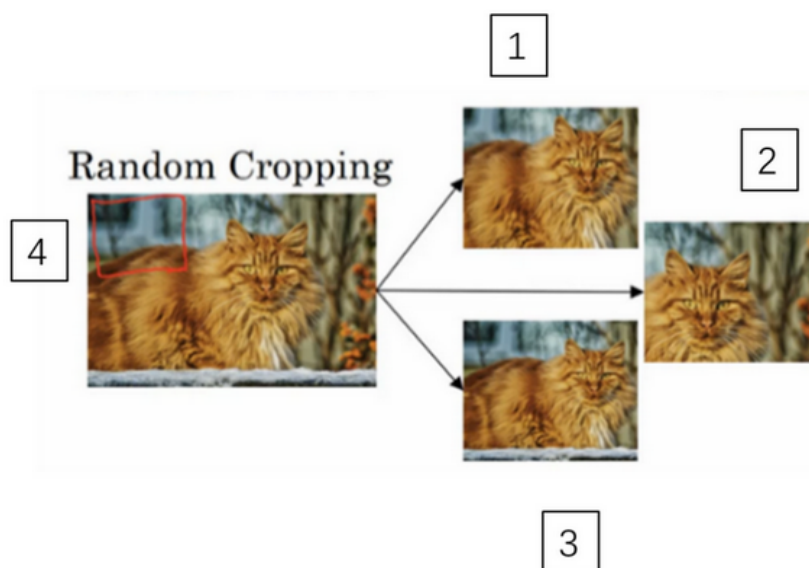


2.10 数据增强（Data augmentation）

大部分的计算机视觉任务使用很多的数据，所以数据增强是经常使用的一种技巧来提高计算机视觉系统的表现。我认为计算机视觉是一个相当复杂的工作，你需要输入图像的像素值，然后弄清楚图片中有什么，似乎你需要学习一个复杂方程来做这件事。在实践中，更多的数据对大多数计算机视觉任务都有所帮助，不像其他领域，有时候得到充足的数据，但是效果并不怎么样。但是，当下在计算机视觉方面，**计算机视觉的主要问题是没办法得到充足的数据**。对大多数机器学习应用，这不是问题，但是对计算机视觉，数据就远远不够。所以这就意味着当你训练计算机视觉模型的时候，数据增强会有所帮助，这是可行的，无论你是使用迁移学习，使用别人的预训练模型开始，或者从源代码开始训练模型。让我们来看一下计算机视觉中常见的数据增强的方法。

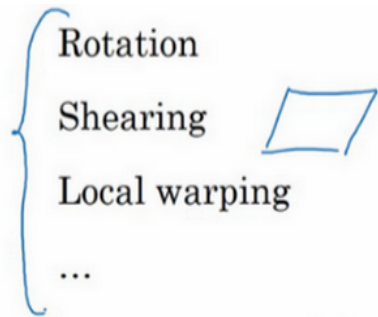


或许最简单的数据增强方法就是**垂直镜像对称**，假如，训练集中有这张图片，然后将其翻转得到右边的图像。对大多数计算机视觉任务，左边的图片是猫，然后镜像对称仍然是猫，如果镜像操作保留了图像中想识别的物体的前提下，这是个很实用的数据增强技巧。



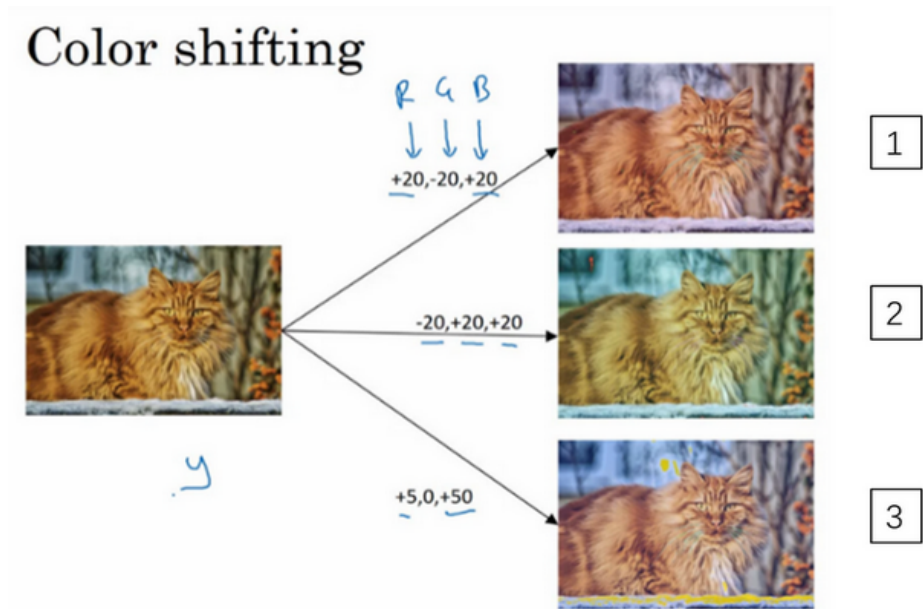
另一个经常使用的技巧是**随机裁剪**，给定一个数据集，然后开始随机裁剪，可能修剪这个（编号 1），选择裁剪这个（编号 2），这个（编号 3），可以得到不同的图片放在数据集中，

你的训练集中有不同的裁剪。随机裁剪并不是一个完美的数据增强的方法，如果你随机裁剪的那一部分（红色方框标记部分，编号 4），这部分看起来不像猫。但在实践中，这个方法还是很实用的，随机裁剪构成了很大一部分的真实图片。



镜像对称和随机裁剪是经常被使用的。当然，理论上，你也可以使用旋转，剪切（**shearing**：此处并非裁剪的含义，图像仅水平或垂直坐标发生变化）图像，可以对图像进行这样的扭曲变形，引入很多形式的局部弯曲等等。当然使用这些方法并没有坏处，尽管在实践中，因为太复杂了所以使用的很少。

第二种经常使用的方法是**彩色转换**，有这样一张图片，然后给 **R**、**G** 和 **B** 三个通道加上不同的失真值。

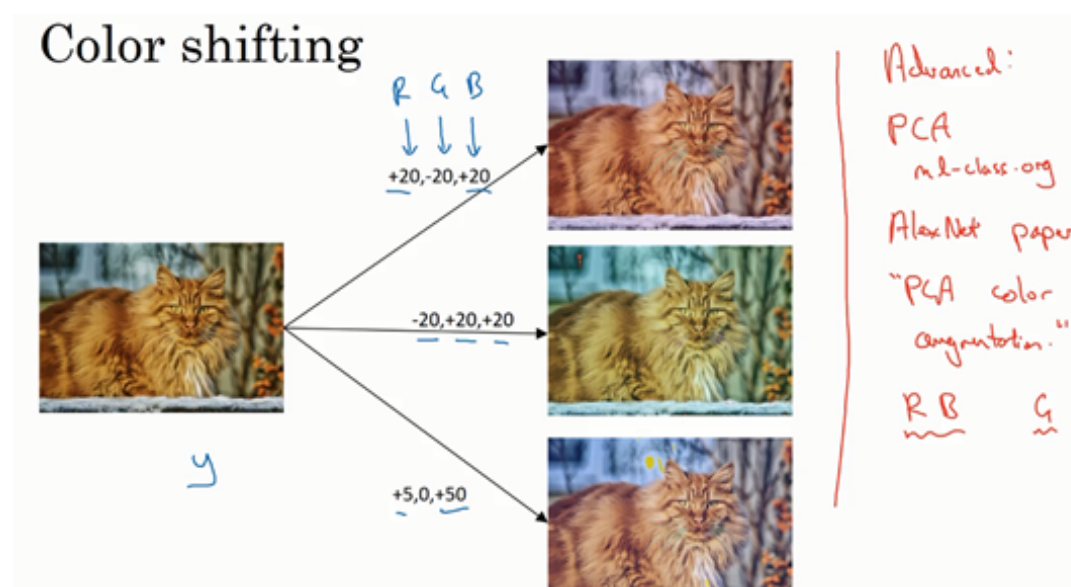


在这个例子中（编号 1），要给红色、蓝色通道加值，给绿色通道减值。红色和蓝色会产生紫色，使整张图片看起来偏紫，这样训练集中就有失真的图片。为了演示效果，我对图片的颜色进行改变比较夸张。在实践中，对 **R**、**G** 和 **B** 的变化是基于某些分布的，这样的改变也可能很小。

这么做的目的就是使用不同的 **R**、**G** 和 **B** 的值，使用这些值来改变颜色。在第二个例子

中（编号 2），我们少用了一点红色，更多的绿色和蓝色色调，这就使得图片偏黄一点。

在这（编号 3）使用了更多的蓝色，仅仅多了点红色。在实践中，**R**、**G** 和 **B** 的值是根据某种概率分布来决定的。这么做的理由是，可能阳光会有一点偏黄，或者是灯光照明有一点偏黄，这些可以轻易的改变图像的颜色，但是对猫的认可，或者是内容的认可，以及标签 **y**，还是保持不变的。所以介绍这些，颜色失真或者是颜色变换方法，这样会使得你的学习算法对照片的颜色更改更具鲁棒性。

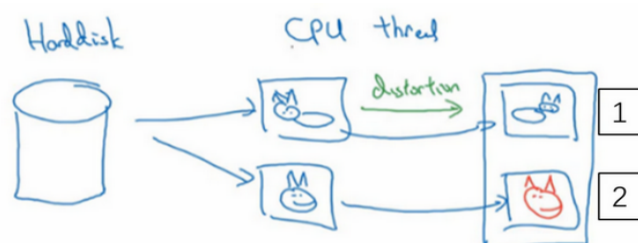


这是对更高级的学习者的一些注意提醒，你可以不理解我用红色标出来的内容。对 **R**、**G** 和 **B** 有不同的采样方式，其中一种影响颜色失真的算法是 **PCA**，即主成分分析，我在机器学习的 **mooc** 中讲过，在 **Coursera ml-class.Org** 机器学习这门课中。但具体颜色改变的细节在 **AlexNet** 的论文中有时被称作 **PCA** 颜色增强，**PCA** 颜色增强的大概含义是，比如说，如果你的图片呈现紫色，即主要含有红色和蓝色，绿色很少，然后 **PCA** 颜色增强算法就会对红色和蓝色增减很多，绿色变化相对少一点，所以使总体的颜色保持一致。如果这些你都不懂，不需要担心，可以在网上搜索你想要了解的东西，如果你愿意的话可以阅读 **AlexNet** 论文中的细节，你也能找到 **PCA** 颜色增强的开源实现方法，然后直接使用它。

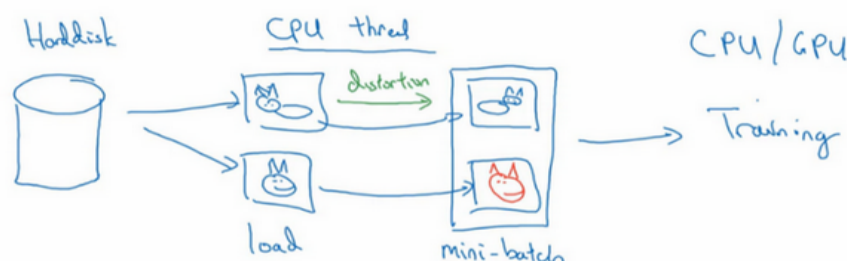


你可能有存储好的数据，你的训练数据存在硬盘上，然后使用符号，这个圆桶来表示你

的硬盘。如果你有一个小的训练数据，你可以做任何事情，这些数据集就够了。



但是你有特别大的训练数据，接下来这些就是人们经常使用的方法。你可能会使用 **CPU** 线程，然后它不停的从硬盘中读取数据，所以你要有一个从硬盘过来的图片数据流。你可以用 **CPU** 线程来实现这些失真变形，可以是随机裁剪、颜色变化，或者是镜像。但是对每张图片得到对应的某一种变形失真形式，看这张图片（编号 1），对其进行镜像变换，以及使用颜色失真，这张图最后会颜色变化（编号 2），从而得到不同颜色的猫。



与此同时，**CPU** 线程持续加载数据，然后实现任意失真变形，从而构成批数据或者最小批数据，这些数据持续的传输给其他线程或者其他的进程，然后开始训练，可以在 **CPU** 或者 **GPU** 上实现训一个大型网络的训练。



常用的实现数据增强的方法是使用一个线程或者是多线程，这些可以用来加载数据，实现变形失真，然后传给其他的线程或者其他进程，来训练这个（编号 2）和这个（编号 1），可以并行实现。

这就是数据增强，与训练深度神经网络的其他部分类似，在数据增强过程中也有一些超

参数，比如说颜色变化了多少，以及随机裁剪的时候使用的参数。与计算机视觉其他部分类似，一个好的开始可能是使用别人的开源实现，了解他们如何实现数据增强。当然如果你想获得更多的不变特性，而其他人的开源实现并没有实现这个，你也可以去调整这些参数。因此，我希望你们可以使用数据增强使你的计算机视觉应用效果更好。