Fully Convolutional Networks for Semantic Segmentation (FCN)

POWERPOINT TEMPLATE RELEASE

INDEX

HYUNHP

INTRODUCTION

DOWNSAMPLING

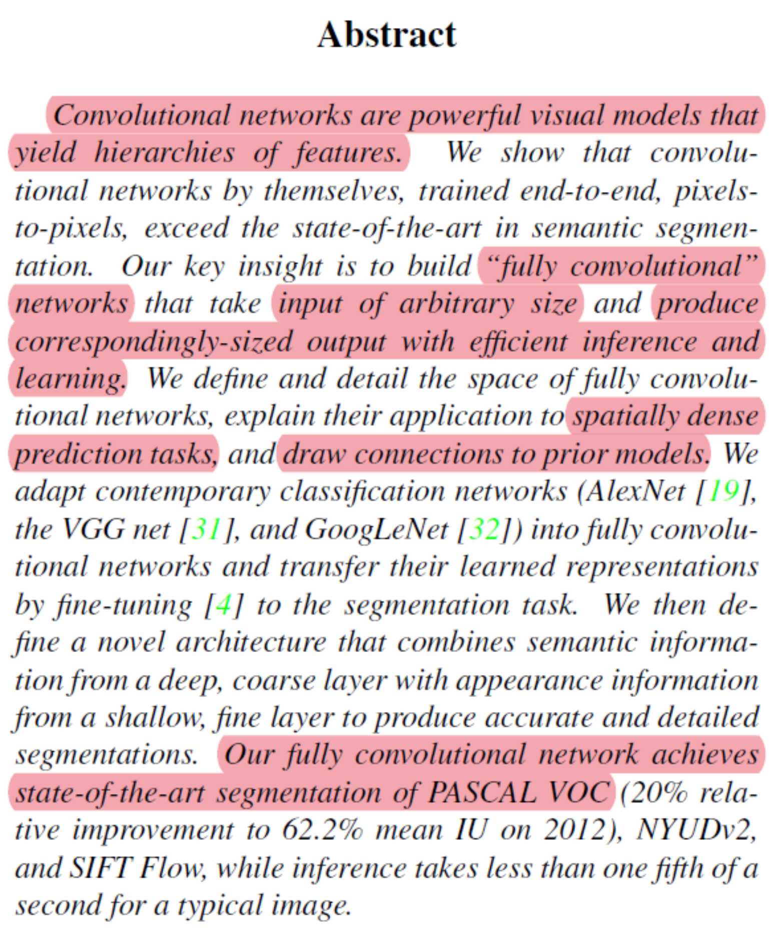
UPSAMPLING

SKIP CONNECTION

SUMMARY

INTRODUCTION

INTRODUCTION



Abstract

Convolutional networks are powerful visual models that yield hierarchies of features. We show that convolutional networks by themselves, trained end-to-end, pixels-to-pixels, exceed the state-of-the-art in semantic segmentation. Our key insight is to build "fully convolutional" networks that take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning. We define and detail the space of fully convolutional networks, explain their application to spatially dense prediction tasks, and draw connections to prior models. We adapt contemporary classification networks (AlexNet [19], the VGG net [31], and GoogLeNet [32]) into fully convolutional networks and transfer their learned representations by fine-tuning [4] to the segmentation task. We then define a novel architecture that combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed segmentations. Our fully convolutional network achieves state-of-the-art segmentation of PASCAL VOC (20% relative improvement to 62.2% mean IU on 2012), NYUDv2, and SIFT Flow, while inference takes less than one fifth of a second for a typical image.

- Start of a deep learning network for image processing began with CNN-based Alexnet outperformed in object detection competition.

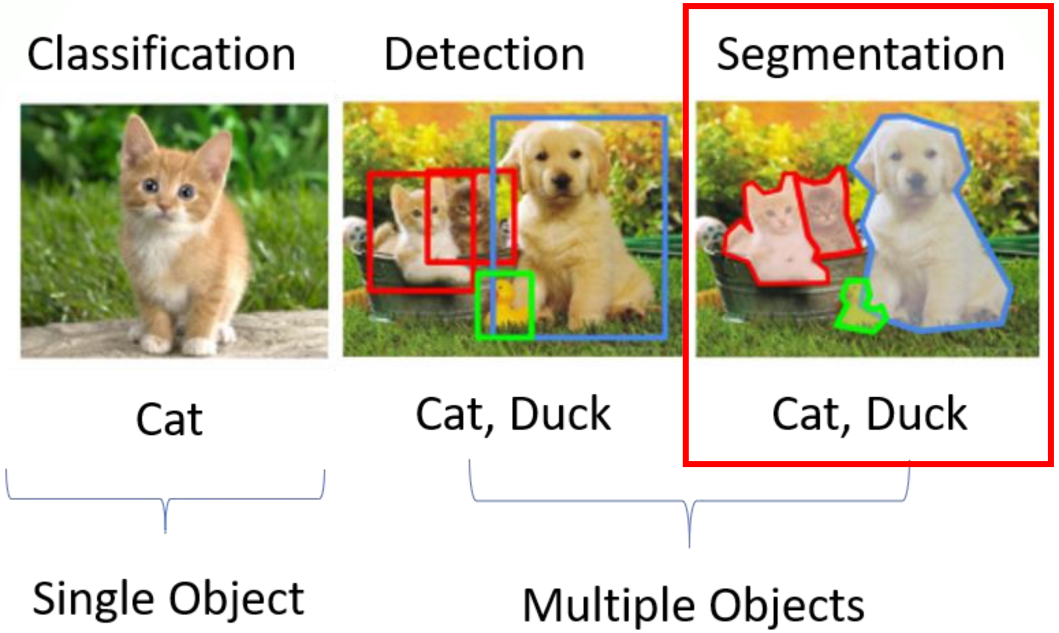
- Starting with Alexnet in 2012, various deep learning-based networks for classification began to be devised.

- With the introduction of VGG and GoogLeNet in 2014 and ResNet in 2015, there has been a breakthrough in **classification performance.**

- **However,** CNN need to have a **uniform output size**, so there is a problem that the **input size must be fixed** due to the **fully connected layer.**

- Solving this limitation, the **Full Convolution Network (FCN)** has achieved performance improvements in the **arbitral input size and segmentation fields.**

WHAT IS SEGMENTATION?



- **FCN** is a **deep learning network structure for segmentation** and is a technique for **grouping and dividing the original images into meaningful parts.**
- This is a complicated issue which **classify the entire pixel of the image** into the correct label for **pixel-by-pixel classification.**

ARCHITECTURE

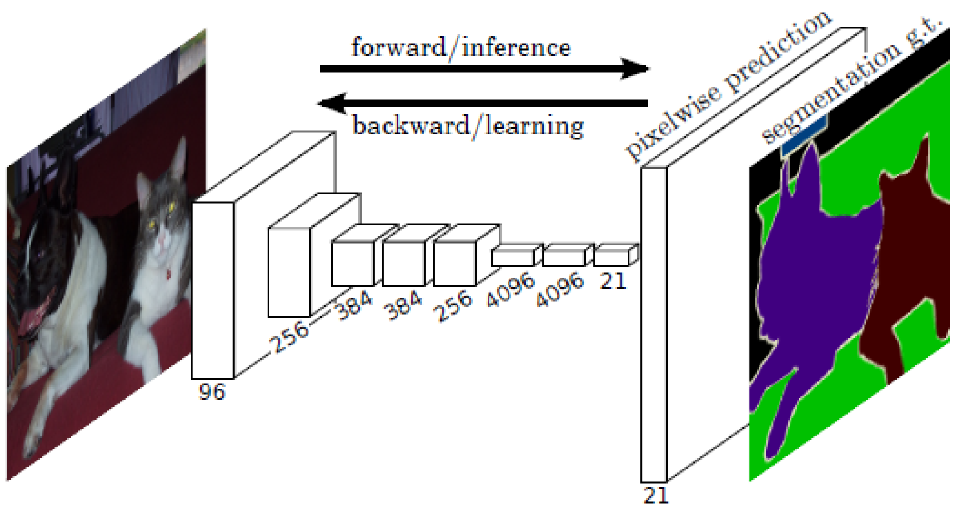


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

- FCN did not use the (fully connected) FC layer used as the classifier in classification but used the **1x1 convolution layer to maintain the location information**.

- By not using the FC layer, not only the location information is maintained, but only the convolution layer is used, so the **input size is no longer restricted (Arbitrary input size).**

- The basic concepts used in the paper are as follows.,
  - **Feature Extraction** : Consists of convolution layers that are often seen in the structure of a typical CNN.
  - **Feature level classification(Downsampling)** : Classification is performed for each pixel of the extracted feature map.
  - **Upsampling** : Increase the size of the original image through the strided transpose convolution.
  - **Segmentation** : Create a segmentation result image using the updated results of each class.

DOWNSAMPLING

Fully Connected Layer in CNN (1)

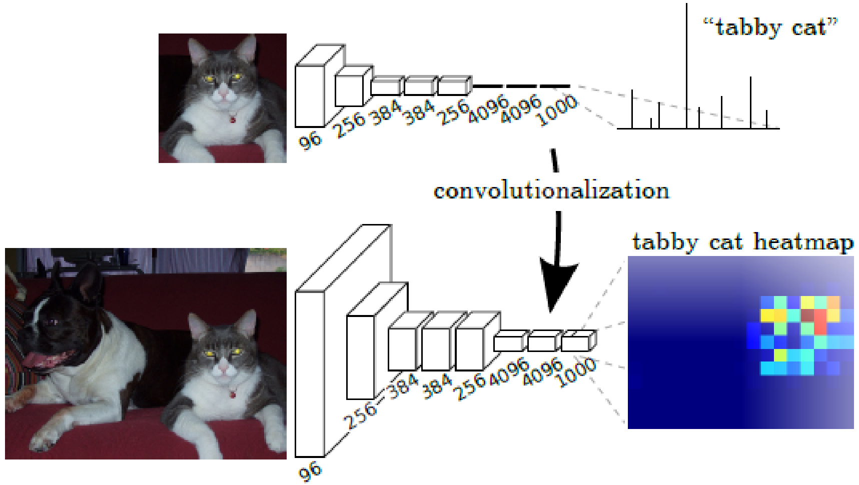


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

- When classification is performed on CNN, if the last FC layer activation function is softmax, the probability value appears and the class corresponding to the largest probability value is determined to correspond to the image.

- In a CNN classification task like this, there is **no information about the location of the object**, but only the probability value of what the object is. In other words, **lose the location of the object.**

- But if look at the input, image has **spatial information** about an object in the image, and as go down to the lower level, there are still spatial information.
- Will lose this kind of spatial information in the middle, and that's **when it becomes a fully connected layer**.

- Because in the fully connected layer, all the nodes are connected. (All the nodes are multiplied by one another.)

Fully Connected Layer in CNN (2)

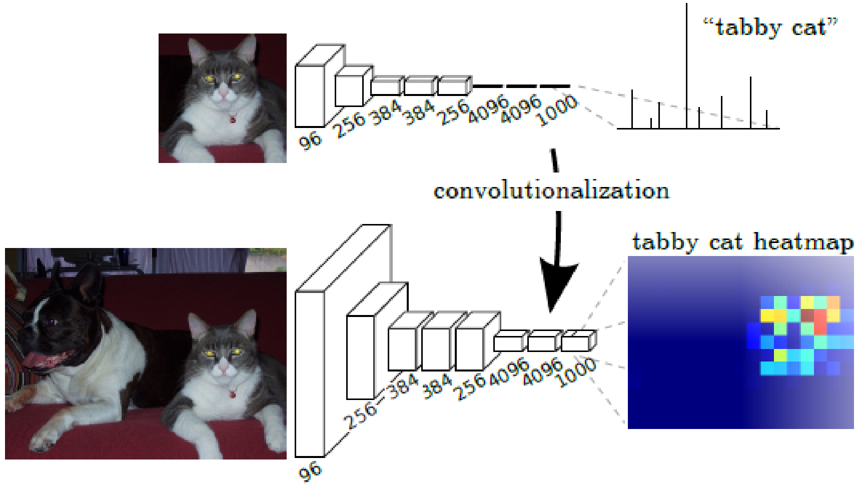


Figure 2. Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

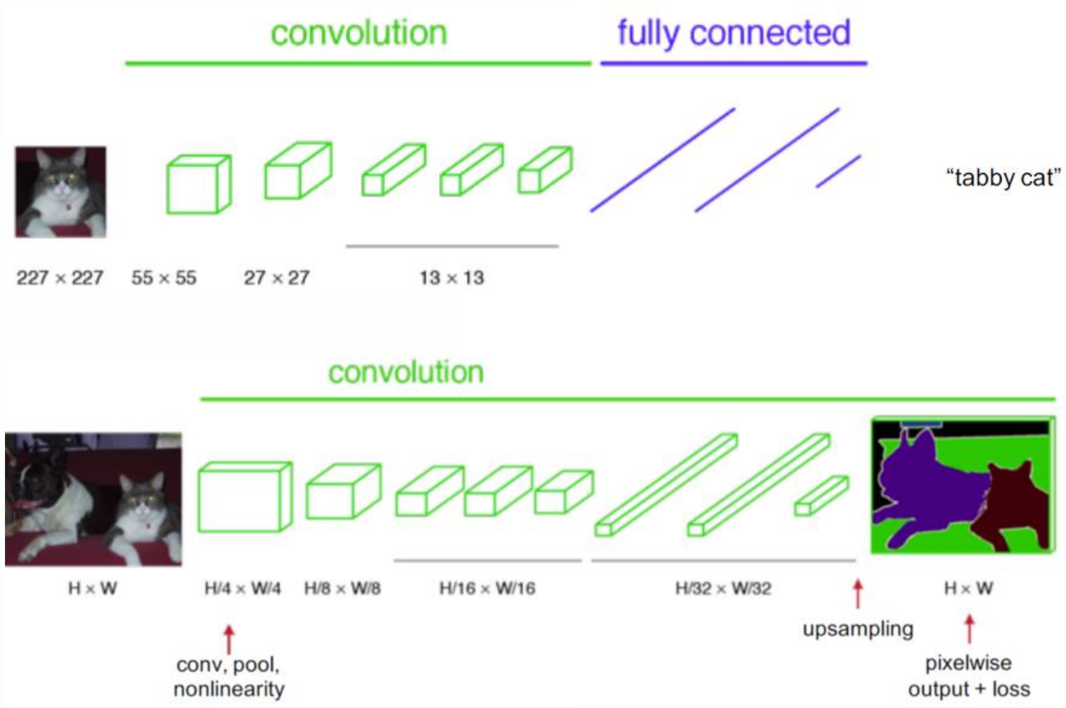
- Therefore**, network in network, 1x1 network (NIN)** is used instead of fully connected layer. **(cf. NIN : Network In Network)**

- NINs have now been widely used for efficient network design **(reducing dimensions and reducing computation)**

- As its name suggests, NIN is acting as a multi-layer perceptionron in the network. (See picture on left)

- Since spatial information must be maintained to process segmentation, if you insert the NIN instead of the fully connected layer position, you can obtain a volume-type output as shown below in the left figure. **If you draw these results in heatmap, you can see that spatial information is maintained.**

Network in Network

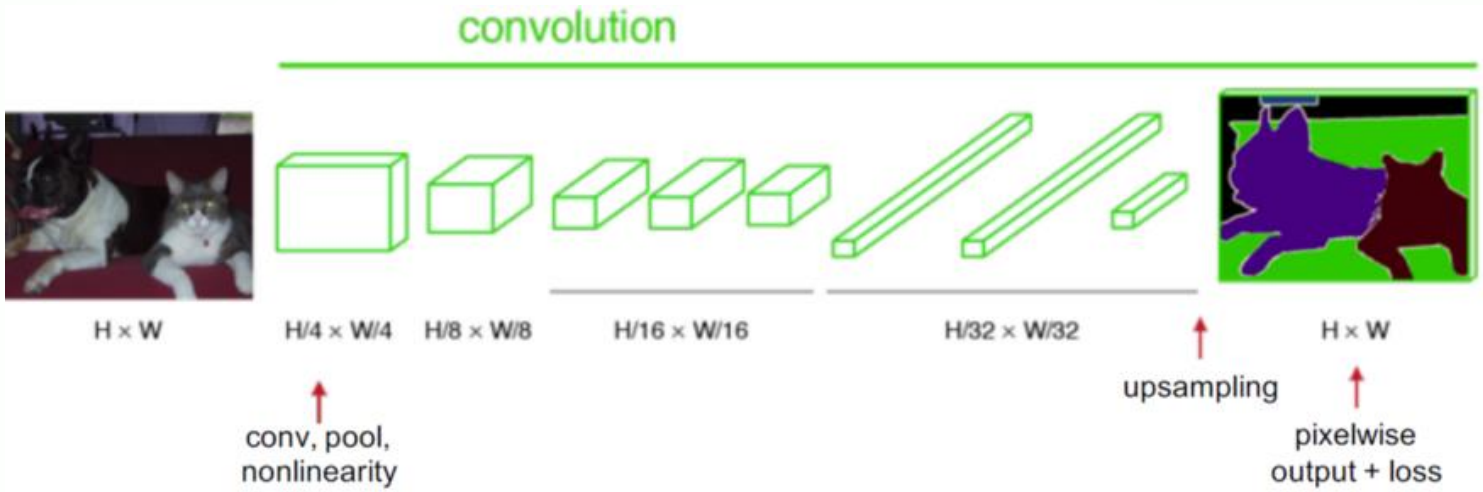


- The network below in the figure on the left is the first step in the way that FCN uses. In other words, the fully connected layer disappeared. Instead, we used **NIN** to shrink the dimension.

- **To restore this to the image size, you need to** do an **upsample** again. First, if you look at what you've done so far, it's like you've acted as an encoder to compress information.

UPSAMPLING

UPSAMPLING



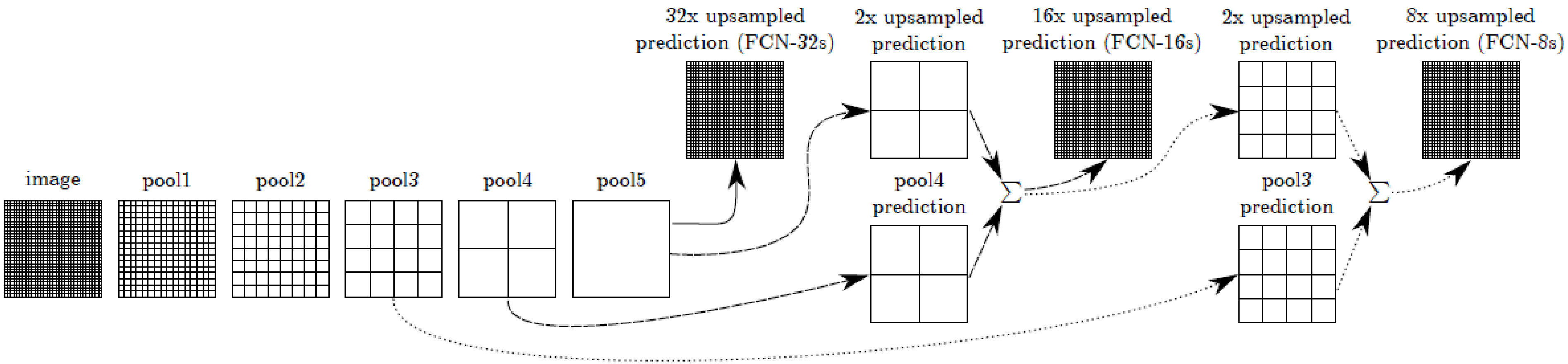
- **Upsampling is to restore the downsampled feature as large as the image size.**

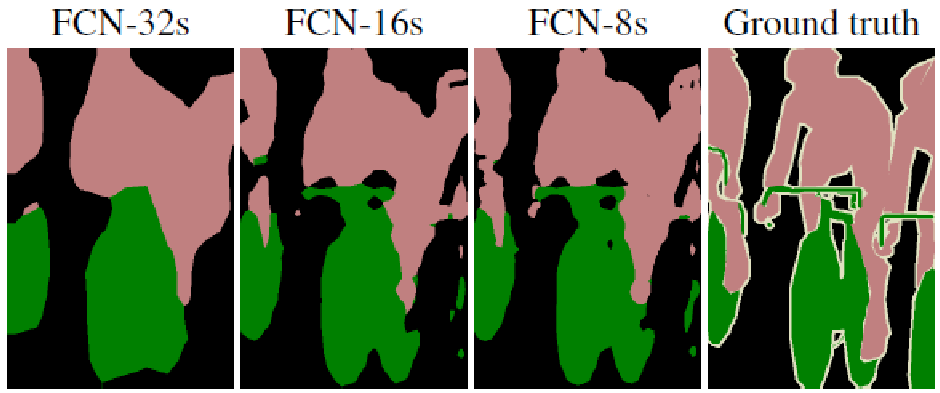
- When you want to re-size a feature, the easiest thing to think of is a method like **bilinear interpolation** (but there is a limit to performance).

- The concept is to do a **deconvolution operation** at the **decoder end**, just as we learn the parameter of the filter when compressing the feature by performing a convolution operation at the encoder end, and as a result, learn the parameter when expanding the feature again.
  - The results of the reference show that this method is much more effective and intuitively more like a deep learning network.

SKIP CONNECTION

SKIP CONNECTION (1)



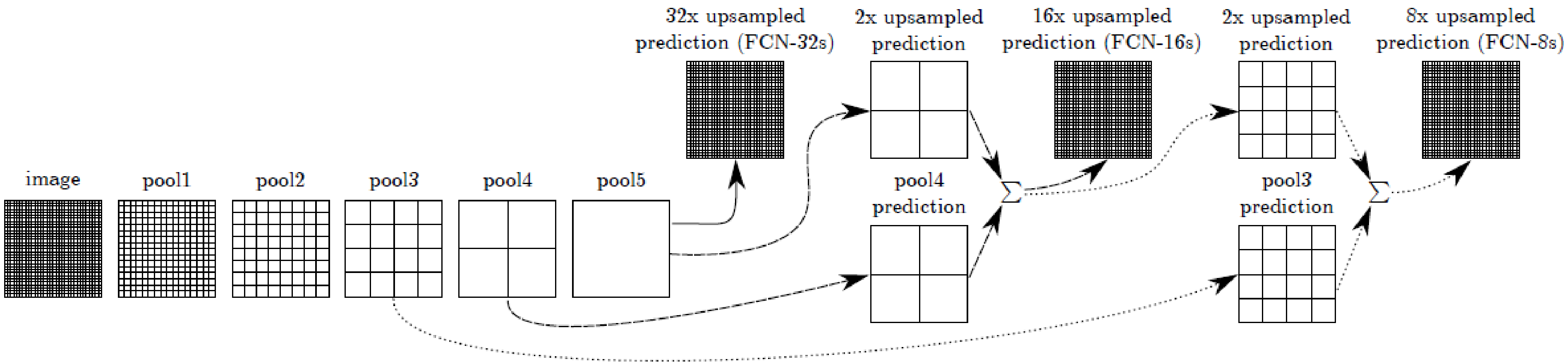


- If you look at the segmentation results on the leftmost side, it is divided by class well, but it feels like it is crushed like low resolution, and the details are incorrectly segmented.

- Therefore, improved the problem by creating a **skip connection** that delivers **high-resolution image information immediately for use in deconvolution.**

- Skip connections are created at each stage according to each symmetrical network structure, and performance is improved when multiple skip connections are inserted.

SKIP CONNECTION (2)

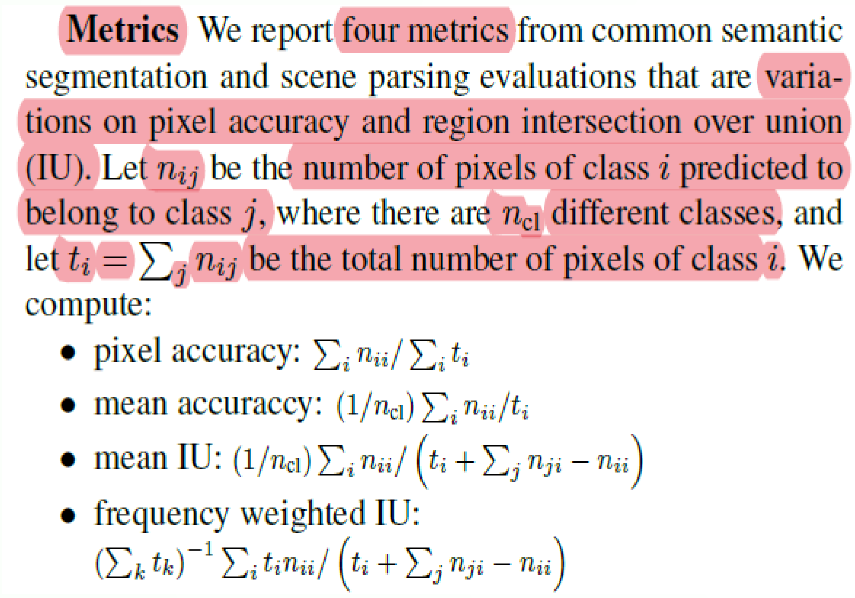


- Above figure shows **FCN-32s, FCN-16s, and FCN-8s**, which are related to which **skip connection** the FCN is connected and **how many times it has upsampled as a result.**

- First, if look **at FCN-32s**, it was upsampled **32 times without skip connection in pool5.**
- Then, if look **at FCN-16s**, you can upsample pool5 twice, and then **sum** with pool4. Upsampling it **16 times and restoring it to its original size** (so its name is FCN-16s).

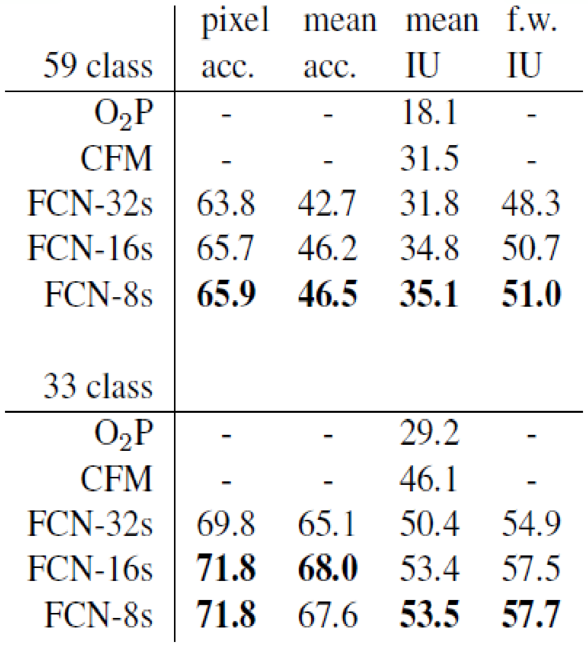
- Go one step further and upsample the sum results again in FCN-16s. And **do pool3 and sum.** It will be **upsampled 8 times and restored to its original size.**

EXPERIMENT RESULTS

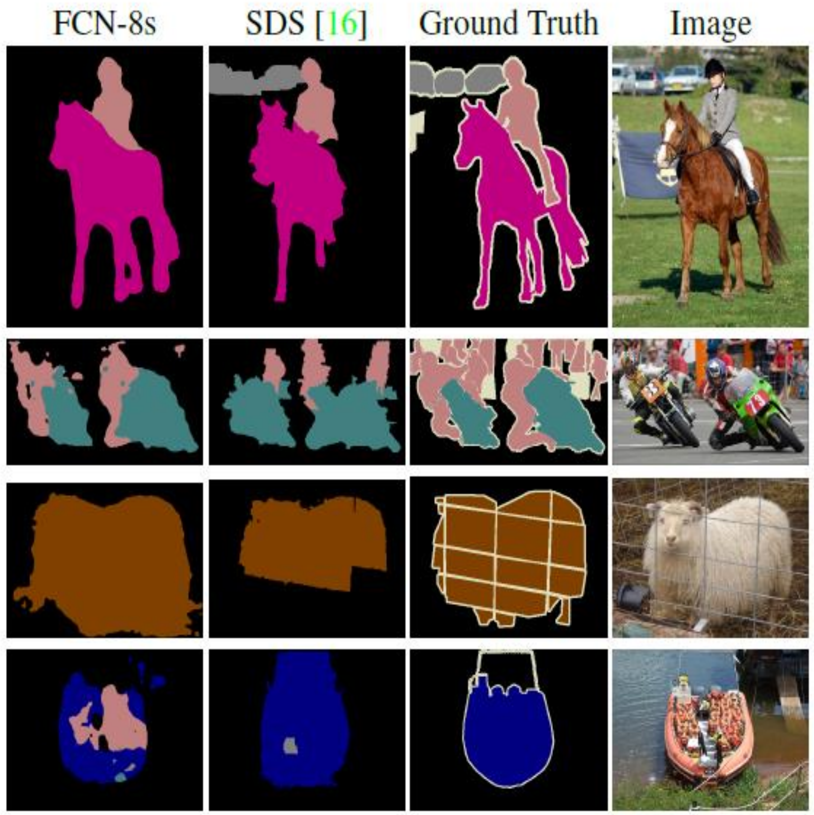


Metrics We report four metrics from common semantic segmentation and scene parsing evaluations that are variations on pixel accuracy and region intersection over union (IU). Let n_{ij} be the number of pixels of class i predicted to belong to class j, where there are n_{cl} different classes, and let $t_i = \sum_j n_{ij}$ be the total number of pixels of class i. We compute:

- pixel accuracy: $\sum_i n_{ii} / \sum_i t_i$
- mean accuraccy: $(1/n_{\text{cl}}) \sum_i n_{ii}/t_i$
- mean IU: $(1/n_{\text{cl}}) \sum_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$
- frequency weighted IU: $\left(\sum_k t_k\right)^{-1} \sum_i t_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$

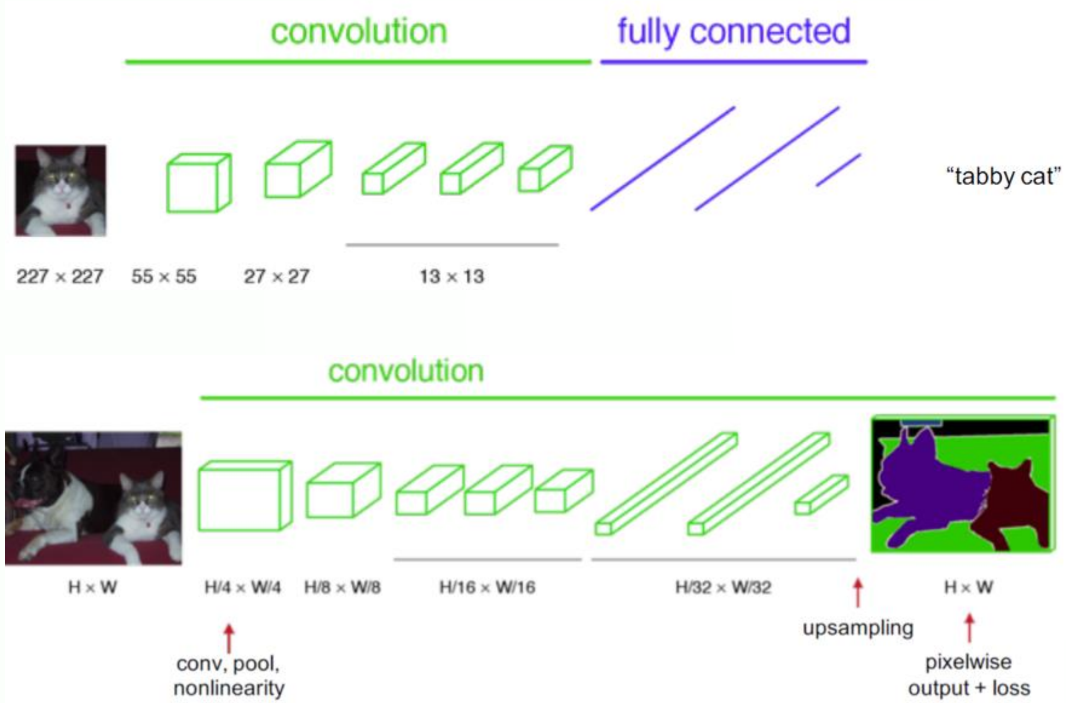


59 class	pixel acc.	mean acc.	mean IU	f.w. IU
O_2P	-	-	18.1	-
CFM	-	-	31.5	-
FCN-32s	63.8	42.7	31.8	48.3
FCN-16s	65.7	46.2	34.8	50.7
FCN-8s	**65.9**	**46.5**	**35.1**	**51.0**
33 class				
O_2P	-	-	29.2	-
CFM	-	-	46.1	-
FCN-32s	69.8	65.1	50.4	54.9
FCN-16s	**71.8**	**68.0**	53.4	57.5
FCN-8s	**71.8**	67.6	**53.5**	**57.7**



SUMMARY

SUMMARY (1)



- The characteristic of a network that has **FC layers** is that it receives only **fixed-sized inputs** due to the nature of FC layers.

- On the other hand, with a **convolution layer**, there **is no limit to the size of the image**, and **spatial information is maintained**, so the 1 pixel value of the last feature map represents 32 x 32 of the original image.

SUMMARY (2)

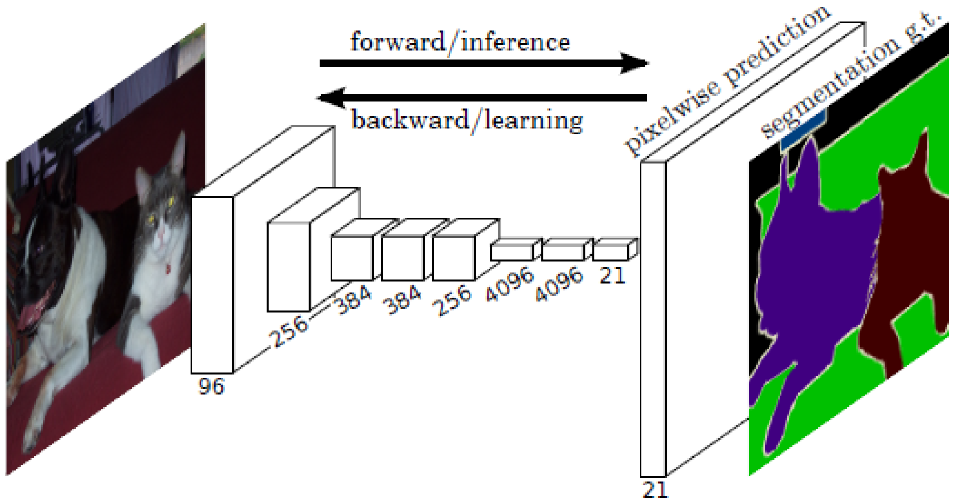
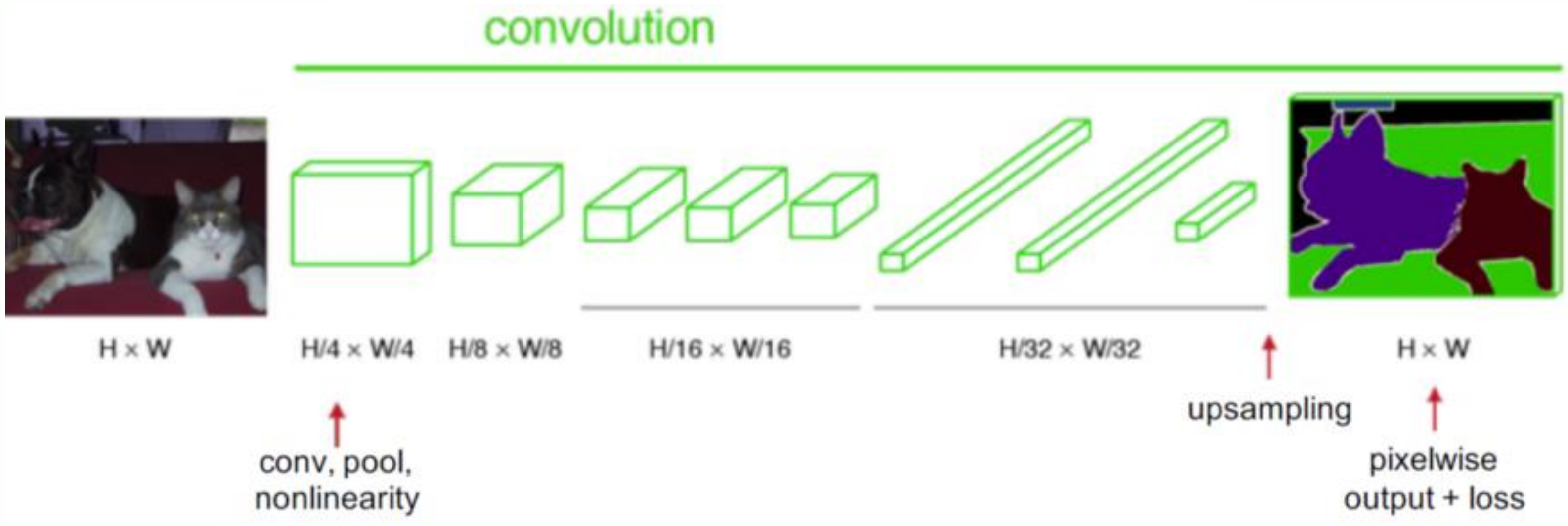


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

- After several steps of convolution layer and pooling layer, the size of the feature map decreases, but the process of re-growing the result of the feature map is necessary to make pixel-wise predictions.

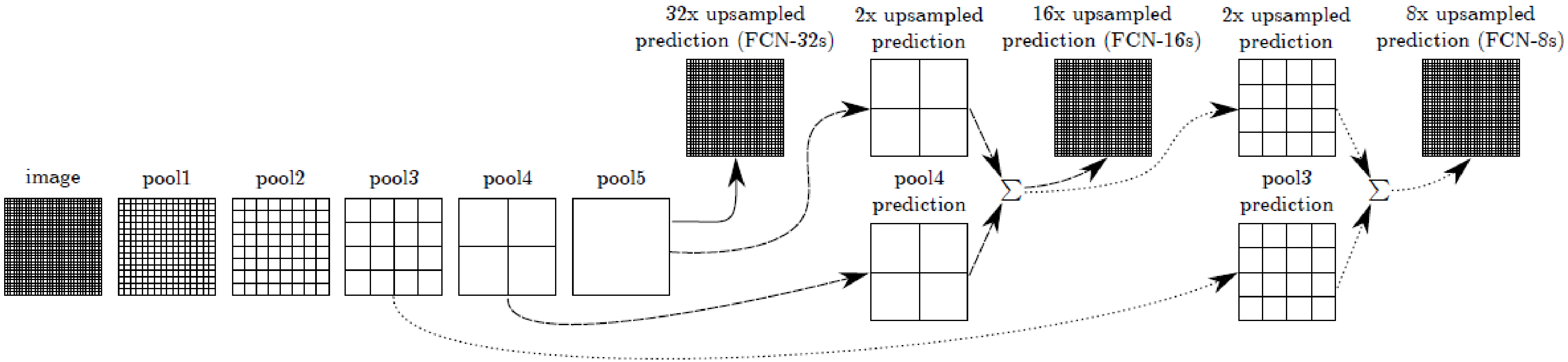
SUMMARY (3)



- Expand the value obtained as a result of the **1x1 convolution** to the size of the **original image** and use the **stride transpose convolution** to determine the parameters of the filter through learning.

- However, simply upsampling the score will limit performance and to solve this problem, paper introduced skip layers.

SUMMARY (4)



- The previous layers contain finer features than the results of the last convolution layer, so you can **get more finer (high-resolution) image information by using the features of the previous layers together.** (FCN-32s < FCN-16s < FCN-8s)

APPENDIX

FCN

HYUNHP

REFERENCE

https://gaussian37.github.io/vision-segmentation-fcn

https://medium.com/@msmapark2/

https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Long_Fully_Convolutional_Networks_2015_CVPR_paper.pdf

https://towardsdatascience.com/review-fcn-semantic-segmentation-eb8c9b50d2d1