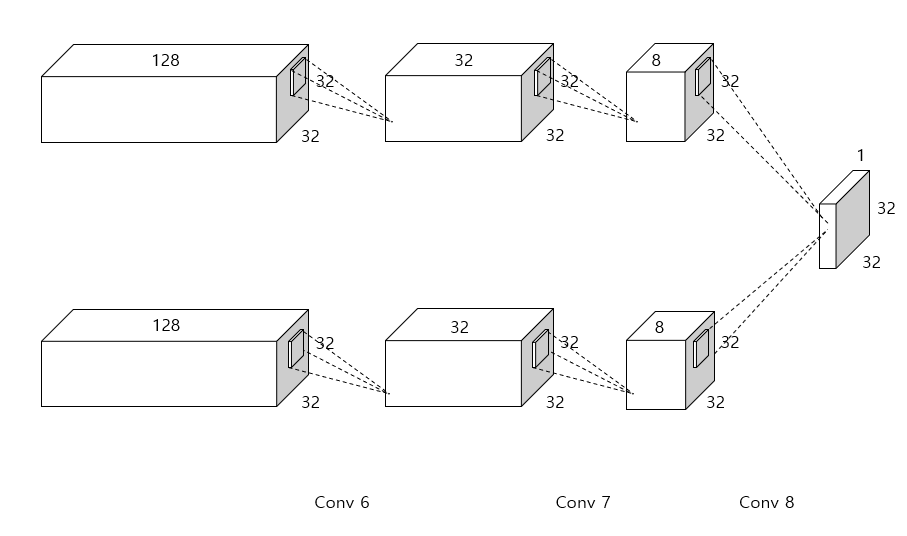


Figure . Alexnet

기존의 Alexnet에서 Fully connected layer를 모두 제거하고 convolution layer를 몇 개 더 추가하여 총 8개의 convolution layer와 2개의 max pool layer 그리고 2개의 local response normalization layer로 구성되어 있다.

3) Preprocessing

먼저 2048x2048x3의 input을 network에 넣기 위해서는 512x512x3으로 줄여야하고 loss 계산을 위해 각 input에서의 ground truth를 만들어 놓아야 한다.

먼저 input의 resize의 경우 Inter Area 보간법을 통해 각 변을 1/4배로 줄였다, Ground truth는 label에 명시되어 있는 좌표정보를 통해 1/8배로 줄였을 경우 그 비율에 해당하는 곳은 1(true) 그 외는 -1(false)로 만들었다.

4) Loss Function

Loss Function은 각 grid에서 ground truth와의 오차 제곱합을 이용하였다. 단, ground truth가 1인 부분 즉 object의 부분은 전체 이미지에서 매우 적으므로 단순히 오차 제곱합으로 loss function을 사용하면 학습결과 모든 grid가 -1로 나오게 된다. 그래서 grid에서 1에 해당하는 부분의 오차에는 <수식 1>에 해당하는 weight를 곱하여 loss function에 더하였다.

수식 . weight

5) Train