

Open in app ↗



Search

Write



# FPN(feature pyramid networks)

Getting free accuracy boost on almost any architecture



Sanchit Tanwar · Follow

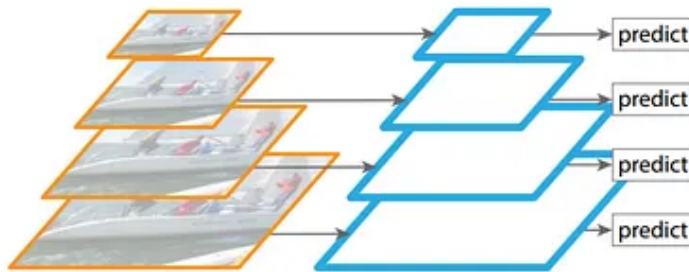
Published in Analytics Vidhya · 7 min read · Jul 10, 2020



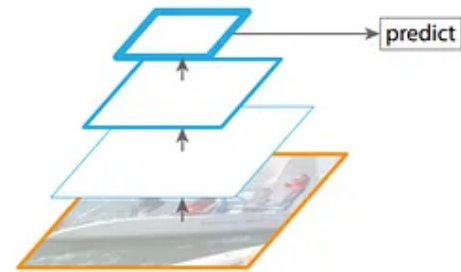
72



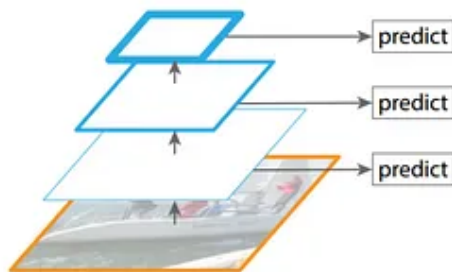
4



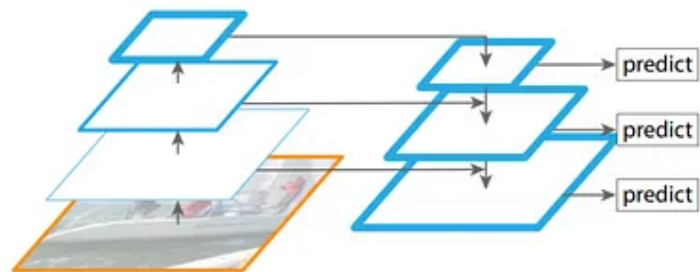
(a) Featurized image pyramid



(b) Single feature map



(c) Pyramidal feature hierarchy



(d) Feature Pyramid Network

Source: FPN Paper

I have planned to read major object detection papers (although I have skimmed through most of them, now I will be reading them in detail, good

enough to write a blog about them). The papers are related to deep learning-based object detection. Feel free to give suggestions or ask doubts will try my best to help everyone. Anyone starting with the field can skip a lot of these papers. I will also write the priority/importance of the papers once I read them all.

I have written the blog considering readers similar to me and still learning. Although I will try my best to write the crux of the paper by understanding paper in depth from various sources including blogs, codes, and videos, in case you find any error feel free to highlight it or add a comment on the blog. I have mentioned the list of papers that I will be covering at the end of the blog.

Let's get started :)

Yah subtitle is correct, FPN is a very simple method that can be used with almost any model to improve results. We will jump into technicalities of the paper soon, but for this blog, there are some pre-requisites. You should have a high-level idea about the following Fast RCNN, Faster RCNN, anchor boxes, knowledge of SSD will come in handy. I have blogs for all these papers as well you can check them(links at the end of this blog). FPN is a relatively simpler if you understand all the prerequisites well.

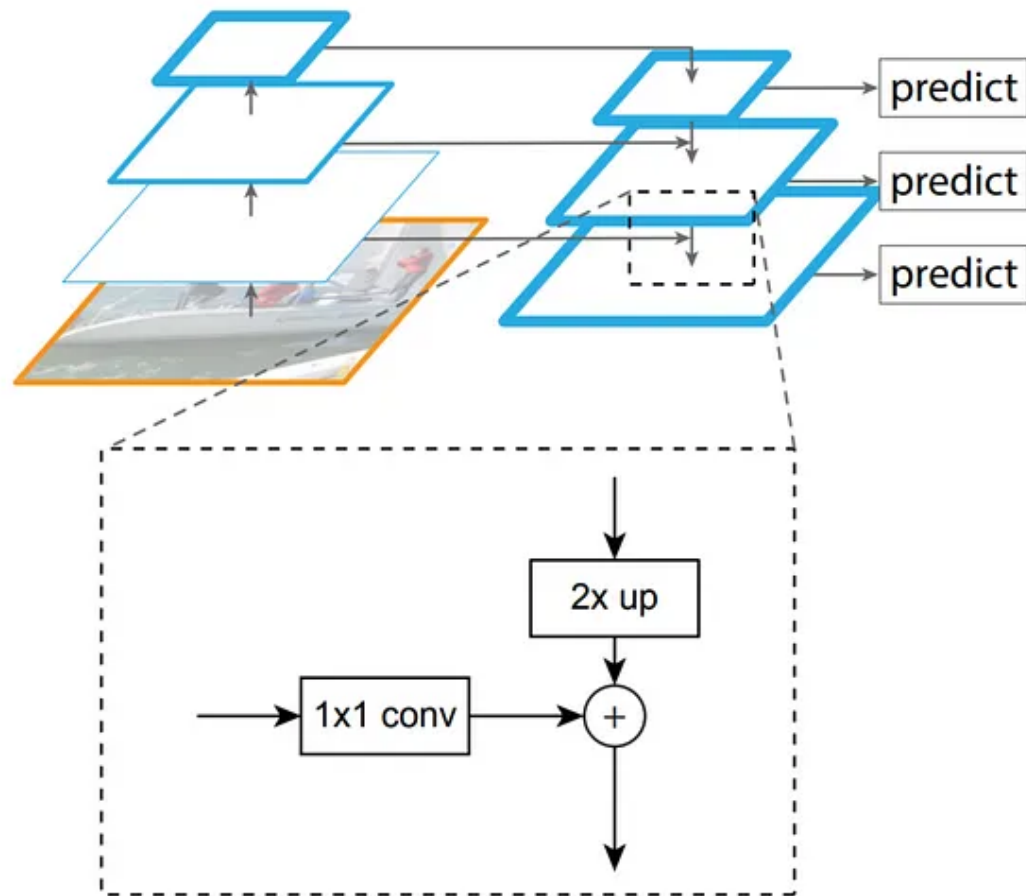
Image pyramids(multiple images of multiple scales) are often used at the time of predictions to improve the results. But computing results using modern deep learning architectures is often an expensive process in terms of both computing and time.

*FPN is based on exploiting the inherent multi-scale pyramidal hierarchy of deep CNN. It is analogous to the difference between RCNN and Fast RCNN, RCNN is a region-based object detector in which first we find ROI's using an algorithm such as selective search and then crop these ROI's(around 2000) from the image and feeding them into CNN to get results and in Fast RCNN the initial layers of CNN are shared for complete image and the ROI cropping is done on the extracted feature map thus saving a lot of time. In the case of FPN, the research is based on exploiting internal multi-scale nature, and the image pyramid is somehow implemented internally to architecture and sharing most parts of the network. We will jump into technical details now.*

CNN is based on the hierarchical structure in which the resolution of the feature map is reduced after each layer but semantics captured by every deeper layer is stronger than the previous layer. The semantically stronger features are spatially coarser because of downsampling. FPN creates an architecture where the semantically stronger features are merged with the features from previous layers(which are subsampled fewer times and thus have more accurate localization information).

The architecture consists of two pathways:

1. Bottom-up pathway (Normal feed-forward CNN)
2. Top-down pathway (New architecture used for merging features)



FPN architecture Source: FPN paper

## Bottom-up pathway (Left pyramid in the above image)

- It is normal feed-forward CNN architecture. In the paper, the authors used Resnet architecture for evaluation of performance. We will first name the layers as C2, C3, C4, C5 which are Conv 2,3,4 and 5 layers in resnet architecture. The size of the feature map after applying C2 is  $\text{image size}/4$  and this spatial dimension is downsampled by a factor of 2 after each layer.

## Top-down pathway (Right pyramid in the above image)

In this pathway, deeper features are merged with lower features using lateral connections. Since the number of channels of layers in the bottom-up

pathway is not the same, a  $1 \times 1$  convolution is applied first to get a fixed number of channels for each layer (this dimension is kept 256 in the paper). The spatial size is also different thus we upsample (2x) the deeper features so that the spatial size of this feature is matched with a higher resolution feature map of the previous layer in the bottom-down pathway. Now the dimensions of two feature maps are the same and they are merged by element-wise addition.

We can understand this with an example. Let's say our image size is  $512 \times 512$ , now the size of feature map after each convolution layer (C2, C3, C4, C5) will be  $[(128 \times 128), (64 \times 64), (32 \times 32), (16 \times 16)]$ . Output number of channels of each layer is  $[256, 512, 1024, 2048]$ . Now we apply a  $1 \times 1$  convolution (with output number of channels = 256) on the outputs of C2, C3, C4, C5 to get an equal number of channels. We will call these intermediate features with the same number of output channels as S2, S3, S4, S5 corresponding to C2, C3, C4, C5. Now S5 is upsampled to  $32 \times 32$  and merged with S4 using element-wise addition. Now, this output will be upsampled to  $64 \times 64$  and will be merged with S3 and so on. We will call outputs from this stage as T2, T3, T4, T5.

To reduce the effect of aliasing because of upsampling a  $3 \times 3$  convolution is applied on T2, T3, T4, T5 to get our final feature maps P2, P3, P4, P5 corresponding to C2, C3, C4, C5. These features are used to generate final classification and regression scores (bbox). The parameters for the head can be shared and separate head gives no added benefit.

That's it in the theory of FPN. But we will see how FPN can be implemented for Faster RCNN and Fast RCNN.

## FPN for Faster RCNN

Faster RCNN uses region proposal network. RPN is used to generate bounding box proposals and these proposals are later used to generate final predictions. RPN is a small network that is implemented over the extracted features of the last layer(C5). A  $3 \times 3$  convolution is applied to this extracted feature followed by two similar  $1 \times 1$  convolution layers (one for classification and other for regression).

RPN is adapted here by simply replacing a single scale feature map with FPN. Thus now RPN is implemented for P2-P5 and not C5 alone. For training of RPN, anchor boxes of multiple scales are used. But since the multi-scale is now inherent in the extracted feature, it is not necessary to have multi-scale anchor boxes on any of the levels. Instead, a single scale anchor box is assigned to each level. The size of the anchor boxes used in the paper is  $\{32^2, 64^2, 128^2, 256^2, 512^2\}$  for  $\{P2, P3, P4, P5, P6\}$ . P6 is introduced here so that a large size anchor box can be used. P6 is subsampled from P5 with stride 2. The anchor boxes of  $\{1:1, 1:2, 2:1\}$  aspect ratios is used.

These anchor boxes are matched with ground truth boxes and the model is trained end to end.

## FPN for Fast RCNN

Implementation of the FPN in fast RCNN is very simple. Fast RCNN uses region proposal techniques such as selective search to generate ROI and uses ROI pooling on the single-scale feature map to get final results. By applying FPN we will have multiple feature maps of different scales and we need a strategy to assign given ROI to feature map(now we have multiple feature maps, which feature map to use for given ROI?).

The feature map used is calculated using:

$$k = \lfloor k_0 + \log_2(\sqrt{wh}/224) \rfloor.$$

Here 224 is training size of the image in the imagenet dataset(resnet used is retained on imagenet).  $k_0$  is the feature map to which ROI of size 224 is assigned,  $w$  and  $h$  is width and height of ROI. The head has shared parameters for each feature map.

## References

1. <https://github.com/potterhsu/easy-fpn.pytorch>
2. [Feature Pyramid Networks for Object Detection.](#)

Peace ...

## List of Papers:

1. [OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks.](#) [[Link to blog](#)]
2. [Rich feature hierarchies for accurate object detection and semantic segmentation\(RCNN\).](#) [[Link to blog](#)]
3. [Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition \(SPPNet\).](#) [[Link to blog](#)]
4. [Fast R-CNN](#) [[Link to blog](#)]
5. [Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.](#) [[Link to blog](#)]

6. [You Only Look Once: Unified, Real-Time Object Detection.](#) [\[Link to blog\]](#)
7. [SSD: Single Shot MultiBox Detector.](#) [\[Link to blog\]](#)
8. [Feature Pyramid Networks for Object Detection.](#) ← You completed this blog.
9. [DSSD: Deconvolutional Single Shot Detector.](#) [\[Link to blog\]](#)
10. [Focal Loss for Dense Object Detection\(Retina net\).](#) [\[Link to blog\]](#)
11. [YOLOv3: An Incremental Improvement.](#) [\[Link to blog\]](#)
12. [SNIPER: Efficient Multi-Scale Training.](#) [\[Link to blog\]](#)
13. [High-Resolution Representations for Labeling Pixels and Regions.](#) [\[Link to blog\]](#)
14. [FCOS: Fully Convolutional One-Stage Object Detection.](#) [\[Link to blog\]](#)
15. [Objects as Points.](#) [\[Link to blog\]](#)
16. [CornerNet-Lite: Efficient Keypoint Based Object Detection.](#) [\[Link to blog\]](#)
17. [CenterNet: Keypoint Triplets for Object Detection.](#) [\[Link to blog\]](#)
18. [CBNet: A Novel Composite Backbone Network Architecture for Object Detection.](#) [\[Link to blog\]](#)
19. [EfficientDet: Scalable and Efficient Object Detection.](#) [\[Link to blog\]](#)
20. [YOLOv4: Optimal Speed and Accuracy of Object Detection.](#) [\[Link to blog\]](#)

[Object Detection](#)[Machine Learning](#)[Deep Learning](#)[Tutorial](#)[Computer Vision](#)





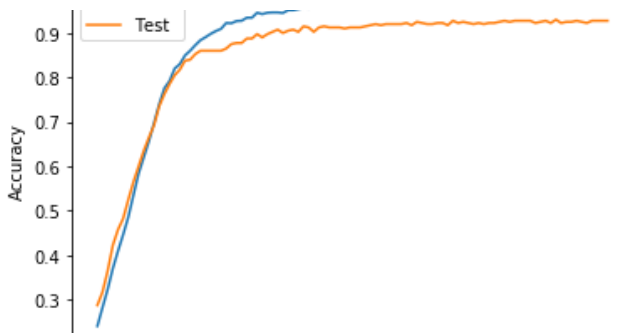
## Written by Sanchit Tanwar


266 Followers · Writer for Analytics Vidhya

Follow

Signup for my live computer vision: [https://bit.ly/cv\\_coursem](https://bit.ly/cv_coursem)

### More from Sanchit Tanwar and Analytics Vidhya



 Sanchit Tanwar in Towards Data Science

## Building our first neural network in keras

In this article, we will make our first neural network(ANN) using keras framework. This...

8 min read · Jun 26, 2019

 669

 14




 Kia Eisinga in Analytics Vidhya

## How to create a Python library

Ever wanted to create a Python library, albeit for your team at work or for some open...

7 min read · Jan 27, 2020


 2.3K

 27





$$R(s, a) + \gamma \sum_{s'} P(s' | s, a)$$

 Shashank Singhal in Analytics Vidhya

## Hadoop : How to install in 5 Steps in Windows 10

An easy to go guide for installing the Hadoop in Windows

7 min read · Mar 28, 2021



198




14



89



 Sanchit Tanwar in Analytics Vidhya

## Bellman Equation and dynamic programming

This is a series of articles on reinforcement learning and if you are new and have not...

3 min read · Jan 29, 2019

See all from Sanchit Tanwar

See all from Analytics Vidhya

## Recommended from Medium

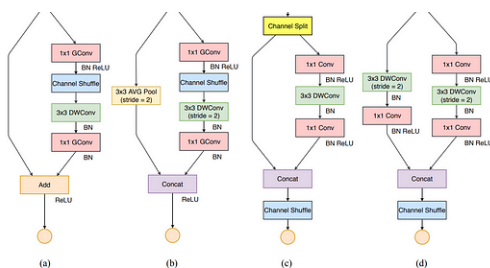
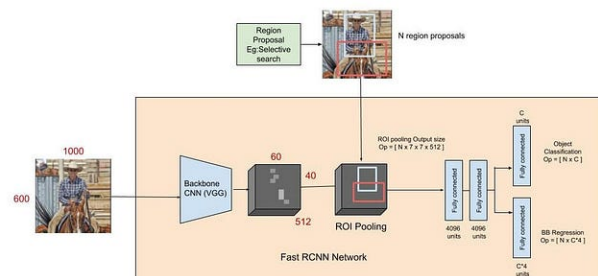


Fig. 3: Building blocks of ShuffleNet v1 [15] and this work. (a): the basic ShuffleNet unit; (b) the ShuffleNet unit for spatial down sampling (2x); (c) our basic unit; (d) our unit for spatial down sampling (2x). DWConv: depthwise convolution, GConv: group convolution.





Sahil Chachra

## Paper Summary — ShuffleNetv2 : Practical Guidelines for Efficient...

In this paper, authors have discussed how using only FLOPs (floating point operations...

6 min read · Aug 26, 2023



103



Everton Gomedede, PhD

## Understanding R-CNN: A Revolution in Object Detection

Introduction

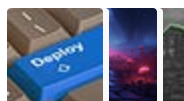
6 min read · Nov 10, 2023



69



### Lists



#### Predictive Modeling w/ Python

20 stories · 912 saves



#### Practical Guides to Machine Learning

10 stories · 1065 saves



#### Natural Language Processing

1201 stories · 676 saves



#### data science and AI

40 stories · 76 saves

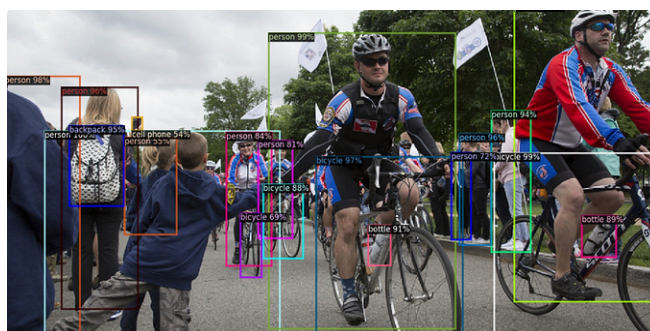


Deepak N R

## Case Study: Strategies for Accurate Small Object Detection i...

Introductory Insights:

6 min read · Sep 25, 2023

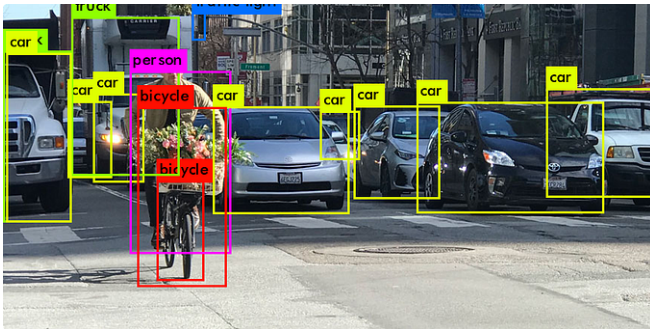


Cagri Goksu Ustundag

## Object Detection Using Detectron2: Training with Custom Dataset

Cagri Goksu Ustundag

7 min read · Oct 11, 2023



Henrique Vedoveli

FAILED

## Metrics Matter: A Deep Dive into Object Detection Evaluation

In the ever-evolving field of computer vision, accurate evaluation metrics are essential for...

7 min read · Sep 15, 2023



See more recommendations