感受野形态对卷积神经网络性能影响的研究

葛一帆[1]，吴建鑫[1]

1. 南京大学软件新技术国家重点实验室, 南京 210023

摘　要： 本文研究卷积神经网络的感受野。不同于大部分的工作，我们分析感受野的形态，而不仅仅是尺寸。给出了感受野的一种形式化定义，发展出一套分析感受野形态的可视化方法。通过对感受野形态的分析，发现广为使用的一些卷积神经网络存在感受野中心偏移和呈“网格”状的问题。我们针对这些问题给出了修改方案，从可视化结果可以看到修改方案消除了这些问题，并通过在分类，物体检测和实例分割等计算机视觉任务上的实验验证了修改方案的有效性。

关键词： 卷积神经网络；感受野；可视化；计算机视觉

Beyond size: how the appearance of the receptive field can affect the performance of a convolutional neural network

GE Yifan[1]，WU Jianxin[1]

1. National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023

Abstract: This paper is on the receptive field of a convolutional neural network. Unlike the vast majority of works, we not only analyze the size of a receptive field but also its appearance. We give a formal definition of the receptive field and develop a method to analyze the appearance of a receptive field with the help of visualization. Based on our analysis, we find that some of the widely-used convolutional neural networks have issues of midpoint shift and “gridding”. We propose a simple cure for these issues and observe that they disappear in the visualization. We validate the effectiveness of the proposed method with experiments on computer vision tasks, including classification, object detection and instance segmentation.

Key words: Convolutional Neural Network; Receptive Field; Visualization; Computer Vision

1　引言

近年来，卷积神经网络在计算机视觉上取得了重大成功。在分类任务上，卷积神经网络甚至达到了超越人类的水平 [1]。在较难的任务，如物体检测 [2]和语义分割 [3]上面，卷积神经网络也有相当不错的表现。

感受野是卷积神经网络中一个很重要的概念，大的感受野也是卷积神经网络相较于传统方法取得更高准确率的重要原因之一。对于卷积神经网络的最终表示或中间表示来说，感受野指的是其用以计算某个输出单元所需的输入层上的对应区域。

在大部分的研究工作中，对感受野的分析局限在讨论它的尺寸（即范围大小）上面。然而，正如 [4]中所揭示的那样，单独的尺寸并不能编码所有信息。我们认为，在分析卷积神经网络感受野的时候，必须同时考虑感受野呈现出的形态。

在本文的第 3 部分中，我们发展了一套通过可视化来观察卷积神经网络感受野形态的工具。利用这套工具，我们分析了 ResNet 的感受野，发现其中存在感受野中心偏移和呈“网格”

基金项目：国家自然科学基金(61772256)

状的问题，这可能会导致卷积神经网络的准确率下降。我们分析发现，这些问题主要是由步长为 2 的卷积导致的，因此提出使用平均池化代替步长为 2 的卷积来进行降采样的修改方案。接着，在第 4 部分中，我们在多个广泛用作基准测试的视觉任务上证实所提方案的有效性。与 ResNet-50 原模型相比，在 ILSVRC2012 [5]上得到了 Top-1 准确率+0.34%的提升，在 COCO2017 [6]物体检测上得到了平均精度+1.40%的提升，在 COCO2017 实例分割上得到了平均精度+1.44%的提升。

2 相关工作

本文的研究受到了 [4]的启发。 [4]提出了“有效感受野”的概念，即，将反向传播时对应于输入图片的偏导数看作其对输出单元影响的程度。本文也采用了类似的切入点和方法，不同的是， [4]将权重以及激活层等的影响也一并考虑在内，而本文想要将这些影响从对感受野的分析中除去。

本文中提出的对卷积神经网络结构的修改方案与 [7]中的 ResNet-D 有相似的部分。不过，[7]并没有给出其修改的动机，是试错型研究，这也就导致其修改方案不彻底；而本文则从观察中发现问题，然后基于此从解决问题的角度设计出修改方案。

3 感受野

3.1 记号定义

表 1：记号

记号	涵义	形式
$\boldsymbol{I}$	输入，图片	矩阵
N	卷积神经网络	可微函数，输入为图片，输出也是一个矩阵
c	输出单元，即 $N(\boldsymbol{I})$ 上某一特定位置（比如，中心）的值	标量，可以看作是根据 N 和 $\boldsymbol{I}$ 计算得到的变量
$\boldsymbol{RF}_{\boldsymbol{I},N,c}$	感受野	矩阵，依赖于 $\boldsymbol{I},N,c$ 的变量，大小和 $\boldsymbol{I}$ 一样

首先定义一下本文使用的记号，见表 1。

其中，$\boldsymbol{RF}_{\boldsymbol{I},N,c}$ 是一个矩阵，是一个用来计算感受野的变量，其 (x,y) 位置的值 $\boldsymbol{RF}_{\boldsymbol{I},N,c}(x,y)$ 表示“输入 $\boldsymbol{I}$ 上 (x,y) 位置的像素经过 N 中的多少条不同路径影响到 c ”。

为了简化讨论，输入 $\boldsymbol{I}$ 和输出 $N(\boldsymbol{I})$ 都看作矩阵，即只有一个通道。本文不考虑卷积神经网络 N 中出现对各通道处理不对称的情况，该简化并不影响 $\boldsymbol{RF}_{\boldsymbol{I},N,c}$。

3.2 感受野的计算

接下来探究如何计算感受野。先考虑一个最简单的情况：

- 输入 $\boldsymbol{I}$ 大小为 5×5；
- 卷积神经网络 N 只包含一个 3×3 卷积；
- c 为输出 $N(\boldsymbol{I})$ 的中心。

则输入 $\boldsymbol{I}$ 只有中心 3×3 大小的区域中每个像素通过 1 条路径影响到 c，其它位置都不参与到对 c 的计算中，即

$$\boldsymbol{RF}_{\boldsymbol{I},N,c}=\begin{pmatrix}0&0&0&0&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&0\\0&0&0&0&0\end{pmatrix}. \tag{1}$$

再考虑 N 有多个层的情况。类似 [4]，容易得到这样一个直观的结论：若 $\frac{\partial c}{\partial \boldsymbol{I}(x,y)}\neq0$，则 $\boldsymbol{RF}_{\boldsymbol{I},N,c}(x,y)>0$。但仅仅有这个结论还不够，因为

1. 可能存在 (x,y) 使得 $\frac{\partial c}{\partial \boldsymbol{I}(x,y)}=0$ 且 $\boldsymbol{RF}_{\boldsymbol{I},N,c}(x,y)>0$；
2. 仅仅有 $\boldsymbol{RF}_{\boldsymbol{I},N,c}(x,y)>0$ 是不能确定 $\boldsymbol{RF}_{\boldsymbol{I},N,c}(x,y)$ 的具体值的，而仅有 1 条路径和多条路径参与计算 c 会产生非常不同的效果。

如果对 N 中的模块和权重等不加限制，就无法通过微分的方法得到更强的结果（即 $\boldsymbol{RF}_{\boldsymbol{I},N,c}(x,y)$ 的具体数值大小）. 我们希望能通过一些操作来改变卷积神经网络 N，得到新的卷积神经网络 N' (把对应的输出单元记为 c')，使得对 $\boldsymbol{I}$ 上的所有像素位置 (x,y)，有

1. $\boldsymbol{RF}_{\boldsymbol{I},N',c'}(x,y)=\boldsymbol{RF}_{\boldsymbol{I},N,c}(x,y)$；
2. $\boldsymbol{RF}_{\boldsymbol{I},N',c'}(x,y)=\frac{\partial c'}{\partial \boldsymbol{I}(x,y)}$。

[4]中提到，如果 N 是个前馈网络，只有卷

积层（无激活函数），卷积层的权重都是 1，偏置都是 0，则它本身就是我们想找的 N'。如果不是这样，那对 N 的改造就应当朝上述性质靠拢。

拿 VGG-16 [8]做一个例子。它的输入是 $224\times224\times3$ 的图片。选定的 c 是最后一层卷积所输出的 $7\times7\times512$ 特征图中第一个通道的中心。对中间经过的模块做如下改动

- 卷积层：将其权重设为全 1，偏置设为全 0；
- ReLU [9]：将它去掉(即变成恒等映射)；
- 最大池化：将其变为权重全 1，偏置全 0 的 depthwise 卷积。

容易看出，通过上述改动得到的卷积神经网络对 VGG-16 来说是符合期望的 N'。对它进行微分，得到的感受野可视化如图 1 所示。在图 1 中，某像素 (x, y) 位于该输出变量的感受野内，当且仅当感受野变量 $\boldsymbol{RF}_{I,N,c}$ 在该像素的值大于 0。

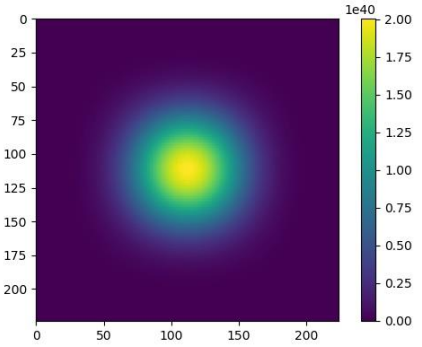
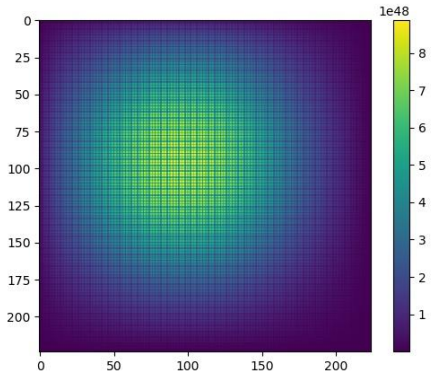


图 1：VGG-16 感受野可视化

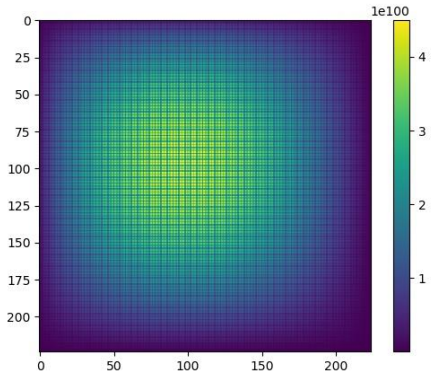
在用深度学习框架（如 PyTorch [10]）实现时，可以将 c' 处的偏导数设为 1，其它全为 0，然后使用反向传播求得输入层的偏导数。

3.3 ResNet [11]中感受野形态存在的问题

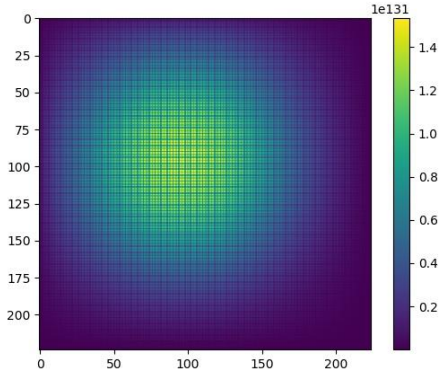
利用所提的方法，还可以可视化 ResNet 的感受野，见图 2。



(a) ResNet-18



(b) ResNet-34



(c) ResNet-50

图 2：ResNet 感受野可视化

可以看到，ResNet 存在两个问题。其一，输出的感受野的中心并不与输入图片的中心重合，存在偏移；其二，感受野呈“网格”状，而不是像 VGG-16 那样平滑过渡。

其它更现代的模型，如 MobileNetV2 [12]和 EfficientNet [13]等，也存在类似的问题。这些问题对于重视准确位置信息的应用，如物体检测和分割，很可能会导致严重的准确率下降，对物体识别也可能有不利影响。

3.3.1 中心偏移现象的成因

ResNet 感受野的中心偏向左上方看起来令人困惑，但其实机制很简单。图 3 是 ResNet 特征图分辨率随着网络前馈变化的过程。感受野中心的偏移就发生在图 3 画绿色框的地方。ResNet 中通过步长为 2 的卷积进行降采样。所以，当特征图的分辨率从14×14变为7×7的时候，输出的中心是(3,3)，对应的14×14特征图上的像素位置是(6,6)，即略偏向左上方。但是，在经过多次步长为 2 的卷积进行降采样后，中心的偏移会变得显著，如图 2 所示。

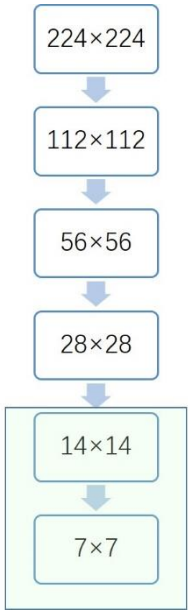


图 3：ResNet 特征图分辨率变化流程

3.3.2 “网格”现象的成因

步长为 2 的卷积引起“网格”现象的机理很好理解。在 ResNet 中，步长为 2 的3×3卷积使得输入该层的特征图各像素参与计算的次数不一样，反映到感受野上就成了“网格”。

一般地，若卷积步长为 1，则不会产生“网格”；若卷积步长＞卷积核大小≥1，则有的点会参与到计算中而有的点不会，会出现“网格”；若卷积核大小＞卷积步长＞1，则有的点加入计算的次数多，有的点少，同样呈现“网格”形态。

3.3.3 本文所提方案

直观上，无论是中心偏移还是“网格”都不是我们希望的：前者可能会导致在物体定位上的系统偏移，后者可能会导致卷积神经网络错误地学习，即，过拟合到本不存在于图片中的纹理。因此，需要修复这两个问题。

在 ResNet 中，这两个问题归根结底都是由步长为 2 的1×1和3×3卷积引起的。一个简单的想法是，将步长为 2 的卷积替换成一个步长为 2 的2×2的平均池化加上一个步长为 1 而卷积核大小不变的卷积。图 4 是这一修改方案的示意图。修改后，每个输出单元对应到输入层上的2×2区域，而由于输入层上的每个像素参与计算的次数相同，“网格”不复存在。

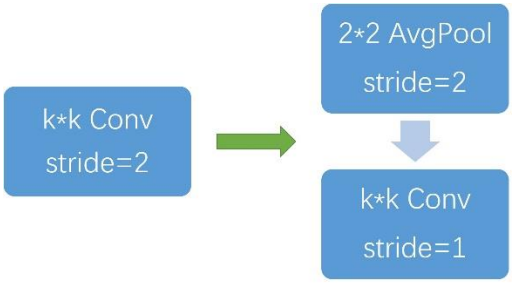
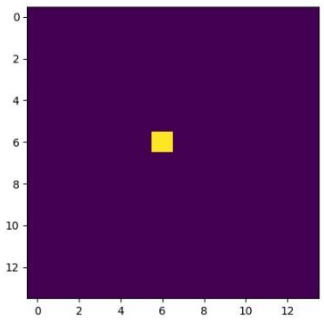
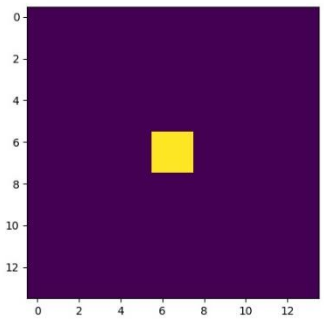


图 4：步长(stride)为 2 的卷积修改方案示意图

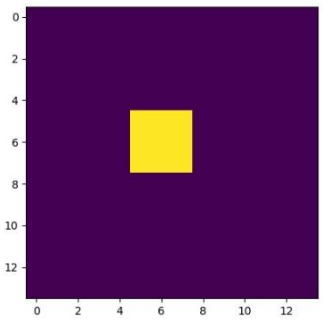
我们可以单独将一个模块看作一个卷积神经网络，观察该模块的感受野，从而分析该模块对整体的感受野的影响。给定输入大小为14×14，步长为 2 的1×1和3×3卷积修改前后的感受野如图 5 所示，其中颜色最深的地方全0，颜色越亮表示数字越大。可以看到，感受野中心偏移的问题消失了，且感受野的非零区域增大了。



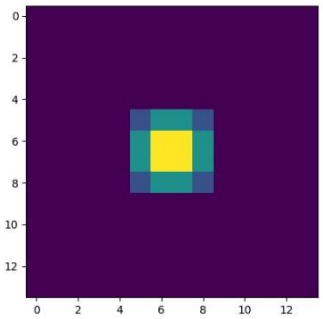
(a) 步长为 2 的 1×1 卷积



(b) 步长为 2 的平均池化加上一个步长为 1 的 1×1 卷积



(c) 步长为 2 的 3×3 卷积



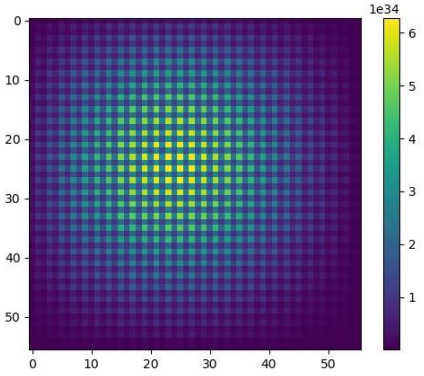
(d) 步长为 2 的平均池化加上一个步长为 1 的 3×3 卷积

图 5：1×1 和 3×3 卷积修改前后感受野对比

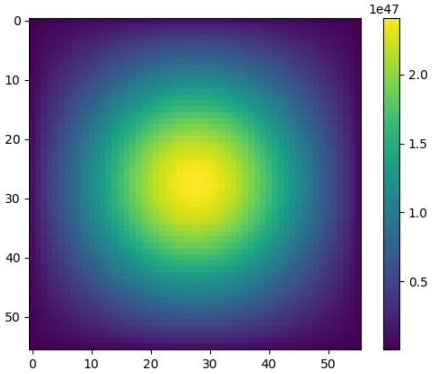
若在整个 ResNet 中所有步长为 2 的 1×1 和 3×3 卷积都做图 4 所示的修改，则可以避免中心偏移，并缓解“网格”问题。（之所以不是避免，是因为 ResNet 中为了快速降低计算量，首先进行了步长为 2 的 7×7 卷积和 3×3 最大池化，这些模块此处不做处理。）

图 6 可视化了本文所提修改方案在 ResNet-18 上的影响。为了更清楚地展现“网格”问题的缓解，将图片经过步长为 2 的 7×7 卷积和 3×3 最大池化这两层降采样之后所得到的特征图看作输入，可视化其后的那些层组成的卷积神经网络的感受野。从图 6 中可以看到，中心偏移和“网格”问题都得到了有效解决。

至此，我们发现了 ResNet 在设计上的一个可能的缺陷，并提出修改方案在形式上解决了问题。但我们仍然不知道这一修改方案是否提升了卷积神经网络的性能，需要通过实验来验证。



(a) 修改前



(b) 修改后

图 6：ResNet-18（去掉前两层降采样）修改前后感受野对比

4 实验

我们在 3 个视觉任务上检验卷积神经网络的性能：ILSVRC2012 [5]分类，COCO2017 [6]物体检测和 COCO2017 实例分割。

我们在 ILSVRC2012 分类任务上训练基于 ResNet-50 的模型，后续两组 COCO2017 实验中使用的主干网络也都基于 ResNet-50。

实验目的是检验图 4 中的修改方案是否提升了卷积神经网络的性能。每个任务上都进行 4 种不同程度的修改：不进行修改，即，使用原模型；只修改 1×1 卷积；只修改 3×3 卷积；同时修改 1×1 和 3×3 卷积。

所有的实验都使用 PyTorch 作为深度学习框架。物体检测和实例分割用到了 Detectron2 [14]代码库。

所有图片预处理中，归一化都按照 ILSVRC2012 的均值和标准差进行。

4.1 实现细节

接下来介绍一些实验的实现细节。

4.1.1 ILSVRC2012 分类

一共训练 90 轮(epoch)，批大小(batch size)为 4096，初始学习率为 1.6。

图片预处理有 3 步，先随机裁剪到 224×224，然后随机水平翻转，最后归一化

优化器使用 SGD，动量(momentum)为 0.9，权重衰减(weight decay)为 1e-4，并开启 Nesterov 动量。

由于批大小比较大，所以训练时学习率先热身(warm-up) 5 轮，以批为单位将学习率从 0 增长到初始学习率，也就是 1.6，其后 25 轮保持 1.6，30 轮保持 1.6e-1，20 轮保持 1.6e-2，10 轮保持 1.6e-3.

4.1.2 COCO2017 物体检测和实例分割

使用在 ILSVRC2012 分类上训练的权重初始化模型进行训练。

除了主干网络和图片归一化外，一切设置与 Detectron2 中预设的配置文件相同：

- 物体检测使用 RetinaNet [2]，配置文件为 retinanet_R_50_FPN_1x.yaml；
- 实例分割使用 Mask R-CNN [3]，配置文件为 mask_rcnn_R_50_FPN_1x.yaml。

4.2 实验结果

1.2.1 ILSVRC2012 分类

实验结果见表 2。从中可以看到，只修改 1×1 卷积能达到最好的 Top-1 准确率，而同时修改 1×1 和 3×3 卷积能达到最好的 Top-5 准确率。只修改 3×3 卷积没有明显与原模型区别开来，Top-1 准确率变低了，Top-5 准确率变高了。

表 2：基于 ResNet-50 的模型在 ILSVRC2012 分类任务上的实验结果

修改		准确率(%)	
1	3	Top-1	Top-5
✗	✗	76.30	93.00
✓	✗	**76.74**	93.28
✗	✓	76.23	93.08
✓	✓	76.44	**93.29**

[7]中提到，把 ResNet-50 的 1×1 卷积如本文一样修改可以达到更好的性能。不过，该工作中对 ResNet-50 的其他部分做了另外的改动，而本实验说明单独修改 1×1 卷积可以带来 ILSVRC2012 分类任务上性能的提高。

4.2.2 COCO2017 物体检测

实验结果见表 3。从中可以看到，3 种不同的修改模型在所有指标上都好于原模型。虽然只修改 3×3 卷积在分类任务上准确率不理想，但它在物体检测任务上比原模型更优。

表 3：主干网络基于 ResNet-50 的 RetinaNet 在 COCO2017 物体检测任务上的实验结果

修改		平均精度(%)					
1	3	AP	AP50	AP75	APs	APm	APl
✗	✗	38.32	59.25	41.29	23.02	41.27	49.20
✓	✗	39.26	60.40	42.93	**23.40**	42.78	49.78
✗	✓	39.23	59.81	42.56	23.37	42.89	50.68
✓	✓	**39.72**	**60.72**	**43.40**	23.34	**43.71**	**50.76**

4.2.3 COCO2017 实例分割

实验结果见表 4。可以看到，情况大致类似于物体检测，修改过的模型均优于未修改的模型。

可以得出结论，就 ResNet-50 而言，本文所提修改方案修复了其设计缺陷，提高了性能。

与分类任务上不同，只修改 3×3 卷积在物体检测和实例分割上都比原模型取得了更好的结果，这可能是因为修改确实使拟合能力有所降低，但由于物体检测和实例分割对感受野的形态很敏感，光滑的感受野带来的增益超过了拟合能力所受的损失。

表 4：主干网络基于 ResNet-50 的 Mask R-CNN 在 COCO2017 实例分割任务上的实验结果

修改		平均精度(%)					
1	3	AP	AP50	AP75	APs	APm	APl
✗	✗	34.65	55.91	36.84	17.02	37.08	49.67
✓	✗	35.57	57.30	37.58	**17.42**	38.21	50.22
✗	✓	35.51	56.90	37.97	17.10	38.25	51.10
✓	✓	**36.09**	**57.67**	**38.66**	17.41	**39.04**	**51.55**

4.3 在其它模型上的尝试

在验证了本文所提修改方案可以提高 ResNet-50 的性能之后，我们在其它具有类似问题的模型上也做了类似的尝试，比如 MobileNetV2 和 EfficientNet。然而，在进行了修改之后，MobileNetV2 和 EfficientNet（以及以它们为主干网络的 RetinaNet）在分类和目标检测任务上性能都有轻微退步。我们分析认为，这可能是由于这些网络采用了 depthwise 卷积，而 ResNet 采用的是经典的卷积算子。

5 结论

我们给出了卷积神经网络感受野的一种形式化定义方式以及利用微分进行计算的方法。使用这套工具，我们发现 ResNet 的感受野中存在中心偏移和呈“网格”状的问题，它们都是由步长为 2 的卷积引起的。然后，我们通过以平均池化取代步长为 2 的卷积进行降采样来解决这些问题。

在 ILSVRC2012 分类，COCO2017 物体检测和 COCO2017 实例分割这 3 个基准测试视觉任务上验证了我们的修改方案能提高 ResNet-50 的性能。我们认为，感受野形态的缺陷在 MobileNetV2 等采用 depthwise 卷积的模型中一样会带来不利影响，其解决方案值得进一步研究。

参考文献

[1] *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.* **He, Kaiming, et al.** s.l. : IEEE Computer Society, 2015. 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015. pp. 1026–1034.

[2] *Focal Loss for Dense Object Detection.* **Lin, Tsung-Yi, et al.** s.l. : IEEE Computer Society, 2017. IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 2999–3007.

[3] *Mask R-CNN.* **He, Kaiming, et al.** s.l. : IEEE Computer Society, 2017. IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 2980–2988.

[4] *Understanding the Effective Receptive Field in Deep Convolutional Neural Networks.* **Luo, Wenjie, et al.** [ed.] Daniel D. Lee, et al. 2016. Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. pp. 4898–4906.

[5] *ImageNet Large Scale Visual Recognition Challenge.* **Russakovsky, Olga, et al.** 2015, Int. J. Comput. Vis., Vol. 115, pp. 211–252.

[6] *Microsoft COCO: Common Objects in Context.* **Lin, Tsung-Yi, et al.** [ed.] David J. Fleet, et al. s.l. : Springer, 2014. Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V. Vol. 8693, pp. 740–755.

[7] *Bag of Tricks for Image Classification with Convolutional Neural Networks.* **He, Tong, et al.** s.l. : Computer Vision Foundation / IEEE, 2019. IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 558–567.

[8] *Very Deep Convolutional Networks for Large-Scale Image Recognition.* **Simonyan, Karen and Zisserman, Andrew.** [ed.] Yoshua Bengio and Yann LeCun. 2015. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

[9] *Rectified Linear Units Improve Restricted Boltzmann Machines.* **Nair, Vinod and Hinton, Geoffrey E.** [ed.] Johannes Fürnkranz and Thorsten Joachims. s.l. : Omnipress, 2010. Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel. pp. 807–814.

[10] *PyTorch: An Imperative Style, High-Performance Deep Learning Library.* **Paszke, Adam, et al.** [ed.] Hanna M. Wallach, et al. 2019. Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada. pp. 8024–8035.

[11] *Deep Residual Learning for Image Recognition.* **He, Kaiming, et al.** s.l. : IEEE Computer Society, 2016. 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 770–778.

[12] *MobileNetV2: Inverted Residuals and Linear Bottlenecks.* **Sandler, Mark, et al.** s.l. : IEEE Computer Society, 2018. 2018 IEEE

Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 4510–4520.

[13] *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.* **Tan, Mingxing and Le, Quoc V.** [ed.] Kamalika Chaudhuri and Ruslan Salakhutdinov. s.l. : PMLR, 2019. Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. Vol. 97, pp. 6105–6114.

[14] **Wu, Yuxin, et al.** Detectron2. *Detectron2.* 2019.