**Fully convolutional Networks**

To understand FCNs, we need to see some CNN concepts. The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers. CNNs are multilayer perceptrons which have been regularized. Usually, MLPs imply fully connected networks, meaning that each neuron in a layer is connected to every single neuron in the next layer. This brings a disadvantage to MLP though: overfitting. This is why CNNs use convoluted layers.

Fully convolutional indicates that the neural network is composed of convolutional layers without any fully-connected layers or MLP usually found at the end of the network.

* The main difference is that the fully convolutional net is learning filters everywhere. Even the decision-making layers at the end of the network are filters.
* Image sizes can also be whatever we want: in MLP, a certain number of inputs is expected, that’s why the images need to have an exact size, relative to the neuron number; but, in FCN, any size can work.
* Spatial information in images is retained more than in fully connected layers, so real world, unconstrained image segmentation has better results.
* Boost in computation efficiency: in AlexNet (the name of a CNN, designed by Alex Krizhevsky) the convolutional layers comprised of 90% of the weights (~representational capacity) but contributed only to 10% of the computation; and the remaining (10% weights => less representation power, 90% computation) was eaten up by fully connected layers.

These are some useful articles that have FCN as their main theme:

1. Fully Convolutional Networks for Semantic Segmentation[[1]](#footnote-1)
2. R-FCN: Object Detection via Region-based Fully Convolutional Networks[[2]](#footnote-2)
3. Fully Convolutional Instance-aware Semantic Segmentation[[3]](#footnote-3)
4. Automatic Pixel‐Level Crack Detection and Measurement Using Fully Convolutional Network[[4]](#footnote-4)
5. A Fast Detection Method via Region‐Based Fully Convolutional Neural Networks for Shield Tunnel Lining Defects[[5]](#footnote-5)
6. Autonomous concrete crack detection using deep fully convolutional neural network[[6]](#footnote-6)
7. Robust Pixel-Level Crack Detection Using Deep Fully Convolutional Neural Networks[[7]](#footnote-7)
8. FULLY CONVOLUTIONAL MULTI-CLASS MULTIPLE INSTANCE LEARNING[[8]](#footnote-8)
9. Fully Convolutional Network for Liver Segmentation and Lesions Detection[[9]](#footnote-9)
10. Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery[[10]](#footnote-10)
11. Efficient method for running Fully Convolutional Networks (FCNs)[[11]](#footnote-11)

Here’s some tools and resources for FCN:

* <https://github.com/veugene/fcn_maker>
* <https://github.com/shekkizh/FCN.tensorflow>
* <https://github.com/MarvinTeichmann/tensorflow-fcn>
* <https://github.com/wkentaro/pytorch-fcn>
* <https://github.com/JihongJu/keras-fcn>

1. <https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Long_Fully_Convolutional_Networks_2015_CVPR_paper.pdf> [↑](#footnote-ref-1)
2. <http://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf> [↑](#footnote-ref-2)
3. <http://openaccess.thecvf.com/content_cvpr_2017/papers/Li_Fully_Convolutional_Instance-Aware_CVPR_2017_paper.pdf> [↑](#footnote-ref-3)
4. <https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12412> [↑](#footnote-ref-4)
5. <https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12367> [↑](#footnote-ref-5)
6. <https://www.sciencedirect.com/science/article/abs/pii/S0926580518306745?via%3Dihub> [↑](#footnote-ref-6)
7. <https://ascelibrary.org/doi/full/10.1061/%28ASCE%29CP.1943-5487.0000854> [↑](#footnote-ref-7)
8. <https://arxiv.org/pdf/1412.7144.pdf> [↑](#footnote-ref-8)
9. <https://link.springer.com/chapter/10.1007/978-3-319-46976-8_9> [↑](#footnote-ref-9)
10. <https://arxiv.org/abs/1606.02585> [↑](#footnote-ref-10)
11. <https://towardsdatascience.com/efficient-method-for-running-fully-convolutional-networks-fcns-3174dc6a692b> [↑](#footnote-ref-11)