

【文章编号】 2096-2835(2019)01-0056-09 DOI:10.3969/j.issn.2096-2835.2019.01.010

# 基于 MobileNet 的轻量级网络构建及其超分辨率图像重建

张焯林, 曹飞龙

(中国计量大学 理学院, 浙江 杭州 310018)

**【摘要】** 目的: 研究轻量级网络的超分辨率重建。方法: 尝试在图像超分辨率重建中引入 MobileNet 网络结构, 并使用 MobileNet v2 网络结构对网络进行改进。结果: 通过将标准的卷积网络分解为深度卷积和逐点卷积操作, 该网络将参数数量和计算量缩减为原来的 1/4 左右。结果显示除了在扩大因子为  $\times 2$  的情况下重建效果有所下降之外, 在其他的尺度上都取得了更好的效果。使用 MobileNet v2 网络结构对网络改进以后, 该网络能在参数数量和计算量增加不多的情况下进一步提升效果, 重建效果超过所有对比方法。结论: 所构建的两个轻量级网络不仅在定性指标上面有更好的结果, 而且在视觉效果上也具一定优势。

**【关键词】** 计量; 超分辨率重建; MobileNet 结构; 深度卷积; 逐点卷积

**【中图分类号】** TB96; TP183

**【文献标志码】** A

## Constructing lightweight neural networks based on MobileNet to reconstruct single image super-resolution

ZHANG Zhuolin, CAO Feilong

(Department of Applied Mathematics, College of Sciences, China Jiliang University, Hangzhou 310018, China)

**[Abstract]** **Aims:** This paper aims to study the image super-resolution of lightweight neural networks. **Methods:** The network structure of MobileNet was introduced into image super-resolution reconstruction to improve the network by using the network structure of MobileNet V2. **Results:** By decomposing the standard convolution network into depthwise convolution and pointwise convolution operation, the number of parameters and computation was reduced to about 1/4 of the original one. The results showed that the reconstruction effect decreased when the scale factor was  $\times 2$  and better results were achieved on other scales. Improved by using MobileNet V2, the network increased the effect with little increase of parameters and computation amount, having the best reconstruction effect. **Conclusions:** This two lightweight networks not only have better results in qualitative indexes, but also they have some advantages in visual effects.

**[Key words]** measurement; super-resolution; MobileNet structure; depthwise convolution; pointwise convolution

**【收稿日期】** 2018-12-26 **《中国计量大学学报》网址:** zgjl.cbpt.cnki.net

**【基金项目】** 国家自然科学基金项目 (No. 61672477)

**【通信作者】** 曹飞龙 (1965-), 男, 教授, 博士后, 主要研究方向为智能计算与应用等。E-mail: feilongcao@gmail.com

自从 AlexNet<sup>[1]</sup> 赢得了 ILSVRC 2012<sup>[2]</sup> 冠军,卷积神经网络在计算机视觉任务当中的应用越来越普遍,在图像分类、目标检测和图像分割等任务中表现出了巨大的优越性。随着计算机硬件计算能力的提升,为了能够实现更高的准确率,人们尝试训练更深更复杂的网络。但是伴随着模型精度一起提升的是计算量、存储空间以及能耗方面的巨大开销。

在实际应用当中,设备的计算能力和存储空间往往是有限的,如移动、车载设备。过于复杂的神经网络对于这些设备来说都是难以接受的。为了能够让深度网络适应这些设备,先前很多模型小型化工作将焦点放在模型的尺寸上。2015 年, Han 等人<sup>[3]</sup> 提出了深度压缩方法,该方法利用冗余卷积核剪枝,低精度浮点数保存权值和哈夫曼编码的方法降低网络的存储容量;2016 年, Szegedy 等人<sup>[4]</sup> 提出 Inception v3,尝试对卷积核进行分解,使用  $1 \times N$  和  $N \times 1$  的卷积核代替  $N \times N$  的卷积核;2016 年, Landola 等人<sup>[5]</sup> 提出 SqueezeNet,采用  $1 \times 1$  卷积核对上层的特征图进行卷积,主要目的是减少特征图的维数。

后来, Sifre 等人<sup>[6]</sup> 在深入研究深度可分离卷积之后提出了小型化网络 MobileNet<sup>[7]</sup>。该网络在保证计算精度不变的基础之上,进一步减少了网络的参数量和计算量。而深度可分离卷积的本质是使用深度卷积和逐点卷积将卷积操作分解为冗余信息更少的稀疏化表达。2018 年, Howard 等人尝试对 MobileNet<sup>[7]</sup> 进行改进,提出了 MobileNet v2<sup>[8]</sup>,主要利用了线性瓶颈层和颠倒残差对网络进行改进,明显地提升了网络的预测精度。

深度学习在图像超分辨率重建中同样被认为是十分有效的,一些传统方法<sup>[9]</sup> 与之相比已经失去了优势。从超分辨率卷积神经网络(Super resolution convolutional neural network, SRCNN)<sup>[10]</sup> 方法到非常深的卷积网络超分辨率(Very deep convolutional neural networks super-resolution, VDSR)<sup>[11]</sup> 方法,再到增强的深度残差网络超分辨率(Enhanced deep residual networks super-resolution, EDSR)<sup>[12-15]</sup> 方法,重建结果被不断地提高,网络也变得复杂化。也有网络专注于加快重建速度,如高效的亚像素卷积神经网络(Efficient sub-pixel convolutional neural net-

work, ESPCN)<sup>[16]</sup> 方法和快速超分辨率卷积神经网络(Fast super resolution convolutional neural network, FSRCNN)<sup>[17]</sup> 方法,它们直接将未插值的低分辨率图片作为输入,减小特征图的尺寸。在网络的最后,将特征图放大重建成超分辨率图像,从而加快了图像的重建速度。本文中我们尝试将 MobileNet<sup>[7]</sup> 运用到 ESPCN<sup>[16]</sup> 图像重建当中去,构建一个小型化的卷积网络,以保证重建效果不变甚至有所提升的情况下,进一步减小网络的权值和计算量。同时,我们也尝试将其改进版本 MobileNet v2<sup>[8]</sup> 应用到该网络中与其进行对比,结果显示能够进一步提高重建效果。

## 1 相关工作

### 1.1 MobileNet

MobileNet<sup>[7]</sup> 的核心层使用了一种称之为深度可分离卷积的操作来替代传统的标准卷积,减少了卷积核的冗余表达。深度可分离卷积<sup>[6]</sup> 可以被分解为深度卷积和卷积核尺寸为  $1 \times 1$  的逐点卷积组合。假设输入特征图和输出特征图的通道数量分别为  $M$  和  $N$ ,图 1 展示了将标准卷积滤波器(a)分解为深度卷积滤波器(b)和逐点卷积滤波器(c)的构成。在计算量和参数数量明显下降之后,卷积网络可以应用在更多的移动端平台。

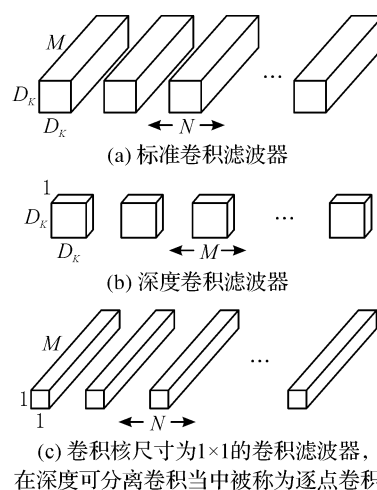


图 1 标准卷积滤波器替换为深度可分离卷积的示意图

Figure 1 Sketch map of depth-separable convolution replacing the standard convolution filter

传统的卷积层以一组尺寸为  $D_F \times D_F \times M$  的特征图  $F$  作为输入,产生一组尺寸为  $D_G \times D_G \times$

$N$  的特征图  $G$  作为输出。这里的  $D_F$  是一张输入方形特征图的宽和高,  $M$  是输入通道的数量,  $D_G$  是方形输出特征图的宽和高,  $N$  是输出特征通道的数量。卷积层中的卷积核  $K$  包含  $D_K \times D_K \times M \times N$  个参数, 这里的  $D_K$  表示一个方形卷积核的边长。一层步长为 1, 包含加边的卷积输出可由下式计算:

$$G_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i-1,l+j-1,m} \quad (1)$$

其计算量为

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \quad (2)$$

标准卷积操作有基于卷积核滤波产生特征图和组合特征图产生新的特征表示的效果。通过使用深度可分离卷积可以将产生和组合步骤被分为两步, 分别用深度卷积和逐点卷积代替, 从而大大减少计算开销。

其中, 每一个输入通道一个滤波器的深度卷积操作可以被写为

$$\hat{G}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,l+j-1,m} \quad (3)$$

这里的  $\hat{K}$  是尺寸为  $D_K \times D_K \times M$ ,  $\hat{K}$  中第  $m$  个滤波器被应用到  $F$  中的第  $m$  张特征图中, 产生  $G$  中第  $m$  张特征图。深度卷积的计算开销是

$$D_K \times D_K \times M \times D_F \times D_F \quad (4)$$

相对于标准卷积来说深度卷积特别高效, 但是它仅仅是对输入通道进行滤波, 而没有组合他们产生新的特征。因此, 为了产生新的特征, 需要使用卷积核尺寸为  $1 \times 1$  的卷积层来对深度卷积的输出进行线性组合。深度可分离卷积总的计算开销是

$$D_K \times D_K \times M \times D_F \times D_F + N \times M \times D_F \times D_F \quad (5)$$

通过将卷积操作分解为滤波和组合两步过程, 我们可以将计算量减少为

$$\frac{D_K \times D_K \times M \times D_F \times D_F + N \times M \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_k^2} \quad (6)$$

MobileNet<sup>[7]</sup> 使用滤波器尺寸为  $3 \times 3$  的深度可分离卷积, 该操作能将卷积的运算量减少 8~9 倍, 同时使预测精度仅有略微的下降。

图 2 是标准卷积层和深度可分离卷积层的结构示意图, 从图中可以看出, MobileNet<sup>[7]</sup> 模型的

核心就是将原本标准的卷积操作分解成一个深度卷积和一个逐点卷积操作, 从而减少参数量。深度卷积和后面的逐点卷积被当成了两个独立的模块, 都在输出结果的部分加入了归一化操作 (Batch normalization, BN)<sup>[18]</sup> 和非线性激活单元 ReLU<sup>[19]</sup> 函数。这样的组合方式代替传统卷积不仅在理论上会更高效, 而且由于大量使用卷积核尺寸为  $1 \times 1$  的卷积操作, 可以直接使用高度优化的函数库来完成, 进一步加快计算速度。

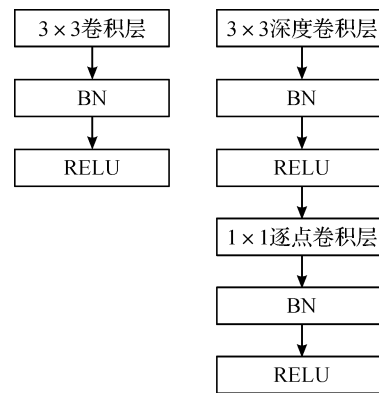


图 2 标准卷积层和深度可分离卷积层的结构示意图

Figure 2 Sketch map of standard convolutional layer and depth separable convolutional layer

## 1.2 MobileNet v2<sup>[8]</sup>

MobileNet v2<sup>[8]</sup> 是在 MobileNet<sup>[7]</sup> 的结构之上进行改进, 在其中加入了两种基本结构, 一种是线性瓶颈层; 另外一种为颠倒残差结构。

### 1.2.1 线性瓶颈

如果将尺寸为  $w \times h \times c$  的一组特征图理解成特征空间上有  $c$  个分量的一个特征向量, 对于我们的任务有用的特征需要分布在这  $c$  维特征空间或其子空间中。显然, 要获得更高的精度, 这些信息随着网络传播需尽可能保留。

易知, 如果一个层变换  $\text{ReLU}(Bx)$  的结果中有一个非零量  $S$ , 对于  $S$  内部的一个点来自于对输入进行线性变换  $B$ , 这表明输入空间对应于全维度输出的映射可用线性变换表示。另一方面, 采用 ReLU 激活函数不可避免地导致输入特征信息的流逝。但是当我们有很多维特征, 某一特征经过 ReLU 函数作用之后丢失了, 相同的特征可能保留在其他维度。

图 3 是一个将低维流形嵌入到高维空间后使用 ReLU 进行变换操作的例子,最左边的输入二维数据  $x$ ,相当于有用的信息。首先,通过一个随机矩阵  $T$  将其映射到不同的高维空间以后,使

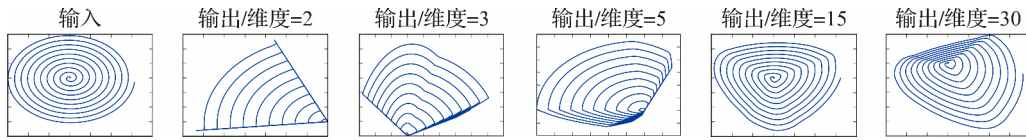


图 3 将低维流形嵌入到高维空间后使用激活函数 ReLU 进行变换操作的例子

Figure 3 Examples of ReLU transformations of low-dimensional manifolds embedded in higher-dimensional spaces

因此,如果输入当中的有用特征可以被嵌入到激活空间当中非常低维的子空间中,那么 ReLU 变换能够基本保留有用信息,但是会引入必要的计算复杂度。于是得出如下两个结论。

1) 如果有用的流形数据经过激活层以后剩下非零量,ReLU 相当于对非零量做了一次非线性变换。

2) 只有当有用的特征落在输入空间中的低维子空间时,ReLU 函数才能够将输入中有用的特征完整保留。

这两个结论提供了优化神经网络的线索,假设有用的特征是低维的,我们可以尝试在深度卷积后插入线性瓶颈层压缩信息后,仍然能捕获完整信息。原文实验证明使用线性层是关键,因为它能防止非线性化破坏太多的信息。

### 1.2.2 颠倒残差

这里的颠倒残差也是一种残差结构,通常残差块的结构是用瓶颈层将特征图通道缩减。但这里的颠倒残差模型是使用瓶颈层将特征通道扩增;然后,使用深度卷积层对特征进行变换;最后,再用瓶颈层对特征进行压缩。在颠倒残差中,深度卷积层仅仅是用来完成特征非线性变换,所以残差连接被添加在两个瓶颈层之间。但总体上,该结构当中残差连接的作用仍然和残差块相同,用来加快收敛、防止退化问题。

如果输入特征的通道维度是  $c$ ,包含了所有有用的信息。该结构首先将特征拓展成  $t \cdot c$  个维度( $t$  大于等于 1),有用的特征将分布在  $t \cdot c$  维空间的子空间当中;然后,通过 ReLU 激活函数能够保留有用特征;最后,为了防止信息损失,去掉了最后的激活函数,只用线性瓶颈层对特征进行压缩。整体结构如图 4。

用  $Y = \text{ReLU}(wx + b)$  变换操作,再利用  $T^{-1}$  变回到 2 维空间,右边的图像展示变换后的结果。从中可以看出,映射到维度越高的特征变换后保留的信息就越多。

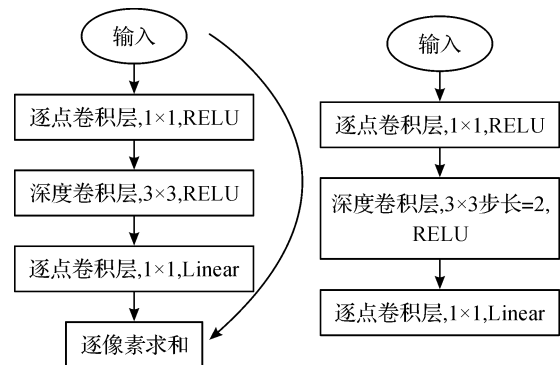


图 4 MobileNet v2 结构示意图

Figure 4 Sketch map of the MobileNet v2

基本的应用结构可以用表 1 进行解释。对于一个尺寸为  $h \times w$  的特征,扩展因子为  $t$ ,核尺寸为  $k$ ,输入通道为  $d_1$ ,输出通道为  $d_2$ 。总的计算量为

$$h \cdot w \cdot d_1 \cdot t(d_1 + k^2 + d_2). \quad (7)$$

表 1 输入通道为  $k$ ,输出通道为  $k'$ ,步长为  $s$ ,扩展因子为  $t$  的瓶颈残差块的结构

Table 1 Bottleneck residual block transforming from  $k$  to  $k'$  channels, with stride  $k'$ , and expansion factor  $t$

输入	操作	输出
$h \times w \times k$	$1 \times 1$ 卷积, ReLU	$h \times w \times (tk)$
$h \times w \times tk$	$3 \times 3$ 深度卷积, ReLU	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	Linear, $1 \times 1$ 卷积	$\frac{h}{s} \times \frac{w}{s} \times k'$

### 1.3 ESPCN<sup>[16]</sup>

该超分辨率重建网络直接使用未经插值的低分辨率图像作为输入,在网络的最后对特征图进行扩大形成输出图像。网络前面三层使用了普通

的卷积层,在网络的最后使用了亚像素卷积层对特征图像素进行重组后得到重建后的超分辨率图像。如果扩大因子为  $r$ ,第三层卷积输出的特征图数量为  $r^2$ ,可以通过最后的亚像素卷积层将特征图重组为超分辨率图像。最后的亚像素卷积层可以表示为

$$I^{\text{SR}} = f(I^{\text{LR}}) = \text{PS}(T_L)。 \quad (8)$$

式(8)中,  $\text{PS}(T_L)$  的本质是一个像素周期性重组操作,将  $H \times W \times C \cdot r^2$  的张量重组为  $rH \times rW \times C$  的张量,整个效果图如图 5。这个操作可以被如下数学公式描述:

$$\text{PS}(T)_{x,y,c} = T_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, c \cdot r \cdot \text{mod}(y,r) + c \cdot \text{mod}(x,r)}。 \quad (9)$$

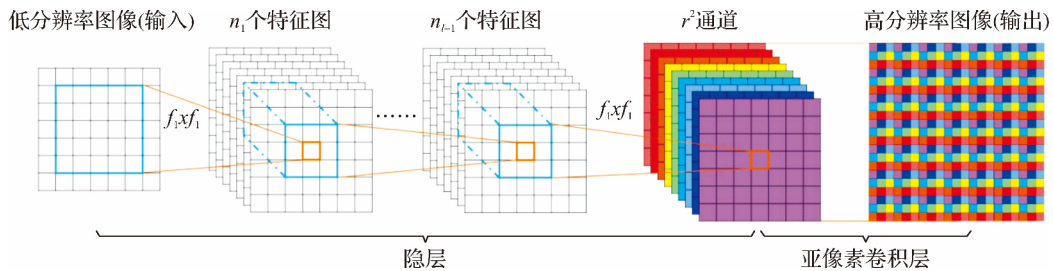


图 5 高效的亚像素卷积神经网络算法流程图

Figure 5 Framework of image super resolution using efficient sub-pixel convolutional neural network

## 2 提出的方法

单图像超分辨率重建的目标是通过一张从高分辨率图像  $I^{\text{HR}}$  下采样得到的低分辨率图像  $I^{\text{LR}}$  估计超分辨率图像  $I^{\text{SR}}$ , 这里的下采样操作包括使用高斯滤波器对高分辨率图像  $I^{\text{HR}}$  进行滤波, 然后对卷积后的图像进行比例为  $r$  的下采样。文章 ESPCN<sup>[16]</sup> 当中, 使用了 3 层卷积神经网络, 在网络的最后通过亚像素卷积层重置特征图像素的位置形成最终的超分辨率图像。我们尝试将 MobileNet<sup>[7]</sup> 引入到网络中, 第一层保持卷积操作不变, 将后两层的卷积操作替换为深度可分离卷积操作, 但是为了适应图像重建任务, 我们将 BN<sup>[18]</sup> 略去。深度卷积用如下公式表示:

$$T_{l+1} = W_{l+1} \circ T_l = (w_1 * t_1, w_2 * t_2, \dots, w_{n_{l+1}} * t_{n_l})。 \quad (10)$$

这里的  $W_l = (w_1, w_2, \dots, w_{n_l})$  和  $T_l = (t_1, t_2, \dots, t_{n_l})$  分别表示第  $l$  层的卷积核和特征图。于是我们的整个神经网络可用如下公式描述:

$$f^l(I^{\text{LR}}; W_l, b_l) = \varphi(W_l * I^{\text{LR}} + b_l); \quad (11)$$

$$f^l(I^{\text{LR}}; W_l, b_l, W_l^1, b_l^1) = \varphi(W_l^1 * \varphi(W_l \circ f^{l-1} + b_l) + b_l^1), l=2, 3; \quad (12)$$

$$I^{\text{SR}} = \text{PS}(f^3)。 \quad (13)$$

这里的  $\varphi$  表示激活函数  $\tanh()$ ,  $*$  表示卷积操作,  $W_l^1$  和  $b_l^1$  表示逐点卷积中大小为  $1 \times 1$  的卷积核和偏置。对于训练集中的高分辨率图像

$I_n^{\text{HR}}, n=1 \dots N$ , 我们通过下采样得到对应的低分辨率图像  $I_n^{\text{LR}}, n=1 \dots N$ , 然后使用上面的网络生成对应的重构图像并计算逐像素平均绝对值误差作为损失函数:

$$\text{loss} = \frac{1}{r^2 HW} \sum_{x=1}^{rH} \sum_{y=1}^{rW} |I_{x,y}^{\text{HR}} - I_{x,y}^{\text{SR}}|。 \quad (14)$$

最后, 我们利用 Adam<sup>[20]</sup> 算法来优化(14)式中的损失函数。具体的过程如下: 假设  $t$  时刻, 目标函数对于参数的一阶导数是  $g_t$ , 那么我们可以先计算一阶矩和二阶矩估计:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad (15)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2。 \quad (16)$$

接下来对两个估计进行矫正计算:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad (17)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}。 \quad (18)$$

最后得到了权值  $\theta_t$  的更新方法:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}。 \quad (19)$$

其中,  $\eta$  为学习速率,  $\beta_1$  为一阶矩估计的指数衰减率,  $\beta_2$  为二阶矩估计的指数衰减率,  $\epsilon$  是非常小的数, 其作用是为了防止分母为 0。

下文使用 MobileNet v2<sup>[8]</sup> 对网络进行进一步改进。因为不同卷积层的特征图数量不一样, 无法在瓶颈层之间进行残差连接, 所以我们将残差

连接移除。同样,我们保留第一层卷积操作不变,仅仅对后面两层进行改进,整个神经网络可用如下公式描述:

$$f^1(I^{LR}; W_1, b_1) = \varphi(W_1 * I^{LR} + b_1); \quad (20)$$

$$f^l(I^{LR}; W_l^1, b_l^1, W_l^2, b_l^2) = W_l^2 * f^l(I^{LR}; W_l^1, b_l^1) + b_l^2, l=2,3;$$

$$\varphi(W_l * \varphi(W_l^1 * f^{l-1} + b_l^1) + b_l^2), l=2,3; \quad (21)$$

$$I^{SR} = PS(f^3). \quad (22)$$

$W_l^1, W_l^2$  和  $b_l^1, b_l^2$  都表示逐点卷积中大小为  $1 \times 1$  的卷积核和偏置。损失函数的构造和网络训练和前面使用 MobileNet<sup>[7]</sup> 改进的网络部分一样。

注:为了叙述方便,下文将使用 MobileNet<sup>[7]</sup> 进行改进后的网络模型称为 MobileNet1,使用 MobileNet v2<sup>[8]</sup> 改进后的网络模型称为 MobileNet2。

两个网络模型的结构如图 6。

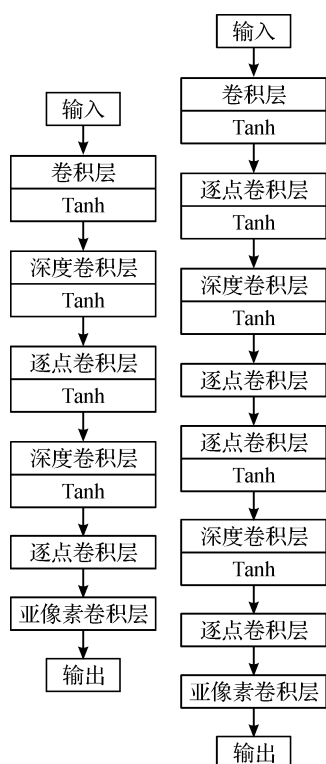


图 6 提出的图像超分辨率重建算法流程图

Figure 6 Framework of our proposal image super resolution

## 3 实验

### 3.1 实验描述

本实验算法在 ESPCN<sup>[16]</sup> 的基础上进行改

进,我们将所提的基于 MobileNet<sup>[7]</sup> 的亚像素卷积神经网络与 ESPCN<sup>[16]</sup> 和三层的超分辨率重建深度网络 SRCNN<sup>[10]</sup> 算法进行比较。

本文的图像超分辨率实验都是在 Matlab2016a 和 Python3.6.5 下操作,所用处理器为 Intel(R)Xeon(R)W-2123 CPU、64GB 内存,Tensorflow 框架<sup>[21]</sup> 下实现,显卡配置为 NVIDIA GeForce GTX 1080。

#### 3.1.1 训练集

本文的所有实验都是在 Yang 等人在文章 ScSr<sup>[22]</sup> 中提出来的 91 张图片训练集下进行的。首先,对于训练集进行预处理操作。因为人类对 Y 通道的变化更为敏感,我们的操作都是在将原始的 RGB 图像转换为 YCbCr 颜色空间以后,在 Y 通道上面进行的。设输入原始的训练图片(已转换为 Y 通道图像)为  $y_1, \dots, y_M$ , 这里的  $M$  为训练图片个数,我们将其作为真实高分辨率图片。对真实的高分辨率图像进行下采样得到对应的低分辨率训练图像  $x_1, \dots, x_M$ , 在实际操作当中可直接用 Matlab 的 `imresize()` 命令完成。然后对训练图像取补丁操作,设扩大因子为  $r$ , 对低分辨率训练图像  $x_1, \dots, x_M$  以步长为  $K$  选取对应像素周围的  $k_1 \times k_2$  邻域为补丁; 对高分辨率训练图像  $y_1, \dots, y_M$  以步长为  $rK$  选取对应像素周围的  $k'_1 \times k'_2$  邻域为补丁。对应的低分辨率补丁和高分辨率补丁分别为  $x_1^p, \dots, x_N^p$  和  $y_1^p, \dots, y_N^p$ , 这里的  $N$  为补丁的个数。最后,使用图像补丁对集合  $\{(x_i^p, y_i^p)\}_{i=1}^N$  作为训练数据进行训练。

#### 3.1.2 测试集

我们使用 3 个数据集作为测试集,“Set5”和“Set14”是在很多论文中用来测试的数据集,分别包含 5 张和 14 张图片,最早由杨建超等人<sup>[22]</sup> 用在 ScSr 当中。“B100”数据集也被使用,该数据集包含了 100 张自然图像,最早,由 Timoft 等人<sup>[23]</sup> 和 Yang<sup>[24]</sup> 用在 Berkerly 语义分割当中。

#### 3.1.3 训练参数

我们把训练低分辨率图像分割成大小为  $17 \times 17$  像素的图像补丁,步长为 9, 高分辨率图像补丁根据扩大因子计算确定。以 32 个图像补丁对为一组使用 Adam<sup>[20]</sup> 进行训练。Adam<sup>[20]</sup> 算法的参数使用默认值 ( $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$ ), 学习速率大小固定为 0.0001。卷积核的

大小与 ESPCN<sup>[16]</sup> 当中的一样,在 MobileNet1 中,特征图数量的组合为(64-64-32-32- $r$ ),卷积核的大小组合为(5-3-1-3-1)。在 MobileNet2 当中,特征图数量的组合为(64-64-64-32-64-64- $r$ ),卷积核的大小组合为(5-1-3-1-1-3-1),其中  $r$  为扩大因子。训练一个模型使用 50 000 个周期,单个模型训练花费时间 1~2 d。

### 3.2 实验结果

#### 3.2.1 对比实验

ESPCN<sup>[16]</sup> 本身使用的是均方误差 (Mean square error, MSE) 作为损失函数进行训练,而我们的方法使用了平均绝对值误差 (Mean absolute error, MAE) 作为损失函数,分别可用  $l_2$  和  $l_1$  范代替。为了证明  $l_1$  范确实对重建效果有提升,我们对比了  $l_1$  和  $l_2$  范在测试集“Set5”和“Set14”上面的重建效果。我们对这两者做了对比实验,比较的指标使用峰值信噪比 (Peak signal-to-noise ratio, PSNR),重建结果如表 2。

表 2 两种损失函数的重建结果对比

Table 2 Comparison of the reconstruction results of the two loss functions

	MobileNet1_ $l_2$	MobileNet2_ $l_1$	MobileNet2_ $l_2$	MobileNet1_ $l_1$
$\times 2$	36.57	36.70	36.57	36.77
Set5 $\times 3$	32.72	32.81	32.79	32.96
$\times 4$	30.33	30.43	30.47	30.66
$\times 2$	32.34	32.44	32.38	32.50
Set14 $\times 3$	29.15	29.20	29.20	29.28
$\times 4$	27.20	27.26	27.28	27.41

从表 2 中可以看出,无论 MobileNet1 还是

MobileNet2,使用  $l_1$  范数作为损失函数的结果在各个数据集上面的表现都要好于  $l_2$  范数。特别是对于 MobileNet2 的提升要更为明显,提升幅度基本超过了 0.1 dB,在数据集 Set5 的扩大因子为  $\times 2$  时甚至能够达到 0.2 dB。可见我们选择平均绝对值误差作为损失函数是合理的。

#### 3.2.2 与其他方法进行比较

我们比较的方法主要包括 SRCNN<sup>[10]</sup>、ESPCN<sup>[16]</sup>,评价指标为 PSNR 和结构相似性 (Structural similarity, SSIM)。在表 3 中,我们列出了在数据集“Set5”、“Set14”和“B100”上面的表现结果。其中,ESPCN<sup>[16]</sup> 算法的结果是使用作者提供的源程序,在 91 张训练集上重新训练得到模型,然后在测试集上面测试的结果。结果数值均为对应数据集上面各张图片评价指标计算结果的平均值。其中字体加粗代表所有方法中最好的结果,加下划线代表第二好的结果。与表 2 的结果进行比较可以看出,同样  $l_2$  是以范数作为损失函数,除了  $\times 2$  的情况下我们的 MobileNet1 模型比 ESPCN<sup>[16]</sup> 重建结果稍差,其他的扩大因子下重建结果都要好于 ESPCN<sup>[16]</sup>,特别是在  $\times 3$  的情形下我们的模型要远远好于 ESPCN<sup>[16]</sup>。从表 3 中我们能够比较得出使用了  $l_1$  范数作为损失函数以后,MobileNet1 的重建效果在各个数据集上面都要好于 ESPCN<sup>[16]</sup> 的重建效果。在指标 PSNR 上面,将模型改进得到的 MobileNet2 能比 MobileNet1 提高 0.05~0.23 dB。同时,其效果在各个数据集上面也都超过了其他所有对比方法的重建结果。

表 3 不同的超分辨率重建算法在数据集上的表现

Table 3 The representation of different super-resolution reconstruction algorithms on datasets

		Bicubic	SRCNN	ESPCN	MobileNet1_ $l_1$	MobileNet2_ $l_1$
Set5	$\times 2$	33.66/0.929 9	36.34/0.952 1	36.61/0.953 1	<u>36.70/0.954 1</u>	<b>36.77/0.954 7</b>
	$\times 3$	30.39/0.868 2	32.39/0.903 3	32.48/0.902 8	<u>32.81/0.910 1</u>	<b>32.96/0.912 1</b>
	$\times 4$	28.42/0.810 4	30.09/0.853 0	30.29/0.855 7	<u>30.43/0.863 9</u>	<b>30.66/0.868 7</b>
Set14	$\times 2$	30.23/0.868 7	32.18/0.903 9	32.36/0.904 7	<u>32.44/0.905 4</u>	<b>32.50/0.906 4</b>
	$\times 3$	27.54/0.773 6	29.00/0.814 5	28.99/0.813 1	<u>29.20/0.817 1</u>	<b>29.28/0.819 1</b>
	$\times 4$	26.00/0.701 9	27.20/0.741 3	27.19/0.742 3	<u>27.26/0.745 6</u>	<b>27.41/0.749 5</b>
B100	$\times 2$	29.56/0.843 1	31.14/0.884 7	31.17/0.885 1	<u>31.21/0.885 9</u>	<b>31.26/0.887 5</b>
	$\times 3$	27.21/0.738 4	<u>28.21/0.780 0</u>	28.09/0.777 9	<u>28.21/0.779 8</u>	<b>28.27/0.782 2</b>
	$\times 4$	25.96/0.667 4	<u>26.71/0.702 2</u>	26.59/0.701 6	<u>26.64/0.703 3</u>	<b>26.76/0.706 9</b>

另外,我们计算出了使用尺寸为  $240 \times 240$  的输入图像进行超分辨率重建后,所有对比算法模

型的参数数量和计算量,如表 4。表 4 表明 MobileNet1 的参数数量和计算量都是最少的,且明



显少于其他的网络。从参数数量和重建效果对比,在同时使用  $l_2$  范的情况下,MobileNet1 在参数和计算量远远少于 SRCNN<sup>[10]</sup> 和 ESPCN<sup>[16]</sup> 的情况下,在  $\times 3$ 、 $\times 4$  扩大因子下重建效果都要更好,在  $\times 2$  下超过了 SRCNN<sup>[10]</sup> 并接近 ESPCN<sup>[16]</sup>。而 MobileNet2 在增加一定量的参数和计算量但计算量仍然少于 ESPCN<sup>[16]</sup> 和 SRCNN<sup>[10]</sup> 后能够使重建效果进一步提升。

表 4 不同网络的参数数量和计算量比较

Table 4 Comparison of the number of parameters and computation of different networks

	SRCNN		ESPCN		MobileNet1		MobileNet2	
	参数数量	计算量	参数数量	计算量	参数数量	计算量	参数数量	计算量
240×240	8 129	518 137 088	22 729	136 689 664	5 001	39 968 576	11 881	114 582 016

为了能够直观地比较各个方法重建图像的效果,我们从每一个测试集中选择一张图片,将其扩大因子  $\times 3$  的重建效果展示在图 7 当中。从图

像(a),(b)当中来看,我们的两个模型重建的图像相比于 SRCNN<sup>[10]</sup> 和 ESPCN<sup>[16]</sup> 图案更加地清晰,内部没有多余的黑色斑点。图像(c)当中各个方法



图 7 图像超分辨率重建算法视觉对比图

Figure 7 Comparison of visual images for image super-resolution reconstruction



的重建结果纹理都出现了融合现象,但是我们提出的方法重建图像的融合程度相比于 ESPCN<sup>[16]</sup>方法要更弱一些。可见,我们的方法不仅在定量的评价标准上面超出了其他的方法,在视觉上面也有一定的优势。

## 4 结 论

本文中,我们在超分辨率重建算法 ESPCN<sup>[16]</sup>中引入 MobileNet<sup>[6]</sup>的网络结构。提出的网络模型 MobileNet1 大大减少了卷积所需的参数数量和计算量,达到了原来的 1/4 左右。从实验结果中来看,该算法除了在  $\times 2$  这个扩增因子当中会使重建效果有所下降以外,在其他的尺度上面都能使重建效果有所提升,并且使用  $l_1$  范之后能使效果更好。当使用 MobileNet2 改进 MobileNet1 时,能够在参数量提升不多的情况下,使重建效果进一步提升。在三个数据集上面超过了比较的方法,提升最多能够达到 0.23 dB。从视觉效果上来看,我们的方法能够使图案更清晰,有较明显的优势。

## 【参 考 文 献】

- [1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]//**International Conference on Neural Information Processing Systems**, New York: ACM,2012:1097-1105.
- [2] RUSSAKOVSKY O, DENG J, SU H, et al. ImageNet large scale visual recognition challenge[J]. **International Journal of Computer Vision**,2014,115(3):211-252.
- [3] HAN S, MAO H, DALLY W J. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding[C]//**International Conference of Learning Representation**. San Juan, Puerto Rico: arXiv, 2016:1-14.
- [4] SZEGEDY C, VANHOUCKE V, IOFFE S, et al. Rethinking the inception architecture for computer vision[C]//**The IEEE Conference on Computer Vision and Pattern Recognition**. Las Vegas: IEEE,2016:2818-2826.
- [5] IANDOLA F N, HAN S, MOSKEWICZ M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and  $< 0.5$  MB model size[C]//**International Conference of Learning Representation**. San Juan: arXiv,2016:1-9.
- [6] SIFRE L. **Rigid-Motion Scattering for Image Classification** [D]. Paris:École Polytechnique,2014.
- [7] HOWARD A G, ZHU M, CHEN B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications[C]//**IEEE Conference on Computer Vision and Pattern Recognition**. Hawaii: IEEE,2017:1-9.
- [8] SANDLER M, HOWARD A, ZHU M, et al. MobileNetV2: Inverted residuals and linear bottlenecks[C]//**The IEEE Conference on Computer Vision and Pattern Recognition**. Salt Lake City: IEEE,2018:4510-4520.
- [9] 夏荷艳,曹飞龙,稀疏表示图像超分辨率重建的自适应方法[J]. **中国计量大学学报**,2017,28(4):478-484.
- [9] XIA H Y, CAO F L. An adaptive method for sparse representation image super-resolution reconstruction[J]. **Journal of China University of Metrology**,2017,28(4):478-484.
- [10] DONG C, LOY C C, TANG X. Learning a deep convolutional network for image super-resolution[C]//**European Conference on Computer Vision**. Berlin: Springer, 2014: 184-199.
- [11] KIM J, LEE J K, LEE K M. Accurate image super-resolution using very deep convolutional networks[C]//**IEEE Conference on Computer Vision and Pattern Recognition**. Las Vegas: IEEE,2016:1646-1654.
- [12] TAI Y, YANG J, LIU X. Image super-resolution via deep recursive residual network[C]//**IEEE Conference on Computer Vision and Pattern Recognition**. Hawaii: IEEE Computer Society,2017:2790-2798.
- [13] HUANG G, LIU Z, WEINBERGER K Q. Densely connected convolutional networks[C]//**IEEE Conference on Computer Vision and Pattern Recognition**. Hawaii: IEEE Computer Society,2017:2261-2269.
- [14] LEDIG C, THEIS L, HUSZÁR F, et al. Photo-realistic single image super-resolution using a generative adversarial network[C]//**IEEE Conference on Computer Vision and Pattern Recognition**. Hawaii: IEEE,2017:105-114.
- [15] LIM B, SON S, KIM H, et al. Enhanced deep residual networks for single image super-resolution [C]//**IEEE Conference on Computer Vision and Pattern Recognition**. Hawaii: IEEE,2017:1132-1140.
- [16] BISHOP R, RUECKERT D, WANG Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network[C]//**IEEE Conference on Computer Vision and Pattern Recognition**. Las Vegas: IEEE Computer Society,2016:1874-1883.
- [17] DONG C, LOY C C, TANG X. Accelerating the super-resolution convolutional neural network [C]//**European Conference on Computer Vision**. Berlin: Springer, 2016: 391-407.
- [18] IOFFE S, SZEGEDY C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. **Journal of Machine Learning Research**,2015,37: 448-456.

(下转第 103 页)

- [D]. Shanghai: Shanghai Jiaotong University, 2012.
- [7] 杨润党, 武殿梁, 范秀敏, 等. 基于约束的虚拟装配运动导航技术[J]. 农业机械学报, 2006, 37(3): 113-118.  
YANG R D, WU D L, FAN X M, et al. Movement navigation technology of virtual assembly based constraint[J]. *Transactions of the Chinese Society for Agricultural Machinery*, 2006, 37(3): 113-118.
- [8] 宋玉春, 肖耘亚. 基于 CMAC 网络的轴承轮毂单元铆合控制算法研究[J]. 制造技术与机床, 2011, 54(5): 155-158.  
SONG Y C, XIAO Y. Study on clinching control algorithm for hub bearing unit based on CMAC[J]. *Manufacturing Technology & Machine Tool*, 2011, 54(5): 155-158.
- [9] 曲杰, 张国杰, 徐小琴. 轮毂轴承单元轴铆合装配的铆头优化设计[J]. 华南理工大学学报(自然科学版), 2016, 44(2): 60-66, 73.  
QU J, ZHANG G J, XU X Q. Optimization design of rivet head for shaft riveting assembly of hub bearing unit[J]. *Journal of South China University of Technology(Natural Science Edition)*, 2016, 44(2): 60-66, 73.
- [10] 单越康, 余松青, 刘月瑶, 等. 基于沟道综合位置的轮毂轴承轴向游隙评价方法[J]. 轴承, 2016, 51(5): 49-53.  
SHAN Y K, YU S Q, LIU Y Y, et al. Evaluation method for axial clearance of hub bearings based on comprehensive position of raceway[J]. *Bearing*, 2016, 51(5): 49-53.
- [11] 吴斌, 史忠植. 一种基于蚁群算法的 TSP 问题分段求解算法[J]. 计算机学报, 2001, 24(12): 1328-1333.  
WU B, SHI Z Z. An ant colony algorithm based partition algorithm for TSP[J]. *Chinese Journal of Computers*, 2001, 24(12): 1328-1333.
- [12] 周明, 孙树栋. 遗传算法原理及应用[M]. 北京: 国防工业出版社, 1999: 10-25.
- [13] 徐锋, 杜军平. 改进蚁群算法在旅游路线规划中的应用研究[J]. 计算机工程与应用, 2009, 45(23): 193-195, 226.  
XU F, DU J P. Study on travel route planning based on improved ant colony algorithm[J]. *Computer Engineering and Applications*, 2009, 45(23): 193-195, 226.
- (上接第 64 页)
- [19] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J]. *Advances in Neural Information Processing Systems*, 2012, 25(2): 1097-1105.
- [20] KINGMA D K, BA J. Adam: A method for stochastic optimization[C]//*International Conference for Learning Representations*. San Diego: arXiv, 2015: 1-15.
- [21] ABADI M, AGARWAL A, BARHAM P, et al. TensorFlow: Large-scale machine learning on heterogeneous systems[C]//*Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*. Savannah, USA: USENIX, 2015: 265-283.
- [22] YANG J, WRIGHT J, HUANG T S, et al. Super-resolution via sparse representation[J]. *IEEE Transaction Image Process*, 2010, 19(11): 2861-2873.
- [23] TIMOFTE R, SMET V D, GOOL L V. A+: Adjusted anchored neighborhood regression for fast super-resolution[C]//*The 12th Asian Conference on Computer Vision*. Berlin: Springer, 2014: 111-126.