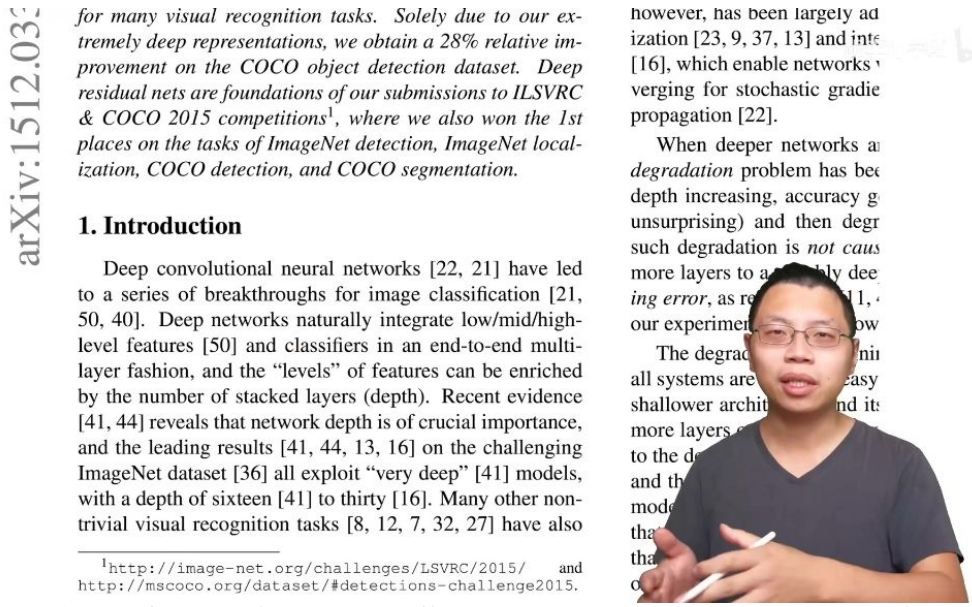
一、深度残差网络 ***00:00***

1. 介绍 00:14

1）深度卷积神经网络的优势 00:22



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

1. Introduction

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 non-trivial visual recognition tasks [8, 12, 7, 32, 27] have also

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

- **多层次特征整合**：深度网络能自然整合低/中/高级特征，通过增加层数(深度)可以丰富特征层次
- **端到端学习**: 采用端到端多层方式同时学习特征和分类器
- **深度重要性**：　网络深度对性能至关重要，ImageNet上领先结果都使用16-30层的"非常深"模型

2）网络深度增加带来的问题 00:40

- **核心问题**: 简单地堆叠更多层是否能持续提升网络性能
- **早期障碍**: 梯度消失/爆炸问题阻碍深层网络收敛

3）梯度消失与爆炸问题及其解决方案 01:02

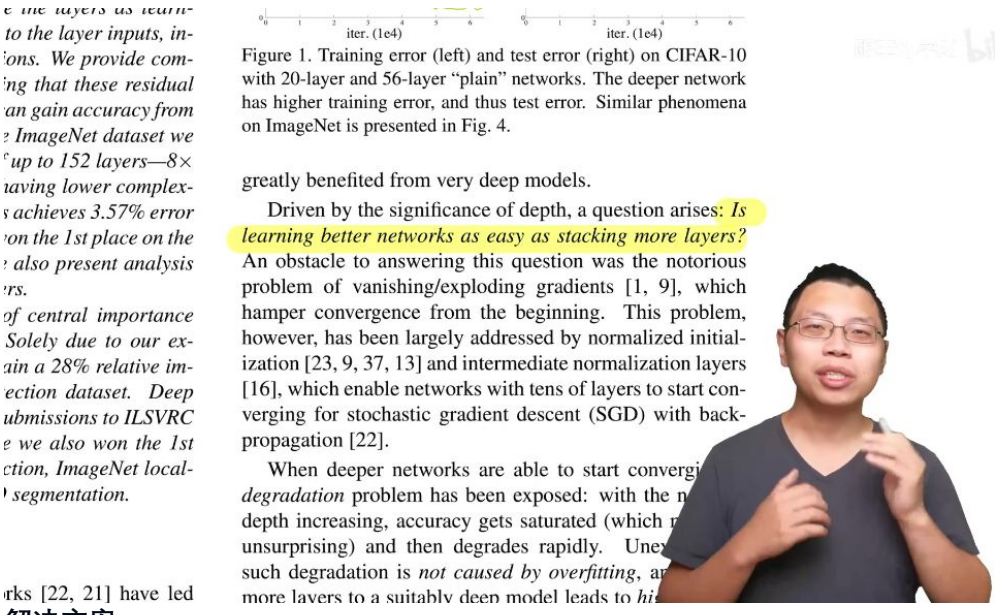


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

greatly benefited from very deep models.

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 back-propagation [22].

- **解决方案**:

- **规范化初始化**: 权重初始化时控制大小范围
- **中间规范化层**: 如Batch Normalization层，控制各层输出的均值和方差

- **效果**: 使数十层的网络能够通过SGD+反向传播收敛

4）网络变深时性能变差的问题 01:48

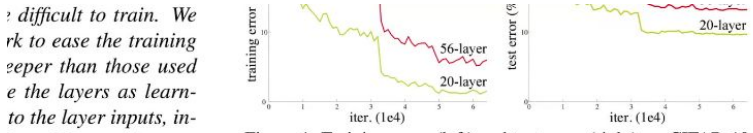
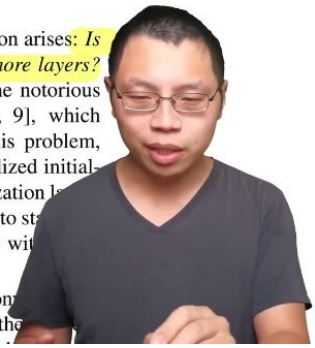


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

greatly benefited from very deep models.

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 back propagation [22].

When deeper networks are able to start converging, a *degradation* problem has been exposed: with the



- **现象**: 当网络深度增加时，准确率会先饱和后快速下降
- **非过拟合**: 训练误差和测试误差同时变差，排除了过拟合的可能性
- **实验证据**: 56层"普通"网络比20层网络在CIFAR-10上表现出更高的训练和测试误差

5）深度模型精度变差的原因分析 02:11

- **优化困难**: 深层网络系统比浅层网络更难优化
- **理论解存在**: 深层网络可以通过构造恒等映射(identity mapping)来保持浅层网络的性能

6）深层网络与浅层网络的关系 02:57

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.

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

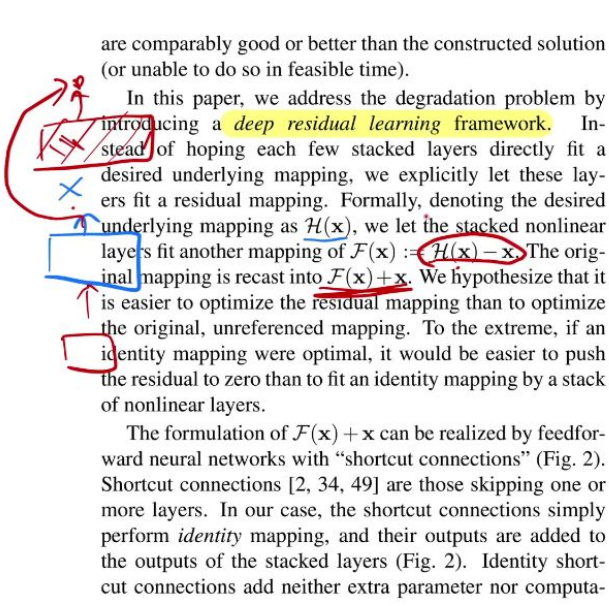
1

- **构造解原理**: 深层网络中新加层可设为恒等映射，其他层复制浅层网络权重
- **实际困难**: 虽然理论上存在这种解，但SGD优化器难以找到这样的解

7）深度残差学习框架的引入 04:24

- **核心思想**: 显式构造恒等映射，确保深层网络不会比浅层网络更差
- **命名**: 称为深度残差学习框架(Deep Residual Learning Framework)

8）残差映射的学习方法 04:50



are comparably good or better than the constructed solution (or unable to do so in feasible time).

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.

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 computa-

it is applicable in other vis

2. Related Work

Residual Representation
[18] is a representation tha
with respect to a dictionar
formulated as a probabilis
of them are powerful shall
trieval and classification [
encoding residu ctors
tive than enc nal
In low-l n
ing Partial q
Multigrid me fo
lems at multi , w
sible for th uti
scale
co
s

- **数学表达**:
  - 期望映射:$H(x)$
  - 残差映射:$F(x)$: = $H(x)$ - x
  - 最终输出:$F(x)$ + x
- **优化优势**: 优化残差映射比直接优化原始映射更容易

9）残差学习模块的构建 06:33

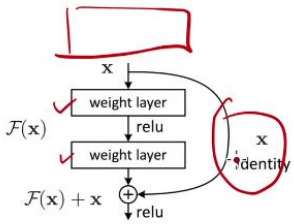
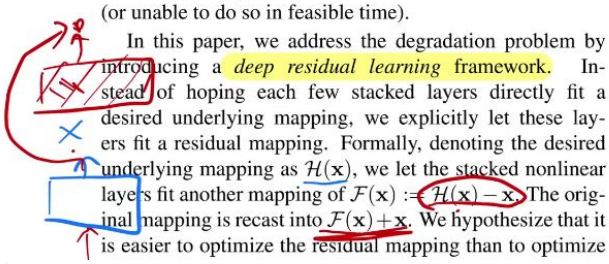


Figure 2. Residual learning: a building block.



are comparably good or better than the constructed solution (or unable to do so in feasible time).

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

ImageNet *test* set, and *w*
2015 classification compe
resentations also have exc
on other recognition tasks
1st places on: ImageNet c
COCO detection, and CO
COCO 2015 competitions.
the residual princi
it is appli r vis

2. Rela

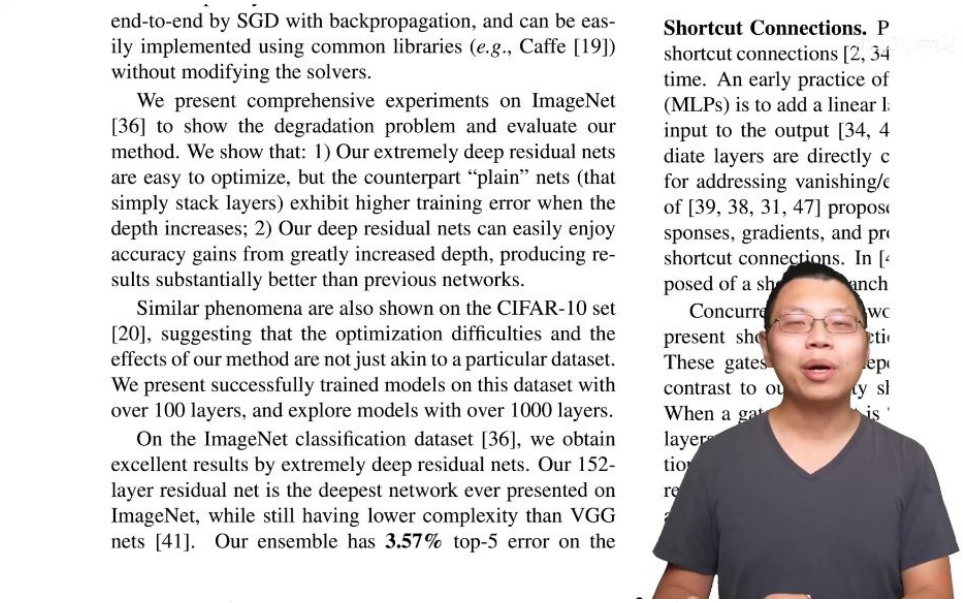
Residual R tation
[18] is ha
wi
f

- **实现方式**: 使用带有"快捷连接"(shortcut connections)的前馈神经网络
- **连接特性**:
  - 跳过一层或多层
  - 执行恒等映射
  - 输出与堆叠层输出相加

10）残差网络的优化与实现 07:42

- **实现优势**:
  - 不增加额外参数
  - 不增加计算复杂度
  - 可使用现有框架(如Caffe)轻松实现
- **训练方式**: 仍可通过SGD+反向传播进行端到端训练

11）实验结果与模型评估 09:02



end-to-end by SGD with backpropagation, and can be easily implemented using common libraries (*e.g.*, Caffe [19]) without modifying the solvers.

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.

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.

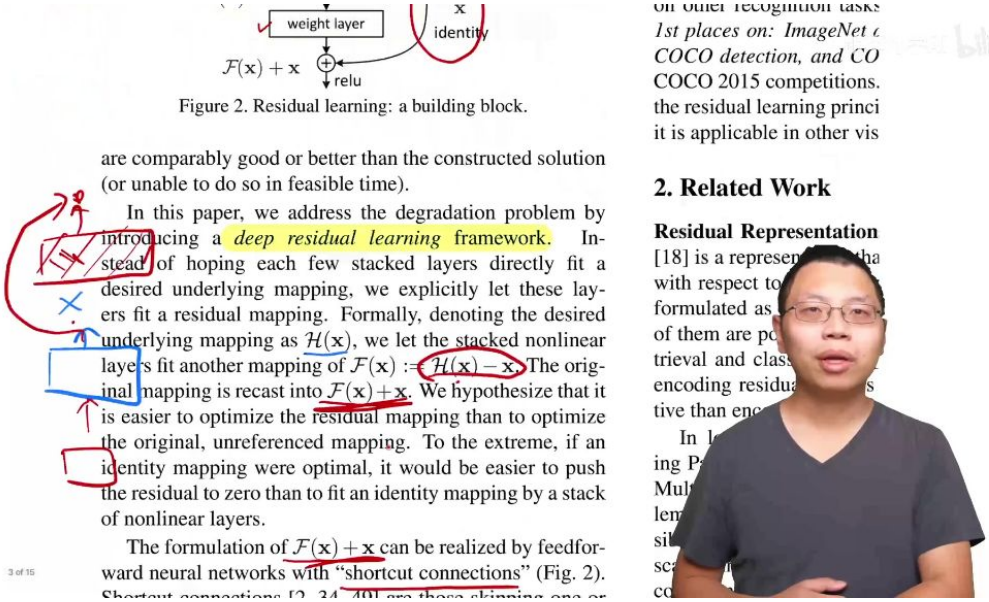
On the ImageNet classification dataset [36], we obtain excellent results by extremely deep residual nets. Our 152-layer residual net is the deepest network ever presented on ImageNet, while still having lower complexity than VGG nets [41]. Our ensemble has **3.57%** top-5 error on the

Shortcut Connections. P
shortcut connections [2, 34
time. An early practice of
(MLPs) is to add a linear l
input to the output [34, 4
diate layers are directly c
for addressing vanishing/e
of [39, 38, 31, 47] propos
sponses, gradients, and pr
shortcut connections. In [
posed of a sh anch
Concurr wo
present sh eti
These gates ep
contrast to ou y sl
When a ga is
layer
tio
re

- **优化优势**: 极深的残差网络容易优化，而普通网络随深度增加训练误差升高
- **深度增益**: 残差网络能从深度增加中获得准确率提升
- **模型规模**:
  - CIFAR-10上训练超过100层的模型
  - 探索超过1000层的模型
- **竞赛成绩**:
  - ImageNet上152层残差网络(当时最深)
  - 集成模型top-5错误率3.57%
  - 在ImageNet检测、定位和COCO检测、分割任务中获得第一名

2. 相关工作 10:44

1）快捷连接的理论与实践 10:46



- **残差映射原理**：通过让堆叠的非线性层拟合残差映射$F(x)$: = $H(x)$ - x，将原始映射重构为$F(x)$ + x，比直接优化原始未参考映射更容易
- **极端情况处理**：若恒等映射最优，将残差推至零比通过非线性层堆叠拟合恒等映射更简单
- **实现方式**：通过带有"快捷连接"的前馈神经网络实现，这些连接会跳过一层或多层

2）机器学习中的残差迭代与boosting 11:02

- **历史渊源**：残差概念在机器学习中早有应用，如线性模型最早通过残差迭代求解，梯度提升(GBDT)通过残差学习网络并叠加弱分类器
- **领域差异**：计算机视觉领域(CVPR)较少回顾机器学习中的残差应用，更关注视觉特定方法

3）经典论文技术借鉴与融合 12:17

- **技术传承**：ResNet并非首个提出快捷连接的文章，如Highway Networks已有类似结构但更复杂(带门控函数)
- **创新本质**：经典工作的价值常在于巧妙组合现有技术解决关键问题，而非完全原创
- **学术规范**：应诚实标注前人工作(to our best knowledge)，若遗漏可后续补充，但需避免故意不引近期相似工作

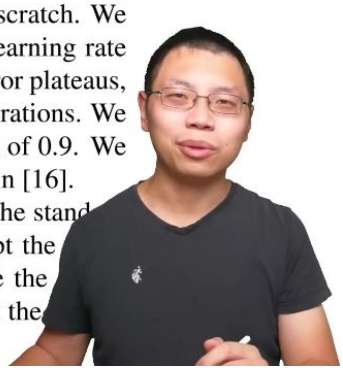
4）残差连接处理输入输出形状不等的方法 15:09

- **零填充法**：在输入输出上添加额外零值使形状匹配
- **投影法**：
  - 全连接层：通过线性变换调整维度
  - 卷积层：使用1×1卷积核调整通道数，步幅为2时可同步减半高宽

5）实验细节：数据增强与权重初始化 16:22

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].

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



- **数据增强**:
  - 短边随机采样于[256,480]范围进行尺度增强
  - 从图像或其水平翻转中随机采样224×224区域
  - 逐像素均值减法处理
  - 使用标准颜色增强(调整亮度/饱和度等)
- **网络配置**:
  - 每个卷积后立即应用批归一化(BN)
  - 权重初始化遵循文献[13]方法
  - 不使用dropout(因无全连接层)

6）实验细节：模型训练与测试方法 19:13

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].

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 $\{224, 256, 384, 480, 640\}$).

4. Experiments

4.1. ImageNet Classification

We evaluate our method on the ImageNet 2012 cation dataset [36] that consists of 1000 classes. The

- **训练参数**:
  - SGD优化器，批量大小256
  - 初始学习率0.1，误差平台时除以10
  - 训练60×10^4次迭代(建议改用epoch计数更合理)
  - 权重衰减0.0001，动量0.9
- **测试方法**:

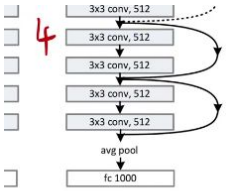
- 标准10-crop测试(从测试图像采样10个子图预测取平均)
- 多尺度测试(短边调整至{224,256,384,480,640}等尺寸)
- 全卷积形式计算得分

- **工程考量**：多尺度测试虽提升精度但计算代价高，实际部署需权衡精度与成本

3. 实验 20:23

1）ImageNet分类 20:34

- 普通网络设计 20:44



res for ImageNet. **Left**: the
OPs) as a reference. **Mid-**
r layers (3.6 billion FLOPs).
arameter layers (3.6 billion
dimensions. **Table 1** shows

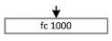
We evaluate our method on the ImageNet 2012 classification dataset [36] that consists of 1000 classes. The models are trained on the 1.28 million training images, and evaluated on the 50k validation images. We also obtain a final result on the 100k test images, reported by the test server. We evaluate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

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 d
tailed architectures.

The results in Table 2 show that the deeper 34-layer
net has higher validation error than the shallower 18
plain net. To reveal the reasons, in Fig. 4 (left) we c
pare their training/validation errors during the training
cedure. We have observed the degradation probl

4

- 5 of 15
- **数据集构成**：使用ImageNet　2012分类数据集，包含1000个类别，128万训练图像，5万验证图像和10万测试图像
- **评估指标**：同时评估top-1和top-5错误率
- **网络结构**：
  - 18层和34层普通网络
  - 34层网络结构如图3中间所示
  - 18层网络结构类似但层数较少



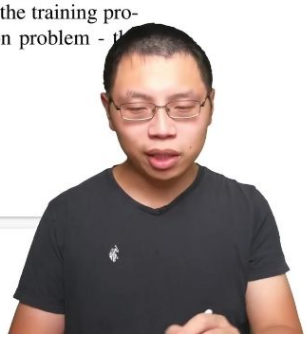
for ImageNet. **Left**: the
) as a reference. **Mid-**
ers (3.6 billion FLOPs).
neter layers (3.6 billion
ensions. **Table 1** shows

We evaluate both top-1 and top-5 error rates.

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.

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 -

4



  -
  - **深度问题**：34层普通网络比18层网络验证错误率更高
  - **训练观察**：图4左侧展示了训练过程中的训练/验证误差对比，观察到退化问题
- 残差网络设计 21:14

18-layer	34-layer	50-layer	101-layer	152-layer
7×7, 64, stride 2				
3×3 max pool, stride 2				
$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
average pool, 1000-d fc, softmax				
1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

ding blocks are shown in brackets (see also Fig. 5), with the numbers of blocks stacked
_1, and conv5_1 with a stride of 2.

- **版本差异**：包含ResNet-18/34/50/101/152五个版本
- **共同结构**：
  - 第一个7×7卷积相同
  - 最后的全局池化和1000维全连接层相同
- **核心差异**：中间重复的卷积层结构不同

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	

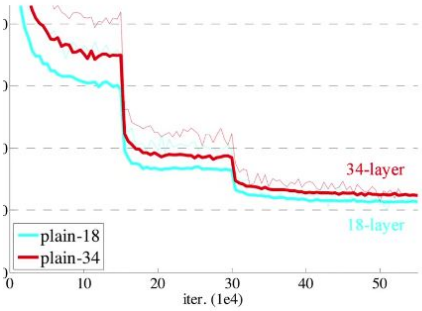
tures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with th
rmed by conv3_1, conv4_1, and conv5_1 with a stride of 2.

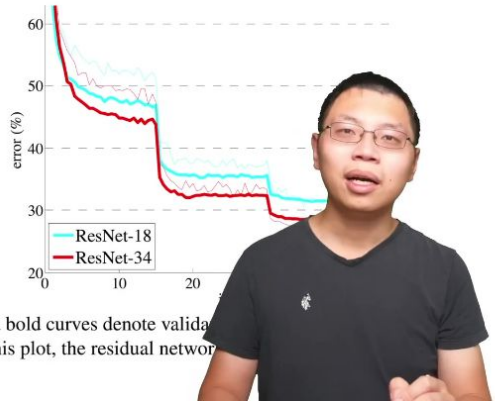
- **34层计算**：3+4+6+3=16个块，每个块2层→32层，加上首尾→34层
- **18层计算**：所有块数变为2→8×2=16层，加上首尾→18层
- **设计原理**：这些层数设计是作者调参得出的超参数

- 训练过程 25:11

conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 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

itectures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of block erformed by conv3_1, conv4_1, and conv5_1 with a stride of 2.





ning on **ImageNet**. Thin curves denote training error, and bold curves denote valida 8 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networ unterparts.

- **学习率调整**：采用×0.1的学习率衰减策略
- **收敛速度**：有残差连接的34层比无残差连接的34层收敛更快
- **训练技巧**：建议晚些调整学习率，让模型有更多微调时间

- 残差连接不同情况的分析 28:25

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	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.

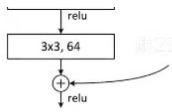


Figure 5. A deeper residu building block (on 56×56 34. Right: a "bottleneck" b

parameter-free we investiga compare thr for increasin free (the sam tion shortcuts shortcut Ta ter th argu ind B

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60

7 of 15

- **三种方案**：
  - A：维度增加时补零
  - B：维度增加时使用1×1卷积投影
  - C：所有连接都使用投影
- **选择依据**：B方案比A略好，C方案计算代价太高，最终选择B方案

- 更深的残差网络 30:10

err. (test)
7.32
5.66
6.8
4.94
4.82
3.57

error is on the

e 2), resulting
Fig. 4 right *vs.*
ess of residual

l/residual nets
-layer ResNet
the net is "not
) solver is still
l this case, the

the bottleneck architectures that are introduced below.

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]. 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.

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-dimensio ends. So identity shortcuts lead to more efficient mo for the bottleneck designs.

50-layer ResNet: We replace each 2-layer block

[4]Deeper *non*-bottleneck ResNets (*e.g.*, Fig. 5 left) also gain

- **设计原理**：使用1×1-3×3-1×1的三层结构替代原来的两层结构
- **计算优势**：虽然通道数增加4倍，但计算复杂度相近
- **50层设计**：在34层基础上加入bottleneck，通道数从64→256→512→1024→2048

me	output size	18-layer	34-layer	50-layer	101-layer	152-layer
	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 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

mageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of bl
onv3_1, conv4_1, and conv5_1 with a stride of 2.

- **101层变化**：将6个块变为23个块
- **152层变化**：将4个块变为8个，23个块变为36个

- 实验结果 35:22

ResNet-34 C	21.53	5.60
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).

method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [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.

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 effectiveness of residual

- **性能提升**：从ResNet-34的21.84% top-1错误率降到ResNet-152的19.38%
- **比赛结果**：使用多裁剪融合后达到3.57% top-5错误率
- **比较优势**：比之前最好结果提升1.6个百分点

2）CIFAR-10数据集上的实验 36:25

- CIFAR-10数据集及实验目的 36:29
  - **数据集特点**： CIFAR-10是一个小型数据集，包含32×32的小尺寸图片，共10类约5万个样本
  - **实验目的**： 通过简单数据集验证残差网络的有效性，便于观察深层网络训练过程中的现象

simple architectures as follows.

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:

output map size	32×32	16×16	8×8
# layers	1+2n	2n	2n
# filters	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),

- 网络架构设计 36:45
  - **基础结构**:
    - 输入32×32图像，减去像素均值
    - 首层为3×3卷积
    - 堆叠6n层3×3卷积，特征图尺寸依次为{32,16,8}，每尺寸对应2n层
    - 滤波器数量依次为{16,32,64}
    - 通过步长为2的卷积进行下采样
    - 全局平均池化+10维全连接+softmax
  - **参数总量**: 总计6n+2个权重层

- **残差连接**: 在3×3卷积层之间添加3n个恒等映射的shortcut连接
- 训练细节与参数设置 36:59

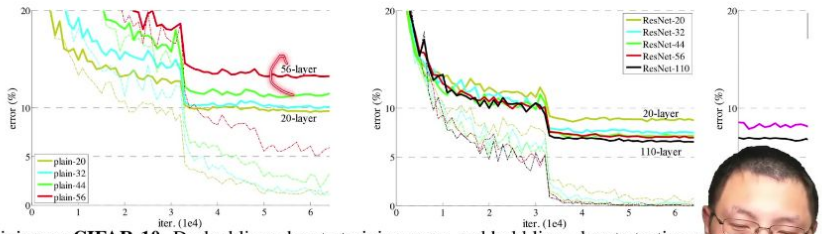
Highway [42, 43]	32	1.25M	8.80
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43** (6.61±0.16)
ResNet	1202	19.4M	7.93

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].

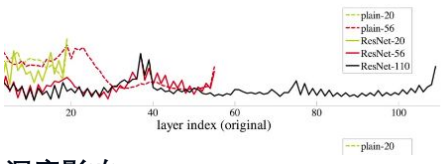
so our residual models have exactly the same depth, width, and number of parameters as the plain counterparts.

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 mi batch size of 128 on two GPUs. We start with a learn rate of 0.1, divide it by 10 at 32k and 48k iterations terminate training at 64k iterations, which is determin a 45k/5k train/val split. We follow the simple data aug

- AR-10 dataset
- **优化参数**:
  - 权重衰减0.0001，动量0.9
  - 批大小128，双GPU训练
  - 初始学习率0.1，在32k和48k迭代时除以10
  - 64k迭代终止训练
- **数据增强**: 采用[24]的简单增强方法，每边填充4像素
- **模型对比**:
  - ResNet-20: 0.27M参数，8.75%错误率
  - ResNet-110: 1.7M参数，6.43%(6.61±0.16)%错误率
  - ResNet-1202: 19.4M参数，7.93%错误率
- 残差连接的作用与效果 38:13



5. Training on **CIFAR-10**. Dashed lines denote training error, and bold lines denote testing er
-110 is higher than 60% and not displayed. **Middle**: ResNets. **Right**: ResNets with 110 and 12



training da
test dat
VGG-
ResNe

Table 7. Object
2007/2012 test

- **深度影响**:
  - 普通网络：56层比20层精度更差
  - 残差网络：深度增加仍保持精度提升
- **残差机制**:
  - 当新增层无用时，残差连接使其输出趋近0

- 实际表现：前100层有效，后900层基本不更新
- 即使训练1202层网络，过拟合现象也不严重

- 层响应的标准差分析 38:50

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

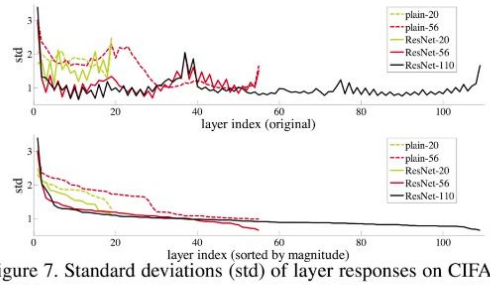


Figure 7. Standard deviations (std) of layer responses on CIFAR-10. The responses are the outputs of each 3×3 layer, after BN and before nonlinearity. **Top**: the layers are shown in their original order. **Bottom**: the responses are ranked in descending order.

training data	07+
test data	VOC 0
VGG-16	73.
ResNet-101	**76.**

Table 7. Object detection mAP 2007/2012 test sets using **baselir** ble 10 and 11 for be

metric	@
VGG-1	
ResNet-1	4

Table 8. Object detec P (
using **baseline**

have simi
overfittir
large (l
such as

networks such as FitNet [35] and Highway [42] (Table 6), yet is among the state-of-the-art results (6.43%, Table 6).

- **Analysis of Layer Responses**. Fig. 7 shows the standard
- **分析方法**: 测量各3×3卷积层在BN后、非线性前的输出标准差
- **关键发现**:
  - 无残差网络：深层响应标准差较大
  - 残差网络：深层响应标准差显著减小
- **理论解释**:
  - 残差连接使无效层的输出自动趋近0
  - 但需注意：有无残差连接本质是不同的模型架构，直接比较需谨慎

3）目标检测 40:02

- 目标检测数据集及检测方法 40:05
  - **数据集表现**：在PASCAL VOC 2007/2012和COCO数据集上使用Faster R-CNN作为检测方法，将VGG-16替换为ResNet-101后获得显著提升
  - **性能提升**：在COCO数据集上标准指标(mAP@[.5,.95])提高了6.0%，相对提升达28%
  - **方法对比**：两种模型的检测实现相同，性能提升完全归因于ResNet-101更好的网络结构
- 目标检测mAP介绍 40:14
  - **mAP定义**：目标检测中最常见的精度指标，表示预测框的平均精度
  - **评估标准**：数值越高越好，在不同IoU阈值下进行评估
  - **性能对比**：从21.2提升到27.2（VGG-16到ResNet-101），从70.4提升到73.8（PASCAL VOC 2012测试集）
- 检测结果讨论 40:46

test data	COCO val		COCO test-dev	
mAP	@.5	@[.5, .95]	@.5	@[.5, .95]
baseline Faster R-CNN (VGG-16)	41.5	21.2		
baseline Faster R-CNN (ResNet-101)	48.4	27.2		
+box refinement	49.9	29.9		
+context	51.1	30.0	53.3	32.2
+multi-scale testing	53.8	32.5	**55.7**	**34.9**
ensemble			**59.0**	**37.4**

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

net	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse
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
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
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**

Detection results on the PASCAL VOC 2007 test set. The baseline is the Faster R-CNN syste
refinement, context, and multi-scale testing in Table 9.

net	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table
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
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
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**

- **改进方法**：在ResNet-101基础上依次添加box refinement、context和multi-scale testing
- **性能提升**：mAP@[.5,.95]从27.2逐步提升到32.5（单模型）和34.9（集成模型）
- **作者观点**：这些结果属于锦上添花，即使不放这些结果也不会影响文章的核心贡献

- 目标检测基线介绍 42:35
  - **方法细节**：使用ImageNet预训练模型初始化，在检测数据上微调
  - **架构差异**：与VGG-16不同，ResNet没有全连接层，采用"Networks on Conv feature maps"(NoC)方法
  - **特征共享**：使用步长≤16像素的卷积层（共91层）计算全图共享的卷积特征图
- 梯度相关解释 44:37
  - **梯度传播**：传统深层网络梯度计算为$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g(x))}{\partial g(x)} \cdot \frac{\partial g(x)}{\partial x}$
  - **梯度消失**：多层连乘导致梯度值越来越小，特别是当梯度值本身较小时
  - **残差优势**：残差连接使梯度计算变为$\frac{\partial}{\partial x}[f(g(x)) + g(x)] = \frac{\partial f(g(x))}{\partial g(x)} + \frac{\partial g(x)}{\partial x}$，避免了纯乘法带来的梯度消失
- 残差连接的优势 45:52
  - **训练动态**：即使深层网络也能保持较大梯度，使优化过程持续进行
  - **收敛特性**：SGD优化的精髓在于保持足够大的梯度，避免过早收敛到次优点
  - **实际表现**：在CIFAR-10上训练1202层网络也能取得不错效果，过拟合问题不严重
- 残差连接与梯度消失问题 46:57
  - **数学解释**：残差连接将梯度乘法变为加法，即使深层网络也能保持有效梯度
  - **训练效率**：浅层网络的梯度直接传递，不受深层网络影响
  - **实际效果**：使得超深层网络（如1000+层）的训练成为可能
- 残差连接与模型收敛 48:50

conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax			
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9

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

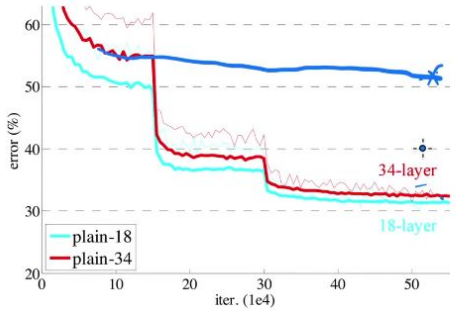
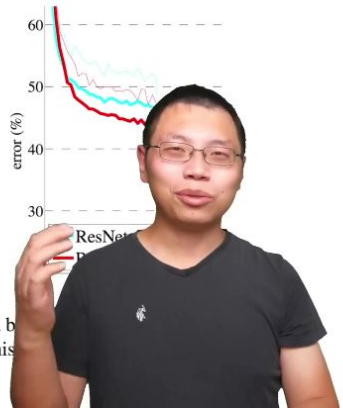


Figure 4. Training on **ImageNet**. Thin curves denote training error, and b
networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this
their plain counterparts.



- **响应分析**：ResNet的层响应标准差普遍小于普通网络，说明残差函数更接近零
- **优化特性**：深层ResNet的响应幅度更小，表明网络确实利用了残差学习的优势
- **训练观察**：18层普通网络和ResNet准确率相当，但ResNet收敛更快

4. 结束 50:11

1）残差连接与模型复杂度 50:18

- **复杂度降低机制**：残差连接的加入虽然增加了参数数量，但通过结构设计显著降低了模型的"内在复杂度"。这种设计使得网络更容易找到简单的解（如某些层权重为零的identity映射）。
- **训练引导作用**：理论上深层网络可以学习到identity映射，但实际训练中难以自发实现。残差连接通过显式结构引导网络朝这个方向优化。
- **过拟合缓解**：模型复杂度的降低意味着网络更倾向于找到简单的解，从而减轻了过拟合现象。这与奥卡姆剃刀原理一致——在同等表现下选择更简单的模型。

2）机器学习中的residual 51:56

[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.

9

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g(x))}{\partial g(x)} \cdot \frac{\partial g(x)}{\partial x}$$

$$\frac{\partial f(g(x)) + g(x)}{\partial x} = 1 + \frac{\partial g(x)}{\partial x}$$

- **A. Object Detection Baselines** 8 in
- **与GBDT的区别**：
  - **操作维度**：梯度提升树（GBDT）在标签空间进行残差计算，而ResNet是在特征空间进行残差连接。

- **实现方式**：GBDT通过串行迭代修正预测误差，ResNet通过跨层连接实现特征修正。
- **理论基础**：GBDT基于函数空间的梯度下降，ResNet基于神经网络的特征表示学习。

3）ResNet文章总结与评价 52:25

- **核心贡献**：
  - **简单有效**：通过极简的残差连接结构解决了深度网络训练难题。
  - **实验突破**：在ImageNet等基准上实现了当时最先进的性能。
- **学术价值**：
  - **启发意义**：虽然理论解释不够完善，但开创性的实验结果为后续研究指明了方向。
  - **研究范式**：示范了优秀科研工作的标准——不必面面俱到，只要在一个关键点上做出实质性突破。
- **后续影响**：
  - **社区发展**：为研究者留下了充足的发展空间，促进了残差结构在各种网络中的变体应用。
  - **工程价值**：提出的基础结构被广泛应用于计算机视觉各个领域。

二、知识小结

知识点	核心内容	关键要点/易混淆点	重要性
残差连接原理	通过shortcut connection实现输入与输出的直接相加，学习残差F(x)=H(x)-x而非直接学习H(x)	• 梯度计算变为加法形式避免消失 • **与Highway Network的区别**：无门控参数更简单	★★★★★
网络退化问题	深层网络训练误差反而增大，**非过拟合导致**（训练/测试误差同步上升）	• 理论上应能学习恒等映射 • 实际SGD优化困难	★★★★
Bottleneck设计	1×1卷积先降维再升维（如256→64→256），**降低计算量**	• 通道数变化时需匹配维度 • 步幅2实现下采样	★★★★☆
ResNet架构变体	18/34层用基础块（2个3×3卷积），50+层用Bottleneck块（3层）	• 34→50层计算量仅增1.8倍 • **152层**在ImageNet达3.57%错误率	★★★★
训练优化策略	• 学习率0.1分阶段衰减 • 不用Dropout • 10-crop测试增强	• 与AlexNet相比：**随机尺寸缩放**替代PCA	★★★
理论解释争议	原始解释：强制浅层网络性能下限 **后续研究**：梯度传播更稳定	• CIFAR-10千层网络仍有效 • 模型复杂度实际降低	★★★☆