Semantic segmentation Paper Summary pre-report

ByeongGeun Shin[1],
[1]Pusan National University.

1 Fully Convolutional Networks for Semantic Segmentation (FCN) Paper Summary Paper Summary

1.1 Abstract

FCN은 end-to-end, pixels-to-pixels 학습이 되는 convolutional network이다. 핵심 아이디어는 임의의 크기로 입력 값을 받고, 그에 해당하는 출력값을 생성하는 'fully convolutional network' 이다. AlexNet[1], GoogLeNet[3], VGGnet[2]과 같은 classification 신경망을 사용하고, 이들을 segmentation task에 맞게 fine-tunning 한다. 그리고 나서 shallow의 정보와 deep의 정보를 결합하는 새로운 구조를 정의한다.

1.2 Introduction

Semantic segmentation은 coarse부터 fine까지 inference를 통해 모든 픽셀에 대해 예측을 생성하는 것이다. 이전까지 semantic segmentation에 사용했던 convnet은 각 픽셀에 레이블링을 했다. 하지만 이 방법은 해결해야할 단점이 있다. 슬라이딩 윈도우 방식이 계산량이 많아 비효율적이고, 픽셀간 관계 정보가 부족하다. FCN(fully convolutional network)는 end-to-end, pixels-to-pixels로 학습한다. 신경망에 있는 upsampling layer는 subsampled pooling과 함께, pixelwise 예측을 가능하게 한다. 이를 통해 FCN은 지역 정보뿐 아니라 전역 정보를 활용하는 방식으로 기존 방법보다 성능을 높일 수 있음을 보인다.

1.3 Fully convolutional networks

FCN은 임의의 크기를 가진 입력값을 취하고, 그에 해당하는 크기의 출력값을 생성한다. classification 신경망을 coarse output을 생성하는 fully convolutional 신경망으로 어떻게 전환하는지 설명한다. pixelwise prediction을 위해, 이 coarse output을 pixel로 연결해야 한다. 마지막으로 upsampling을 위한 deconvolution layer를 소개한다.

1.3.1 Adapting classifiers for dense prediction

LeNet, AlexNet과 같은 recognition 신경망은 fixed-sized 입력값을 받아서 공간정보가 없는 출력값을 생성한다. 이러한 신경망의 fully connected layers는 공간 정보를 제거한다. 하지만 이 fully connected layers는 전체 입력 값을 다루는 kernel을 가진 convolution으로 볼 수 있다. 이것을 fully convolutional 신경망으로 전환하면, classification 신경망은 임의의 크기의 입력값을 받아서 heat map을 생성할 수 있게 된다.

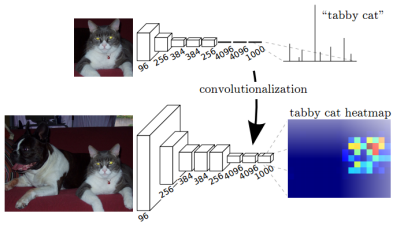


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

Figure 1: Adapting classifiers for dense prediction process.

1.3.2 Shift-and-stich is filter rarefaction

입력값이 conv + pooling을 통과하면 크기가 감소한다. 이를 복원하는 방법으로 shift-and-stich 방법을 검토하는데, upsampling이 더 효과적으로 판단하여 shift-and-stich 방방법은 사용하지 않는다.

1.3.3 Upsampling is backwards strided convolution

couarse output을 dense pixel로 연결하는 또다른 방법은 보간법(interpolation) 이다. upsampling을 위해 bilinear interpolation을 사용한다.

1.3.4 Patchwise training is loss sampling

patchwise training과 fully convolutional training을 비교해서 설명한다. patchwise training은 전체 이미지에서 crop된 부분을 이용하는 것이다. 반면에 fully convolutional training은 전체 이미지를 입력으로 받아 학습하는 것이다. pathchwise training을 사용하면, cass imbalance 문제가 발생하고, fully convolutional training이 속도와 효율성 면에서 더 좋다고 한다.

1.4 Segmentation Architecture

classification 신경망을 FCN으로 변경하고, upsampling과 pixelwise loss를 위해 구조를 수정한다. 그리고 prediction을 개선하기 위해 coarse, semantic, local, appearance 정보를 결합하는 skip architecture을 제안한다. per-pixel multinimial logistic loss로 학습하고, mean pixel intersection over union의 표준 metric으로 평가한다. 학습은 ground truth에서 벗어나는 pixel을 무시한다.

1.4.1 From classifier to dense FCN

backbone은 VGG-16을 사용한다. 그리고, 마지막 classifier layer를 버리고, 이것을 fully convolution으로 변경한다. 원래 이미지 크기로 맞춰주기 위해 coarse output에 upsampling을 수행하는 deconvolution layer 이후의 coarse output locater에 21 차원을 가진 1x1 convolution을 추가한다. 21 차원은 배경을 포함한 PASCAL classes를 예측한다.

1.4.2 Combining what and where

segmentation을 위한 새로운 fully convolutional network를 정의한다. layer를 결합하고 출력값의 공간적인 정보를 개선한다.

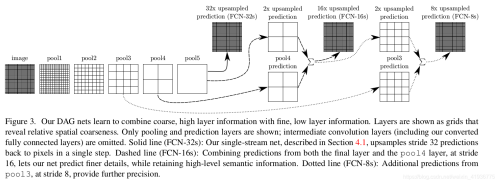


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Layers are shown as grids that reveal relative spatial coarseness. Only pooling and prediction layers are shown; intermediate convolution layers (including our converted fully connected layers) are omitted. Solid line (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Dashed line (FCN-16s): Combining predictions from both the final layer and the pool4 layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Dotted line (FCN-8s): Additional predictions from pool3, at stride 8, provide further precision.

Figure 2: segmentation FCN.

fully convolutionalized classifiers는 segmentation을 위해 fintuning할 수 있다. 높은 standard metric을 얻더라도, 이 출력값은 coarse를 만족하지 않는다. FCN-32s를 보면, 픽셀들이 뭉쳐져 있는 것을 확인할 수 있다. 마지막 prediction 레이어에서 32 pixel stride은 upsampled output에서의 scale을 제한한다.

이를 해결하기 위해서, final prediction layer를 lower layer와 결합하는 link를 추가한다. lower layer를 higher layer와 연결한다. fine layer를 coarse layer와 연결하는 것은 model이 local predictions을 예측하게 한다.

16 pixel stride layer로부터 예측함으로써, output stride를 반으로 분할한다. 그리고 추가적인 class prediction을 생성하기 위해 pool4의 위에 1x1 convolution layer를 추가한다. 이 출력값을 stride32인 conv7에서 계산된 예측값을 x2 upsampling 한뒤에 더한다. 마지막으로 입력 이미지의 크기로 upsampling 한다. 이것이 FCN-16s의 과정이다. 다시 이것을 x2 upsample 한뒤에 pool3의 출력값과 더해주면 FCN-8s가 된다.

1.5 Experiments & Conclusion

Pascal VOC, NYUDv2, SIFT Flow 데이터셋을 통해 FCN의 성능을 평가하며, FCN이 픽셀 단위의 정확도에서 기존 모델보다 우수함을 보였다. 특히 FCN은 지역적으로 섬세한 부분도 잘 인식하여 높은 정확도를 달성했고, 실험을 통해 FCN의 효과적인 학습과 예측 성능을 입증했다. 다양한 해상도에서의 성능을 평가하여 FCN의 일반화 성능도 검증하였다.

Table 2. Comparison of skip FCNs on a subset of PASCAL VOC2011 validation[7]. Learning is end-to-end, except for FCN-32s-fixed, where only the last layer is fine-tuned. Note that FCN-32s is FCN-VGG16, renamed to highlight stride.

	pixel acc.	mean acc.	mean IU	f.w. IU
FCN-32s-fixed	83.0	59.7	45.4	72.0
FCN-32s	89.1	73.3	59.4	81.4
FCN-16s	90.0	75.7	62.4	83.0
FCN-8s	**90.3**	**75.9**	**62.7**	**83.2**

Figure 3: FCNs PASCALVOC2011 Validation.

FCN은 완전한 컨볼루션 구조를 이용해 이미지 분할 문제를 해결하는 새로운 방식의 모델을 제안했고, 실험 결과를 통해 FCN의 효율성과 정확성이 입증되었다.

2 Learning Deconvolution Network for Semantic Segmentation Paper Summary

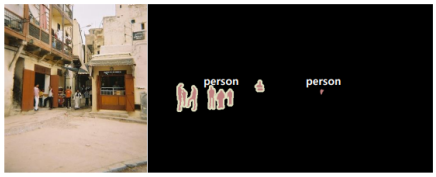
2.1 Abstract

DeconvNet은 2015년도 CVPR에 소개된 논문으로 FCN의 한계를 극복한 논문이다. FCN의 경우에는 큰 Object와 작은 Object를 구분을 못하는 문제가 있었다. 이를 극복하기 위해 논문에선 Layer를 더 깊게 쌓음으로서 Deconvolution 구조를 복잡하게 사용한다. 또한 unpooling layers에서 Maxpooling 과 Transposed Convoltuion을 같이 도입하면서, Maxpooling은 디테일한 부분을 Transposed Convolution은 내부의 값을 채우는 작용을 한다. 두가지의 Contribution을 통해서, 성능적인 측면에서도 SOTA를 달성하고 특히 작은 Object와 큰 Object도 잘 맞추는 효과가 생겼다.

2.2 FCNs Limitation

기존의 FCN 네트워크는 기존에 정의된 고정된 Receptive field를 가진다. 그렇기에 object의 크기가 receptive field 대비 크거나 작은 경우에 대해서 잘 못맞추는 경향을 보이는 문제가 있다. 이는 고정된 Receptive Field도 있지만, Maxpooling과 Convolution에 의한 문제도 있다. Maxpooling의 5개 연산과 Convolution으로 크기가 줄어들고 그 과정에서 정보의 손실이 발생하기에 크기가 작은 사람의 경우는 정보의 손실이 많이 발생한다. 이를 해결해주기위해서 FCN에서는 Skip Connection을 적용해서 FCN16s, 8s을 사용해서 Maxpooling에 의해서 정보가 손실되기전의 특징맵을 활용하는 모습을 보였지만, 디테일한 모습보다 포괄적인 모습으로 나오게 되었다.

(a) Inconsistent labels due to large object size



(b) Missing labels due to small object size

Figure 4: Fully Convolutional Networks의 한계.

2.3 Deconvolutional Network

2.3.1 Architecture

DeconvNet의 구조는 크게 Convolution network 부분가 Deconvolution network 부분으로 나뉘고, 둘이 대칭인 모습을 보이고 있다. 각 부분의 특징을 생각해보면 Convolution은 input image의 feature를 추출해서 multidimensional feature representation으로 변환하는 역할을 수행한다. 이때, 위의 구성은 VGG 16-layer net에서 마지막 classification layer를 제거한 형태이다.

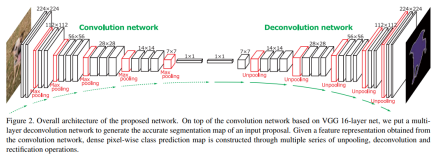


Figure 2. Overall architecture of the proposed network. On top of the convolution network based on VGG 16-layer net, we put a multi-layer deconvolution network to generate the accurate segmentation map of an input proposal. Given a feature representation obtained from the convolution network, dense pixel-wise class prediction map is constructed through multiple series of unpooling, deconvolution and rectification operations.

Figure 5: Deconvolutional Network Architecture.

Encoder는 Conv과 Max Pooling 연산을 수행하며 하나의 Conv은 Convolution →BatchNorm → ReLU 구조를 가진다. Decoder의 Deconv과 Un Pooling 연산을 수행하며 하나의 Deconv은 Transposed Convolution → BatchNorm → ReLU 구조를 가지고 있다.

2.3.2 Unpooling

Convolution에서 Pooling은 대표값을 추출해서 noisy activation을 걸러주는 역할을 하지만, spatial information을 잃어버리는 문제점을 가지고 있다. 이러한 문제를 해결하기 위해서, Pooling 시의 활성화된 위치를 기억하고 해당 값을 복원하는 방법을 사용하였다.

2.3.3 Deconvolution

Unpooling layer의 output은 크지만, sparse activation map 이라는 문제점이 있다.(대부분의 값이 0으로 비활성화 되어있음). deconvolution layer는 이러한 sparse를 dense activation map 으로 조밀하게 만드는 특징을 가진다. 이러한 deconvolution layers는 unpooling layer의 size를 유지하므로, input object의 모양을 재구성한다. convolution layer와 유사하게 deconvolution layer의 hierarchical structure는 different level of shape details를 포착하기 위해 사용된다. lower layer의 필터는 전반적인 모양을 잡고, higher layer는 디테일한 모양을 잡는 역할이다.

2.4 Training

2.4.1 Batch Normalization

DNN은 Internal-covariate-shift 때문에 최적화하기 어렵고, 이를 해결하기 위해서 batch normalization을 적용한다.모든 layer는 표준 정규분포을 통해서 정규화된다.

2.4.2 Two-stage Training

normalization이 local optimal를 탈출하는데 도움을 주지만, semantic segmentation의 공간은 학습 데이터의 수에 비해서는 크고 instance segmentation을 수행하는 deconvolution의 장점은 사라진다. 이를 위해 1 stage의 쉬운 예제(object proposals 알고리즘으로 객체가 있을많나 영역을 자르고, 실제 정답 object를 crop하여 이를 중앙으로 하는 bounding box)를 학습하고 2 stage는 1 stage에서 잘라낸 이미지들 중 실제 정답을 crop하기 전에 실제 정답과 잘 겹치는 것들을 활용하여 2차 학습을 진행한다.

2.5 Inference

DeconvNet은 개별의 instance에 대해서 semantic segmentation을 수행한다. 먼저, 개별 instance를 생성하기 위해 input image를 window sliding을 통해서 충분한 수의 candidate proposals를 만든다. 이후, 이에 대해 semantic segmentation을 수행하고 proposals에 대해서 나온 모든 결과를 aggregate해서 전체 이미지에 대한 결과를 생성한다. 추가적으로, FCN과 앙상블시에 성능이 향상된다.

2.5.1 Aggregating Instance-wise Segmentation Map

몇몇 Proposals는 부정확한 예측을 가지고 있기에, aggregation동안에 suppress 한다. Pixel-wise Maximum or average를 통해서 충분히 robust한 결과를 만든다. 이후, output map에 fully-connected CRF를 적용한다.

2.5.2 Ensemble with FCN

DeconvNet은 fine-details를 잘 잡는 반면에, FCN은 overall shape를 추출하는데 강점을 가지고 있다.instance wise prediction은 object의 various scales을 다루고, FCN은 coarse scale에서의 context를 잡는데 강점이 있다. 둘을 독립적으로 시행후에 Ensemble 하고, CRF를 적용하면 가장 좋은 결과가 나온다.

2.6 Experiments

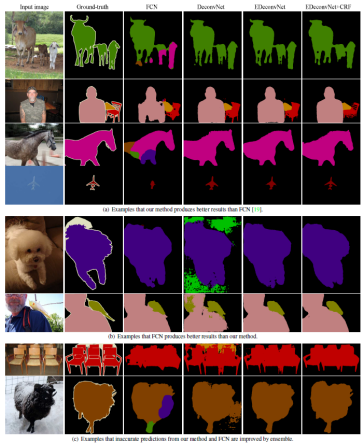
PASCAL VOC 2012 segmentation dataset과 Microsoft COCO 으로 실험한 결과 아래와 같이 좋은 결과를 얻었다.

Table 1. Evaluation results on PASCAL VOC 2012 test set. (Asterisk (*) denotes the algorithms that also use Microsoft COCO for training.)

Method	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mean
Hypercolumn [11]	88.9	68.4	27.2	68.2	47.6	61.7	76.9	72.1	71.1	24.3	59.3	44.8	62.7	59.4	73.5	70.6	52.0	63.0	38.1	60.0	54.1	59.2
MSRA-CFM [3]	87.7	75.7	26.7	69.5	48.8	65.6	81.0	69.2	73.3	30.0	68.7	51.5	69.1	68.1	71.7	67.5	50.4	66.5	44.4	58.9	53.5	61.8
FCN8s [19]	91.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
TTI-Zoomout-16 [20]	89.8	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3	64.4
DeepLab-CRF [1]	93.1	84.4	**54.5**	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	**59.7**	82.2	50.4	73.1	63.7	71.6
DeconvNet	92.7	85.9	42.6	78.9	62.5	66.6	87.4	77.8	79.5	26.3	73.4	60.2	70.8	76.5	79.6	77.7	58.2	77.4	52.9	75.2	59.8	69.6
DeconvNet+CRF	92.9	87.8	41.9	80.6	63.9	67.3	**88.1**	78.4	81.3	25.9	73.7	61.2	72.0	77.0	79.9	78.7	59.5	78.3	**55.0**	75.2	61.5	70.5
EDeconvNet	92.9	88.4	39.7	79.0	63.0	67.7	87.1	**81.5**	84.4	27.8	76.1	61.2	78.0	79.3	83.1	79.3	58.0	82.5	52.3	80.1	64.0	71.7
EDeconvNet+CRF	93.1	**89.9**	39.3	79.7	63.9	**68.2**	87.4	81.2	**86.1**	28.5	**77.0**	62.0	79.0	**80.3**	**83.6**	80.2	58.8	**83.4**	54.3	**80.7**	65.0	**72.5**
* WSSL [21]	93.2	85.3	36.2	**84.8**	61.2	67.5	84.7	81.4	81.0	**30.8**	73.8	53.8	77.5	76.5	82.3	**81.6**	56.3	78.9	52.3	76.6	63.3	70.4
* BoxSup [2]	**93.6**	86.4	35.5	79.7	**65.2**	65.2	84.3	78.5	83.7	30.5	76.2	**62.6**	**79.3**	76.1	82.1	81.3	57.0	78.2	**55.0**	72.5	**68.1**	71.0

Figure 6: PASCAL VOC 2012 and Microsoft COCO Eval.

FCN-8s와의 앙상블 및 CRF를 적용한 모델(EDeconvNet + CRF)이 가장 높은 성능을 기록하였다.



(a) Examples that our method produces better results than FCN [19].

(b) Examples that FCN produces better results than our method.

(c) Examples that inaccurate predictions from our method and FCN are improved by ensemble.

Figure 7: DeconvNet, FCN, EDeconvNet, EDeconvNet + CRF Diff.

DeconvNet은 FCN보다 fine segmentation을 생성할 수 있고 multi-scale object를 다룰 수 있지만, 가끔 noisy한 결과를 보인다.

3 Semantic segmentation Practice

3.1 Intro

Semantic segmentation은 이미지의 각 픽셀을 사전에 정의된 카테고리로 분류하는 컴퓨터 비전 기술이다. 이는 이미지의 전체를 하나의 레이블로 분류하는 이미지 분류(image classification)나 객체의 경계와 위치를 파악하는 객체 탐지(object detection)와는 달리, 이미지에 나타난 객체와 배경을 픽셀 단위로 이해하도록 돕는다. 본 보고서는 Fully Convolutional Network (FCN)와 한계를 극복하는 Deconvolutional Network 논문을 활용한 semantic segmentation 구현 과정을 설명하고, 이를 평가한 결과를 다룬다.

3.2 Practice Method

3.2.1 Fully Convolutional Network

기존의 Convolutional Neural Network (CNN) 모델(예: AlexNet, VGGNet)은 고정된 크기의 입력을 받아 출력으로 확률 벡터를 제공한다. 이러한 모델은 완전 연결 계층(Fully Connected Layer)을 사용하며, 이 계층은 공간적 좌표 정보를 버리고 고정된 차원의 특징만을 학습한다. Semantic segmentation 문제에서는 이미지의 공간적 정보를 보존하는 것이 중요하기 때문에 FCN은 Fully Connected Layer를 Convolution Layer로 변환함으로써 공간 정보를 유지한다.

3.2.2 Upsampling

FCN은 Convolution 및 Pooling 연산으로 인해 해상도가 줄어든 feature map을 입력 크기로 복원하기 위해 업샘플링 기술을 사용한다. Bilinear Interpolation 기법은 단순한 선형 보간법으로 해상도를 복원한다. 비학습 기반 방식으로 간단하지만 최적의 성능을 제공하지는 못한다. Transposed Convolution (Deconvolution) 기법은 학습 가능한 필터를 사용하여 입력 해상도를 높이는 방법이다. 표준 Convolution이 입력을 압축하는 것과 달리, Transposed Convolution은 입력의 정보를 다수의 공간 위치로 확산시킨다. Dilated Convolution 기법은 커널 요소 간의 간격을 늘리는 'dilation rate'를 도입하여 receptive field를 확장한다. 추가적인 파라미터 없이 더 넓은 맥락 정보를 포착할 수 있다.

3.2.3 Dataset

본 실험에서는 PASCAL VOC 2012 데이터셋을 사용하였다. 실제 실습에서 사용된 데이터셋은 20개의 객체 클래스와 배경 클래스로 제한하고, 각 이미지에 대해 픽셀 단위의 라벨이 제공된다. 이 데이터셋은 Semantic segmentation 연구 및 벤치마크에 널리 사용되는 데이터셋으로, 학습 및 평가에 적합하다.

3.3 Mission Implementation

3.3.1 FCN-8s Model

FCN-8s 모델은 VGGNet 기반의 Fully Convolutional Network로, 해상도가 줄어든 feature map을 점진적으로 업샘플링하여 원본 이미지 크기로 복원한다.

3.3.2 예측 계층 (Prediction Layers)

FCN의 주요 특징은 Fully Connected Layer를 1×1 Convolution Layer로 변환하는 것이다. 이를 통해 각 위치에 대해 클래스별 점수를 예측한다. 주요 계층은 다음과 같다.

Predict 1: 1×1 Conv (in: 4096, out: n_{class})

Predict 2: 1×1 Conv (in: 512, out: n_{class}, weight: 0.01)

Predict 3: 1×1 Conv (in: 256, out: n_{class}, weight: 0.0001)

3.3.3 업샘플링 계층 (Upsampling Layers)

해상도를 복원하기 위해 Transposed Convolution을 사용한다. 이는 학습 가능한 파라미터를 통해 최적의 복원 필터를 학습한다.

Deconv 1: 4×4 Transposed Conv (in: n_{class}, out: n_{class}, stride: 2, bias: False)

Deconv 2: 4×4 Transposed Conv (in: n_{class}, out: n_{class}, stride: 2, bias: False)

Deconv 3: 16×16 Transposed Conv (in: n_{class}, out: n_{class}, stride: 8, bias: False)

3.4 Training

모델 학습에는 Cross-Entropy Loss를 사용하였다. 이 손실 함수는 픽셀 단위로 예측 클래스와 실제 라벨 간의 차이를 최소화하도록 학습을 유도한다. 학습 손실은 100번의 반복(iteration)마다 출력하였고, 손실이 점진적으로 감소하는 경향을 보여 학습이 정상적으로 진행됨을 확인하였다.

지정했던 Max Iter 20000에서의 최종 Loss는 0.527이었고, 최소 Loss는 19500 Iter에서의 0.507 이었다. 아래 그림을 보렴 오차가 수렴하는 것을 볼 수 있기에 더 이상의 훈련을 진행하진 않았다.

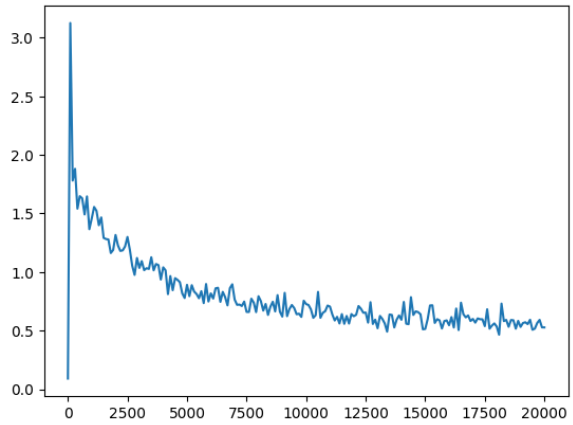


Figure 8: Training Loss.

3.5 Evaluation and Visualization

3.5.1 Evaluation

모델 성능은 1449개의 검증용 이미지에 대해 mIoU(Mean Intersection over Union) 지표를 사용해 평가하였다. 이 지표는 예측된 분류 영역과 실제 라벨 간의 겹치는 정도를 측정한다.

$$\text{IoU(Class i)} = \frac{\text{True Positive(Class i)}}{\text{True Positive(Class i)} + \text{False Positive(Class i)} + \text{False Negative(Class i)}}$$

최종 mIoU는 모든 클래스의 IoU 평균값으로 계산된다. 1449개 검증 데이터에 대해 모델의 최종 결과는 0.4321로 나타났다.

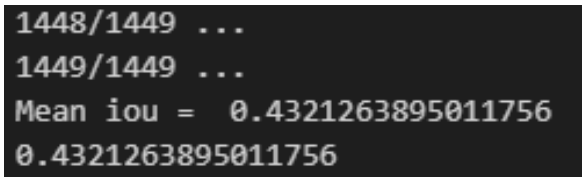


Figure 9: Validation mIoU.

3.5.2 Inference

제공된 decode label 함수를 사용하여 모델 출력 이미지를 시각화하였다. 이를 통해 각 픽셀이 올바른 클래스로 분류되었는지 시각적으로 확인하였다. 모델은 대부분의 객체 위치를 정확히 탐지하였지만, 경계 부분에서 약간의 오분류가 관찰되었다. 이는 해상도 복원 과정에서 발생하는 정보 손실 때문일 가능성이 크다.

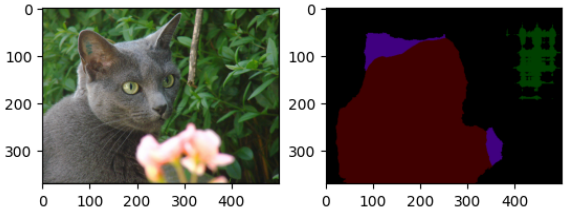


Figure 10: Test Inference.

3.6 Conclusion

본 실험에서는 FCN-8s 구조와 Deconv Upsampling을 활용한 semantic segmentation 모델을 구현하고, 이를 PASCAL VOC 2012 데이터셋으로 평가하였다. 이를 통해 FCN+Upsamling은 공간 정보를 보존하며 공간 분류 문제를 해결하는 데 효과적임을 확인하였다. 향후 연구에서는 더 정교한 업샘플링 기술을 도입하거나 더 좋은 모델이나 대규모 데이터셋을 사용한 사전 학습을 통해 성능을 향상시킬 수 있을 것이다.

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105, 2012.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.