遥感图像分类

1.3 VGGNet

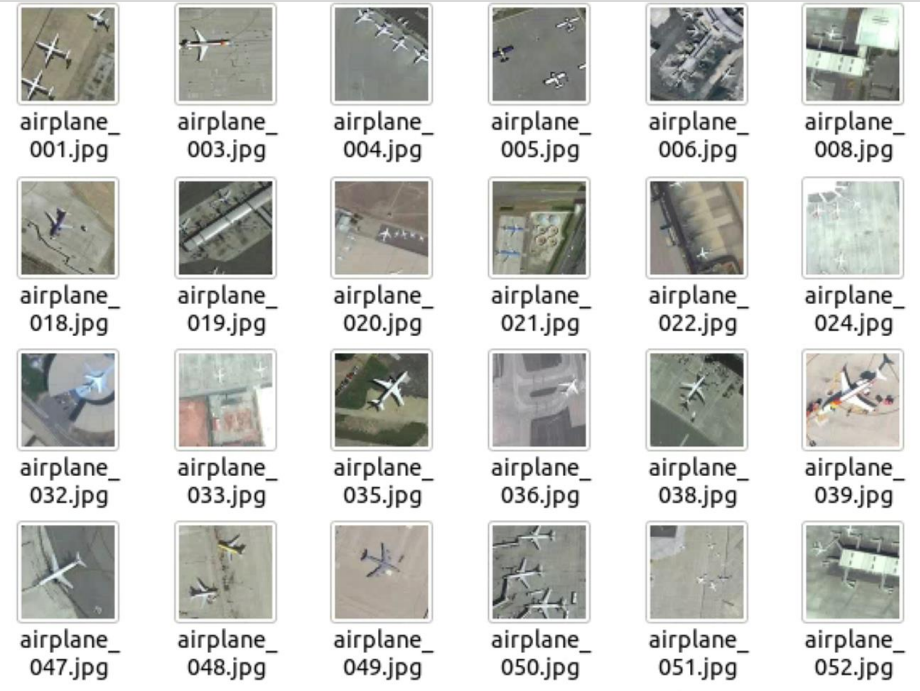
@tm9161

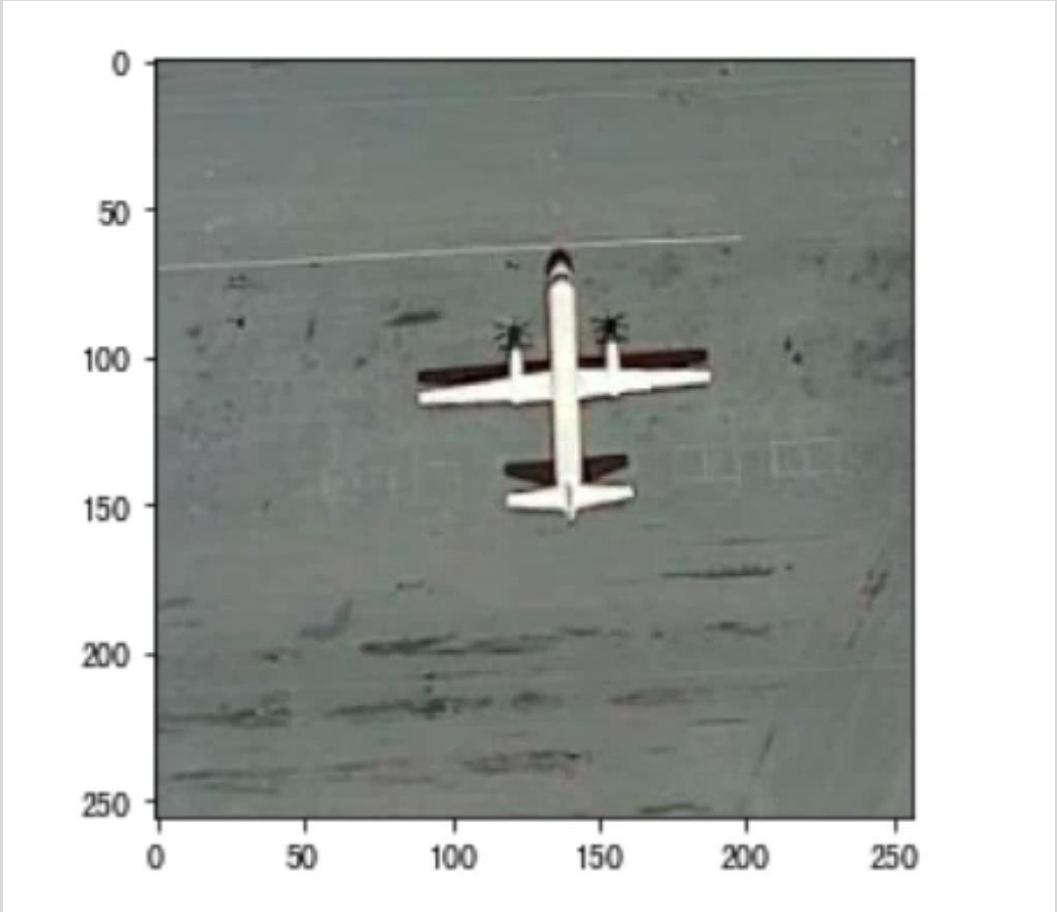
L3

VGGNet

1. VGGNet 结构与创新
2. VGGNet 训练与预测
3. 迁移学习训练VGGNet

遥感图像数据集





包含31500张遥感图像（45类*700张），256x256像素的彩色图。

本次使用其中的5类，划分每类630张为训练集，70张为测试集。

载入数据

1.按路径读取

2.预处理
a.归一化
b.水平翻转
c.批大小
d.随机
e.尺寸
f.独热编码

```
train_dir = 'sat2/train'
test_dir = 'sat2/val'

im_size = 224
batch_size = 32
```

```
train_images = ImageDataGenerator(rescale = 1/255,horizontal_flip=True)
test_images = ImageDataGenerator(rescale = 1/255)
#归一化
```

```
train_gen = train_images.flow_from_directory(directory=train_dir,
                                             batch_size=batch_size,
                                             shuffle=True,
                                             target_size=(im_size, im_size),
                                             class_mode='categorical')
#按路径载入图片，批处理大小，随机，尺寸，读热编码
```

```
Found 3150 images belonging to 5 classes.
```

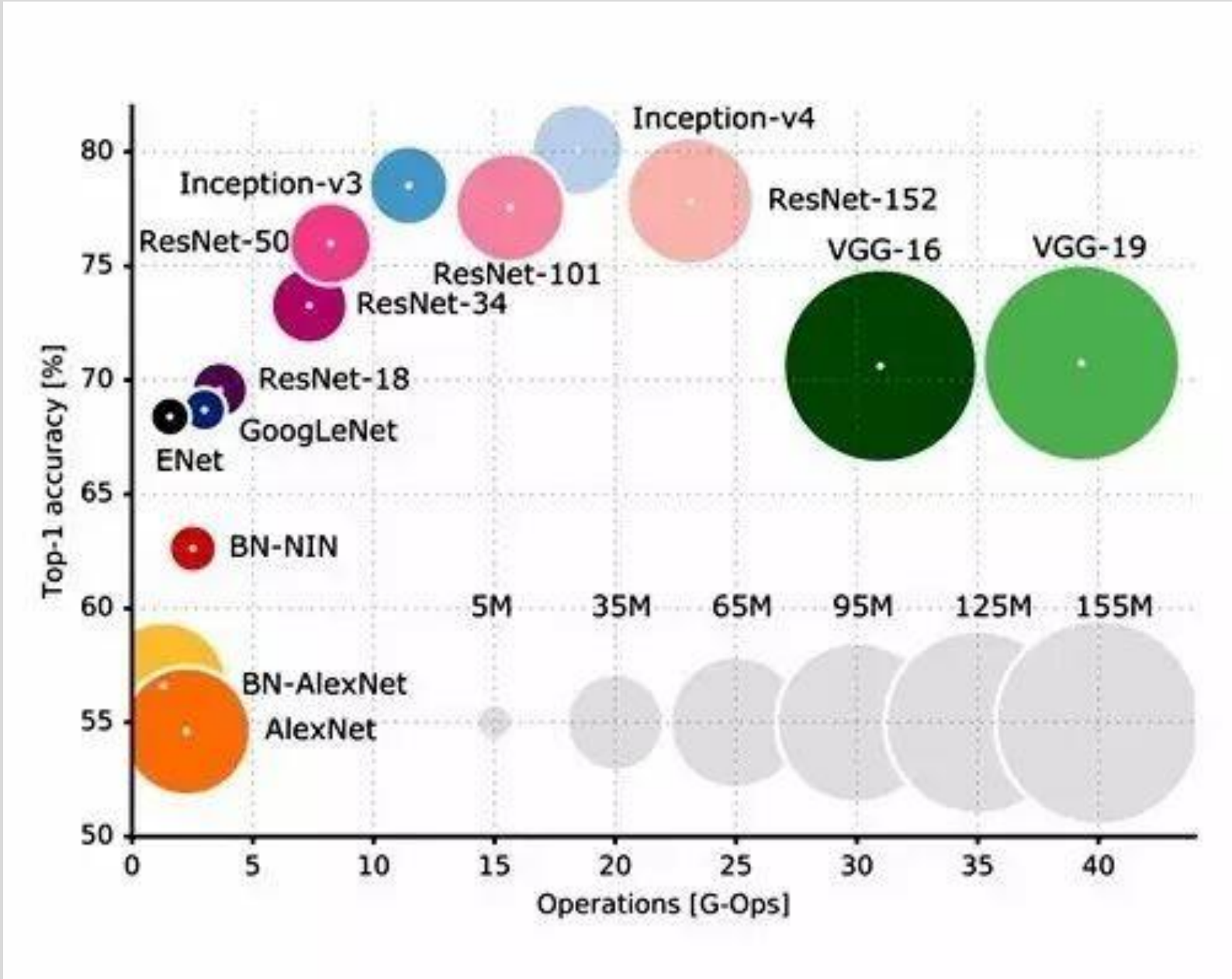
```
val_gen = test_images.flow_from_directory(directory=test_dir,
                                          batch_size=batch_size,
                                          shuffle=False,
                                          target_size=(im_size, im_size),
                                          class_mode='categorical')
#按路径载入图片，批处理大小，随机，尺寸，读热编码
```

```
Found 350 images belonging to 5 classes.
```

ILSVRC

ImageNet Large Scale Visual Recognition Challenge是近年来机器视觉领域最受追捧也是最具权威的学术竞赛之一，代表了图像领域的最高水平。

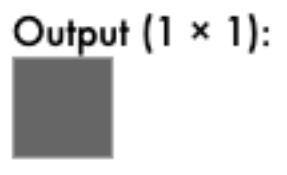
2014年 ImageNet挑战赛的亚军是来自牛津大学视觉几何团队的VGGNet。这个卷积网络是一个**简单而优雅**的架构，只有7.3%的误差率。



感受视野 Receptive Field

定义：输出层一个元素对应输入层区域的大小。
计算：感受视野=（上一层感受视野 - 1）*步长 +卷积核尺寸

VGGNet提出：
堆叠两个3*3卷积核替代一个5*5卷积核；
堆叠三个3*3卷积核替代一个7*7卷积核。



相同感受视野，训练参数量减少。

https://ezyang.github.io/convolution-visualizer/index.html

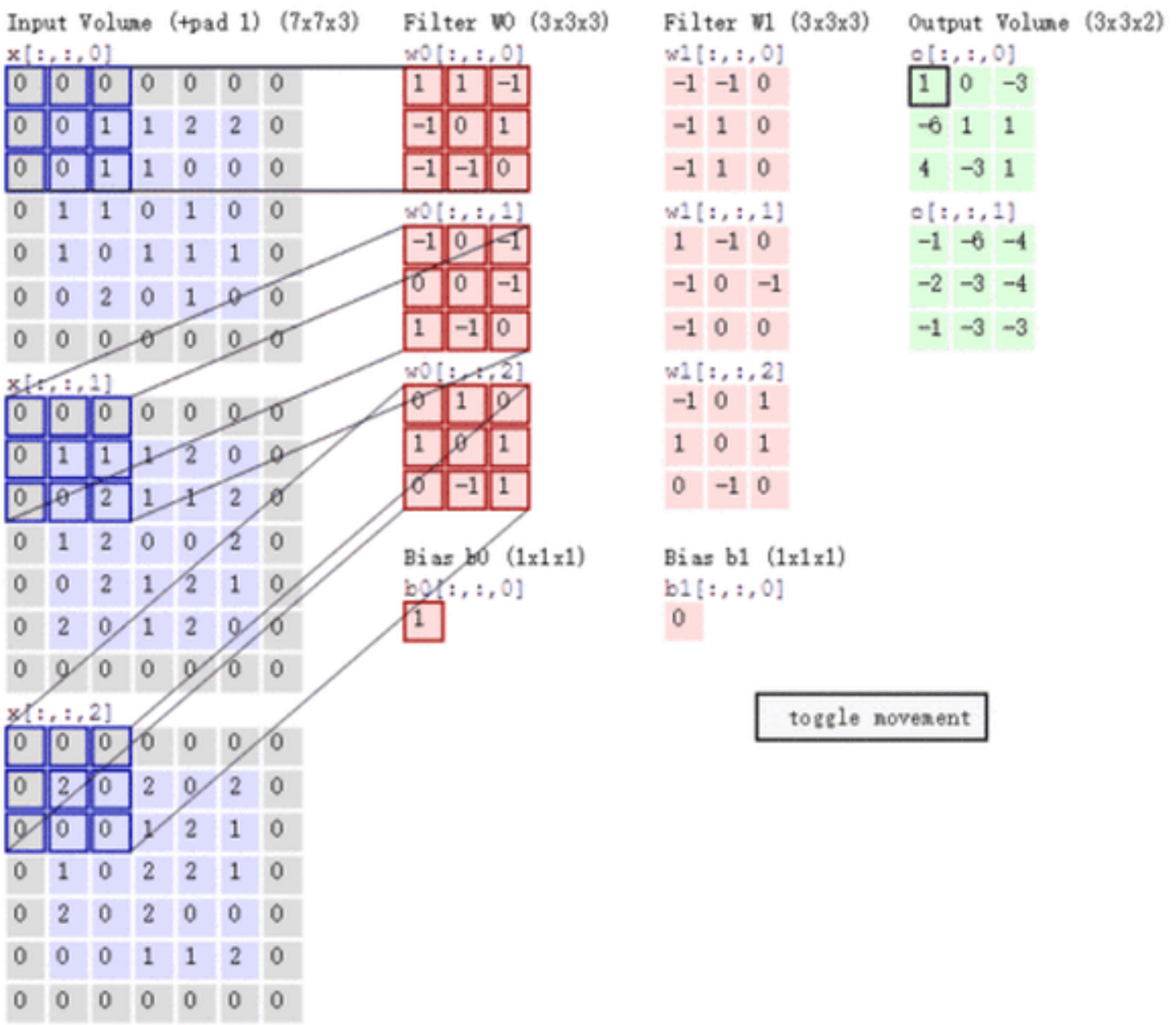
参数 Param

卷积层：
（卷积参数（卷积核各部分）+偏置参数）*卷积核的个数

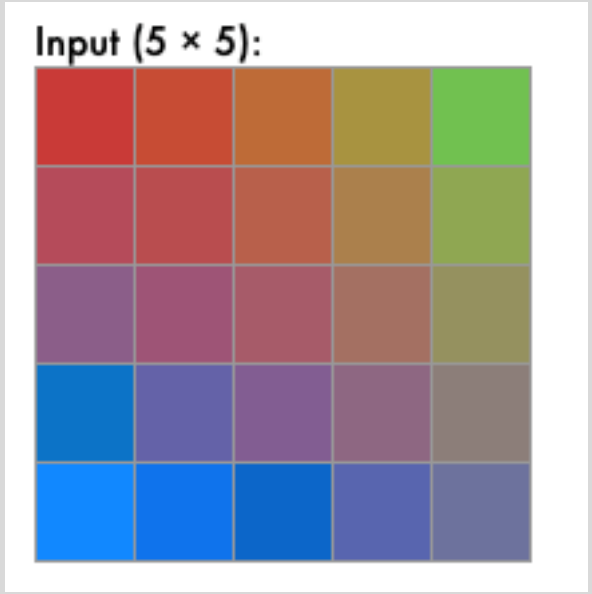
（3层*3x3大小 +1*偏置）*2个卷积核

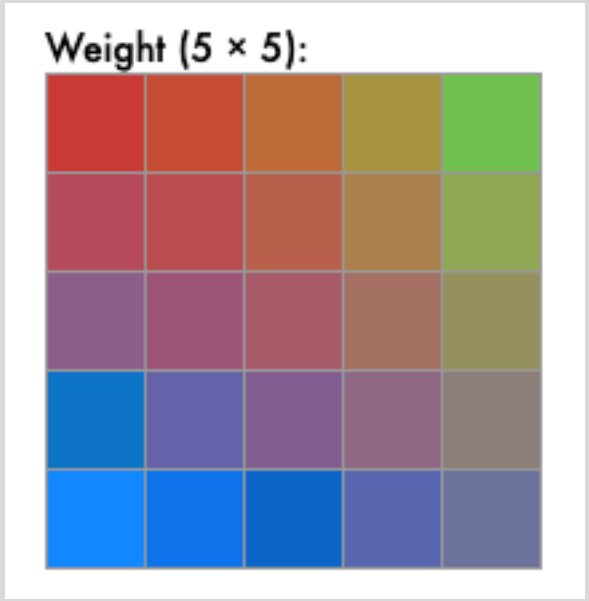
池化层：无需要训练参数

全连接层：神经元连接权重+偏置参数



感受视野 Receptive Field



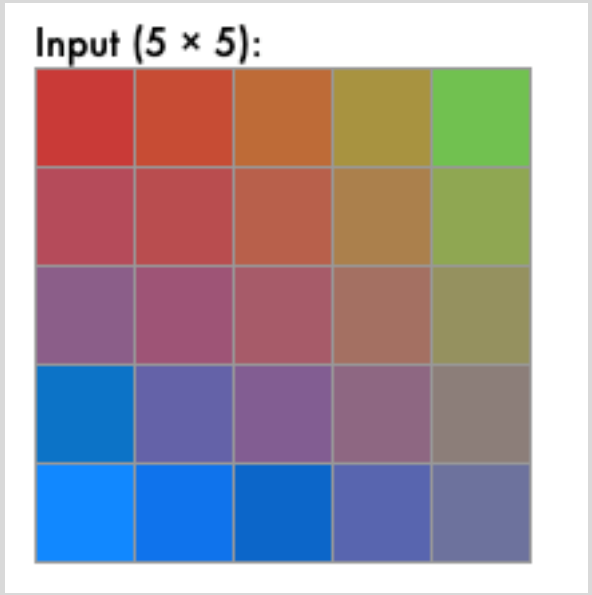


卷积后尺寸 = （输入图像大小-卷积核大小+加边像素数）/步长 +1
卷积后尺寸 = （5-5+0）/1 +1 = 1

感受视野 = 5；计算： （1-1）*1+5

参数 = 5*5+1 = 26

感受视野 Receptive Field

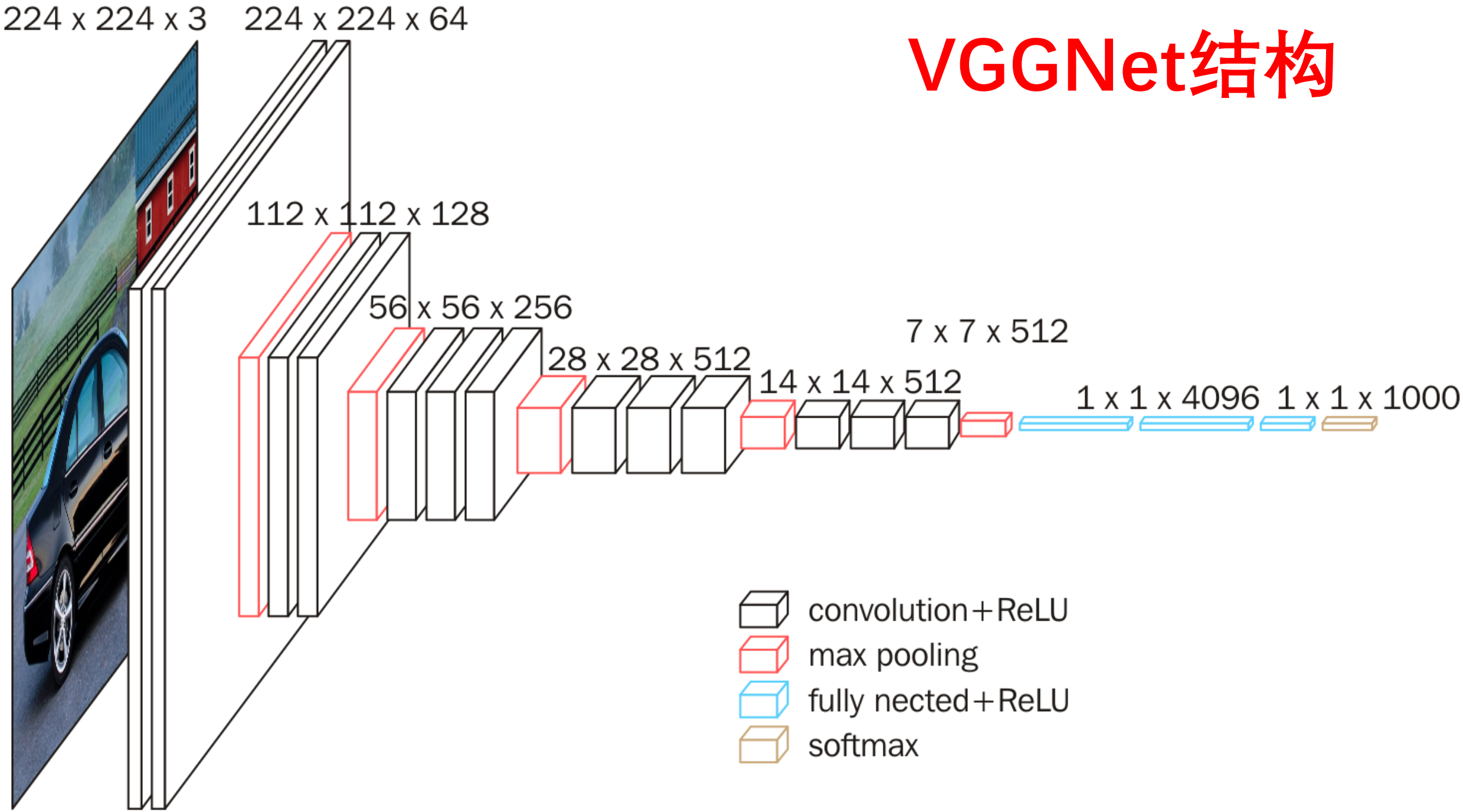


卷积后尺寸 = （输入图像大小-卷积核大小+加边像素数）/步长 +1
卷积后尺寸 = （5-3+0）/1 +1 = 3
卷积后尺寸 = （3-3+0）/1 +1 = 1

感受视野 = 5；计算：层1： （1-1）*1+3 = 3 ； 层2： （3-1）*1+3 = 5

参数 = （3*3+1 ）*2 = 20

VGGNet结构


224 x 224 x 3
224 x 224 x 64
112 x 112 x 128
56 x 56 x 256
28 x 28 x 512
14 x 14 x 512
7 x 7 x 512
1 x 1 x 4096
1 x 1 x 1000
convolution+ReLU
max pooling
fully nected+ReLU
softmax

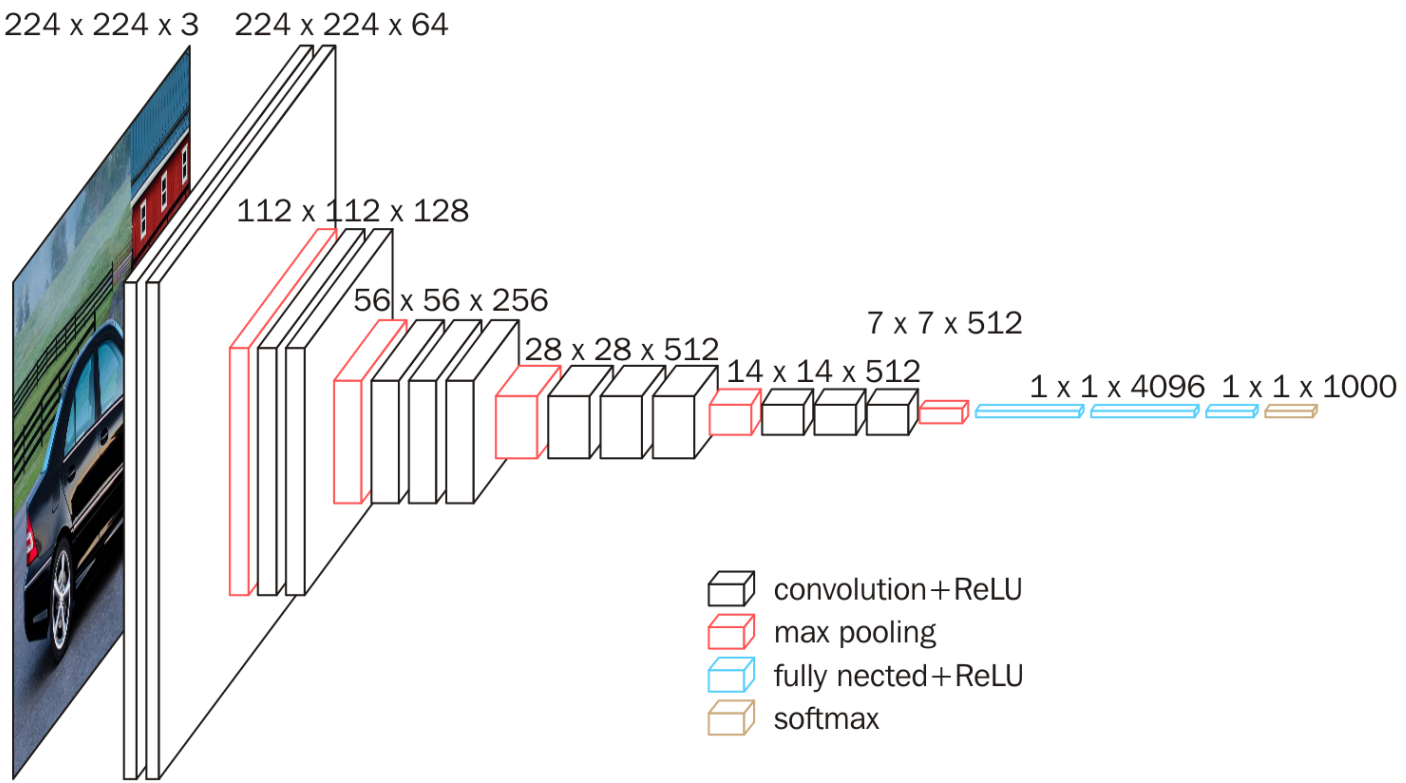
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

conv3：卷积核大小为3*3，步长为1，加边为1。

输出 = （224 – 3 + 2*1）/1 +1 =224 （长宽不变）

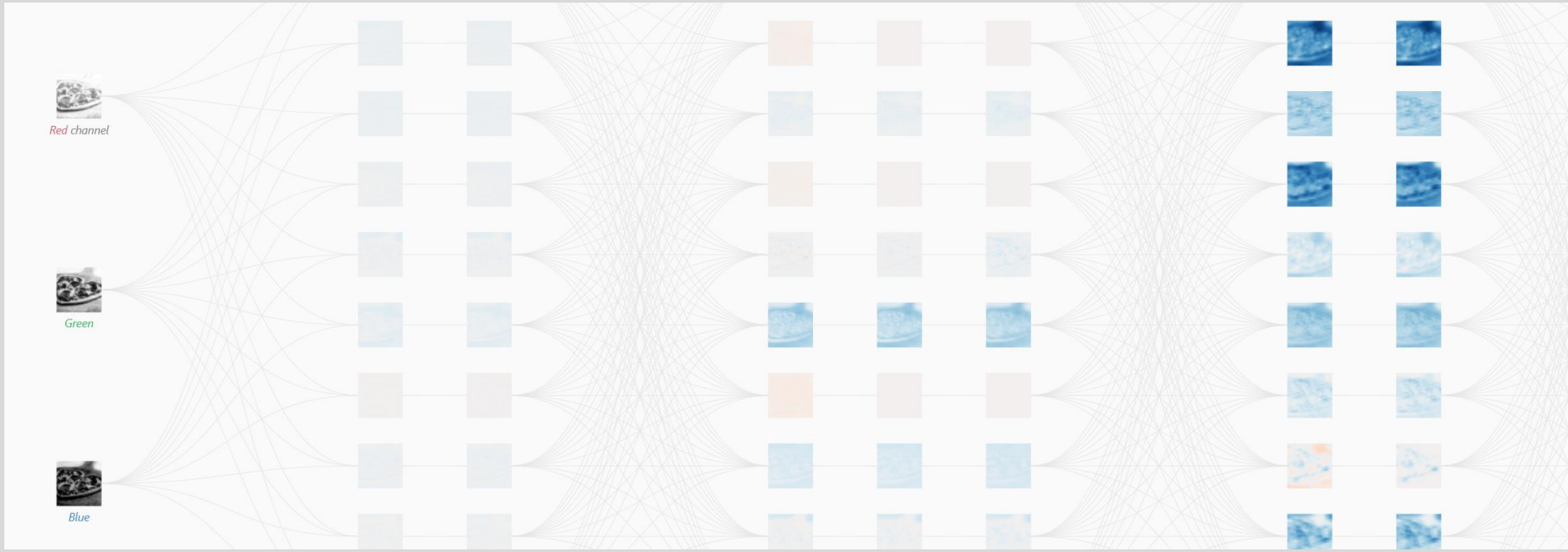
maxpool：池化核大小为2*2，步长为2。

输出 = （224 – 2 +0）/2 +1 = 112（长宽减半）



迁移学习 Transfer Learning

启发：卷积过程提取特征，具有通用性。
优势：速度快，小数据集适用复杂模型。



https://poloclub.github.io/cnn-explainer/

迁移学习 Transfer Learning

1.载入预训练模型，训练所有参数。
2.载入预训练模型，固定部分训练参数，训练部分参数。

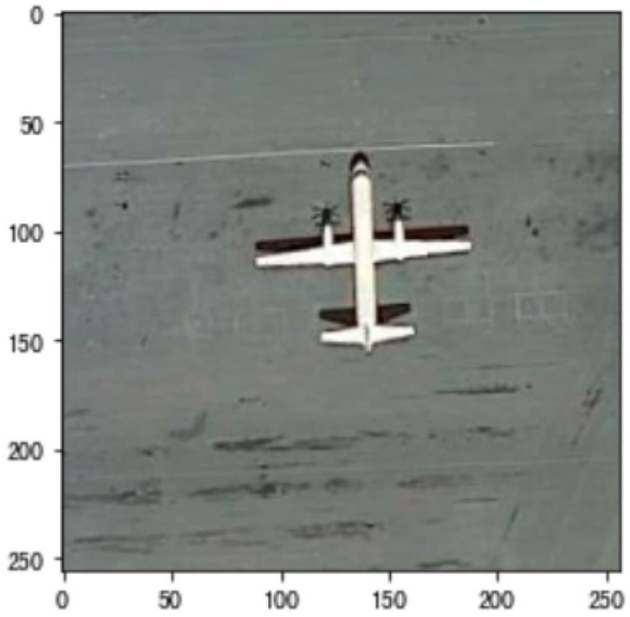
```
vgg = tf.keras.applications.VGG16(include_top=False, weights='imagenet',input_shape=(224, 224, 3))
vgg.trainable = False
#迁移学习 去掉全连接层 加载权重 输入尺寸
```

```
model = tf.keras.Sequential()
#VGG-11/16
```

```
model.add(vgg)
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Dense(1024, activation='relu'))
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Dense(1024, activation='relu'))
model.add(tf.keras.layers.Dense(5, activation='softmax'))
```

图片读取&预处理

```
In [3]: img = cv2.imread('1.jpg',1)
        #读取图片

In [4]: plt.imshow(img)

Out[4]: <matplotlib.image.AxesImage at 0x7fdb782b0ac0>
```



1.图片读取：cv2.imread

2.图片大小调整：cv2.resize

3.图片维度调整：reshape

4.归一化：/255

```
In [5]: img.shape

Out[5]: (256, 256, 3)

In [6]: img = cv2.resize(img,(224,224))
        img = img.reshape(1,224,224,3)
        img = img/255
        #图片预处理

In [7]: img.shape

Out[7]: (1, 224, 224, 3)
```

模型预测

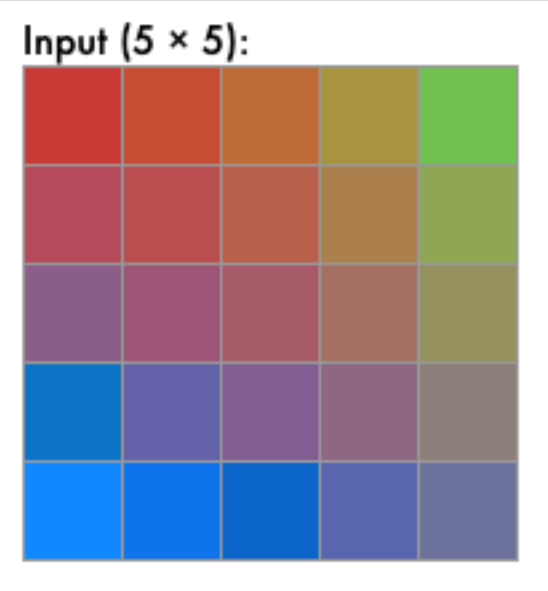
```
In [8]: predict = new_model.predict(img)

In [9]: predict
Out[9]: array([[9.9936479e-01, 7.3952382e-09, 2.1086180e-07, 6.3472008e-04,
                1.7892945e-07]], dtype=float32)

In [10]: label = ['airplane','bridge','palace','ship','stadium']

In [11]: label[np.argmax(predict)]
Out[11]: 'airplane'
```

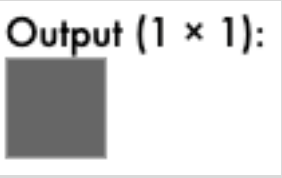
总结：

1. 堆叠两个3*3卷积核替代5*5卷积核；堆叠三个3*3卷积核替代7*7卷积核。目的：相同感受视野，减少参数量。

2. 增加网络深度，提升性能。

3. 计算资源问题。





总结：

4. 迁移学习：载入预训练模型，固定卷积层，训练全连接层。

```
vgg = tf.keras.applications.VGG16(include_top=False, weights='imagenet',input_shape=(224, 224, 3))
vgg.trainable = False
#迁移学习 去掉全连接层 加载权重 输入尺寸
```

```
model = tf.keras.Sequential()
#VGG-11/16
```

```
model.add(vgg)
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Dense(1024, activation='relu'))
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Dense(1024, activation='relu'))
model.add(tf.keras.layers.Dense(5, activation='softmax'))
```

参考资料：

1. VGG网络详解及感受野的计算
https://www.bilibili.com/video/BV1q7411T7Y6

2. Very Deep Convolutional Networks for Large-Scale Visual Recognition
https://www.robots.ox.ac.uk/~vgg/research/very_deep/

3. Transfer Learning
https://cs231n.github.io/transfer-learning

4. What is a Convolutional Neural Network?
https://poloclub.github.io/cnn-explainer/