Learning to extract semantic structure from documents using multimodal fully convolutional neural networks
(2017 - CVPR)

A Resume

November 12, 2018

Abstract

This paper introduces an end-to-end multimodal fully convolutional neural net (MFCN) for document semantic segmentation, i.e. using a unified model capable of simultaneously identifying both *appearance-based* and *semantics-based* classes.

This is an active area of reaseach called DSSE (Document semantic structure extraction), considered as a pixel wise segmentation problem, i.e. each pixel is labeled as background, figure, table, paragraph, section heading, list, caption, etc.

Why ? motivation and hypothesis

The goal of DSSE is to understand images and documents, spliting the documents into regions of interest and recognizing the role of each one, this is done in two steps, page segmentation; distinguishing regions, and structure analysis; categorizing each region into semantically-relevent classes.

This work's contributions :
- Proposing a multimodal end-to-end network for simultanously doing both tasks, eliminating the need for designing hand-crafted features, and proposing a robust DSSE system capable of disambiguating false indentifications by laveraging the textual information and incomporating them in the CNN architecture as well.
- Support both supervised and auxiliary training for better representation learning.
- A synthetic data generation process used for creating a large-scale dataset, giving a real advantage considering that other document understanding datasets are limited have limited size and a lack if fine-grained semantic labels, to be used for training the supervised part of the MFCN model.

The methods

The model : Multimodal fully convolutional network

The proposed model (Figure 1) contains four parts :

- An encoder that learns a hierarchy of features representation,
- A decoder that outputs segmentation masks,
- An auxiliary decoder for reconstruction,
- A bridge merging visual and textual representation.

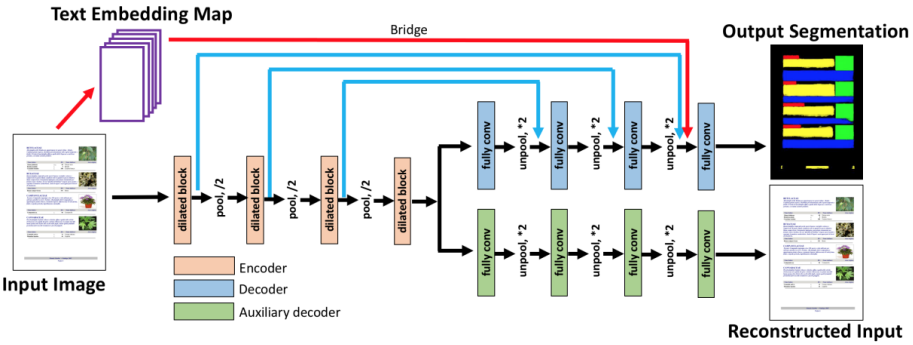

Figure 1: The MFCN model with four different parts

The model thus supports a supervised training for pixel wise segmentation and two unsupervised learning tasks to improve the learned documment representation (a reconstruction task and and a consistensy task) Details about the model :

The first part

Both the encoder and decoder parts follow the same stucture as in [1], which is as follows:

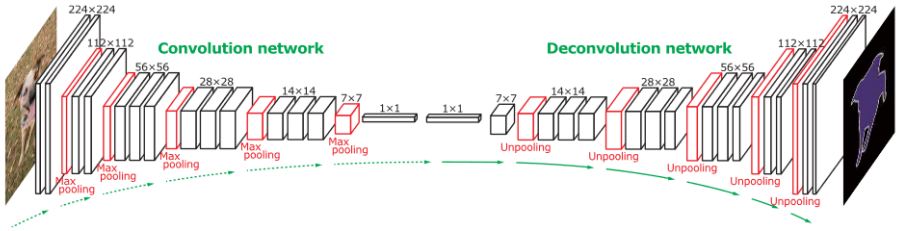

Figure 2: Deconvolutional network for semantic segmentation, based on VGG 16-layer net.

Encoder & Decoder. A fully convolutional network based on VGG-16 (FCN is a CNN without any fully connected layers, suppressing any need for having static input images i.e. only one size), followed by a deep deconvolution network, which is composed of deconvolution, unpooling, and ReLU layers. The trained network is applied to individual object proposals to obtain instance-wise segmentations, which are combined for the final semantic segmentation; it is free from scale issues found in the original FCN-based methods and identifies finer details of an object (the problem with the first versions of FCN is the coarse output images due to the use of simple upsampling methodes).

Side notes : Deconv and unpool

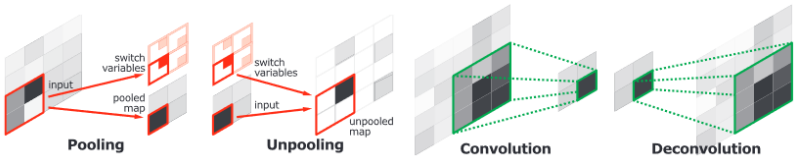

Figure 3: Unpooling and deconvolution

Deconvolution: The deconvolution layers densify the sparse activations obtained by unpooling through convolution-like operations with multiple learned filters. However, contrary to convolutional layers, which connect multiple input activations within a filter window to a single activation, deconvolutional layers associate a single input activation with multiple outputs.

Unpooling : Pooling in convolution network is designed to filter noisy activations in a lower layer by abstracting activations in a receptive field with a single representative value. Although it helps classification by retaining only robust activations in upper layers, spatial information within a receptive field is lost during

pooling, which may be critical for precise localization that is required for semantic segmentation. This issue is resolved by using unpooling layers in deconvolution network, which perform the reverse operationof pooling and reconstruct the original size of activations.

Skip connections. The usage of low level features can help in the indentification of certain regions (e.g bullets for lists), however with the downsimpling performed by the max pooling layers, the FCN performs poorly on small objects. So to use both the low level features (i.e. that are unaware of of the object high-level semantic information due to a small receptive field) and high level features (not necessarily aligned with the object bounderies due to the translation invariance of CNNs), an alternative implementation of skip connection are used (a modfied version of the skip connections used in [2]).

Side notes : Skip connection

Architectures for object instance segmentation such as DeepMask, predict masks using only upper-layer CNN features, resulting in coarse pixel masks. Common skip architectures are equivalent to making independent predictions from each layer and averaging the results, such an approach is not well suited for object instance segmentation. In [2] they propose to augment feedforward nets with a novel top-down refinement approach. The resulting bottom-up/top-down architecture is capable of efficiently generating high-fidelity object masks.

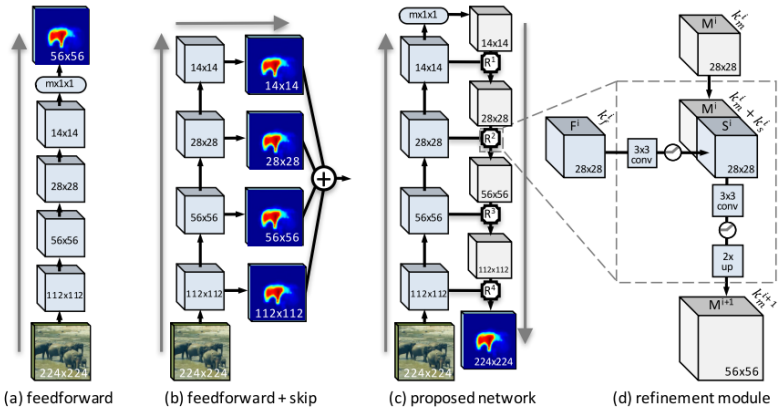

Figure 4: The skip connections used in the top down approach

This CNN architecture efficiently merges the spatially rich information from low-level features with the high-level object knowledge encoded in upper network layers. Rather than generating independent outputs from multiple network layers, first generates a coarse mask encoding in a feedforward manner, which is simply a semantically meaningful feature map with multiple channels, then refines it by successively integrating information from earlier layers by a refinement module and stacking successive modules together into a top-down refinement process (Figure 4 c and d).

In this work they use the same skip connection (blue arrows in Figure 1), but instead of using an upsampling (Figure 4 d), they use unpooling to perserve more spatial information.

Dialated networks. Giving that it is difficult to indentify some object only using some parts of them (e.g. list vs paragraphs without seeing the bulets), the receptive field must be large enough to capture the significant parts. For this, they propose dilated convolution block, each one contains five dilated convolutions layers with different dilation d (5 dilations : $d = 1, 2, 4, 8, 16$) and 3x3 kernels.

Side notes : Dilated convolutions

Let l be a dilation factor and let $*_l$ be defined as: $(F *_l k)(p) = \sum_{s+lt=p} F(s)k(t)$

The familiar discrete convolution is simply the 1-dilated convolution.

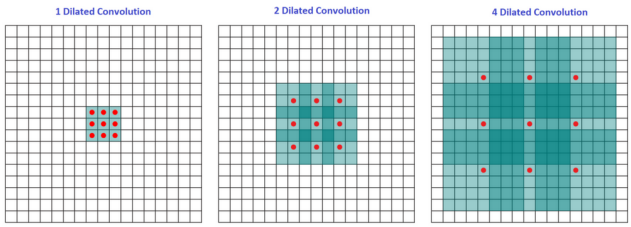

Figure 5: Three different types of dilation

The Second part, Text embedding

To have both the textual and visual representations, a text embedding map is created and then fed to our multimodal model. A sentence is the minimum unit that conveys semantic meaning and is represented as a low dimensional vector, and built by averaging the embeddings of individual words.

For each pixel inside the area of a sentence, the corresponding embedding is used as an input, pixel beloning to the same sentence share the same embedding. For a document of size $H \times W$, the result is an embedding map of size $N \times H \times W$ if the learned embedding are of N-dimensional vectors.

Side notes : Skip-grams

Given a specific word in the middle of a sentence (the input word), look at the words nearby and pick one at random. The network is going to tell us the probability for every word in our vocabulary of being the nearby word that we chose. Nearby word refers to the "window size" parameter to the algorithm. A typical window size might be 5, meaning 5 words behind and 5 words ahead (10 in total).

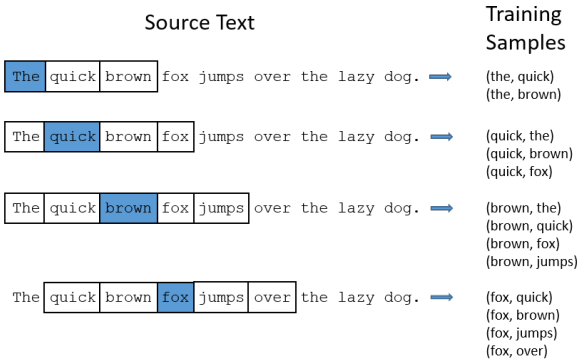

Figure 6: Skip-gram inputs

The network is going to learn the statistics from the number of times each pairing shows up. So, for example, the network is probably going to get many more training samples of (Soviet, Union) than it is of (Soviet, Sasquatch). When the training is finished, if you give it the word Soviet as input, then it will output a much higher probability for Union or Russia than it will for Sasquatch.

The embedding is learned using a skip-gram model, for V number of words, and ω a one-hot vector representing a word, the training objective is to find a N-dimentional vector (N ¡¡ V) vector representation for each word. More formally, given a sequence of words $[\omega_1, \omega_2, \omega_3, ...\omega_T]$, we maximise the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-C\leq j\leq C, j\neq 0} logP(\omega_{t+j}|\omega_t)$$

Where T is the length of the sequence and C is the size if the context window, the probability of outputting a word ω_0 given an input word ω_i is defined as softmax :

$$P(\omega_0|\omega_i) = \frac{exp(v\prime_{\omega_0}^T v_{\omega_i})}{\sum_{\omega=1}^{V} exp(v\prime_{\omega}^T v_{\omega_i})}$$

Where v_ω and $v\prime_\omega$ are the input and output N-dimentional vector representation of ω.

The third part, Unsupervised tasks

To not only use synthetic documents (that are quite limited in their layouts variations), they also define two unsupervised tasks to make use of real documents and have better representation learning.

- **Reconstructon task.** learning better representations improves performance of supervised tasks, thus we have a reconstruction loss at intermediate features as an auxilary decoder, that exists only during the trainging phase, for $a_l, l = 1, 2, ...L$ as the activations of the l^{th} layer of the encoder with a feature map of size $C_l \times H_l \times W_l$, and a_0 the input image, the auxiliary decoder D_{rec} attempts to reconstruct a hierachy of features $\{\tilde{a}_l\}$. Reconstruction loss is defined as :

$$L_{rec}^{(l)} = \frac{1}{C_l H_l W_l} \|a_l - \tilde{a}_l\|_2^2$$

- **Consistency task.** By redering the commands in the PDF files, we can obtain the bounding boxes of each region, to encourage the intra-region consistency, we define the consistency loss L_{cons}, let $p_{(i,j)}$ $(i = 1, 2, ...W)$ be the activatons at location (i, j), in a feature map of size $(C \times H \times W)$, and b the rectangular area of the bounding box of size $(H_b \times W_b)$, L_{cons} is defined as:

$$L_{cons} = \frac{1}{H_b W_b} \sum_{(i,j)\in b} \|p_{(i,j)} - p^{(b)}\|_2^2$$

$$p^{(b)} = \frac{1}{H_b W_b} \sum_{(i,j)\in b} p_{(i,j)}$$

The consistency loss is differentiable and can be optimized using stochastic gradient descent (see the paper for the differentiation)

Synthetic document generation

To adress the issue of the lack of rich datasets for document segmentation, they created a data engine capable of generating large-scale, pixel wise annotated documents.

An automated and random layout of partial data scraped from the web, in which paragraphs, figures, tables, captions, section headings, and lists are arranged to make up single, double, or triple-column PDFs, the images and their captions are taken from COCO dataset. Randomly sampled sentences from 2016 English Wikipedia dump, Section headings are the contents blocks in Wikipedia, For lists, all the items come from the same wikipedia page, and the captions are the ones associated with the COCO image or the class name found in the web image search.

The results and the implications

Implementation details: All kernels are 3x3 with a stride of 1, the pooling and unpooling are of size 2x2, a batch norm layer is used after each convolution and before each non-linear function. Per-channel mean substraction and resizing each image ¡ 384 pixels, for synthetic documents both per-pixel classification loss and unsupervised loss are used, for real documents onlu the unsupervised losses.

For text embedding each word is represented as a 128-dimensional vector, and the skip-model is trained on 2016 English wikipedia dumps, a post processing step is applied as a cleanup strategy for segment masls.

Results : The methods uses both visual and textual information for document segmentation, with an unsupervised tasks giving the posibility to used unlabeled data, both the multimodal approach and unsupervised tasks can help improved the state of the art results as shown is the paper.

References

[1] Hyeonwoo Noh, Seunghoon Hong and Bohyung Han, *Learning Deconvolution Network for Semantic Segmentation*

[2] Pedro O. Pinheiro, Ronan Collobert, Piotr Dollar, *Learning to Segment Object Candidates*