

# 没钱买华为P30？这个图像超分辨率项目帮你「拍」出高清照片

机器之心 2019-03-27

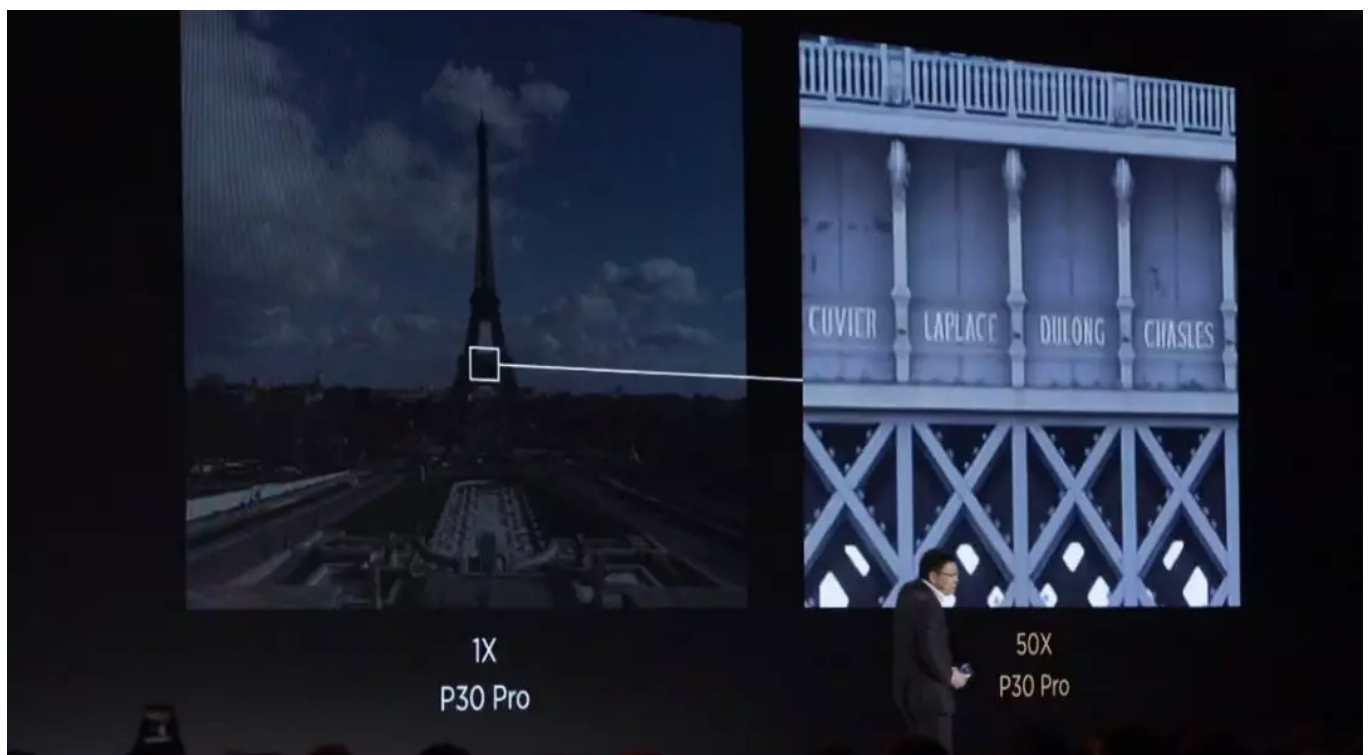
选自GitHub

项目作者：Francesco Cardinale等

机器之心编译

华为刚刚发布的 P30「望远镜」手机能在几十米外拍到埃菲尔上的人名，确实令人佩服，但其售价也是令人望而生畏。那么，不买华为手机、高级单反就拍不到充满细节的高清照片了吗？

相机不够算法凑，拥有超级拍照能力的手机也离不开算法的加持。本文介绍的 图像超分辨率项目 可以帮你补齐相机镜头的短板。



华为 P30 发布会上展示的埃菲尔铁塔高清远距离照片。

今天，一位 Reddit 网友贴出了自己基于 Keras 的图像超分辨率项目，可以让照片放大后依然清晰。先来看一下效果。

## Image Super-Resolution (ISR)



放大数倍后，照片中的蝴蝶（蛾子？）依然没有失真，背上的绒毛清晰可见。

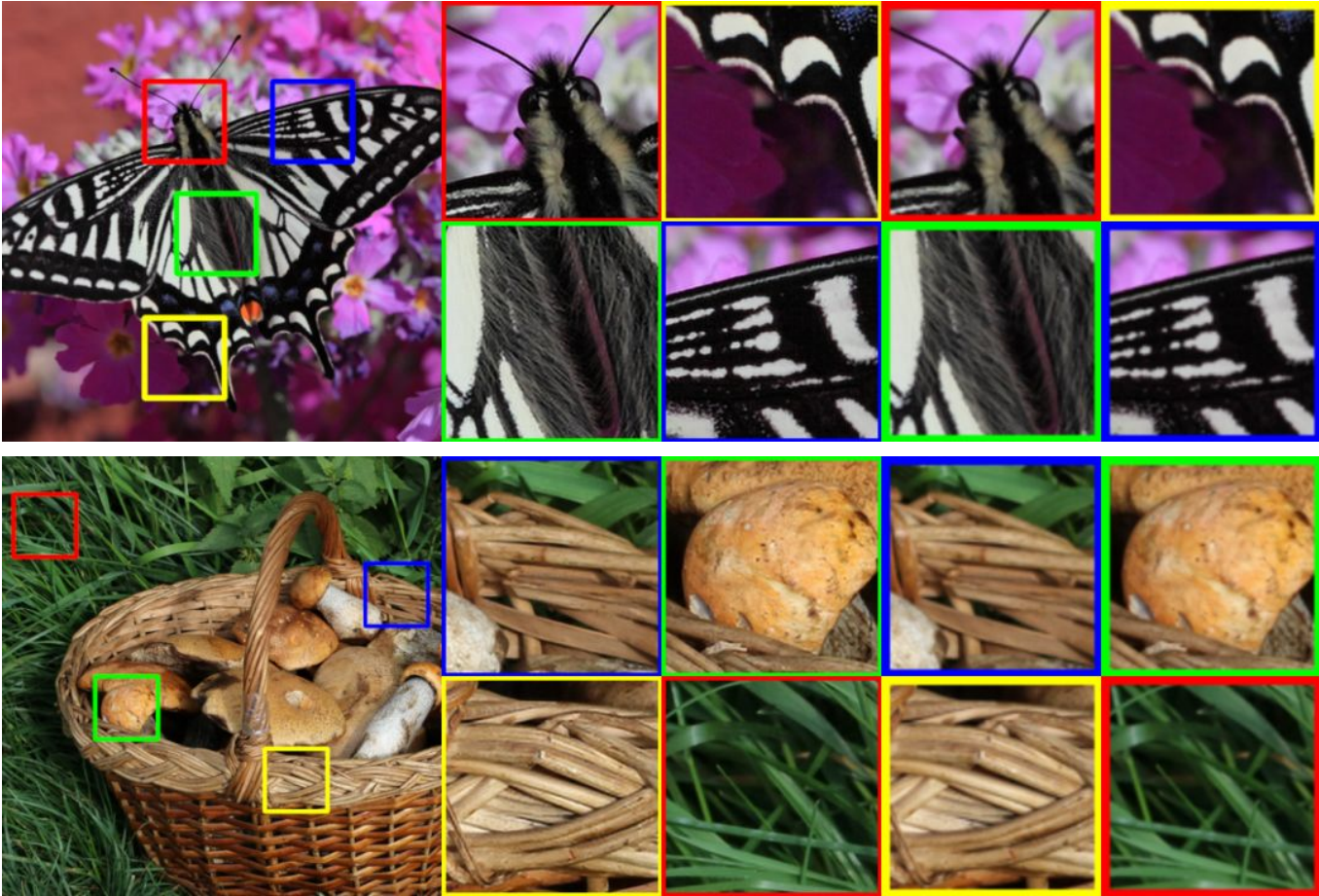
作者表示，该项目旨在改善低分辨率图像的质量，使其焕然一新。使用该工具可以对图像进行超级放缩，还能很容易地在 RDN 和 GAN 上进行实验。

该项目包含不同残差密集网络的 Keras 实现，它们可用于高效的单图像超分辨率（ISR）。同时作者还提供了各种文档资料以帮助训练模型，包括如何使用对抗损失组件训练这些网络。

### 项目示例

这些示例使用的放大因子（upscaling factor）为 2，即像素数扩大两倍。大家可在 sample\_weights 中查看生成示例图像的权重，它们存储在 git lfs 上。如要下载这些权重，你需要先复制该 repo，然后运行 git lfs pull。

左图为原始的低分辨率图像，中间图为该网络的输出结果，右图为使用 GIMP bicubic scaling 得到的基线模型放大结果。



下面是不同方法作用于噪声图像的效果对比，这些方法分别是：使用 bicubic scaling 的基线模型、使用像素级内容损失函数训练的 RDN 网络，以及使用 VGG19 内容压缩数据集和损失函数进行重训练的 RDN 网络。该 repo 包含这些模型的权重。



Bicubic up-scaling（基线模型）的输出结果示例。





使用像素级内容损失函数训练的 RDN 网络的输出结果示例。



使用 VGG 内容和对抗损失组件训练的 RDN 网络的输出结果示例。

## 超分辨率项目有什么

前面展示的超分辨率效果都是根据该项目实现的不同模型做出来的。超分辨率希望根据已有的图像信息重构出缺失的图像细节，它通常借助卷积神经网络抽取图像信息，再通过转置卷积将这些信息扩展到希望获得的图像分辨率。

在这个项目中，作者新增了很多模块与特征，例如使用 VGG 与 GAN 实现真实的放大图像。该项目主要实现的是 RDN 与 RRDN 网络，且同时还提供了预训练权重和 Colab 教程。不论是训练还是推断，根据这些资料我们都可以快速上手。

此外，该项目目前已经可以发布到 PyPI 上了，因此安装也只需键入 pip 命令即可。

总而言之，整个项目实现了三个超分辨率网络，且采用了 Keras 版的 VGG-19 作为特征抽取模块。如下所示为三个超分辨率网络的相关研究：

- Residual Dense Network for Image Super-Resolution(Zhang et al. 2018, arXiv:1802.08797)
- ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks(Wang et al. 2018, arXiv:1809.00219)
- Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network(SRGANS, Ledig et al. 2017, arXiv:1609.04802)

如果我们想要生成上面那样的高清图，该项目还提供了一系列的资源：

- 文档：<https://idealo.github.io/image-super-resolution/>
- 代码：<https://github.com/idealo/image-super-resolution/>
- Colab 推断代码：[https://colab.research.google.com/github/idealo/image-super-resolution/blob/master/notebooks/ISR\\_Prediction\\_Tutorial.ipynb](https://colab.research.google.com/github/idealo/image-super-resolution/blob/master/notebooks/ISR_Prediction_Tutorial.ipynb)
- Colab 训练代码：[https://colab.research.google.com/github/idealo/image-super-resolution/blob/master/notebooks/ISR\\_Training\\_Tutorial.ipynb](https://colab.research.google.com/github/idealo/image-super-resolution/blob/master/notebooks/ISR_Training_Tutorial.ipynb)

## 超分辨率项目怎么用

你可以选择两种方式安装图像超分辨率（ISR）包。

从 PyPI 中安装 ISR（推荐）：

```
pip install ISR
```

从 GitHub 源安装 ISR：

```
git clone https://github.com/idealo/image-super-resolution
cd image-super-resolution
python setup.py install
```

## 预测

如果我们需要扩展低像素图像，简单两步就能借助 ISR 执行超分辨率。首先加载图像并做一定的预处理：

```
import numpy as np
from PIL import Image

img = Image.open('data/input/test_images/sample_image.jpg')
lr_img = np.array(img)/255.
lr_img = np.expand_dims(lr_img, axis=0)
```

加载模型并执行预测:

```
from ISR.models import RDN

rdn = RDN(arch_params={'C':6, 'D':20, 'G':64, 'G0':64, 'x':2})
rdn.model.load_weights('weights/rdn-C6-D20-G64-G064-x2_enhanced-e219.hdf5')

sr_img = rdn.model.predict(lr_img)[0]
sr_img = sr_img.clip(0, 1) * 255
sr_img = np.uint8(sr_img)
Image.fromarray(sr_img)
```

## 训练

如果需要使用你的数据集重新训练超分辨率模型，那我们也只需要改一改参数。如下首先创建模型:

```
from ISR.models import RRDN
from ISR.models import Discriminator
from ISR.models import Cut_VGG19

lr_train_patch_size = 40
layers_to_extract = [5, 9]
scale = 2
hr_train_patch_size = lr_train_patch_size * scale

rrdn = RRDN(arch_params={'C':4, 'D':3, 'G':64, 'G0':64, 'T':10, 'x':scale}, patch_size=
f_ext = Cut_VGG19(patch_size=hr_train_patch_size, layers_to_extract=layers_to_extract)
discr = Discriminator(patch_size=hr_train_patch_size, kernel_size=3)
```

创建 Trainer 对象，并将训练的各种配置传递到该对象中:

```
from ISR.train import Trainer

loss_weights = {
    'generator': 0.0,
    'feat_extr': 0.0833,
    'discriminator': 0.01,
}

trainer = Trainer(
```

```
generator=rrdn,  
discriminator=discr,  
feature_extractor=f_ext,  
lr_train_dir='low_res/training/images',  
hr_train_dir='high_res/training/images',  
lr_valid_dir='low_res/validation/images',  
hr_valid_dir='high_res/validation/images',  
loss_weights=loss_weights,  
dataname='image_dataset',  
logs_dir='./logs',  
weights_dir='./weights',  
weights_generator=None,  
weights_discriminator=None,  
n_validation=40,  
lr_decay_frequency=30,  
lr_decay_factor=0.5,  
T=0.01,  
)
```

开始训练：

```
trainer.train(  
    epochs=80,  
    steps_per_epoch=500,  
    batch_size=16,  
)
```

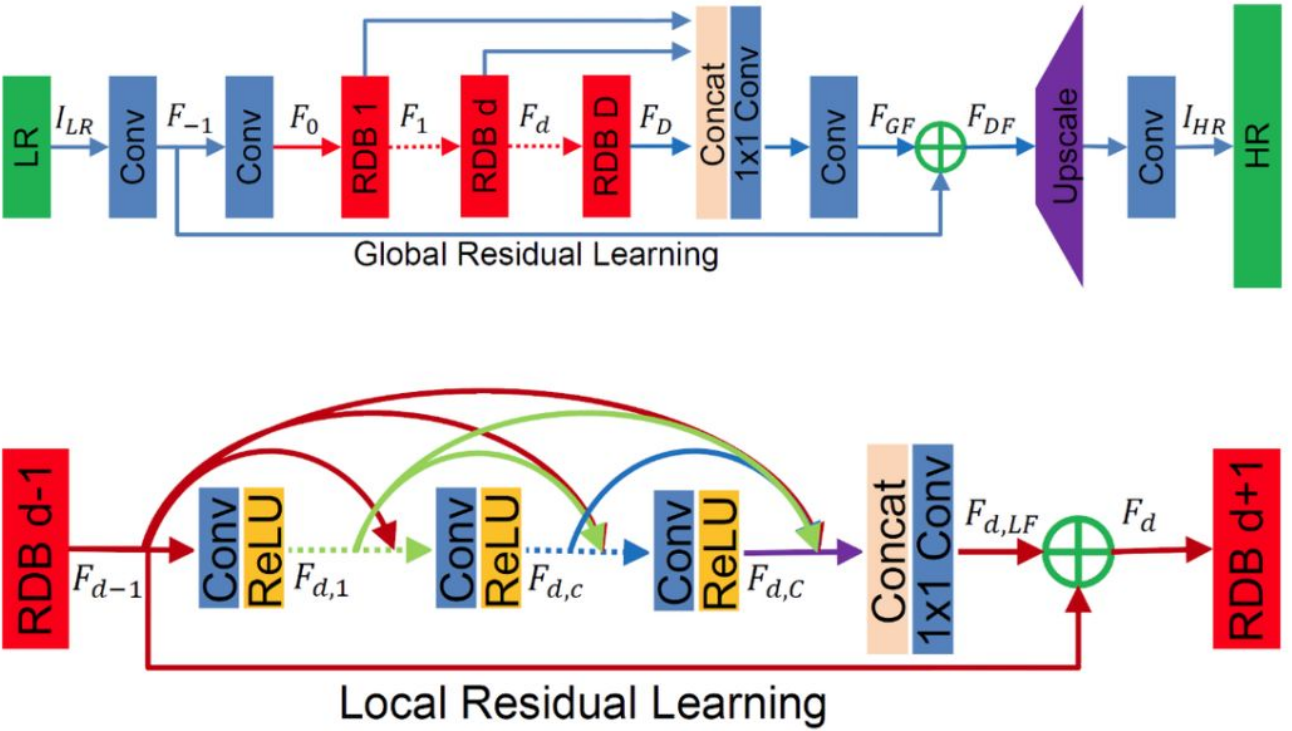
## 网络架构与超参数

实际上，如果我们需要重新训练，那么还需要了解具体的参数都表示什么。这一部分介绍了各超分辨率网络的架构与对应超参数。

### RDN 网络架构

RDN 网络架构的主要参数如下：

- D：残差密集块（RDB）数量
- C：RDB 内部堆叠的卷积层数量
- G：RDB 内部每一卷积层的特征图数量



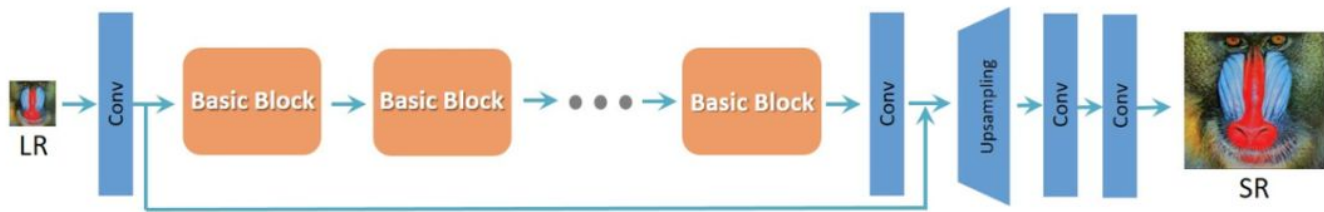
图源：<https://arxiv.org/abs/1802.08797>

RRDN 网络架构

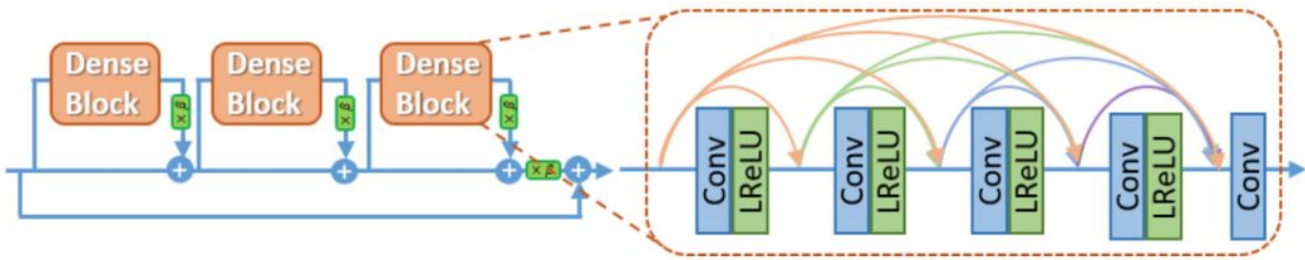
RRDN 架构的主要参数如下：

- T：残差密集块内的残差数量（RRDB）
- D：每一 RRDB 内部的残差密集块（RDB）的数量
- C：RDB 内部堆叠的卷积层数量
- G：RDB 内部每一卷积层的特征图数量





Residual in Residual Dense Block (RRDB)



图源：<https://arxiv.org/abs/1809.00219>

本文为机器之心编译，转载请联系本公众号获得授权。

✂-----

加入机器之心（全职记者 / 实习生）：[hr@jiqizhixin.com](mailto:hr@jiqizhixin.com)

投稿或寻求报道：[content@jiqizhixin.com](mailto:content@jiqizhixin.com)

广告 & 商务合作：[bd@jiqizhixin.com](mailto:bd@jiqizhixin.com)

喜欢此内容的人还喜欢

瞄准GPT-3落地难题，首个千亿中文大模型「盘古」问世，专攻企业级应用  
机器之心

市场监管总局依法对美团涉嫌垄断行为立案调查  
侠客岛

你为什么总是羡慕别人？  
不打烊画社

