Literature Review on *"Fully Convolutional Networks for Semantic Segmentation"* by Jonathan Long ∗ Evan Shelhamer ∗ Trevor Darrell

Simon Tian, jt886@cornell.edu, November 2nd, 2023

1. Introduction

Semantic segmentation (SS) is a labelling method specific to computer vision (CV) algorithms. Instead of identifying the object borders and features within a given image, the SS method will assign labels to every pixel that belongs to the same object category. Typically, the SS method's objective is achieved by relying on hand-made features[1] and structured prediction techniques. Nevertheless, the introduction of deep learning Convolutional[2] Neural Networks (CNNs) and Fully Convoluted Networks (FCNs) has dramatically improved the performance of the SS method. Particularly, FCNs are seen as a significant breakthrough in this area, addressing the limitations of previous architectures by enabling pixel-wise labelling in a computationally efficient manner.

"Fully Convolutional Networks for Semantic Segmentation" by Long, Shelhamer, and Darrell [1] assessed the shortcomings of the typical CNN 'classification' architecture and introduced the FCN architecture and its deconvolutional variation for performing the SS method. Finally, through extensive experiments, Long, Shelhamer and Darrel demonstrated that their FCN approach surpassed its predecessors, marking a significant advancement for the SS method.

Figure 1: Image Labeling (Left) vs. Semantic Segmentation (Right) [2]

---

[1] A feature category, eg. descriptors, are representations or a characterization of a portion of an image, typically centered around a point of interest or key point.

[2] Convolution is a mathematical operation that combines two functions to produce a third function, representing how one function modifies or is influenced by the other.

2. CNNs and Image Classification

Convolutional Neural Networks were first invented in the 1980s. It later gained popularity in the 2010s in image classification, object classification, and face recognition, as the computational capabilities of Graphical Processing Units (GPUs) dramatically increased.

The success of CNNs, particularly after AlexNet's[3] win in the ImageNet Large Scale Visual Recognition Challenge in 2012, paved the way for leveraging these architectures for other CV tasks. Models like VGG[4], GoogLeNet[5], and ResNet[6] improved accuracy on image classification tasks and provided deeper architectures that captured richer hierarchical features.

CNNs utilize a layered structure to extract significant features from images efficiently, especially for image classifications. Through multiple layers of convolution and pooling, CNNs learn representations that capture a wide range of features, from basic edges to complex objects. These learned features have outperformed traditional hand-crafted features and remained largely popular in image classification until today.

3. From Classification to Segmentation

As image classification techniques gained their presence in autonomous driving, medical imaging, augmented reality, etc., the deficiencies of such techniques began to emerge. For instance, image classification is like walking in a library and only being able to identify the genres. On the other hand, the SS method is like putting a label on each page of every book in the library, so the researcher is able to tell the theme of a particular page instantly.

The transition from classification to segmentation was not clear-cut. The challenge was to retain spatial information[7], which is typically lost in fully connected layers in the CNNs. Some earlier attempts involved patch-based classification or labelling super-pixels. However, these methods lacked the capability to capture fine details or scale invariances[8].

---

[3] A variation of CNN proposed by Alex Krizhevsky. The outperformance of AlexNet brought deep learning neural networks into popularity. [3]

[4] Visual Geometry Group, a neural network with high accuracy yet slower execution.

[5] GoogLeNet is deeper than its cohorts and more efficient to execute.

[6] ResNet exchanges information between layers, making it easier to learn identity functions.

[7] The location information of specific pixels.

[8] This means the network is insensitive to the input image size.

4. Prior Progress on Image Segmentation

Preceding the deep learning era, segmentation methods included using texture, contour, and color cues, with the advent of machine learning providing approaches such as Support Vector Machines (SVMs)[9] and decision forests. Yet, these methods were limited by the need for feature engineering and the lack of computation power for applying classifiers at a pixel/regional level. Some of the earlier segmentation attempts include:

- **Hand-crafted features**: Prior to the deep learning revolution, image segmentation relied on extracting hand-crafted features like color, texture, and shape. Techniques like Texton Forests[10], Graph Cuts[11], and Random Forests were used in combination with these features for segmentation.
- **Region-based methods**: Methods like Selective Search[12] and Regions with CNN features (R-CNN)[13] combined region proposals with CNNs for object detection, and could be adapted for segmentation.

5. From CNN to FCN Approaches

Despite CNNs demonstrating adequacies for SS methods, there was a need for architectures that were end-to-end trainable and efficiently utilized the spatial information present in an image. FCNs began to come into play and address this gap in needs.

To compare the two networks, the primary differences between CNNs and FCNs in terms of their architecture and characteristics are:

Table 1: Comparison between CNN and FCN Methods

Category	CNN	FCN
Purpose and Output	Designed for image **classification**, CNNs provide a singular output label for an entire image.	Tailored for **SS methods**, FCNs produce a dense output where each pixel is labelled according to its corresponding object.
Fully Connected Layers	Contains **fully connected layers** towards the end, which expects fixed-size inputs and flattens	The fully connected layers are replaced with **convolutional layers**. This modification retains the spatial

[9] A supervised machine learning algorithm aiming to define the optimal hyperplane separating datapoint classes.
[10] An image classification technique based on image micro-patterns or texture elements.
[11] A background separation method to see pixels as nodes, to find boundaries between foreground and background.
[12] A segmentation method to separate images into minuscule pieces, then re-combine based on visual similarity.
[13] A technique to first propose potential regions using selective search, then clarify using extracted features by CNN.

	spatial hierarchies, producing a singular output (label).	information, allowing FCNs to produce spatially dense outputs (label for each pixel indicating class information).
Input Size Flexibility	Typically requires **fixed-size input** images.	Can handle **inputs of varying sizes** due to the absence of fully connected layers.
Deconvolutional Layers[14]	Does not have deconvolutional (or transposed convolutional) layers as standard.	Introduces deconvolutional layers to upsample feature maps, enabling them to generate an output of the **same size as the input image**.
Skip Connections (in some FCN variants)	Does not typically use skip connections.	Incorporates skip connections to merge feature maps between earlier and deeper layers. This combines both border contexts and finer details, improving segmentation precision.
End-to-End Training	Typically employed for image classification, there the process of image to label is in one go.	Typically incorporated for the SS method, labelling each pixel of the image in one go without intermediate steps.
Versatility	CNNs are predominantly used for image classification.	While FCNs are primarily for SS methods, their foundational principles can be applied to tasks like object detection and instance segmentation.
Efficiency	Might require region proposals or sliding window mechanisms. The output is a single label instead of a spatial map.	Process the image in a **single forward pass** and yield a pixel-wise output, which is more computationally efficient.

Additionally, *Fig. 2* visually illustrates the difference between the outputs of CNN and FCN architecture.

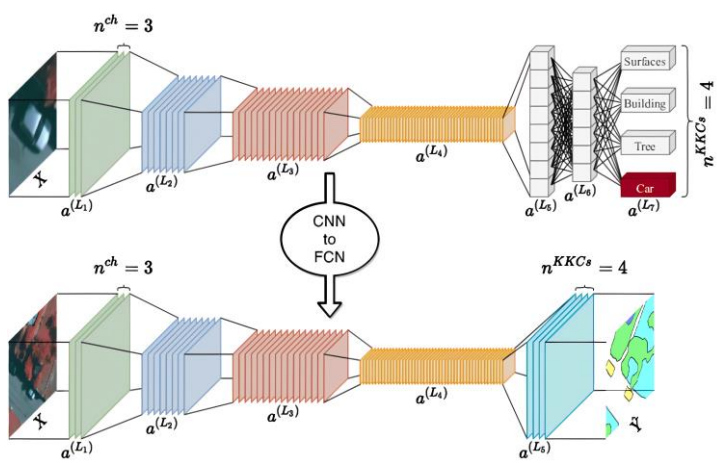

Figure 2: A comparison between CNN and FCN Architecture [4]

[14] Deconvolutional layer is a new approach introduced in this paper.

6. The Composition of FCN

To accomplish the desired SS objectives, FCNs can be mathematically described as follows:

1. **Convolutional Layer**:
   - Convolution is the fundamental operation of FCNs. The convolutional layer acts like a set of small, movable flashlights scanning across an image. As each flashlight (or filter) scans a portion of the image, it tries to identify certain features or patterns—like edges, textures, or colors. By moving across the entire image and using multiple flashlights (filters), the layer builds a map of where each feature is located. This map is called the feature map.
   - Given an input matrix (image or feature map) I and a kernel (filter) K, the convolution operation at a location $O(x, y)$ is given by the summation of element-wise multiplications between K and where K overlaps with I:

     $$O(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I(i, j) \cdot K(x - i, y - i)$$

     Where:

     $O(x, y)$: $Output\ value\ at\ location\ (x, y)\ in\ the\ output\ feature\ map.$

     $I(i, j)$: $Input\ feature\ map\ or\ image\ value\ at\ location\ (i, j).$

     K: $Kernel\ or\ filter\ used\ for\ the\ convolution.$
   - In practice, the kernel K is usually of a small size, e.g., 3×3.
2. **Fully Convolutional Adaptation**:
   - CNNs look at the entire image at once and provide a single label, like identifying the image as a "cat" or a "dog." However, FCNs need to make decisions for every pixel. Therefore, FCNs adapt fully convolutional layers to maintain the spatial layout of the image, ensuring all section of the image retains their original context and position.
   - Mathematically, a fully connected layer that would produce an output O from input I using weight W. The relationship is given by:

     $$O = W \cdot I$$

     Where:

     O: $Output\ value.$

     I: $Input\ feature\ map\ or\ vector.$

     W: $Weight\ matrix\ (interpreted\ as\ 1 \times 1\ convolution\ in\ FCN).$

- Typically, in FCNs, the convolution operation has a dimension of 1×1.

3. **Pooling and Strided Convolution**:
   - Imagine looking at an intricate painting closely. While the fine details are preserved, it is hard to see the whole picture. Pooling is like taking a few steps back to get a more general view. It shrinks and simplifies the feature map, retaining only the most important information. This increases the network efficiency and reduces overfitting.
   - Mathematically, given input *I* and pooling function *f*, the pooled output *O* at location (*x*,*y*) is given by the function:

     $$O(x, y) = f\big(I(2x : 2x + 1, 2y : 2y + 1)\big)$$[15]

     Where:

     $O(x, y)$: $Output\ value\ at\ location\ (x, y)\ in\ the\ pooled\ output.$

     I: $Input\ feature\ map.$

     f: $Pooling\ function\ (typically\ max\ or\ average).$
   - Pooling layers reduce spatial dimensions. If a 2×2 max-pooling operation is employed, then for every 2×2 block in the input, the maximum value becomes the output.
4. **Upsampling (Deconvolution)**:
   - If pooling is like zooming out of an image, then upsampling is like zooming back in. The size is often reduced for efficiency when FCNs process an image; once finished, to label every pixel in the original image, the image needs to be in its original size. Upsampling achieves this by filling in details and expanding the reduced image.
   - Deconvolutional layers (often called transposed convolutions) are used to upscale the feature maps. They can be seen as the reverse operation of convolution.
5. **Skip Connections**:
   - As an FCN processes an image, the deeper layers capture more abstract and wide-ranging features, while the earlier layers capture fine details. Skip connections combine the detailed information from earlier layers with the broader context from deeper layers. This helps in producing sharp and accurate segmentation results.

---

[15] This indicates taking a small 2x2 region from I starting at position $(2x, 2y)$ and spanning one unit in both x and y directions. The ":" operator indicates a range from say, $2x\ to\ 2x + 1$

- The combination of different layers is defined as an element-wise addition:

$$O = O_1 + O_2$$

Where:

O: $Combined\ output.$

$O_1\ and\ O_2$: $Outputs\ from\ different\ depth\ of\ layers\ of\ the\ network.$

6. **Activation Functions**:
   - After identifying features in the convolutional layers, the network needs a way to decide which features are important and which ones are not. The activation function does this by preserving all the positive values and zeroing negative values. The activation function is like a gatekeeper that only allows meaningful information.
   - A common activation function is Rectified Linear Unit (ReLU), which is defined as:

$$f(x) = \max(0, x)$$

Where:

$f(x)$: $Activated\ output\ value$

x: $Input\ value$

   - Other functions like sigmoid or tanh can also be employed.
7. **Softmax and Pixel-wise Classification**:
   - In the final layer, the FCN must decide what object each pixel belongs to, like "cat," "dog," or "background." The SoftMax function helps with this by converting raw scores for each category into probabilities. It's like taking the network's initial guesses and refining them to be more confident and accurate.
   - The conversion from scores to probabilities for each object class can be described as:

$$P(c) = \frac{e^{s_c}}{\sum_i e^{s_i}}$$

Where:

$P(c)$: $Probability\ of\ the\ pixel\ belonging\ to\ class.$

s_c : $Raw\ score\ for\ class.$

s_i : $Raw\ scores\ for\ all\ classes.$

   - This provides a probability distribution over classes for each pixel.

In essence, FCNs bridged the gap between the demand for new SS approaches, and the shortcomings of conventional CNNs. Its feature-rich output allows the computation and prediction process to be efficient, accurate, and versatile.

7. Key Contributions of the FCN Paper

The paper *"Fully Convolutional Networks for Semantic Segmentation"* by Long, Shelhamer, and Darrell made several key contributions to the field of CV, specifically in pushing the SS frontier. Here are the primary contributions:

1. **Introduction of FCNs for SS methods**:
   - Prior to this paper, SS often involved patch-based methods or region proposals. The authors introduced a novel framework by adapting CNN for the pixel-wise SS method. This adaptation involved converting the fully connected layers into convolutional layers, allowing the network to make dense predictions.
2. **End-to-End Training**:
   - FCNs are trained end-to-end, meaning the entire model is trained in a single process without the need for multiple stages or components. This ensures a direct mapping from raw pixels to their corresponding class labels.
3. **Incorporation of Deconvolutional Layers**:
   - To recover the spatial loss during pooling operations, the authors introduced upsampling layers. These layers enlarge the feature maps, allowing network outputs with the same spatial dimensions as the input image. The weights of these upsampling layers can be learned, enabling more accurate pixel-wise predictions.
4. **Skip Architectures for Combining Layers of Different Resolutions**:
   - The authors proposed a method to combine feature maps from intermediate layers with those from deeper layers. This mixing of coarse and fine information from different network depths refines the predictions, yielding in higher accuracy.
5. **Outstanding Performance on Benchmarks**:
   - The proposed FCN models achieved exceptional results on several benchmark datasets at the time, such as PASCAL VOC. This validated the effectiveness of the approach and established FCNs as a key for SS tasks, as shown in Table 2:

Table 2: FCN and Cohort Dataset Accuracy Comparison [1]

Dataset	Cohort Best Performance (% accuracy)	FCN Performance (% accuracy)
PASCAL VOC	52.6	62.7
NYUDv2	64.3	65.4
SIFT Flow	90.8	94.3

6. **Input Size Flexibility**:
    - Typical CNNs often require fixed-size inputs due to their fully connected layers. However, FCNs can handle images of varying sizes. This is ideal for many real-world applications where input images can have different dimensions.
7. **Efficiency in Prediction**:
    - FCNs process images in a single pass without the need for region proposals or sliding window techniques, making the approach computationally efficient.
8. **Generalization across Tasks**:
    - The authors demonstrated that FCNs can be fine-tuned for different tasks beyond the SS method, like object detection, highlighting the versatility and potential of the architecture.

8. Post-FCN Works:

After FCNs, the idea of using deep learning for the SS method gained popularity, and several architectures emerged to improve upon FCN's results. Here are a few notable ones:

1. **SegNet**:
    - Developed by researchers from the University of Cambridge, SegNet uses an encoder-decoder architecture similar to FCN but introduces some modifications. The encoder captures the context, while the decoder refines the segmentation using this contextual information.
    - An important feature is the use of pooling indices. During max-pooling in the encoder, the location of the maximum value within each pool is recorded. These data points are later used by the decoder to perform upsampling, preserving spatial information and reducing the number of parameters.
2. **U-Net**:
    - Originated from biomedical imaging, this architecture also uses an encoder-decoder structure and skip connections. These skip connections transfer feature maps from the encoder directly to the decoder, aiding in precise localization.
    - U-Net has been particularly successful in medical image segmentation due to its efficiency and effectiveness with small datasets.

In addition to FCNs, the use of dilated convolutions became popular as a method to maintain the total parameters while increasing the receptive field of convolutional layers.

1. **DeepLabv2**:
    - Developed by researchers at Google, DeepLabv2 introduced the use of dilated/atrous convolutions. By adjusting the dilation rate, the receptive field (input size) can be expanded without increasing the kernel size or the number of parameters.
    - This allows the model to capture larger objects or structures in the image.

Following the FCN, these works continued to push the boundaries of semantic segmentation among many others, leveraging deep learning's potential to address the challenges inherent in this task. Each new method is built upon previous knowledge, incorporating novel strategies and techniques to further improve performance and adaptability across various CV applications.

In conclusion, the publication *"Fully Convolutional Networks for Semantic Segmentation"* revolutionized semantic segmentation by introducing an efficient, end-to-end trainable architecture that leveraged the entire depth of CNNs. The principles laid out by this work continue to influence modern CV architectures and methodologies in semantic segmentation.

References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," CVF Open Access, https://openaccess.thecvf.com/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html (accessed Nov. 3, 2023).

[2] R. Tedrake, "Robotic manipulation," Ch. 9 - Object Detection and Segmentation, https://manipulation.csail.mit.edu/segmentation.html (accessed Nov. 2, 2023).

[3] pinecone.io, "Alexnet and ImageNet: The Birth of Deep Learning," Pinecone, https://www.pinecone.io/learn/series/image-search/imagenet/ (accessed Nov. 2, 2023).

[4] H. N. Oliveira, C. Silva, G. L. S. Machado, and K. Nogueira , Fully Convolutional Open Set Segmentation, https://www.researchgate.net/publication/342520093_Fully_Convolutional_Open_Set_Segmentation (accessed Nov. 3, 2023).