**Machine Learning**

OpenCV

OpenCV is a cross-platform computer vision library that runs on Linux, Windows and Mac OS operating systems. It's lightweight and efficient -- it's made up of a bunch of C functions and a handful of C++ classes, but it also provides a Python interface to implement many general algorithms in image processing and computer vision. The following code tries to use some simple filters, including image smoothing, Gaussian blur, and so on.

### **Scikit-image**

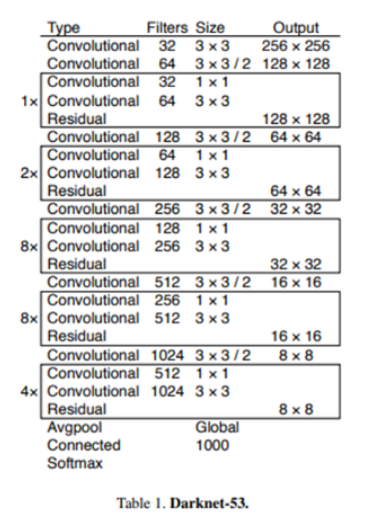
scikit-image is an image processing library based on scipy that processes images as numpy arrays. For example, you can change the scale of an image using scikit-image, which provides functions such as rescale, resize, and downscale local mean.



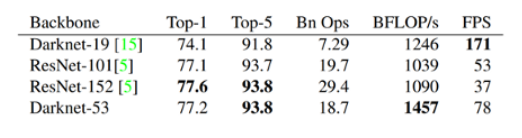
**Yolov3**

Based on YOLOv2, YOLOv3 improved the backbone of the network, used multi-scale feature graphs for detection, and improved multiple independent Logistic regression classifiers to replace softmax for class classification prediction.

YOLOv3 proposed a new backbone network: Darknet-53, from the 0th layer to the 74th layer, a total of 53 convolutional layers, the rest are Resnet layers. Compared with Darknet-19, Darknet-53 removed all maxpooling Layer, adding more 1X1 and 3X3 convolutional layers, but because deepening the number of network layers can easily cause the gradient to disappear or explode, so Darknet-53 has added the Residual block in ResNet to solve the gradient problem. From the following figure The Darknet-53 architecture can see that a total of 23 residual blocks have been added.



Due to the deepening of the network layers, Darknet-53 is much slower than Darknet-19, but the processing speed of Darknet-53 is 78fps, which is still much faster than ResNet with the same precision, and YOLO3 still maintains high performance.



YOLOv3 draws on the FPN method and uses multi-scale feature maps to detect objects of different sizes to improve the prediction ability of small objects.

