Deep Residual Learning for Image Recognition

深度殘差學習於圖像識別

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun

Microsoft Research 微軟研究院{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract 摘要

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers—8× deeper than VGG nets [41] but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet *test* set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers.

更深的神經網絡更難訓練。我們提出了一個殘差學習框架，以簡化比以前使用的網絡深得多的網絡的訓練。我們明確地將層重新表述為參照層輸入的學習殘差函數，而不是學習無參照的函數。我們提供了綜合的實證證據，顯示這些殘差網絡更容易優化，並且可以從顯著增加的深度中獲得準確性。在 ImageNet 數據集上，我們評估了深度達到 152 層的殘差網絡，比 VGG 網絡 [41] 深 8 層，但仍具有較低的複雜度。這些殘差網絡的集成在 ImageNet 測試集上達到了 3.57% 的錯誤率。這一結果在 ILSVRC 2015 分類任務中獲得了第一名。我們還對 CIFAR-10 的 100 層和 1000 層進行了分析。

The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions[1]

[1] http://image-net.org/challenges/LSVRC/2015/ and http://mscoco.org/dataset/#detections-challenge2015.

http://image-net.org/challenges/LSVRC/2015/ 和 http://mscoco.org/dataset/#detections-challenge2015。

表示的深度對於許多視覺識別任務至關重要。僅由於我們極深的表示，我們在 COCO 物體檢測數據集上取得了 28% 的相對改進。深度殘差網絡是我們提交至 ILSVRC 和 COCO 2015 競賽的基礎 [1]

, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

，我們在 ImageNet 檢測、ImageNet 定位、COCO 檢測和 COCO 分割任務中也獲得了第一名。

1 Introduction 1 介紹

Deep convolutional neural networks [22, 21] have led to a series of breakthroughs for image classification [21, 50, 40]. Deep networks naturally integrate low/mid/high-level features [50] and classifiers in an end-to-end multi-layer fashion, and the "levels" of features can be enriched by the number of stacked layers (depth). Recent evidence [41, 44] reveals that network depth is of crucial importance, and the leading results [41, 44, 13, 16] on the challenging ImageNet dataset [36] all exploit "very deep" [41] models, with a depth of sixteen

[41] to thirty [16]. Many other nontrivial visual recognition tasks [8, 12, 7, 32, 27] have also greatly benefited from very deep models.

深度卷積神經網絡 [22, 21] 已經在圖像分類 [21, 50, 40] 領域取得了一系列突破。深度網絡自然地將低層、中層和高層特徵 [50] 與分類器以端對端的多層方式整合，且通過堆疊層數（深度）來豐富“層級”特徵。近期證據 [41, 44] 顯示網絡深度至關重要，挑戰性的 ImageNet 數據集 [36] 上的領先結果 [41, 44, 13, 16] 都利用了“非常深”的 [41] 模型，其深度範圍從十六層 [41] 到三十層 [16]。許多其他非平凡的視覺識別任務 [8, 12, 7, 32, 27] 也大大受益於非常深的模型。

Driven by the significance of depth, a question arises: *Is learning better networks as easy as stacking more layers?* An obstacle to answering this question was the notorious problem of vanishing/exploding gradients [1, 9], which hamper convergence from the beginning. This problem, however, has been largely addressed by normalized initialization [23, 9, 37, 13] and intermediate normalization layers [16], which enable networks with tens of layers to start converging for stochastic gradient descent (SGD) with backpropagation [22].

由於深度的重要性，提出了一個問題：學習更深的網絡是否和堆疊更多層一樣簡單？回答這個問題的一個障礙是著名的消失/爆炸梯度問題[1, 9]，它從一開始就妨礙了收斂。然而，這個問題已經在很大程度上通過正規化初始化[23, 9, 37, 13]和中間正規化層[16]得到了解決，使得具有數十層的網絡可以開始收斂於隨機梯度下降 (SGD) 與反向傳播[22]。

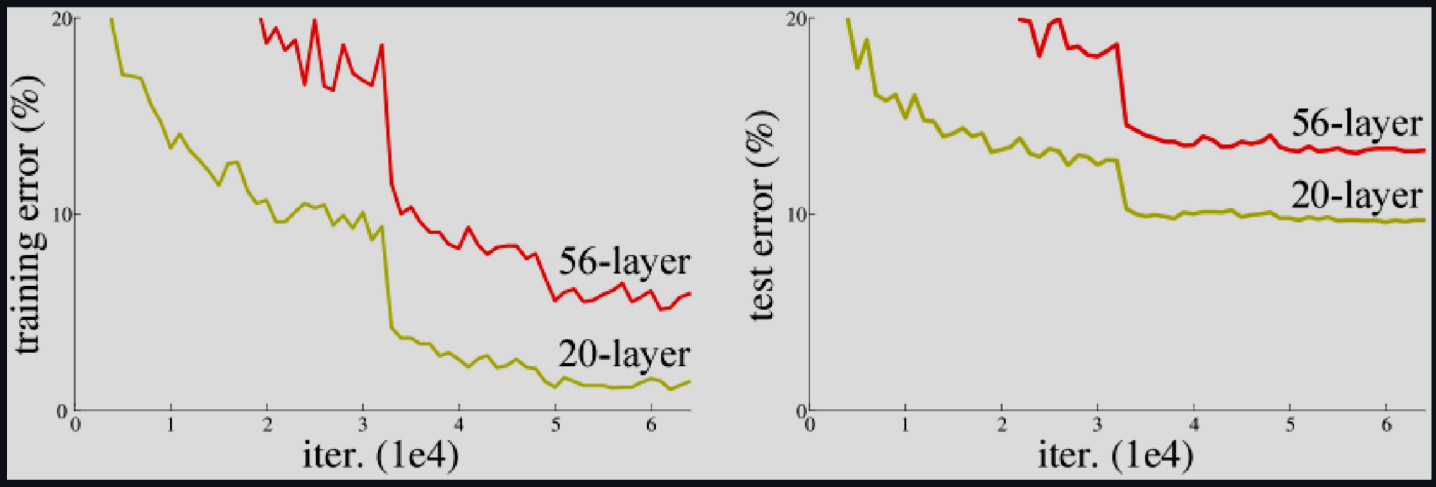

Figure 1: Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

圖 1：在 CIFAR-10 上，20 層和 56 層「普通」網絡的訓練誤差（左）和測試誤差（右）。較深的網絡有較高的訓練誤差，從而有較高的測試誤差。圖 4 展示了在 ImageNet 上的類似現象。

When deeper networks are able to start converging, a *degradation* problem has been exposed: with the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly. Unexpectedly, such degradation is *not caused by overfitting*, and adding more layers to a suitably deep model leads to *higher training error*, as reported in [11, 42] and thoroughly verified by our experiments. Fig. 1 shows a typical example.

當更深的網絡開始收斂時，暴露出了一個降級問題：隨著網絡深度的增加，準確度會達到飽和（這可能並不令人意外），然後迅速下降。出乎意料的是，這種降級並不是由於過擬合引起的，向一個合適深度的模型中添加更多層會導致更高的訓練誤差，正如[11, 42]報告所述，並且我們的實驗也徹底驗證了這一點。圖 1 顯示了一個典型的例子。

The degradation (of training accuracy) indicates that not all systems are similarly easy to optimize. Let us consider a shallower architecture and its deeper counterpart that adds more layers onto it. There exists a solution *by construction* to the deeper model: the added layers are *identity* mapping, and the other layers are copied from the learned shallower model. The existence of this constructed solution indicates that a deeper

model should produce no higher training error than its shallower counterpart. But experiments show that our current solvers on hand are unable to find solutions that are comparably good or better than the constructed solution (or unable to do so in feasible time).

訓練準確度的退化顯示出並非所有系統都同樣容易優化。我們來考慮一個較淺的架構及其更深的對應模型，後者在其上添加了更多層。對於更深的模型，存在一個由構造決定的解：添加的層是恆等映射，其他層則是從已學到的較淺模型中複製而來。這個構造解的存在表明，更深的模型應該不會比其較淺的對應模型產生更高的訓練誤差。但實驗顯示，我們目前手頭的求解器無法找到與這個構造解相媲美或更好的解（或在可行時間內無法做到）。

In this paper, we address the degradation problem by introducing a *deep residual learning* framework. Instead of hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping. Formally, denoting the desired underlying mapping as $\mathcal{H}(\mathbf{x})$, we let the stacked nonlinear layers fit another mapping of $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$. The original mapping is recast into $\mathcal{F}(\mathbf{x}) + \mathbf{x}$. We hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. To the extreme, if an identity mapping were optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers.

在本文中，我們通過引入深度殘差學習框架來解決退化問題。我們不再希望每幾層堆疊的層直接擬合所需的基本映射，而是明確地讓這些層擬合一個殘差映射。形式上，將所需的基本映射表示為 $\mathcal{H}(\mathbf{x})$，我們讓堆疊的非線性層擬合另一個映射 $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$。原始映射被重新表述為 $\mathcal{F}(\mathbf{x}) + \mathbf{x}$。我們假設優化殘差映射比優化原始的、未參考的映射更容易。極端情況下，如果單位映射是最佳的，則將殘差推至零會比通過一堆非線性層擬合單位映射更容易。

The formulation of $\mathcal{F}(\mathbf{x}) + \mathbf{x}$ can be realized by feedforward neural networks with "shortcut connections" (Fig. 2). Shortcut connections [2, 34, 49] are those skipping one or more layers. In our case, the shortcut connections simply perform *identity* mapping, and their outputs are added to the outputs of the stacked layers (Fig. 2). Identity shortcut connections add neither extra parameter nor computational complexity. The entire network can still be trained end-to-end by SGD with backpropagation, and can be easily implemented using common libraries (*e.g.*, Caffe [19]) without modifying the solvers.

$\mathcal{F}(\mathbf{x}) + \mathbf{x}$ 的公式可以通過具有「捷徑連接」的前饋神經網絡來實現（見圖 2）。捷徑連接 [2, 34, 49] 是那些跳過一層或多層的連接。在我們的情況下，捷徑連接僅執行恒等映射，其輸出會加到堆疊層的輸出上（見圖 2）。恒等捷徑連接既不增加額外的參數，也不增加計算複雜度。整個網絡仍然可以通過帶有反向傳播的 SGD 進行端到端訓練，並且可以使用常見庫（例如 Caffe [19]）輕鬆實現，而無需修改求解器。

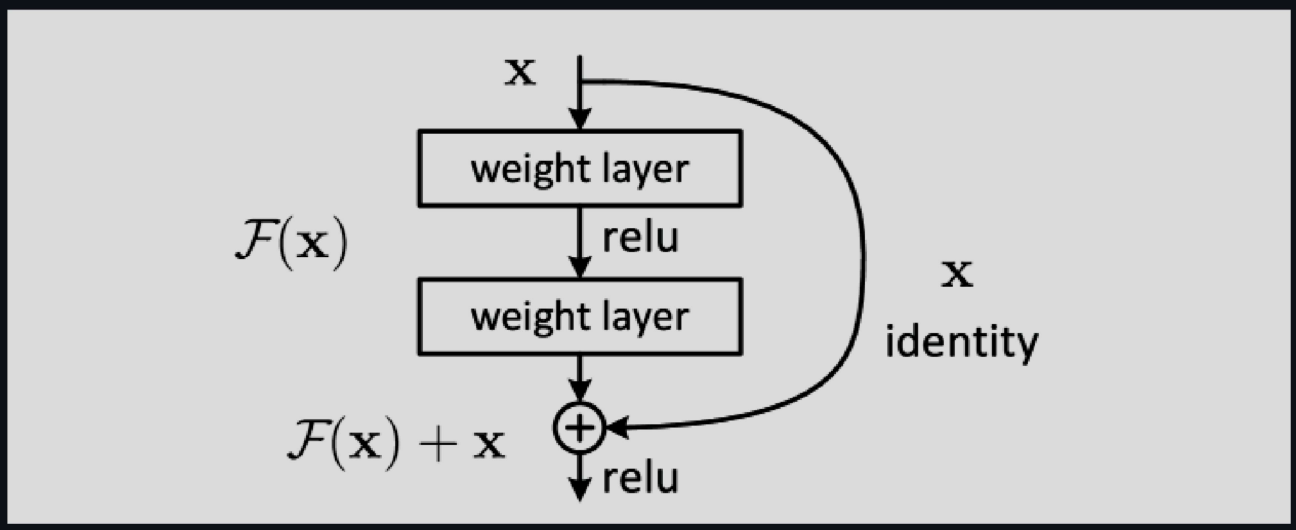

Figure 2: Residual learning: a building block.

圖 2：殘差學習：一個構建模塊。

We present comprehensive experiments on ImageNet [36] to show the degradation problem and evaluate our method. We show that: 1) Our extremely deep residual nets are easy to optimize, but the counterpart "plain" nets (that simply stack layers) exhibit higher training error when the depth increases; 2) Our deep residual nets can easily enjoy accuracy gains from greatly increased depth, producing results substantially better than previous networks.

我們在 ImageNet [36] 上進行了全面的實驗，以展示降級問題並評估我們的方法。我們展示了：1) 我們的極深殘差網絡易於優化，但相對的「普通」網絡（僅僅堆疊層）在深度增加時會顯示出更高的訓練誤差；2) 我們的深層殘差網絡能夠輕鬆地從大幅增加的深度中獲得準確率的提升，產生比之前的網絡顯著更好的結果。

Similar phenomena are also shown on the CIFAR-10 set [20], suggesting that the optimization difficulties and the effects of our method are not just akin to a particular dataset. We present successfully trained models on this dataset with over 100 layers, and explore models with over 1000 layers.

類似的現象也在 CIFAR-10 集合 [20] 上顯示，這表明優化困難和我們的方法效果不僅僅與特定數據集相關。我們展示了在這個數據集上成功訓練的模型，層數超過 100 層，並探索了層數超過 1000 層的模型。

On the ImageNet classification dataset [36], we obtain excellent results by extremely deep residual nets. Our 152-layer residual net is the deepest network ever presented on ImageNet, while still having lower complexity than VGG nets [41]. Our ensemble has **3.57%** top-5 error on the ImageNet *test* set, and *won the 1st place in the ILSVRC 2015 classification competition*. The extremely deep representations also have excellent generalization performance on other recognition tasks, and lead us to further *win the 1st places on: ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation* in ILSVRC & COCO 2015 competitions. This strong evidence shows that the residual learning principle is generic, and we expect that it is applicable in other vision and non-vision problems.

在 ImageNet 分類數據集[36]上，我們通過極其深的殘差網絡取得了優異的結果。我們的 152 層殘差網絡是迄今為止在 ImageNet 上提出的最深的網絡，但其複雜度仍低於 VGG 網絡[41]。我們的集成模型在 ImageNet 測試集上的 top-5 錯誤率為 3.57%，並在 ILSVRC 2015 分類競賽中獲得了第一名。這些極深的表示在其他識別任務上也展現了優秀的泛化性能，並使我們進一步在以下比賽中獲得了第一名：ImageNet 檢測、ImageNet 定位、COCO 檢測和 COCO 分割，這些都是 ILSVRC & COCO 2015 競賽中的獎項。這一強有力的證據表明殘差學習原則具有普遍性，我們期望它適用於其他視覺和非視覺問題。

2 Related Work 2 相關工作

Residual Representations. In image recognition, VLAD [18] is a representation that encodes by the residual vectors with respect to a dictionary, and Fisher Vector [30] can be formulated as a probabilistic version [18] of VLAD. Both of them are powerful shallow representations for image retrieval and classification [4, 48]. For vector quantization, encoding residual vectors [17] is shown to be more effective than encoding original vectors.

殘差表示。在影像識別中，VLAD [18] 是一種通過相對於字典的殘差向量進行編碼的表示，Fisher Vector [30] 可以被表述為 VLAD 的概率版本 [18]。這兩者都是影像檢索和分類 [4, 48] 的強大淺層表示。對於向量量化，編碼殘差向量 [17] 被證明比編碼原始向量更有效。

In low-level vision and computer graphics, for solving Partial Differential Equations (PDEs), the widely used Multigrid method [3] reformulates the system as subproblems at multiple scales, where each subproblem is responsible for the residual solution between a coarser and a finer scale. An alternative to Multigrid is hierarchical basis preconditioning [45, 46], which relies on variables that represent residual vectors between two scales. It has been shown [3, 45, 46] that these solvers converge much faster than standard solvers that are unaware of the residual nature of the solutions. These methods suggest that a good reformulation or pre-

conditioning can simplify the optimization.

在低階視覺和計算機圖形學中，為了解決偏微分方程（PDEs），廣泛使用的多重網格方法 [3] 將系統重新表述為多個尺度上的子問題，其中每個子問題負責粗糙尺度和細緻尺度之間的殘差解。多重網格的替代方案是分層基預處理 [45, 46]，該方法依賴於表示兩個尺度之間殘差向量的變數。已經顯示 [3, 45, 46] 這些解算器比對殘差性質不知情的標準解算器收斂速度快得多。這些方法表明，一個好的重新表述或預處理可以簡化優化過程。

Shortcut Connections. Practices and theories that lead to shortcut connections [2, 34, 49] have been studied for a long time. An early practice of training multi-layer perceptrons (MLPs) is to add a linear layer connected from the network input to the output [34, 49]. In [44, 24], a few intermediate layers are directly connected to auxiliary classifiers for addressing vanishing/exploding gradients. The papers of [39, 38, 31, 47] propose methods for centering layer responses, gradients, and propagated errors, implemented by shortcut connections. In [44], an "inception" layer is composed of a shortcut branch and a few deeper branches.

快捷連接。導致快捷連接的實踐和理論 [2, 34, 49] 已經被研究了很長時間。早期訓練多層感知器 (MLPs) 的一種做法是添加一個從網絡輸入到輸出的線性層 [34, 49]。在 [44, 24] 中，一些中間層直接連接到輔助分類器，以解決消失/爆炸梯度問題。[39, 38, 31, 47] 的論文提出了通過快捷連接來中心化層響應、梯度和傳播錯誤的方法。在 [44] 中，一個"起始"層由一個快捷分支和幾個更深層的分支組成。

Concurrent with our work, "highway networks" [42, 43] present shortcut connections with gating functions [15]. These gates are data-dependent and have parameters, in contrast to our identity shortcuts that are parameter-free. When a gated shortcut is "closed" (approaching zero), the layers in highway networks represent *non-residual* functions. On the contrary, our formulation always learns residual functions; our identity shortcuts are never closed, and all information is always passed through, with additional residual functions to be learned. In addition, highway networks have not demonstrated accuracy gains with extremely increased depth (*e.g.*, over 100 layers).

與我們的工作並行的是，"高速公路網絡"[42, 43]提供了帶有閘控功能的捷徑連接[15]。這些閘控是數據依賴的並且具有參數，與我們的身份捷徑（無參數）不同。當閘控捷徑"關閉"（接近零）時，高速公路網絡中的層表示非殘差函數。相反，我們的公式總是學習殘差函數；我們的身份捷徑從不會關閉，所有信息總是會通過，並學習額外的殘差函數。此外，高速公路網絡在極度增加深度（例如，超過 100 層）時未顯示出準確性增益。

3 Deep Residual Learning 3Deep 殘差學習

3.1 Residual Learning 3.1 殘差學習

Let us consider $\mathcal{H}(\mathbf{x})$ as an underlying mapping to be fit by a few stacked layers (not necessarily the entire net), with $\mathbf{x}$ denoting the inputs to the first of these layers. If one hypothesizes that multiple nonlinear layers can asymptotically approximate complicated functions[2]

[2]This hypothesis, however, is still an open question. See [28].

然而，這一假設仍然是個開放性問題。見 [28]。

讓我們考慮 $\mathcal{H}(\mathbf{x})$ 作為需要由幾個堆疊層（不一定是整個網絡）擬合的基礎映射，其中 $\mathbf{x}$ 表示這些層的第一層的輸入。如果假設多個非線性層可以漸近地逼近複雜的函數 [2]

, then it is equivalent to hypothesize that they can asymptotically approximate the residual functions, *i.e.*, $\mathcal{H}(\mathbf{x}) - \mathbf{x}$ (assuming that the input and output are of the same dimensions). So rather than expect stacked layers to approximate $\mathcal{H}(\mathbf{x})$, we explicitly let these layers approximate a residual function $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$. The original function thus becomes $\mathcal{F}(\mathbf{x}) + \mathbf{x}$. Although both forms should be able to asymptotically approximate

the desired functions (as hypothesized), the ease of learning might be different.

，那麼這等同於假設它們可以漸近地逼近殘差函數，即 $\mathcal{H}(\mathbf{x})-\mathbf{x}$ （假設輸入和輸出具有相同的維度）。因此，與其期望堆疊層來逼近 $\mathcal{H}(\mathbf{x})$ ，我們明確地讓這些層來逼近一個殘差函數 $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x})-\mathbf{x}$ 。原始函數因此變成 $\mathcal{F}(\mathbf{x})+\mathbf{x}$ 。雖然這兩種形式應該能夠漸近地逼近所需的函數（如假設的那樣），但學習的難易程度可能會有所不同。

This reformulation is motivated by the counterintuitive phenomena about the degradation problem (Fig. 1, left). As we discussed in the introduction, if the added layers can be constructed as identity mappings, a deeper model should have training error no greater than its shallower counterpart. The degradation problem suggests that the solvers might have difficulties in approximating identity mappings by multiple nonlinear layers. With the residual learning reformulation, if identity mappings are optimal, the solvers may simply drive the weights of the multiple nonlinear layers toward zero to approach identity mappings.

這種重新表述是由於對降級問題的反直覺現象的驅動（見圖 1 左）。正如我們在引言中所討論的，如果添加的層可以構建為恒等映射，則較深的模型應該具有不大於其較淺對應物的訓練誤差。降級問題表明，解算器可能在通過多個非線性層來近似恒等映射時會遇到困難。通過殘差學習的重新表述，如果恒等映射是最佳的，解算器可以簡單地將多個非線性層的權重驅動為零，以接近恒等映射。

In real cases, it is unlikely that identity mappings are optimal, but our reformulation may help to precondition the problem. If the optimal function is closer to an identity mapping than to a zero mapping, it should be easier for the solver to find the perturbations with reference to an identity mapping, than to learn the function as a new one. We show by experiments (Fig. 7) that the learned residual functions in general have small responses, suggesting that identity mappings provide reasonable preconditioning.

在實際情況中，身份映射不太可能是最佳解，但我們的重新公式化可能有助於預處理問題。如果最佳函數比零映射更接近身份映射，則對於解算器而言，參照身份映射尋找擾動應該比將函數作為新函數來學習更容易。我們通過實驗（圖 7）顯示，學習到的殘差函數通常具有較小的響應，這表明身份映射提供了合理的預處理。

3.2 Identity Mapping by Shortcuts

3.2 透過捷徑的身份映射

We adopt residual learning to every few stacked layers. A building block is shown in Fig. 2. Formally, in this paper we consider a building block defined as:

我們將殘差學習應用於每幾層堆疊的層。建築塊如圖 2 所示。正式地，在本文中，我們考慮一個定義為：

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}. \tag{1}$$

Here $\mathbf{x}$ and $\mathbf{y}$ are the input and output vectors of the layers considered. The function $\mathcal{F}(\mathbf{x}, \{W_i\})$ represents the residual mapping to be learned. For the example in Fig. 2 that has two layers, $\mathcal{F} = W_2\sigma(W_1\mathbf{x})$ in which σ denotes ReLU [29] and the biases are omitted for simplifying notations. The operation $\mathcal{F}+\mathbf{x}$ is performed by a shortcut connection and element-wise addition. We adopt the second nonlinearity after the addition (*i.e.*, $\sigma(\mathbf{y})$, see Fig. 2).

這裡 $\mathbf{x}$ 和 $\mathbf{y}$ 是考慮的層的輸入和輸出向量。函數 $\mathcal{F}(\mathbf{x}, \{W_i\})$ 代表需要學習的殘差映射。對於圖 2 中的示例，該示例有兩層， $\mathcal{F} = W_2\sigma(W_1\mathbf{x})$ 其中 σ 表示 ReLU [29]，並且為簡化符號而省略了偏置。操作 $\mathcal{F}+\mathbf{x}$ 是通過捷徑連接和逐元素加法來完成的。我們在加法後採用第二個非線性（即 $\sigma(\mathbf{y})$ ，見圖 2）。

The shortcut connections in Eqn.(1) introduce neither extra parameter nor computation complexity. This is not only attractive in practice but also important in our comparisons between plain and residual networks. We can fairly compare plain/residual networks that simultaneously have the same number of parameters, depth, width, and computational cost (except for the negligible element-wise addition).

公式（1）中的快捷連接不會引入額外的參數或計算複雜度。這不僅在實踐中具有吸引力，而且在我們對比普通網絡和殘差網絡時也非常重要。我們可以公平地比較擁有相同參數數量、深度、寬度和計算成本的普通/殘差網絡（忽略微不足道的逐元素加法）。

The dimensions of $\mathbf{x}$ and $\mathcal{F}$ must be equal in Eqn.(1). If this is not the case (*e.g.*, when changing the input/output channels), we can perform a linear projection W_s by the shortcut connections to match the dimensions:

在方程式（1）中，$\mathbf{x}$ 和 $\mathcal{F}$ 的尺寸必須相等。如果不是這樣（例如，在改變輸入/輸出通道時），我們可以通過捷徑連接進行線性投影 W_s 以匹配尺寸：

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}. \tag{2}$$

We can also use a square matrix W_s in Eqn.(1). But we will show by experiments that the identity mapping is sufficient for addressing the degradation problem and is economical, and thus W_s is only used when matching dimensions.

我們也可以在公式(1)中使用一個方陣 W_s 。但我們將通過實驗展示，身份映射足以解決退化問題並且經濟，因此 W_s 只有在匹配維度時才使用。

The form of the residual function $\mathcal{F}$ is flexible. Experiments in this paper involve a function $\mathcal{F}$ that has two or three layers (Fig. 5), while more layers are possible. But if $\mathcal{F}$ has only a single layer, Eqn.(1) is similar to a linear layer: $\mathbf{y} = W_1\mathbf{x} + \mathbf{x}$, for which we have not observed advantages.

殘差函數 $\mathcal{F}$ 的形式是靈活的。本文中的實驗涉及一個具有兩層或三層的函數 $\mathcal{F}$ （見圖 5），但也可以有更多層。但是如果 $\mathcal{F}$ 只有單層，公式(1) 就類似於一個線性層： $\mathbf{y} = W_1\mathbf{x} + \mathbf{x}$ ，我們尚未觀察到其優勢。

We also note that although the above notations are about fully-connected layers for simplicity, they are applicable to convolutional layers. The function $\mathcal{F}(\mathbf{x}, \{W_i\})$ can represent multiple convolutional layers. The element-wise addition is performed on two feature maps, channel by channel.

我們還注意到，雖然上述符號是針對全連接層為了簡便而提出的，但它們同樣適用於卷積層。函數 $\mathcal{F}(\mathbf{x}, \{W_i\})$ 可以表示多個卷積層。逐元素加法是在兩個特徵圖上進行的，按通道進行。

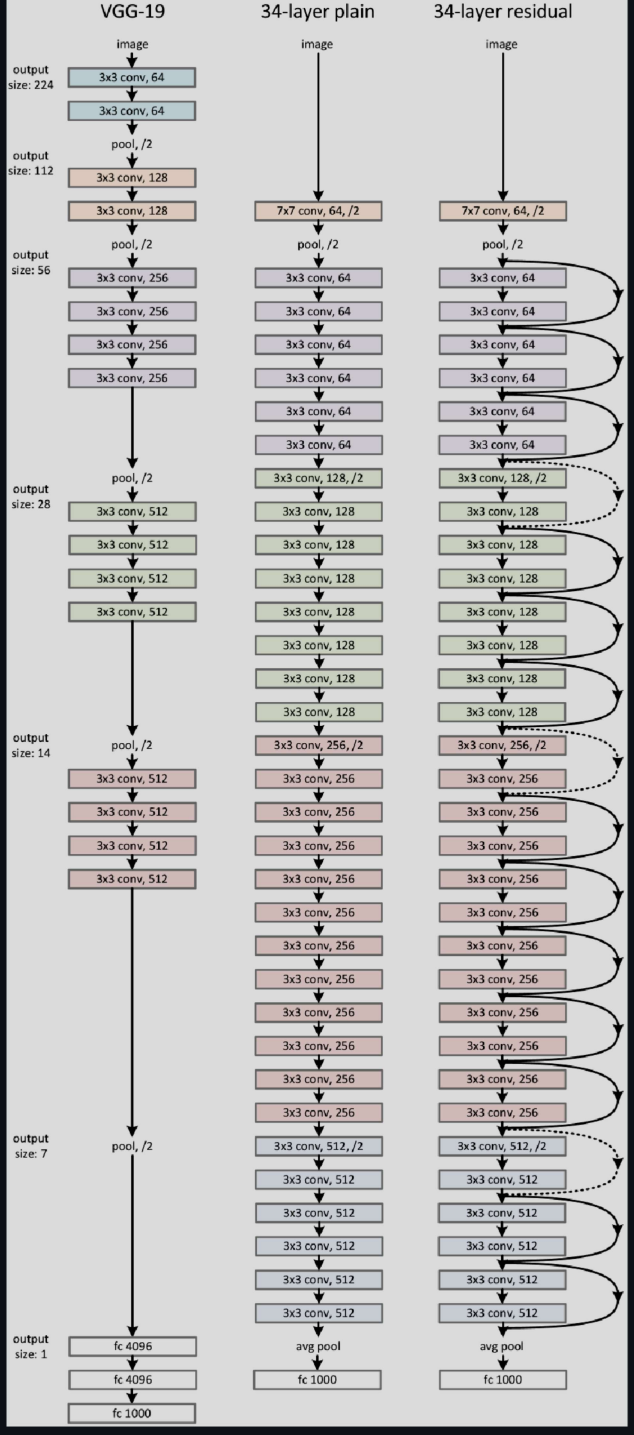

Figure 3: Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

圖 3：ImageNet 的網絡架構示例。左：VGG-19 模型 [41]（19.6 億 FLOPs）作為參考。中：具有 34 個參數層的普通網絡（3.6 億 FLOPs）。右：具有 34 個參數層的殘差網絡（3.6 億 FLOPs）。虛線快捷方式增加了維度。表 1 顯示了更多詳細資訊和其他變體。

3.3 Network Architectures 3.3 網絡架構

We have tested various plain/residual nets, and have observed consistent phenomena. To provide instances for discussion, we describe two models for ImageNet as follows.

我們已測試了各種普通/殘差網絡，並觀察到了相似的現象。為了提供討論的實例，我們描述了兩個 ImageNet 模型，如下所示。

Plain Network. Our plain baselines (Fig. 3, middle) are mainly inspired by the philosophy of VGG nets [41] (Fig. 3, left). The convolutional layers mostly have 3×3 filters and follow two simple design rules: (i) for the same output feature map size, the layers have the same number of filters; and (ii) if the feature map size is halved, the number of filters is doubled so as to preserve the time complexity per layer. We perform down-sampling directly by convolutional layers that have a stride of 2. The network ends with a global average pooling layer and a 1000-way fully-connected layer with softmax. The total number of weighted layers is 34 in Fig. 3 (middle).

Plain Network。我們的普通基線（圖 3，中間）主要受到 VGG 網絡[41]（圖 3，左）哲學的啟發。卷積層大多擁有 3x3 的濾波器，並遵循兩個簡單的設計原則：（i）對於相同的輸出特徵圖大小，層具有相同數量的濾波器；（ii）如果特徵圖大小減半，濾波器的數量加倍，以保持每層的時間複雜度。我們通過具有 2 的步幅的卷積層直接進行降採樣。網絡以一個全局平均池化層和一個具有 softmax 的 1000-way 全連接層結束。圖 3（中間）的加權層總數為 34。

It is worth noticing that our model has *fewer* filters and *lower* complexity than VGG nets [41] (Fig. 3, left). Our 34-layer baseline has 3.6 billion FLOPs (multiply-adds), which is only 18% of VGG-19 (19.6 billion FLOPs).

值得注意的是，我們的模型比 VGG 網絡 [41] (圖 3 左側) 擁有更少的濾波器和更低的複雜度。我們的 34 層基準模型擁有 36 億 FLOPs（乘加運算），僅為 VGG-19（196 億 FLOPs）的 18%。

Residual Network. Based on the above plain network, we insert shortcut connections (Fig. 3, right) which turn the network into its counterpart residual version. The identity shortcuts (Eqn.(1)) can be directly used when the input and output are of the same dimensions (solid line shortcuts in Fig. 3). When the dimensions increase (dotted line shortcuts in Fig. 3), we consider two options: (A) The shortcut still performs identity mapping, with extra zero entries padded for increasing dimensions. This option introduces no extra parameter; (B) The projection shortcut in Eqn.(2) is used to match dimensions (done by 1×1 convolutions). For both options, when the shortcuts go across feature maps of two sizes, they are performed with a stride of 2.

殘差網絡。基於上述的普通網絡，我們插入了快捷連接（圖 3，右），這使得網絡轉變為其對應的殘差版本。當輸入和輸出的維度相同時，可以直接使用身份快捷連接（公式（1），圖 3 中的實線快捷連接）。當維度增加時（圖 3 中的虛線快捷連接），我們考慮兩個選項：（A）快捷連接仍執行身份映射，並為增加的維度填充額外的零條目。此選項不引入額外參數；（B）使用公式（2）中的投影快捷連接來匹配維度（通過 1×1 卷積完成）。對於這兩個選項，當快捷連接跨越兩種大小的特徵圖時，它們以步幅 2 進行。

layer name 層名稱	output size 輸出大小	18-layer 18 層	34-layer 34 層	50-layer 50 層	101-layer 101 層	152-layer 152 層
conv1	112 × 112	7 × 7, 64, stride 2 7 × 7, 64, 步幅 2				
conv2_x	56 × 56	3 × 3 max pool, stride 2 3 × 3 最大池化，步幅 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28 × 28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14 × 14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7 × 7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1 × 1	average pool, 1000-d fc, softmax 平均池化、1000-d fc、softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1: Architectures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of blocks stacked. Downsampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2.

表 1: ImageNet 架構。建構區塊顯示在括號中（另見圖 5），數字表示堆疊的區塊數量。降採樣由 conv3_1、conv4_1 和 conv5_1 進行，步幅為 2。

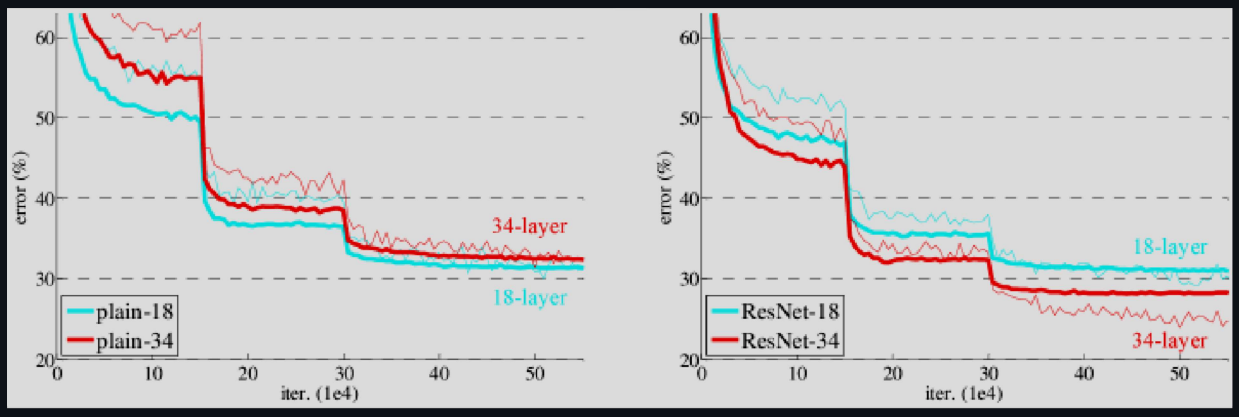

Figure 4: Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

圖 4：在 ImageNet 上的訓練。細曲線表示訓練誤差，粗曲線表示中心裁剪的驗證誤差。左：18 層和 34 層的普通網絡。右：18 層和 34 層的 ResNets。在這個圖中，殘差網絡與其普通對應網絡相比沒有額外的參數。

3.4 Implementation 3.4 實作

Our implementation for ImageNet follows the practice in [21, 41]. The image is resized with its shorter side randomly sampled in [256, 480] for scale augmentation [41]. A 224×224 crop is randomly sampled from an image or its horizontal flip, with the per-pixel mean subtracted [21]. The standard color augmentation in [21

] is used. We adopt batch normalization (BN) [16] right after each convolution and before activation, following [16]. We initialize the weights as in [13] and train all plain/residual nets from scratch. We use SGD with a mini-batch size of 256. The learning rate starts from 0.1 and is divided by 10 when the error plateaus, and the models are trained for up to 60×10^4 iterations. We use a weight decay of 0.0001 and a momentum of 0.9. We do not use dropout [14], following the practice in [16].

我們對於 ImageNet 的實作遵循 [21, 41] 中的做法。影像的短邊會在 [256, 480] 中隨機取樣以進行尺度擴增 [41]。從影像或其水平翻轉中隨機取樣 224 × 224 的裁剪，並減去每個像素的均值 [21]。使用 [21] 中的標準顏色擴增。我們在每個卷積層後及激活前採用批量正規化 (BN) [16]，遵循 [16]。我們按照 [13] 中的方法初始化權重，並從頭開始訓練所有的普通/殘差網絡。我們使用 mini-batch 大小為 256 的 SGD。學習率從 0.1 開始，當誤差穩定時除以 10，模型訓練最多 60×10^4 次迭代。我們使用 0.0001 的權重衰減和 0.9 的動量。我們不使用 dropout [14]，遵循 [16] 中的做法。

In testing, for comparison studies we adopt the standard 10-crop testing [21]. For best results, we adopt the fully-convolutional form as in [41, 13], and average the scores at multiple scales (images are resized such that the shorter side is in $\{224, 256, 384, 480, 640\}$).

在測試中，為了進行比較研究，我們採用標準的 10-crop 測試 [21]。為了獲得最佳結果，我們採用完全卷積的形式，如 [41, 13] 所示，並在多個尺度下平均分數（圖像被調整大小，使得較短的一邊為 $\{224, 256, 384, 480, 640\}$ ）。

4 Experiments 4 實驗

4.1 ImageNet Classification

4.1 ImageNet 分類

We evaluate our method on the ImageNet 2012 classification dataset [36] that consists of 1000 classes. The models are trained on the 1.28 million training images, and evaluated on the 50k validation images. We also obtain a final result on the 100k test images, reported by the test server. We evaluate both top-1 and top-5 error rates.

我們在 ImageNet 2012 分類數據集 [36] 上評估我們的方法，該數據集包含 1000 類。模型在 128 萬張訓練圖像上進行訓練，並在 50k 驗證圖像上進行評估。我們還在 100k 測試圖像上獲得最終結果，由測試伺服器報告。我們評估了 top-1 和 top-5 錯誤率。

Plain Networks. We first evaluate 18-layer and 34-layer plain nets. The 34-layer plain net is in Fig. 3 (middle). The 18-layer plain net is of a similar form. See Table 1 for detailed architectures.

純網絡。我們首先評估 18 層和 34 層純網絡。34 層純網絡如圖 3（中間）所示。18 層純網絡形式類似。詳細架構見表 1。

The results in Table 2 show that the deeper 34-layer plain net has higher validation error than the shallower 18-layer plain net. To reveal the reasons, in Fig. 4 (left) we compare their training/validation errors during the training procedure. We have observed the degradation problem - the 34-layer plain net has higher *training* error throughout the whole training procedure, even though the solution space of the 18-layer plain network is a subspace of that of the 34-layer one.

表 2 中的結果顯示，較深的 34 層純網絡的驗證錯誤高於較淺的 18 層純網絡。為了揭示原因，在圖 4（左）中，我們比較了它們在訓練過程中的訓練/驗證錯誤。我們觀察到了降級問題——雖然 18 層純網絡的解決空間是 34 層純網絡解決空間的子空間，但 34 層純網絡在整個訓練過程中具有更高的訓練錯誤。

	plain 簡單	ResNet ResNet (殘差網絡)
18 layers 18 層	27.94	27.88

34 layers 34 層	28.54	**25.03**

Table 2: Top-1 error (%, 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

表 2: ImageNet 驗證集上的 Top-1 錯誤率（%，10 種裁剪測試）。這裡的 ResNet 與其普通版本相比沒有額外參數。圖 4 顯示了訓練過程。

We argue that this optimization difficulty is *unlikely* to be caused by vanishing gradients. These plain networks are trained with BN [16], which ensures forward propagated signals to have non-zero variances. We also verify that the backward propagated gradients exhibit healthy norms with BN. So neither forward nor backward signals vanish. In fact, the 34-layer plain net is still able to achieve competitive accuracy (Table 3), suggesting that the solver works to some extent. We conjecture that the deep plain nets may have exponentially low convergence rates, which impact the reducing of the training error[3]

[3]We have experimented with more training iterations (3×) and still observed the degradation problem, suggesting that this problem cannot be feasibly addressed by simply using more iterations.

我們已經嘗試了更多的訓練迭代次數（3 × ），仍然觀察到了退化問題，這表明僅僅使用更多的迭代次數無法切實解決這個問題。

我們認為這個優化困難不太可能是由於梯度消失造成的。這些普通網絡使用了批量歸一化（BN）進行訓練，這確保了前向傳播的信號具有非零的方差。我們也驗證了使用 BN 時，反向傳播的梯度顯示出健康的範數。因此，前向和反向信號都沒有消失。事實上，34 層的普通網絡仍然能夠達到具有競爭力的準確度（表 3），這表明解算器在某種程度上是有效的。我們推測，深層的普通網絡可能具有指數級低的收斂速度，這影響了訓練誤差的減少。

. The reason for such optimization difficulties will be studied in the future.

這種優化困難的原因將在未來進行研究。

Residual Networks. Next we evaluate 18-layer and 34-layer residual nets (*ResNets*). The baseline architectures are the same as the above plain nets, expect that a shortcut connection is added to each pair of 3×3 filters as in Fig. 3 (right). In the first comparison (Table 2 and Fig. 4 right), we use identity mapping for all shortcuts and zero-padding for increasing dimensions (option A). So they have *no extra parameter* compared to the plain counterparts.

殘差網絡。接下來我們評估 18 層和 34 層殘差網絡（ResNets）。基線架構與上述簡單網絡相同，只是每對 3 × 3 濾波器之間添加了一個捷徑連接，如圖 3（右）所示。在第一次比較中（表 2 和圖 4 右），我們對所有捷徑使用恆等映射，並對增加維度使用零填充（選項 A）。因此，它們與簡單對應網絡相比沒有額外的參數。

We have three major observations from Table 2 and Fig. 4. First, the situation is reversed with residual learning – the 34-layer ResNet is better than the 18-layer ResNet (by 2.8%). More importantly, the 34-layer ResNet exhibits considerably lower training error and is generalizable to the validation data. This indicates that the degradation problem is well addressed in this setting and we manage to obtain accuracy gains from increased depth.

我們從表 2 和圖 4 中有三個主要觀察結果。首先，殘差學習的情況是顛倒的——34 層 ResNet 比 18 層 ResNet 更好（提高了 2.8%）。更重要的是，34 層 ResNet 顯示出顯著較低的訓練誤差，並且對驗證數據具有良好的泛化能力。這表明在這種設置中，退化問題得到了很好的解決，我們成功地從增加深度中獲得了準確率的提升。

Second, compared to its plain counterpart, the 34-layer ResNet reduces the top-1 error by 3.5% (Table 2), resulting from the successfully reduced training error (Fig. 4 right *vs.* left). This comparison verifies the effec-

tiveness of residual learning on extremely deep systems.

第二，與其普通版本相比，34 層的 ResNet 將 top-1 錯誤率降低了 3.5%（表 2），這是由於成功降低了訓練錯誤（圖 4 右側對比左側）。這一比較驗證了殘差學習在極深系統上的有效性。

Last, we also note that the 18-layer plain/residual nets are comparably accurate (Table 2), but the 18-layer ResNet converges faster (Fig. 4 right *vs.* left). When the net is "not overly deep" (18 layers here), the current SGD solver is still able to find good solutions to the plain net. In this case, the ResNet eases the optimization by providing faster convergence at the early stage.

最後，我們還注意到 18 層的普通/殘差網路在準確度上相當（表 2），但 18 層的 ResNet 收斂速度更快（圖 4 右側對比左側）。當網路「不過於深」時（這裡是 18 層），目前的 SGD 求解器仍能為普通網路找到良好的解決方案。在這種情況下，ResNet 通過在早期階段提供更快的收斂來簡化優化過程。

model	top-1 err. top-1 錯誤	top-5 err. top-5 錯誤
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50 ResNet-50 ---	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Table 3: Error rates (%, **10-crop** testing) on ImageNet validation. VGG-16 is based on our test. ResNet-50/101/152 are of option B that only uses projections for increasing dimensions.

表 3: 在 ImageNet 驗證上的錯誤率（%，10-crop 測試）。VGG-16 基於我們的測試。ResNet-50/101/152 是選擇 B，只使用投影來增加維度。

method 方法	top-1 err. top-1 錯誤率	top-5 err. top-5 錯誤率
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13] PReLU-net [13]（參考文獻 13）	21.59	5.71
BN-inception [16] BN-inception [16]（參考文獻 16）	21.99	5.81
ResNet-34 B ResNet-34 B（ResNet-34 B）	21.84	5.71
ResNet-34 C ResNet-34 C（ResNet-34 C）	21.53	5.60
ResNet-50 ResNet-50（ResNet-50）	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4: Error rates (%) of **single-model** results on the ImageNet validation set (except † reported on the test set).

表 4：ImageNet 驗證集上單模型結果的錯誤率（百分比）（不包括 † 報告的測試集結果）。

method 方法	top-5 err. (test) top-5 錯誤率 (測試)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66

VGG [41] (v5)	6.8
PReLU-net [13] PReLU-net [13] (參考文獻 13)	4.94
BN-inception [16] BN-inception [16] (參考文獻 16)	4.82
ResNet (ILSVRC'15)	**3.57**

Table 5: Error rates (%) of **ensembles**. The top-5 error is on the test set of ImageNet and reported by the test server.

表 5: 集成的錯誤率 (%) 。top-5 錯誤率是基於 ImageNet 的測試集，並由測試伺服器報告。

Identity *vs.* Projection Shortcuts. We have shown that parameter-free, identity shortcuts help with training. Next we investigate projection shortcuts (Eqn.(2)). In Table 3 we compare three options: (A) zero-padding shortcuts are used for increasing dimensions, and all shortcuts are parameter-free (the same as Table 2 and Fig. 4 right); (B) projection shortcuts are used for increasing dimensions, and other shortcuts are identity; and (C) all shortcuts are projections.

身份 vs. 投影捷徑。我們已經展示了無參數的身份捷徑有助於訓練。接下來，我們研究投影捷徑 (方程式 (2))。在表 3 中，我們比較了三種選擇：(A) 零填充捷徑用於增加維度，且所有捷徑都是無參數的 (與表 2 和圖 4 右側相同) ；(B) 投影捷徑用於增加維度，其他捷徑為身份捷徑；以及 (C) 所有捷徑都是投影。

Table 3 shows that all three options are considerably better than the plain counterpart. B is slightly better than A. We argue that this is because the zero-padded dimensions in A indeed have no residual learning. C is marginally better than B, and we attribute this to the extra parameters introduced by many (thirteen) projection shortcuts. But the small differences among A/B/C indicate that projection shortcuts are not essential for addressing the degradation problem. So we do not use option C in the rest of this paper, to reduce memory/time complexity and model sizes. Identity shortcuts are particularly important for not increasing the complexity of the bottleneck architectures that are introduced below.

表 3 顯示，所有三個選項都明顯優於普通對應項。B 稍微優於 A。我們認為這是因為 A 中的零填充維度確實沒有殘餘學習。C 稍微優於 B，我們將此歸因於許多 (十三個) 投影捷徑引入的額外參數。然而，A/B/C 之間的小差異表明，投影捷徑對於解決降級問題並非必不可少。因此，我們在本文其餘部分中不使用選項 C，以降低記憶體/時間複雜度和模型大小。身份捷徑對於不增加下面介紹的瓶頸架構的複雜度尤其重要。

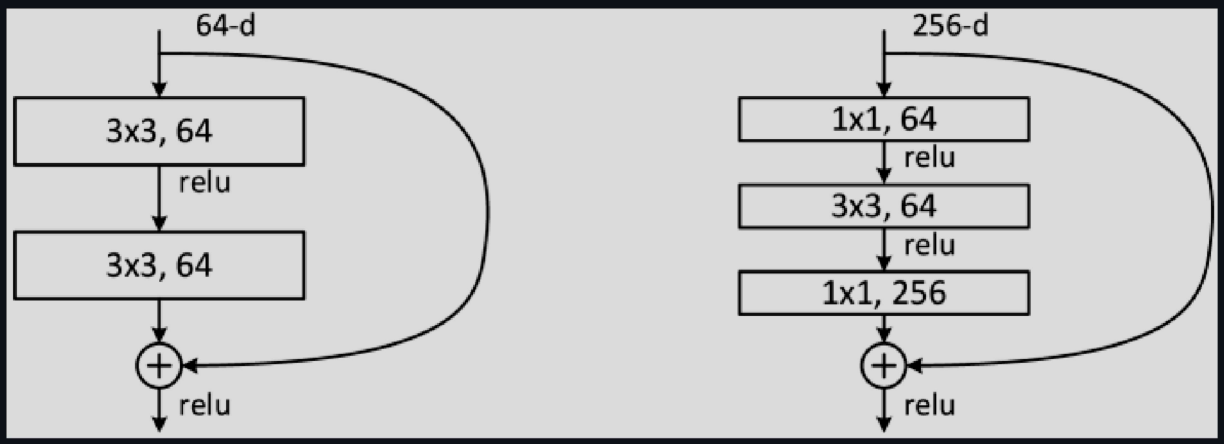

Figure 5: A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

圖 5：ImageNet 的更深殘差函數 $\mathcal{F}$ 。左側：如圖 3 中 ResNet-34 所示的建構塊 (在 56 × 56 特徵圖上) 。右側：ResNet-50/101/152 的"瓶頸"建構塊。

Deeper Bottleneck Architectures. Next we describe our deeper nets for ImageNet. Because of concerns on the training time that we can afford, we modify the building block as a *bottleneck* design[4]

[4]Deeper *non*-bottleneck ResNets (*e.g.*, Fig. 5 left) also gain accuracy from increased depth (as shown on CIFAR-10), but are not as economical as the bottleneck ResNets. So the usage of bottleneck designs is mainly due to practical considerations. We further note that the degradation problem of plain nets is also witnessed for the bottleneck designs.

更深層的非瓶頸 ResNet（例如圖 5 左）也能從增加的深度中獲得準確度（如 CIFAR-10 所示），但其經濟性不如瓶頸 ResNet。因此，瓶頸設計的使用主要是出於實際考量。我們進一步指出，普通網絡的降級問題在瓶頸設計中也有所見。

更深層瓶頸架構。接下來我們描述我們的更深層網絡用於 ImageNet。由於對我們能承受的訓練時間的顧慮，我們將建構模塊修改為瓶頸設計 [4]

. For each residual function $\mathcal{F}$, we use a stack of 3 layers instead of 2 (Fig. 5). The three layers are 1×1, 3×3, and 1×1 convolutions, where the 1×1 layers are responsible for reducing and then increasing (restoring) dimensions, leaving the 3×3 layer a bottleneck with smaller input/output dimensions. Fig. 5 shows an example, where both designs have similar time complexity.

。對於每個殘差函數 $\mathcal{F}$ ，我們使用 3 層堆疊而非 2 層（圖 5）。這三層分別是 1×1、3×3 和 1×1 的卷積，其中 1×1 層負責減少然後再增加（恢復）維度，留下 3×3 層作為具有較小輸入/輸出維度的瓶頸。圖 5 顯示了一個例子，其中兩種設計具有類似的時間複雜度。

The parameter-free identity shortcuts are particularly important for the bottleneck architectures. If the identity shortcut in Fig. 5 (right) is replaced with projection, one can show that the time complexity and model size are doubled, as the shortcut is connected to the two high-dimensional ends. So identity shortcuts lead to more efficient models for the bottleneck designs.

無參數的恒等捷徑對於瓶頸架構特別重要。如果圖 5（右）中的恒等捷徑被投影取代，可以顯示時間複雜度和模型大小都會翻倍，因為捷徑連接到兩個高維端。因此，恒等捷徑使瓶頸設計的模型更加高效。

50-layer ResNet: We replace each 2-layer block in the 34-layer net with this 3-layer bottleneck block, resulting in a 50-layer ResNet (Table 1). We use option B for increasing dimensions. This model has 3.8 billion FLOPs.

50 層 ResNet：我們將 34 層網路中的每個 2 層區塊替換為這個 3 層瓶頸區塊，從而得到 50 層 ResNet（表 1）。我們使用選項 B 來增加維度。此模型擁有 38 億 FLOPs。

101-layer and 152-layer ResNets: We construct 101-layer and 152-layer ResNets by using more 3-layer blocks (Table 1). Remarkably, although the depth is significantly increased, the 152-layer ResNet (11.3 billion FLOPs) still has *lower complexity* than VGG-16/19 nets (15.3/19.6 billion FLOPs).

101 層和 152 層 ResNet：我們通過使用更多的 3 層塊（表 1）來構建 101 層和 152 層 ResNet。值得注意的是，儘管深度顯著增加，152 層 ResNet（11.3 億 FLOPs）的複雜度仍低於 VGG-16/19 網絡（15.3/19.6 億 FLOPs）。

The 50/101/152-layer ResNets are more accurate than the 34-layer ones by considerable margins (Table 3 and 5). We do not observe the degradation problem and thus enjoy significant accuracy gains from considerably increased depth. The benefits of depth are witnessed for all evaluation metrics (Table 3 and 5).

50/101/152 層的 ResNet 比 34 層的模型在準確度上有顯著的提升（表 3 和 5）。我們未觀察到降級問題，因此從顯著增加的深度中獲得了顯著的準確度提升。深度的好處在所有評估指標中均有所體現（表 3 和 5）。

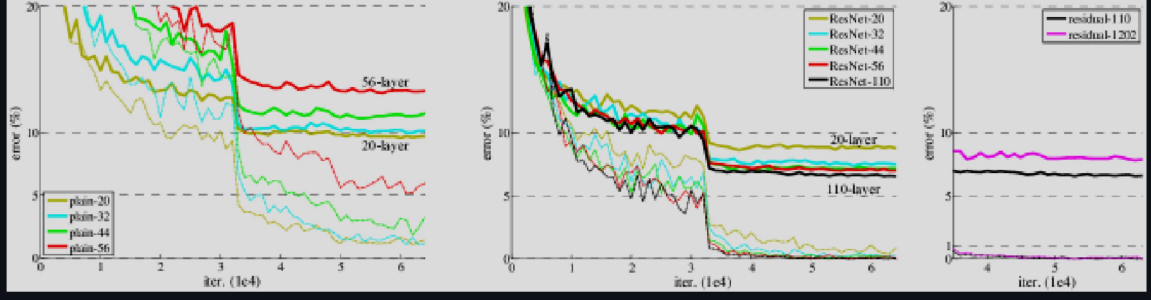

Figure 6: Training on **CIFAR-10**. Dashed lines denote training error, and bold lines denote testing error. **Left**: plain networks. The error of plain-110 is higher than 60% and not displayed. **Middle**: ResNets. **Right**: ResNets with 110 and 1202 layers.

圖 6：在 CIFAR-10 上的訓練。虛線表示訓練錯誤，粗線表示測試錯誤。左側：普通網絡。plain-110 的錯誤率高於 60%，因此未顯示。中間：ResNets。右側：擁有 110 層和 1202 層的 ResNets。

Comparisons with State-of-the-art Methods. In Table 5 we compare with the previous best single-model results. Our baseline 34-layer ResNets have achieved very competitive accuracy. Our 152-layer ResNet has a single-model top-5 validation error of 4.49%. This single-model result outperforms all previous ensemble results (Table 5). We combine six models of different depth to form an ensemble (only with two 152-layer ones at the time of submitting). This leads to **3.57%** top-5 error on the test set (Table 5). *This entry won the 1st place in ILSVRC 2015.*

與最先進方法的比較。在表 5 中，我們與先前最佳的單模型結果進行比較。我們的基線 34 層 ResNets 已達到非常具有競爭力的準確度。我們的 152 層 ResNet 在單模型的 top-5 驗證錯誤為 4.49%。這個單模型結果超越了所有先前的集成結果（表 5）。我們將六個不同深度的模型組合成一個集成（提交時只有兩個 152 層的模型）。這在測試集上導致了 3.57% 的 top-5 錯誤（表 5）。這一結果在 ILSVRC 2015 中獲得了第一名。

4.2 CIFAR-10 and Analysis 4.2 CIFAR-10 與分析

We conducted more studies on the CIFAR-10 dataset [20], which consists of 50k training images and 10k testing images in 10 classes. We present experiments trained on the training set and evaluated on the test set. Our focus is on the behaviors of extremely deep networks, but not on pushing the state-of-the-art results, so we intentionally use simple architectures as follows.

我們對 CIFAR-10 資料集[20]進行了更多的研究，該資料集包含 50k 訓練圖像和 10k 測試圖像，分為 10 個類別。我們展示了在訓練集上訓練並在測試集上評估的實驗。我們專注於極深網絡的行為，而不是推動最新的成果，因此我們故意使用以下簡單架構。

The plain/residual architectures follow the form in Fig. 3 (middle/right). The network inputs are 32×32 images, with the per-pixel mean subtracted. The first layer is 3×3 convolutions. Then we use a stack of $6n$ layers with 3×3 convolutions on the feature maps of sizes $\{32, 16, 8\}$ respectively, with $2n$ layers for each feature map size. The numbers of filters are $\{16, 32, 64\}$ respectively. The subsampling is performed by convolutions with a stride of 2. The network ends with a global average pooling, a 10-way fully-connected layer, and softmax. There are totally $6n$+2 stacked weighted layers. The following table summarizes the architecture:

純淨/殘差架構遵循圖 3（中/右）的形式。網絡輸入為 32 × 32 圖像，並去除每個像素的均值。第一層是 3 × 3 卷積。然後，我們使用一堆 $6n$ 層，對大小為 {32, 16, 8} 的特徵圖進行 3 × 3 卷積，每個特徵圖大小有 2 n 層。濾波器的數量分別為 {16, 32, 64} 。子採樣通過步幅為 2 的卷積進行。網絡以全局平均池化、10 路全連接層和 softmax 結束。總共有 6 n +2 層堆疊的加權層。下表總結了架構：

output map size 輸出地圖大小	32×32	16×16	8×8
# layers # 層	1+2n	2n	2n
# filters # 過濾器	16	32	64

When shortcut connections are used, they are connected to the pairs of 3×3 layers (totally $3n$ shortcuts). On this dataset we use identity shortcuts in all cases (*i.e.*, option A), so our residual models have exactly the same depth, width, and number of parameters as the plain counterparts.

當使用捷徑連接時，它們連接到 3 × 3 層的對（總共 $3n$ 個捷徑）。在這個數據集上，我們在所有情況下使用恒等捷徑（即選項 A），因此我們的殘差模型在深度、寬度和參數數量上完全與普通模型相同。

method 方法			error (%) 錯誤 (%)
Maxout [10]			9.38
NIN [25]			8.81
DSN [24]			8.22
	# layers # 層數	# params # 參數	
FitNet [35]	19	2.5M	8.39
Highway [42, 43] 高速公路 [42, 43]	19	2.3M	7.54 (7.72±0.16)
Highway [42, 43] 高速公路 [42, 43]	32	1.25M	8.80
ResNet ResNet (殘差網絡)	20	0.27M	8.75
ResNet ResNet (殘差網絡)	32	0.46M	7.51
ResNet ResNet (殘差網絡)	44	0.66M	7.17
ResNet ResNet (殘差網絡)	56	0.85M	6.97
ResNet ResNet (殘差網絡)	110	1.7M	**6.43** (6.61±0.16)
ResNet ResNet (殘差網絡)	1202	19.4M	7.93

Table 6: Classification error on the **CIFAR-10** test set. All methods are with data augmentation. For ResNet-110, we run it 5 times and show "best (mean±std)" as in [43].

表 6：CIFAR-10 測試集上的分類錯誤。所有方法均使用數據擴增。對於 ResNet-110，我們運行了 5 次，並顯示"最佳（均值 ± 標準差）"，如 [43] 所示。

We use a weight decay of 0.0001 and momentum of 0.9, and adopt the weight initialization in [13] and BN [16] but with no dropout. These models are trained with a mini-batch size of 128 on two GPUs. We start with a learning rate of 0.1, divide it by 10 at 32k and 48k iterations, and terminate training at 64k iterations, which is determined on a 45k/5k train/val split. We follow the simple data augmentation in [24] for training: 4 pixels are padded on each side, and a 32×32 crop is randomly sampled from the padded image or its horizontal flip. For testing, we only evaluate the single view of the original 32×32 image.

我們使用 0.0001 的權重衰減和 0.9 的動量，並採用[13]中的權重初始化和[16]中的批量正規化，但不使用 dropout。這些模型在兩個 GPU 上以 128 的小批量大小進行訓練。我們從 0.1 的學習率開始，在 32k 和 48k 次迭代時將其除以 10，並在 64k 次迭代時終止訓練，這是基於 45k/5k 的訓練/驗證拆分確定的。我們遵循[24]中的簡單數據增強進行訓練：每邊填充 4 個像素，並從填充的圖像或其水平翻轉中隨機抽取 32x32 的裁剪圖像。測試時，我們僅評估原始 32x32 圖像的單一視圖。

We compare $n = \{3, 5, 7, 9\}$, leading to 20, 32, 44, and 56-layer networks. Fig. 6 (left) shows the behaviors of the plain nets. The deep plain nets suffer from increased depth, and exhibit higher training error when going deeper. This phenomenon is similar to that on ImageNet (Fig. 4, left) and on MNIST (see [42]), suggesting that such an optimization difficulty is a fundamental problem.

我們比較了 $n = \{3, 5, 7, 9\}$，導致 20、32、44 和 56 層網絡。圖 6（左）顯示了普通網絡的行為。深層普通網絡在增加深度時會遭遇更高的訓練誤差。這一現象與 ImageNet（圖 4，左）和 MNIST（見 [42]）上的情況類似，表明這種優化困難是一個基本問題。

Fig. 6 (middle) shows the behaviors of ResNets. Also similar to the ImageNet cases (Fig. 4, right), our ResNets manage to overcome the optimization difficulty and demonstrate accuracy gains when the depth increases.

圖 6（中間）顯示了 ResNets 的行為。與 ImageNet 案例（圖 4，右側）類似，我們的 ResNets 成功克服了優化困難，並且在深度增加時展現了準確度的提升。

We further explore $n = 18$ that leads to a 110-layer ResNet. In this case, we find that the initial learning rate of 0.1 is slightly too large to start converging[5]

[5]With an initial learning rate of 0.1, it starts converging ($< 90\%$ error) after several epochs, but still reaches similar accuracy.

使用初始學習率 0.1，它在幾個時代後開始收斂（ $< 90\%$ 錯誤），但仍達到類似的準確度。

我們進一步探討 $n = 18$ 這導致了 110 層的 ResNet。在這種情況下，我們發現初始學習率 0.1 稍微過大，無法開始收斂 [5] 。

. So we use 0.01 to warm up the training until the training error is below 80% (about 400 iterations), and then go back to 0.1 and continue training. The rest of the learning schedule is as done previously. This 110-layer network converges well (Fig. 6, middle). It has *fewer* parameters than other deep and thin networks such as FitNet [35] and Highway [42] (Table 6), yet is among the state-of-the-art results (6.43%, Table 6).

。因此，我們使用 0.01 來預熱訓練，直到訓練錯誤低於 80%（約 400 次迭代），然後再回到 0.1 繼續訓練。其餘的學習計劃與之前所做的相同。這個 110 層的網絡收斂良好（圖 6，中間）。它比其他深且薄的網絡如 FitNet [35] 和 Highway [42]（表 6）具有更少的參數，卻仍然位於最先進的結果之一（6.43%，表 6）。

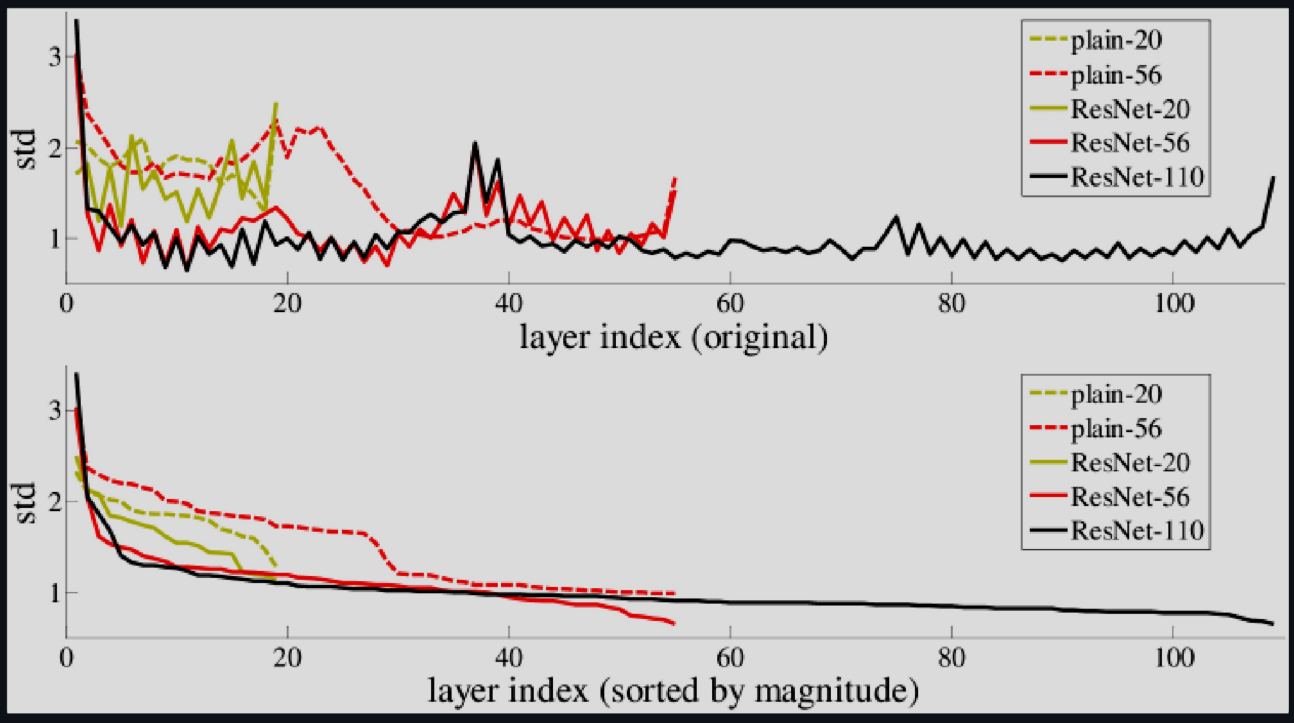

Figure 7: Standard deviations (std) of layer responses on CIFAR-10. The responses are the outputs of each 3×3 layer, after BN and before nonlinearity. **Top**: the layers are shown in their original order. **Bottom**: the responses are ranked in descending order.

圖 7：CIFAR-10 上層響應的標準差（std）。響應是每個 3×3 層在 BN 之後和非線性之前的輸出。上方：層按其原始順序顯示。下方：響應按降序排列。

Analysis of Layer Responses. Fig. 7 shows the standard deviations (std) of the layer responses. The responses are the outputs of each 3×3 layer, after BN and before other nonlinearity (ReLU/addition). For

ResNets, this analysis reveals the response strength of the residual functions. Fig. 7 shows that ResNets have generally smaller responses than their plain counterparts. These results support our basic motivation (Sec.3.1) that the residual functions might be generally closer to zero than the non-residual functions. We also notice that the deeper ResNet has smaller magnitudes of responses, as evidenced by the comparisons among ResNet-20, 56, and 110 in Fig. 7. When there are more layers, an individual layer of ResNets tends to modify the signal less.

層響應分析。圖 7 顯示了層響應的標準差（std）。響應是每個 3 × 3 層的輸出，在 BN 之後和其他非線性（ReLU/加法）之前。對於 ResNets，這個分析揭示了殘差函數的響應強度。圖 7 顯示 ResNets 的響應通常比其普通對應物小。這些結果支持我們的基本動機（第 3.1 節），即殘差函數可能通常比非殘差函數更接近於零。我們還注意到，較深的 ResNet 具有較小的響應幅度，這在圖 7 中的 ResNet-20、56 和 110 的比較中得到了證明。當層數增加時，ResNets 中的單個層對信號的修改傾向於更少。

Exploring Over 1000 layers. We explore an aggressively deep model of over 1000 layers. We set $n = 200$ that leads to a 1202-layer network, which is trained as described above. Our method shows *no optimization difficulty*, and this 10^3-layer network is able to achieve *training error* $< 0.1\%$ (Fig. 6, right). Its test error is still fairly good (7.93%, Table 6).

探索超過 1000 層。我們探索了一個深度達到 1000 層以上的激進模型。我們設定 $n = 200$ ，這導致了一個 1202 層的網絡，並按照上述描述進行訓練。我們的方法顯示出沒有優化困難，這個 10^3 層的網絡能夠實現訓練錯誤 $< 0.1\%$（圖 6，右側）。其測試錯誤仍然相當良好（7.93%，表 6）。

But there are still open problems on such aggressively deep models. The testing result of this 1202-layer network is worse than that of our 110-layer network, although both have similar training error. We argue that this is because of overfitting. The 1202-layer network may be unnecessarily large (19.4M) for this small dataset. Strong regularization such as maxout [10] or dropout [14] is applied to obtain the best results ([10, 25, 24, 35]) on this dataset. In this paper, we use no maxout/dropout and just simply impose regularization via deep and thin architectures by design, without distracting from the focus on the difficulties of optimization. But combining with stronger regularization may improve results, which we will study in the future.

但在這些極深層模型上仍然存在開放性問題。這個 1202 層的網絡測試結果比我們的 110 層網絡差，儘管兩者的訓練誤差相似。我們認為這是因為過擬合。對於這個小數據集來說，1202 層的網絡可能過於龐大（19.4M）。在這個數據集上，應用了強正則化方法如 maxout [10] 或 dropout [14] 以獲得最佳結果（[10, 25, 24, 35] ）。在本文中，我們不使用 maxout/dropout，而是通過深層而纖細的架構設計來簡單地施加正則化，並專注於優化困難。然而，與更強的正則化結合可能會改善結果，我們將在未來進行研究。

4.3 Object Detection on PASCAL and MS COCO

4.3 PASCAL 和 MS COCO 上的物件偵測

training data 訓練數據	07+12	07++12
test data 測試數據	VOC 07 test VOC 07 測試	VOC 12 test VOC 12 測試
VGG-16	73.2	70.4
ResNet-101	**76.4**	**73.8**

Table 7: Object detection mAP (%) on the PASCAL VOC 2007/2012 test sets using **baseline** Faster R-CNN. See also Table 11 and 11 for better results.

表 7：使用基線 Faster R-CNN 在 PASCAL VOC 2007/2012 測試集上的物件檢測 mAP（%）。另見表 11 和 11 以獲得更好的結果。

metric 指標	mAP@.5	mAP@[.5, .95]
VGG-16	41.5	21.2
ResNet-101	**48.4**	**27.2**

Table 8: Object detection mAP (%) on the COCO validation set using **baseline** Faster R-CNN. See also Table 9 for better results.

表 8：在 COCO 驗證集上使用基線 Faster R-CNN 進行物件偵測的 mAP（%）。另見表 9 以獲得更佳結果。

Our method has good generalization performance on other recognition tasks. Table 8 and 8 show the object detection baseline results on PASCAL VOC 2007 and 2012 [5] and COCO [26]. We adopt *Faster R-CNN* [32] as the detection method. Here we are interested in the improvements of replacing VGG-16 [41] with ResNet-101. The detection implementation (see appendix) of using both models is the same, so the gains can only be attributed to better networks. Most remarkably, on the challenging COCO dataset we obtain a 6.0% increase in COCO's standard metric (mAP@[.5, .95]), which is a 28% relative improvement. This gain is solely due to the learned representations.

我們的方法在其他識別任務上具有良好的泛化性能。表 8 顯示了在 PASCAL VOC 2007 和 2012 [5] 以及 COCO [26]上的物體檢測基準結果。我們採用 Faster R-CNN [32]作為檢測方法。在這裡，我們關心的是用 ResNet-101 取代 VGG-16 [41] 的改進。使用這兩個模型的檢測實現（見附錄）是相同的，因此這些增益只能歸因於更好的網絡。最顯著的是，在具有挑戰性的 COCO 數據集上，我們在 COCO 的標準指標（mAP@[.5, .95]）上取得了 6.0%的增長，這是 28%的相對改進。這一增益完全歸因於學到的表示。

Based on deep residual nets, we won the 1st places in several tracks in ILSVRC & COCO 2015 competitions: ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation. The details are in the appendix.

基於深度殘差網絡，我們在 ILSVRC 和 COCO 2015 比賽中的幾個領域中獲得了第一名：ImageNet 檢測、ImageNet 定位、COCO 檢測和 COCO 分割。詳細信息見附錄。

References

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[2] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[3] W. L. Briggs, S. F. McCormick, et al. *A Multigrid Tutorial*. Siam, 2000.

[4] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, pages 303–338, 2010.

[6] S. Gidaris and N. Komodakis. Object detection via a multi-region & semantic segmentation-aware cnn model. In *ICCV*, 2015.

[7] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[10] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *arXiv:1302.4389*, 2013.

[11] K. He and J. Sun. Convolutional neural networks at constrained time cost. In *CVPR*, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] S. Ioffe and C. Szegedy.
Batch normalization: Accelerating deep network training by reducing internal covariate shift.
In *ICML*, 2015.

[17] H. Jegou, M. Douze, and C. Schmid.
Product quantization for nearest neighbor search.
TPAMI, 33, 2011.

[18] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid.
Aggregating local image descriptors into compact codes.
TPAMI, 2012.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell.
Caffe: Convolutional architecture for fast feature embedding.
arXiv:1408.5093, 2014.

[20] A. Krizhevsky.
Learning multiple layers of features from tiny images.
Tech Report, 2009.

[21] A. Krizhevsky, I. Sutskever, and G. Hinton.
Imagenet classification with deep convolutional neural networks.
In *NIPS*, 2012.

[22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel.
Backpropagation applied to handwritten zip code recognition.
Neural computation, 1989.

[23] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller.
Efficient backprop.
In *Neural Networks: Tricks of the Trade*, pages 9–50. Springer, 1998.

[24] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu.
Deeply-supervised nets. *arXiv:1409.5185*, 2014.

[25] M. Lin, Q. Chen, and S. Yan. Network in network.
arXiv:1312.4400, 2013.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick.
Microsoft COCO: Common objects in context. In *ECCV*. 2014.

[27] J. Long, E. Shelhamer, and T. Darrell.
Fully convolutional networks for semantic segmentation.
In *CVPR*, 2015.

[28] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio.
On the number of linear regions of deep neural networks.
In *NIPS*, 2014.

[29] V. Nair and G. E. Hinton.
Rectified linear units improve restricted boltzmann machines.

In *ICML*, 2010.

[30] F. Perronnin and C. Dance.
Fisher kernels on visual vocabularies for image categorization.
In *CVPR*, 2007.

[31] T. Raiko, H. Valpola, and Y. LeCun.
Deep learning made easier by linear transformations in perceptrons.
In *AISTATS*, 2012.

[32] S. Ren, K. He, R. Girshick, and J. Sun.
Faster R-CNN: Towards real-time object detection with region proposal networks.
In *NIPS*, 2015.

[33] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun.
Object detection networks on convolutional feature maps.
arXiv:1504.06066, 2015.

[34] B. D. Ripley. *Pattern recognition and neural networks*.
Cambridge university press, 1996.

[35] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio.
Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al.
Imagenet large scale visual recognition challenge.
arXiv:1409.0575, 2014.

[37] A. M. Saxe, J. L. McClelland, and S. Ganguli.
Exact solutions to the nonlinear dynamics of learning in deep linear neural networks.
arXiv:1312.6120, 2013.

[38] N. N. Schraudolph.
Accelerated gradient descent by factor-centering decomposition.
Technical report, 1998.

[39] N. N. Schraudolph. Centering neural network gradient factors.
In *Neural Networks: Tricks of the Trade*, pages 207–226.
Springer, 1998.

[40] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun.
Overfeat: Integrated recognition, localization and detection using convolutional networks.
In *ICLR*, 2014.

[41] K. Simonyan and A. Zisserman.
Very deep convolutional networks for large-scale image recognition.
In *ICLR*, 2015.

[42] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv:1505.00387*, 2015.

[43] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. *1507.06228*, 2015.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[45] R. Szeliski. Fast surface interpolation using hierarchical basis functions. *TPAMI*, 1990.

[46] R. Szeliski. Locally adapted hierarchical basis preconditioning. In *SIGGRAPH*, 2006.

[47] T. Vatanen, T. Raiko, H. Valpola, and Y. LeCun. Pushing stochastic gradient towards second-order methods–backpropagation learning with transformations in nonlinearities. In *Neural Information Processing*, 2013.

[48] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[49] W. Venables and B. Ripley. Modern applied statistics with s-plus. 1999.

[50] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In *ECCV*, 2014.

Appendix A Object Detection Baselines

附錄 A 物件偵測基準

In this section we introduce our detection method based on the baseline Faster R-CNN [32] system. The models are initialized by the ImageNet classification models, and then fine-tuned on the object detection data. We have experimented with ResNet-50/101 at the time of the ILSVRC & COCO 2015 detection competitions.

在這一節中，我們介紹了基於基線 Faster R-CNN [32] 系統的檢測方法。模型由 ImageNet 分類模型初始化，然後在物體檢測數據上進行微調。我們在 ILSVRC 和 COCO 2015 檢測比賽期間，對 ResNet-50/101 進行了實驗。

Unlike VGG-16 used in [32], our ResNet has no hidden fc layers. We adopt the idea of "Networks on Conv feature maps" (NoC) [33] to address this issue. We compute the full-image shared conv feature maps using those layers whose strides on the image are no greater than 16 pixels (*i.e.*, conv1, conv2_ x, conv3_x, and conv4_x, totally 91 conv layers in ResNet-101; Table 1). We consider these layers as analogous to the 13 conv layers in VGG-16, and by doing so, both ResNet and VGG-16 have conv feature maps of the same total stride (16 pixels). These layers are shared by a region proposal network (RPN, generating 300 proposals) [32] and a Fast R-CNN detection network [7]. RoI pooling [7] is performed before conv5_1. On this RoI-pooled feature, all layers of conv5_x and up are adopted for each region, playing the roles of VGG-16's fc layers. The final classification

layer is replaced by two sibling layers (classification and box regression [7]).

與[32]中使用的 VGG-16 不同，我們的 ResNet 沒有隱藏的 fc 層。我們採用了「卷積特徵圖上的網絡」(NoC) [33]的思想來解決這個問題。我們使用那些對圖像的步幅不超過 16 像素的層來計算完整圖像共享的卷積特徵圖（即 conv1、conv2_x、conv3_x 和 conv4_x，共 91 個卷積層在 ResNet-101 中；表 1）。我們將這些層視為類似於 VGG-16 中的 13 個卷積層，這樣，ResNet 和 VGG-16 擁有相同總步幅的卷積特徵圖（16 像素）。這些層由區域提議網絡（RPN，生成 300 個提議）[32]和 Fast R-CNN 檢測網絡[7]共享。在 conv5_1 之前執行 RoI 池化[7]。在這個 RoI 池化的特徵上，每個區域都使用 conv5_x 及以上的所有層，擔任 VGG-16 的 fc 層的角色。最終的分類層被兩個兄弟層（分類和邊框回歸[7]）取代。

For the usage of BN layers, after pre-training, we compute the BN statistics (means and variances) for each layer on the ImageNet training set. Then the BN layers are fixed during fine-tuning for object detection. As such, the BN layers become linear activations with constant offsets and scales, and BN statistics are not updated by fine-tuning. We fix the BN layers mainly for reducing memory consumption in Faster R-CNN training.

對於 BN 層的使用，預訓練後，我們在 ImageNet 訓練集上計算每一層的 BN 統計數據（均值和方差）。然後，在對物體檢測進行微調時，BN 層會被固定。因此，BN 層變成具有固定偏移量和縮放的線性激活函數，且 BN 統計數據不會在微調過程中更新。我們主要固定 BN 層是為了減少 Faster R-CNN 訓練中的記憶體消耗。

PASCAL VOC

Following [7, 32], for the PASCAL VOC 2007 *test* set, we use the 5k *trainval* images in VOC 2007 and 16k *trainval* images in VOC 2012 for training ("07+12"). For the PASCAL VOC 2012 *test* set, we use the 10k *trainval+test* images in VOC 2007 and 16k *trainval* images in VOC 2012 for training ("07++12"). The hyper-parameters for training Faster R-CNN are the same as in [32]. Table 8 shows the results. ResNet-101 improves the mAP by > 3% over VGG-16. This gain is solely because of the improved features learned by ResNet.

根據 [7, 32]，對於 PASCAL VOC 2007 測試集，我們使用 VOC 2007 中的 5k 訓練驗證圖像和 VOC 2012 中的 16k 訓練驗證圖像進行訓練（"07+12"）。對於 PASCAL VOC 2012 測試集，我們使用 VOC 2007 中的 10k 訓練驗證+測試圖像和 VOC 2012 中的 16k 訓練驗證圖像進行訓練（"07++12"）。Faster R-CNN 的訓練超參數與 [32] 中相同。表 8 顯示了結果。ResNet-101 比 VGG-16 提高了 > 3% 的 mAP。這一增益完全是因為 ResNet 學習到的改進特徵。

MS COCO

The MS COCO dataset [26] involves 80 object categories. We evaluate the PASCAL VOC metric (mAP @ IoU = 0.5) and the standard COCO metric (mAP @ IoU = .5:.05:.95). We use the 80k images on the train set for training and the 40k images on the val set for evaluation. Our detection system for COCO is similar to that for PASCAL VOC. We train the COCO models with an 8-GPU implementation, and thus the RPN step has a mini-batch size of 8 images (*i.e.*, 1 per GPU) and the Fast R-CNN step has a mini-batch size of 16 images. The RPN step and Fast R-CNN step are both trained for 240k iterations with a learning rate of 0.001 and then for 80k iterations with 0.0001.

MS COCO 資料集 [26] 涉及 80 個物體類別。我們評估 PASCAL VOC 指標（mAP @ IoU = 0.5）和標準 COCO 指標（mAP @ IoU = .5:.05:.95）。我們使用訓練集中的 80k 圖像進行訓練，並使用驗證集中的 40k 圖像進行評估。我們的 COCO 偵測系統類似於 PASCAL VOC。我們用 8 GPU 實現訓練 COCO 模型，因此 RPN 步驟的迷你批量大小為 8 張圖像（即每 GPU 1 張），Fast R-CNN 步驟的迷你批量大小為 16 張圖像。RPN 步驟和 Fast R-CNN 步驟都訓練了 240k 次迭代，學習率為 0.001，然後再訓練 80k 次迭代，學習率為 0.0001。

Table 8 shows the results on the MS COCO validation set. ResNet-101 has a 6% increase of mAP@[.5, .95] over VGG-16, which is a 28% relative improvement, solely contributed by the features learned by the better network. Remarkably, the mAP@[.5, .95]'s absolute increase (6.0%) is nearly as big as mAP@.5's (6.9%). This sug-

gests that a deeper network can improve both recognition and localization.

表 8 顯示了在 MS COCO 驗證集上的結果。ResNet-101 的 mAP@[.5, .95] 比 VGG-16 增加了 6%，這是 28% 的相對改進，完全是由更佳網絡學到的特徵貢獻的。值得注意的是，mAP@[.5, .95] 的絕對增量（6.0%）幾乎與 mAP@.5 的增量（6.9%）相當。這表明，更深層的網絡可以改進識別和定位能力。

Appendix B Object Detection Improvements

附錄 B 物件偵測改進

For completeness, we report the improvements made for the competitions. These improvements are based on deep features and thus should benefit from residual learning.

為了完整性，我們報告了比賽中所做的改進。這些改進基於深度特徵，因此應該從殘差學習中受益。

training data 訓練數據	COCO train COCO 訓練		COCO trainval COCO 訓練驗證	
test data 測試數據	COCO val COCO 驗證		COCO test-dev COCO 測試-開發	
mAP	@.5	@[.5, .95]	@.5	@[.5, .95]
baseline Faster R-CNN (VGG-16) **基線 Faster R-CNN (VGG-16)**	41.5	21.2		
baseline Faster R-CNN (ResNet-101) **基線 Faster R-CNN (ResNet-101)**	48.4	27.2		
+box refinement +框選精煉	49.9	29.9		
+context +上下文	51.1	30.0	53.3	32.2
+multi-scale testing +多尺度測試	53.8	32.5	**55.7**	**34.9**
ensemble 集成			**59.0**	**37.4**

Table 9: Object detection improvements on MS COCO using Faster R-CNN and ResNet-101.

表 9：使用 Faster R-CNN 和 ResNet-101 在 MS COCO 上的物件偵測改進。

system 系統	**net 網絡**	**data 數據**	**mAP**	**areo**	**bike**	**bird**	**boat**	**bot-tle**	**bus 公車**	**car 車**	**cat 貓**	**chair 椅子**	**cow 牛**	**tab le 表格**	**dog**	**horse**	**mbike**	**per-son**	**plant**	**sheep**	**sofa**	**tra**
baseline 基準線	VGG-16	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.
baseline 基準線	ResNet-101	07+12	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	**89.8**	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.
baseline+++ 基線+++	ResNet-101	COCO+07+12	**85.6**	**90.0**	**89.6**	**87.8**	**80.8**	**76.1**	**89.9**	**89.9**	89.6	**75.5**	**90.0**	**80.7**	**89.6**	**90.3**	**89.1**	**88.7**	**65.4**	**88.1**	**85.6**	**89.**

Table 10: Detection results on the PASCAL VOC 2007 test set. The baseline is the Faster R-CNN system. The system “baseline+++” include box refinement, context, and multi-scale testing in Table 9.

表 10：在 PASCAL VOC 2007 測試集上的檢測結果。基準線為 Faster R-CNN 系統。系統「baseline+++」包括表 9 中的框精緻、上下文和多尺度測試。

system 系統	net 網絡	data 數據	mAP	areo	bike	bird	boat	bottle	bus 公車	car 車	cat 貓	chair 椅子	cow 牛	table 表格	dog	horse	mbike	person	plant	sheep	sofa	tr
baseline 基準線	VGG-16	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81
baseline 基準線	ResNet-101	07++12	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84
baseline+++ 基線+++	ResNet-101	COCO+07++12	**83.8**	**92.1**	**88.4**	**84.8**	**75.9**	**71.4**	**86.3**	**87.8**	**94.2**	**66.8**	**89.4**	**69.2**	**93.9**	**91.9**	**90.9**	**89.6**	**67.9**	**88.2**	**76.8**	**90**

Table 11: Detection results on the PASCAL VOC 2012 test set (http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=4). The baseline is the Faster R-CNN system. The system "baseline+++" include box refinement, context, and multi-scale testing in Table 9.

表 11：PASCAL VOC 2012 測試集的檢測結果（http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=4）。基線為 Faster R-CNN 系統。系統"baseline+++"包括框體精煉、上下文和多尺度測試，見表 9。

MS COCO

Box refinement. Our box refinement partially follows the iterative localization in [6]. In Faster R-CNN, the final output is a regressed box that is different from its proposal box. So for inference, we pool a new feature from the regressed box and obtain a new classification score and a new regressed box. We combine these 300 new predictions with the original 300 predictions. Non-maximum suppression (NMS) is applied on the union set of predicted boxes using an IoU threshold of 0.3 [8], followed by box voting [6]. Box refinement improves mAP by about 2 points (Table 9).

框架細化。我們的框架細化部分遵循了[6]中的迭代定位。在 Faster R-CNN 中，最終輸出的是一個與提議框不同的回歸框。因此，在推斷時，我們從回歸框中池化出一個新的特徵，並獲得新的分類分數和新的回歸框。我們將這 300 個新預測與原始的 300 個預測結合起來。對預測框的聯合集合應用非最大抑制（NMS），使用 0.3 的 IoU 閾值[8]，然後進行框投票[6]。框架細化使 mAP 提高了約 2 個點（表 9）。

Global context. We combine global context in the Fast R-CNN step. Given the full-image conv feature map, we pool a feature by global Spatial Pyramid Pooling [12] (with a "single-level" pyramid) which can be implemented as "RoI" pooling using the entire image's bounding box as the RoI. This pooled feature is fed into the post-RoI layers to obtain a global context feature. This global feature is concatenated with the original per-region feature, followed by the sibling classification and box regression layers. This new structure is trained end-to-end. Global context improves mAP@.5 by about 1 point (Table 9).

全球上下文。我們在 Fast R-CNN 步驟中結合了全球上下文。給定完整圖像的卷積特徵圖，我們通過全球空間金字塔池化 [12]（使用"單層"金字塔）來池化特徵，這可以作為"RoI"池化來實現，使用整個圖像的邊界框作為 RoI。這個池化特徵被輸入到後 RoI 層，以獲得全球上下文特徵。這個全球特徵與原始每區域特徵連接，然後進行兄弟分類和框回歸層。這個新結構是端到端訓練的。全球上下文將 mAP@.5 提高了約 1 點（表 9）。

Multi-scale testing. In the above, all results are obtained by single-scale training/testing as in [32], where the image's shorter side is $s = 600$ pixels. Multi-scale training/testing has been developed in [12, 7] by selecting a scale from a feature pyramid, and in [33] by using maxout layers. In our current implementation, we have performed multi-scale *testing* following [33]; we have not performed multi-scale training because of limited time. In addition, we have performed multi-scale testing only for the Fast R-CNN step (but not yet for the RPN step). With a trained model, we compute conv feature maps on an image pyramid, where the image's shorter

sides are $s \in \{200, 400, 600, 800, 1000\}$. We select two adjacent scales from the pyramid following [33]. RoI pooling and subsequent layers are performed on the feature maps of these two scales [33], which are merged by maxout as in [33]. Multi-scale testing improves the mAP by over 2 points (Table 9).

多尺度測試。在上述情況下，所有結果都是通過單尺度訓練/測試獲得的，如[32]所示，其中圖像的較短邊為 $s = 600$ 像素。多尺度訓練/測試在[12, 7]中通過從特徵金字塔中選擇一個尺度進行開發，在[33]中通過使用 maxout 層進行開發。在我們目前的實現中，我們按照[33]進行了多尺度測試；由於時間有限，我們尚未進行多尺度訓練。此外，我們僅對 Fast R-CNN 步驟進行了多尺度測試（但尚未對 RPN 步驟進行測試）。使用訓練好的模型，我們在圖像金字塔上計算卷積特徵圖，其中圖像的較短邊為 $s \in \{200, 400, 600, 800, 1000\}$。我們按照[33]從金字塔中選擇兩個相鄰的尺度。對這兩個尺度的特徵圖進行 RoI 池化和後續層操作[33]，這些特徵圖通過 maxout 進行合併，如[33]所示。多尺度測試使 mAP 提高了超過 2 點（表 9）。

Using validation data. Next we use the 80k+40k trainval set for training and the 20k test-dev set for evaluation. The test-dev set has no publicly available ground truth and the result is reported by the evaluation server. Under this setting, the results are an mAP@.5 of 55.7% and an mAP@[.5, .95] of 34.9% (Table 9). This is our single-model result.

使用驗證數據。接下來，我們使用 80k+40k 的 trainval 集進行訓練，並使用 20k 的 test-dev 集進行評估。test-dev 集沒有公開的真實標籤，結果由評估伺服器報告。在這種設置下，結果是 mAP@.5 為 55.7%，mAP@[.5, .95]為 34.9%（表 9）。這是我們的單模型結果。

Ensemble. In Faster R-CNN, the system is designed to learn region proposals and also object classifiers, so an ensemble can be used to boost both tasks. We use an ensemble for proposing regions, and the union set of proposals are processed by an ensemble of per-region classifiers. Table 9 shows our result based on an ensemble of 3 networks. The mAP is 59.0% and 37.4% on the test-dev set. *This result won the 1st place in the detection task in COCO 2015.*

集成。在 Faster R-CNN 中，系統被設計為學習區域提議和物體分類器，因此可以使用集成來提升這兩個任務。我們使用集成來提出區域，並且提議的聯集由每區域分類器的集成進行處理。表 9 顯示了我們基於 3 個網絡集成的結果。mAP 在 test-dev 集合上為 59.0% 和 37.4%。這一結果在 COCO 2015 的檢測任務中獲得了第一名。

PASCAL VOC

We revisit the PASCAL VOC dataset based on the above model. With the single model on the COCO dataset (55.7% mAP@.5 in Table 9), we fine-tune this model on the PASCAL VOC sets. The improvements of box refinement, context, and multi-scale testing are also adopted. By doing so we achieve 85.6% mAP on PASCAL VOC 2007 (Table 11) and 83.8% on PASCAL VOC 2012 (Table 11)[6]

[6] http://host.robots.ox.ac.uk:8080/anonymous/3OJ4OJ.html, submitted on 2015-11-26.

http://host.robots.ox.ac.uk:8080/anonymous/3OJ4OJ.html，提交日期為 2015-11-26。

我們根據上述模型重新檢視了 PASCAL VOC 數據集。使用在 COCO 數據集上的單一模型（表 9 中為 55.7% mAP@.5），我們在 PASCAL VOC 數據集上對這個模型進行了微調。還採用了框體細化、上下文以及多尺度測試的改進。這樣我們在 PASCAL VOC 2007（表 11）上達到了 85.6% mAP，在 PASCAL VOC 2012（表 11）上達到了 83.8% mAP [6]

. The result on PASCAL VOC 2012 is 10 points higher than the previous state-of-the-art result [6].

. 在 PASCAL VOC 2012 上的結果比之前的最新技術成果高出 10 分 [6]。

ImageNet Detection ImageNet 偵測

	val2	test 測試
GoogLeNet [44] (ILSVRC'14)	**-**	**43.9**
our single model (ILSVRC'15) 我們的單一模型 (ILSVRC'15)	60.5	58.8
our ensemble (ILSVRC'15) 我們的集成模型 (ILSVRC'15)	**63.6**	**62.1**

Table 12: Our results (mAP, %) on the ImageNet detection dataset. Our detection system is Faster R-CNN [32] with the improvements in Table 9, using ResNet-101.

表 12: 我們在 ImageNet 偵測數據集上的結果 (mAP, %)。我們的偵測系統是 Faster R-CNN [32]，使用了表 9 中的改進，並且使用了 ResNet-101。

The ImageNet Detection (DET) task involves 200 object categories. The accuracy is evaluated by mAP@.5. Our object detection algorithm for ImageNet DET is the same as that for MS COCO in Table 9. The networks are pre-trained on the 1000-class ImageNet classification set, and are fine-tuned on the DET data. We split the validation set into two parts (val1/val2) following [8]. We fine-tune the detection models using the DET training set and the val1 set. The val2 set is used for validation. We do not use other ILSVRC 2015 data. Our single model with ResNet-101 has 58.8% mAP and our ensemble of 3 models has 62.1% mAP on the DET test set (Table 12). *This result won the 1st place in the ImageNet detection task in ILSVRC 2015*, surpassing the second place by **8.5 points** (absolute).

ImageNet 檢測（DET）任務涉及 200 個物件類別。準確度通過 mAP@.5 來評估。我們的 ImageNet DET 物件檢測演算法與 MS COCO 在表 9 中的演算法相同。這些網絡在 1000 類別的 ImageNet 分類集上進行預訓練，並在 DET 資料上進行微調。我們根據 [8] 將驗證集分為兩部分（val1/val2）。我們使用 DET 訓練集和 val1 集來微調檢測模型。val2 集用於驗證。我們不使用其他 ILSVRC 2015 資料。我們的單一模型 ResNet-101 在 DET 測試集上的 mAP 為 58.8%，而我們的 3 模型集成在 DET 測試集上的 mAP 為 62.1%（表 12）。這一結果在 ILSVRC 2015 的 ImageNet 檢測任務中獲得了第 1 名，超過第二名 8.5 分（絕對值）。

Appendix C ImageNet Localization

附錄 C ImageNet 定位

LOC method 方法	LOC network 網路	testing 測試	LOC error LOC 錯誤 on GT CLS 在 GT CLS 上	classification 分類 network 網路	top-5 LOC error 前五名 LOC 錯誤 on predicted CLS 在預測 CLS 上
VGG's [41] VGG 的 [41]	VGG-16	1-crop	33.1 [41] 33.1 [41]		
RPN	ResNet-101	1-crop 1-裁剪	13.3		
RPN	ResNet-101	dense 密集	11.7		
RPN	ResNet-101	dense 密集		ResNet-101	14.4
RPN+RCNN	ResNet-101	dense 密集		ResNet-101	**10.6**
RPN+RCNN	ensemble 集成	dense 密集		ensemble 集成	**8.9**

Table 13: Localization error (%) on the ImageNet validation. In the column of "LOC error on GT class" ([41]), the ground truth class is used. In the "testing" column, "1-crop" denotes testing on a center crop of 224×224

pixels, “dense” denotes dense (fully convolutional) and multi-scale testing.

表 13：ImageNet 驗證集上的定位誤差（%）。在「LOC error on GT class」的欄位中（[41]），使用的是真實類別。在「testing」欄位中，「1-crop」表示在 224 × 224 像素的中心裁剪上進行測試，「dense」表示密集（完全卷積）和多尺度測試。

The ImageNet Localization (LOC) task [36] requires to classify and localize the objects. Following [40, 41], we assume that the image-level classifiers are first adopted for predicting the class labels of an image, and the localization algorithm only accounts for predicting bounding boxes based on the predicted classes. We adopt the “per-class regression” (PCR) strategy [40, 41], learning a bounding box regressor for each class. We pre-train the networks for ImageNet classification and then fine-tune them for localization. We train networks on the provided 1000-class ImageNet training set.

ImageNet 位置標定（LOC）任務 [36] 需要對物體進行分類和定位。根據 [40, 41]，我們假設首先採用圖像級分類器來預測圖像的類別標籤，定位算法僅僅根據預測的類別來預測邊界框。我們採用“每類回歸”（PCR）策略 [40, 41]，為每個類別學習一個邊界框回歸器。我們先對網絡進行 ImageNet 分類的預訓練，然後對其進行定位的微調。我們在提供的 1000 類 ImageNet 訓練集上訓練網絡。

Our localization algorithm is based on the RPN framework of [32] with a few modifications. Unlike the way in [32] that is category-agnostic, our RPN for localization is designed in a *per-class* form. This RPN ends with two sibling 1×1 convolutional layers for binary classification (*cls*) and box regression (*reg*), as in [32]. The *cls* and *reg* layers are both in a *per-class* from, in contrast to [32]. Specifically, the *cls* layer has a 1000-d output, and each dimension is *binary logistic regression* for predicting being or not being an object class; the *reg* layer has a 1000×4-d output consisting of box regressors for 1000 classes. As in [32], our bounding box regression is with reference to multiple translation-invariant “anchor” boxes at each position.

我們的定位演算法基於 [32] 的 RPN 框架，並進行了一些修改。與 [32] 中的無類別感知方法不同，我們的定位 RPN 設計為每個類別的形式。這個 RPN 以兩個兄弟 1 × 1 卷積層結束，用於二分類 (cls) 和框回歸 (reg)，與 [32] 相同。cls 和 reg 層均為每個類別的形式，與 [32] 不同。具體來說，cls 層的輸出為 1000 維，每個維度為二元邏輯回歸，用於預測是否為物體類別；reg 層的輸出為 1000 × 4 維，由 1000 類別的框回歸器組成。與 [32] 相同，我們的邊界框回歸是參考每個位置的多個平移不變的「錨點」框。

As in our ImageNet classification training (Sec. 3.4), we randomly sample 224×224 crops for data augmentation. We use a mini-batch size of 256 images for fine-tuning. To avoid negative samples being dominate, 8 anchors are randomly sampled for each image, where the sampled positive and negative anchors have a ratio of 1:1 [32]. For testing, the network is applied on the image fully-convolutionally.

如同我們在 ImageNet 分類訓練（第 3.4 節）中所做的，我們隨機抽取 224 × 224 的裁剪進行數據增強。我們使用 256 張圖像的迷你批次大小進行微調。為了避免負樣本佔主導地位，每張圖像隨機抽取 8 個錨點，其中抽取的正負錨點比例為 1:1 [32]。在測試時，網絡以完全卷積的方式應用於圖像。

method 方法	top-5 localization err top-5 本地化錯誤	
	val 置信度	test 測試
OverFeat [40] (ILSVRC’13)	30.0	29.9
GoogLeNet [44] (ILSVRC’14)	-	26.7
VGG [41] (ILSVRC’14)	26.9	25.3
ours (ILSVRC’15) ours (ILSVRC’15) 我們（ILSVRC’15）	**8.9**	**9.0**

Table 14: Comparisons of localization error (%) on the ImageNet dataset with state-of-the-art methods.

Table 14:Comparisons of localization error (%) on the ImageNet dataset with state-of-the-art methods. 表 14：在 ImageNet 數據集上與最先進方法的定位錯誤 (%) 比較

Table 13 compares the localization results. Following [41], we first perform "oracle" testing using the ground truth class as the classification prediction. VGG's paper [41] reports a center-crop error of 33.1% (Table 13) using ground truth classes. Under the same setting, our RPN method using ResNet-101 net significantly reduces the center-crop error to 13.3%. This comparison demonstrates the excellent performance of our framework. With dense (fully convolutional) and multi-scale testing, our ResNet-101 has an error of 11.7% using ground truth classes. Using ResNet-101 for predicting classes (4.6% top-5 classification error, Table 5), the top-5 localization error is 14.4%.

表 13 比較了定位結果。根據[41]，我們首先使用真實類別作為分類預測進行"oracle"測試。VGG 的論文[41]報告了使用真實類別的中心裁剪錯誤為 33.1%（表 13）。在相同設置下，我們使用 ResNet-101 網絡的 RPN 方法顯著降低了中心裁剪錯誤至 13.3%。這一比較顯示了我們框架的優異性能。使用密集（全卷積）和多尺度測試，我們的 ResNet-101 在使用真實類別時的錯誤為 11.7%。使用 ResNet-101 進行類別預測（4.6% top-5 分類錯誤，表 5），top-5 定位錯誤為 14.4%。

The above results are only based on the *proposal network* (RPN) in Faster R-CNN [32]. One may use the *detection network* (Fast R-CNN [7]) in Faster R-CNN to improve the results. But we notice that on this dataset, one image usually contains a single dominate object, and the proposal regions highly overlap with each other and thus have very similar RoI-pooled features. As a result, the image-centric training of Fast R-CNN [7] generates samples of small variations, which may not be desired for stochastic training. Motivated by this, in our current experiment we use the original R-CNN [8] that is RoI-centric, in place of Fast R-CNN.

上述結果僅基於 Faster R-CNN [32] 中的提議網絡 (RPN)。可以使用 Faster R-CNN 中的檢測網絡 (Fast R-CNN [7]) 來改善結果。但我們注意到，在這個數據集上，一張圖片通常包含單一主導物體，且提議區域彼此高度重疊，因此具有非常相似的 RoI-pooled 特徵。因此，Fast R-CNN [7] 的圖像中心訓練生成的小變化樣本，可能不適合隨機訓練。受到這個啟發，在我們目前的實驗中，我們使用原始的 R-CNN [8]，這是 RoI-中心的，來取代 Fast R-CNN。

Our R-CNN implementation is as follows. We apply the per-class RPN trained as above on the training images to predict bounding boxes for the ground truth class. These predicted boxes play a role of class-dependent proposals. For each training image, the highest scored 200 proposals are extracted as training samples to train an R-CNN classifier. The image region is cropped from a proposal, warped to 224×224 pixels, and fed into the classification network as in R-CNN [8]. The outputs of this network consist of two sibling fc layers for *cls* and *reg*, also in a per-class form. This R-CNN network is fine-tuned on the training set using a mini-batch size of 256 in the RoI-centric fashion. For testing, the RPN generates the highest scored 200 proposals for each predicted class, and the R-CNN network is used to update these proposals' scores and box positions.

我們的 R-CNN 實現如下。我們在訓練圖像上應用如上所述的每類 RPN 以預測對應真實類別的邊界框。這些預測框充當類別依賴的提案。對於每張訓練圖像，提取最高分的 200 個提案作為訓練樣本，用於訓練 R-CNN 分類器。從提案中裁剪圖像區域，變形為 224 x 224 像素，並如 R-CNN [8] 所述輸入到分類網絡中。該網絡的輸出由兩個兄弟 fc 層組成，分別用於 cls 和 reg，也以每類的形式存在。這個 R-CNN 網絡在訓練集上進行微調，使用 256 的迷你批次大小以 RoI 為中心的方式進行。測試時，RPN 為每個預測類別生成最高分的 200 個提案，R-CNN 網絡用於更新這些提案的分數和框位置。

This method reduces the top-5 localization error to 10.6% (Table 13). This is our single-model result on the validation set. Using an ensemble of networks for both classification and localization, we achieve a top-5 localization error of 9.0% on the test set. This number significantly outperforms the ILSVRC 14 results (Table 14), showing a 64% relative reduction of error. *This result won the 1st place in the ImageNet localiza-*

tion task in ILSVRC 2015.

這種方法將前五名定位錯誤率降低至 10.6%（表 13）。這是我們在驗證集上的單模型結果。使用分類和定位的網絡集成，我們在測試集上達到了 9.0%的前五名定位錯誤率。這一數字顯著超過了 ILSVRC 14 的結果（表 14），顯示出錯誤相對減少了 64%。這一結果在 ILSVRC 2015 的 ImageNet 定位任務中獲得了第一名。