# TensorFlow和DLTK在生物医学图像分析中的应用

作者：Martin Rajchl，S.Ira Ktena和Nick Pawlowski - 伦敦帝国学院

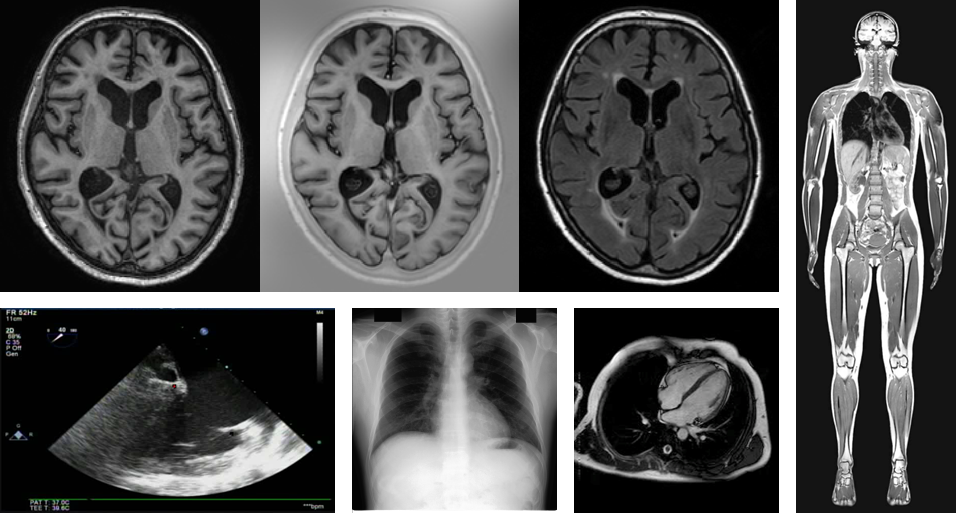
，医学影像学深度学习工具包扩展到可以对生物医学影像进行深度学习。它提供专业的操作和功能、模型的实现（如本博客所用）和。

这篇博客文章是对生物医学图像深度学习的一个快速介绍，在这里我们将演示当前工程问题的一些问题和解决方案，并向您展示如何为您的问题建立和运行一个原型。



### 概述

什么是生物医学图像分析？为什么需要它？生物医学图像是对人体在不同尺度（即微观、宏观等）上的测量。它们有多种成像方式（如CT扫描仪、超声波机等），可测量人体的物理特性（如放射密度、X射线的不透明度）。这些图像由领域专家（如放射科医生）为临床任务（如诊断）进行解释，并对医生的决策产生重大影响。



生物医学图像通常是体积图像（3D），有时具有额外的时间维度（4D）和/或多通道（4-5D）（例如多序列MR图像）。生物医学图像的变化与自然图像（如照片）的变化大不相同，因为临床方案的目的是分层图像的获取方式（如患者仰卧、头部不倾斜等）。在他们的分析中，我们的目的是检测细微的差异（即一些小区域指示异常发现）。

为什么是计算机视觉和机器学习？长期以来，计算机视觉方法被用来自动分析生物医学图像。最近出现的深度学习已经取代了许多其他机器学习方法，因为它避免了手工工程特征的创建，从而从过程中消除了一个关键的错误源。此外，GPU的快速推理速度加快了全网络的速度，使我们能够将分析扩展到前所未有的数据量（例如）。

我们能在生物医学成像中使用深度学习库吗？为什么要创建DLTK？创建的主要原因是将此域的专用工具包括在外。虽然许多深度学习库向开发人员公开了低级操作（例如张量乘法等），但由于它们在体积图像（例如可微的三维上采样层等）上的使用而缺少许多高级专业操作，并且由于图像的附加空间维度，我们可以运行到内存中问题（例如，在float32中存储1k个CT图像数据库的单个副本，图像尺寸为512x512x256个体素，约268gb）。由于采集的性质不同，有些图像需要特殊的预处理（如强度归一化、偏置场校正、去噪、空间归一化/配准等）。

### 文件格式、标题和读取图像

虽然许多成像设备供应商以标准格式生成图像，将体积保存在一系列二维切片中，但许多分析库依赖于更适合于计算和与医学图像交互的格式。我们使用（或.nii格式），最初是为大脑成像开发的，但在DLTK和本教程中广泛用于大多数其他体积图像。此格式和其他格式保存的信息是重建图像容器并将其定向到物理空间所必需的信息。

为此，它需要专业标题信息，我们将通过几个属性进行深入学习：

* 尺寸和大小存储有关如何重建图像的信息（例如，使用尺寸向量将体积转换为三维）。
* 数据类型
* 体素间距（也是体素的物理尺寸，通常以mm为单位）
* 物理坐标系原点
* 方向

为什么这些属性很重要？网络将在体素空间中训练，这意味着我们将创建形状和维度的张量[批量大小，dx，dy，dz，通道/特征]，并将其馈送到网络。该网络将在该体素空间中进行训练，并假设所有图像（也就是看不见的测试图像）都在该空间中正常化，或者可能有问题需要推广。在该体素空间中，特征提取器（例如卷积层）将假设体素维度是各向同性的（即在每个维度中都是相同的），并且所有图像的方向都是相同的。

但是，由于大多数图像都是描绘物理空间的，因此我们需要将该物理空间转换为公共体素空间：

如果所有图像的方向都相同（有时我们需要注册以实现图像的空间规格化：签出），我们可以通过

所有这些信息都是存储在.nii头中的向量。

读取.nii images：有几个库可以读取.nii文件并访问头信息并对其进行解析，以获得作为数组的重建图像容器。我们选择了一个python包装库，它允许我们导入额外的图像过滤器来进行预处理和其他任务：

### 数据I/O注意事项

根据训练数据库的大小，有几个选项可以将.nii图像数据输入到网络图中。每种方法在速度方面都有特定的权衡，并且在训练过程中可能成为瓶颈。我们将介绍三种选择：

内存和馈送字典：我们可以创建一个tf.placeholder到网络图，并在训练期间通过feed-dict馈送它。我们从磁盘读取所有.nii文件，用python（c.f.load\_data（））处理它们，并将所有训练示例存储在内存中，我们从中获取：

TLDR：这种直接方法通常是最快和最容易实现的，因为它避免了从磁盘上连续读取数据，但是需要将训练示例（和验证示例）的整个数据库保存在内存中，这对于更大的数据库或更大的图像文件是不可行的。

使用TFRecords数据库：对于图像卷上的大多数深度学习问题，训练示例数据库太大，无法装入内存。TFRecords格式允许序列化训练示例，并将其存储在具有快速写访问（即并行数据读取）的磁盘上：

该格式可以直接与TensorFlow接口，并可以直接集成到tf.graph中的训练循环中：

TLDR:TFRecords是从磁盘访问文件的快速方法，但需要存储整个培训数据库的另一个副本。如果我们的目标是使用几个TB大小的数据库，这可能会令人望而却步。

使用本地python生成器：最后，我们可以使用python生成器，创建read\_fn（）来直接加载图像数据…

以及tf.data.Dataset.from\_generator（）来对示例进行排队：

TLDR：它避免创建图像数据库的额外副本，但是比TFRecords慢得多，因为生成器无法并行读取和映射函数。

Speed benchmarking&选择方法：我们运行这三种方法，将.nii文件读取到TensorFlow，并比较加载和馈送固定大小示例数据库所需的时间。所有代码和结果都可以在这里找到。

最快的方法显然是在5.6秒内通过占位符从内存中输入数据，随后是31.1秒的TFRecords和123.5秒的python生成器从磁盘读取的未优化数据。然而，只要训练过程中的前向/后向传递是计算瓶颈，数据I/O的速度就可以忽略不计。

### 数据规范化

与自然图像一样，我们可以规范化生物医学图像数据，但是方法可能略有不同。标准化的目的是去除已知数据中的一些变化（例如，不同的受试者姿势或图像对比度的差异等），从而简化我们感兴趣的细微差异的检测（例如，病理学的存在）。在这里，我们将讨论最常见的规范化形式：

体素强度标准化：这种形式高度依赖于成像方式，数据是用。典型的是定性图像的标准（例如加权脑MR图像，其中对比度高度依赖于采集参数，通常由专家设置）。如果我们采用这样的统计方法，我们使用的统计数据来自一个完整的单卷，而不是整个数据库。

与此相反，定量成像测量物理量（例如CT成像中的射电密度，其中不同扫描仪的强度是可比较的），并受益于剪裁和/或重新缩放，如（例如到[-1,1]）。

空间标准化：图像方向的标准化避免了模型必须学习所有可能的方向，这在很大程度上减少了所需的训练图像数量（请参见标题属性的重要性以了解图像的方向）。此外，我们还考虑了体素间距，即使是从同一台扫描仪获取的图像，其间距也可能会有所不同。这可以通过重新采样到各向同性分辨率来实现：

如果需要进一步的标准化，我们可以使用医学图像注册包（例如，等等）并将图像注册到同一空间，以便图像之间的体素位置相互对应。分析大脑结构磁共振图像（如T1加权磁共振图像）的一个典型步骤是将训练数据库中的所有图像注册到参考标准，如平均图谱（如图谱）。根据注册方法的自由度，这也可以对大小（仿射注册）或形状（变形注册）进行正规化。这两种变体很少使用，因为它们删除了图像中的一些信息（即形状信息或大小信息），这些信息可能对分析很重要（例如，大心脏可能预示着心脏病）。

### 数据扩充

通常情况下，可用的数据量是有限的，有些变化是不包括在内的。一些例子包括：

* 软组织器官，其中存在广泛的正常形状。
* 病理学，如癌症病变，在形状和位置上有很大的不同
* 免费的超声波图像，那里有很多可能的视图

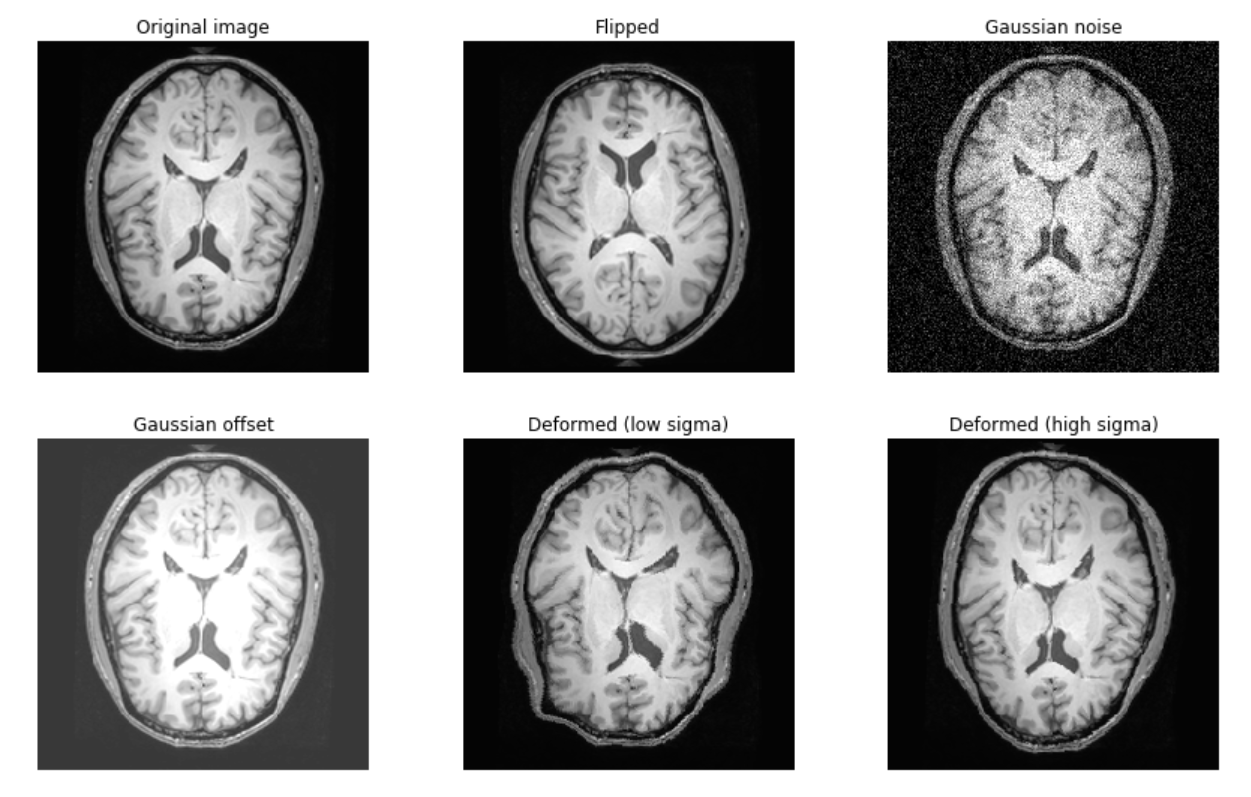
为了正确地推广到不可见的测试用例，我们通过模拟数据的变化来增强训练图像，我们的目标是对这些变化具有鲁棒性。与标准化方法类似，我们区分强度和空间增强：

强度增强示例：

* 在训练图像中加入噪声
* 添加随机偏移或对比度以处理图像之间的差异

空间增强示例：

* 将图像张量向期望对称的方向翻转（例如，在脑部扫描时向左/向右翻转）
* 随机变形（例如用于模拟器官形状的差异）
* 沿轴旋转（例如，用于模拟不同超声视角）
* 随机修剪和补丁训练



关于扩充和数据I/O的重要说明：根据需要或有帮助的扩充，有些操作仅在python中可用（例如），这意味着如果使用使用原始TensorFlow的读取方法（即TFRecords或tf.placeholder），则需要对它们进行预计算并存储到磁盘，从而大大增加了训练数据库的规模。

### 阶级平衡

领域专家解释（例如，手动分割或疾病分类）是从医学图像中监督学习的一项要求。通常，图像级别（例如疾病级别）或体素级别（例如分割）标签在相同的比率下不可用，这意味着网络在训练期间将看不到来自每个级别的相同数量的示例。如果类比率有点相似（例如，对于二元分类情况，为30/70），则这对准确性没有很大影响。然而，由于大多数损失是整个批次的平均成本，网络将首先学习正确预测最常出现的类别（例如，背景或正常情况，这通常是可用的更多示例）。

训练过程中的类不平衡将对罕见现象（如图像分割中的小损伤）产生较大影响，并在很大程度上影响测试精度。

为了避免这种下降，有两种典型的方法来解决数据集中的类不平衡：

* 通过采样实现类平衡：这里，我们的目标是在采样过程中纠正所看到的示例的频率。这可以通过以下方法实现：a）从每个类中抽取等量的样本；b）过度抽取代表类；c）过度抽取不太频繁的类。在DLTK中，我们有一个a）的实现，可以找到。我们对图像卷中的随机位置进行采样，并考虑一个提取的示例（如果它包含我们要查找的类）。
* 通过损失函数进行类平衡：与典型的体素平均损失（例如分类交叉熵、L2等）相比，我们可以a）使用固有平衡的损失函数（例如，所有类的平均骰子系数）或b（例如，中值频率重加权交叉熵）。

### 示例应用程序亮点

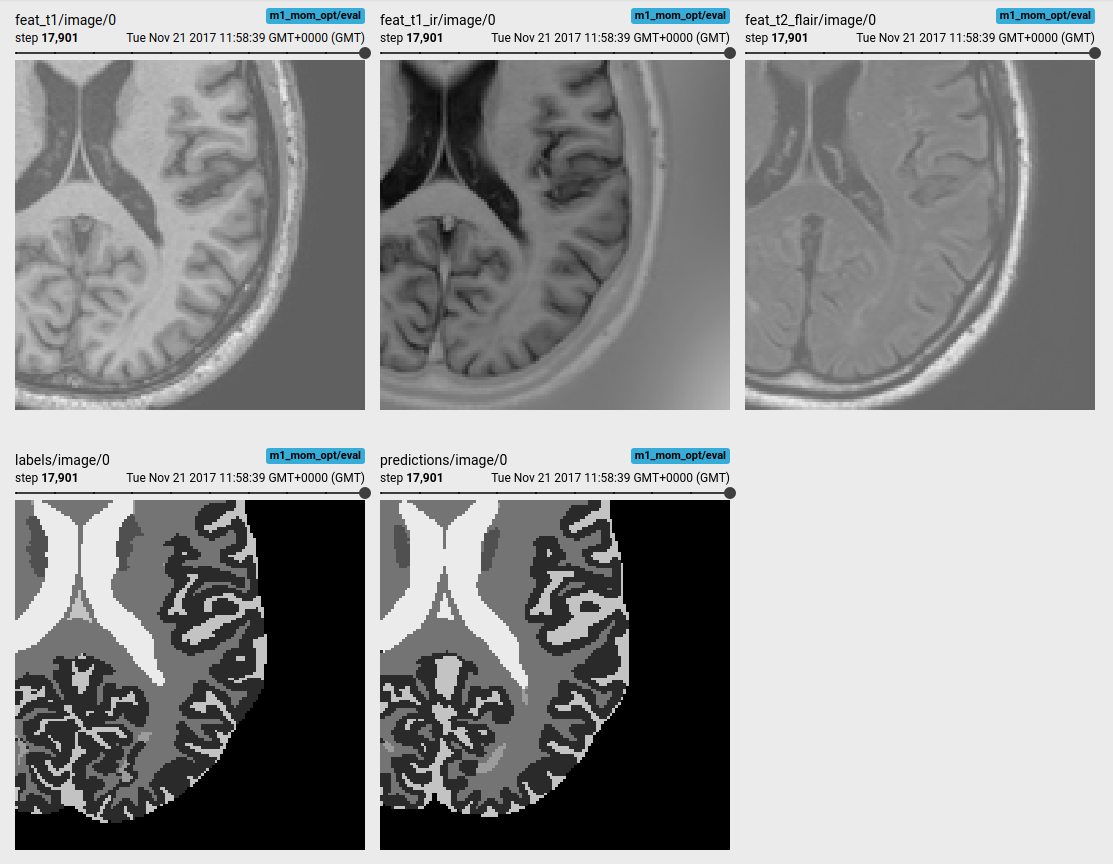
利用本文提供的所有基本知识，我们现在可以研究如何构建完整的应用程序，以便使用TensorFlow深入学习医学图像。我们已经使用深层神经网络实现了几个典型的应用程序，并将介绍其中的一些，让您了解现在可以尝试解决哪些问题。

注意：这些示例应用程序学习了一些有意义的东西，但它们是为演示目的而不是高性能实现而构建的。

#### 示例数据集

我们提供以下所有示例。对于大多数情况（包括上面的演示），我们使用。对于图像分割，我们下载了，您需要先注册，然后才能下载。

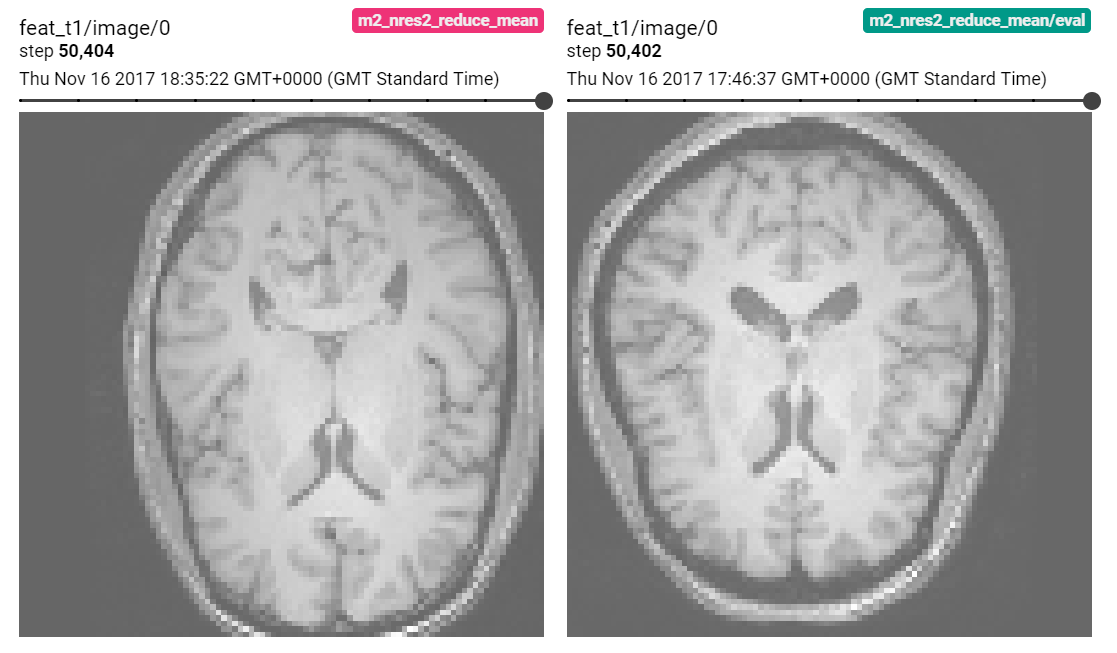
#### 多通道脑MR图像分割



此图像分割应用程序学习从小（N=5）MR brain challenge数据集上的多序列MR图像（T1加权、T1反转恢复和T2 FLAIR）预测脑组织和白质病变。它使用一个3D U-Net类网络作为特征抽取器，跟踪TensorBoard中每个标签的骰子系数精度。

代码和说明可以找到。

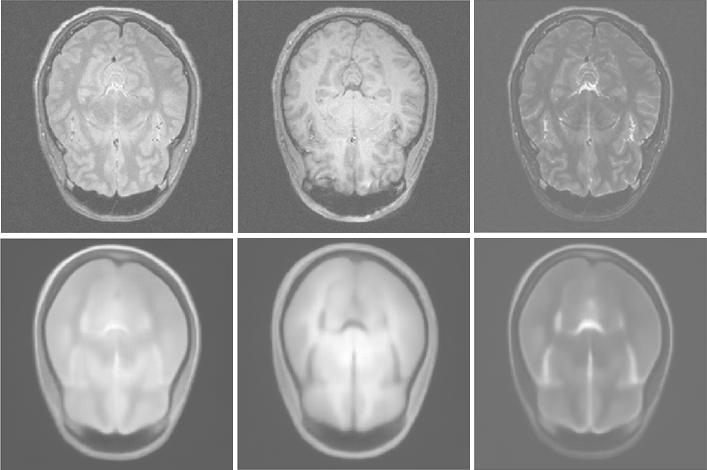
#### T1加权脑磁共振图像的年龄回归和性别分类



两个采用可伸缩3D ResNet架构的类似应用程序从IXI数据库的T1加权脑MR图像中学习预测受试者的年龄（回归）或性别（分类）。这两个应用程序之间的主要区别是损失函数：当我们训练回归网络，将年龄预测为具有L2损失（预测年龄和实际年龄之间的均方差）的连续变量时，我们使用分类交叉熵损失来预测性别类别。

这些应用程序的代码和说明可以在以下位置找到：。

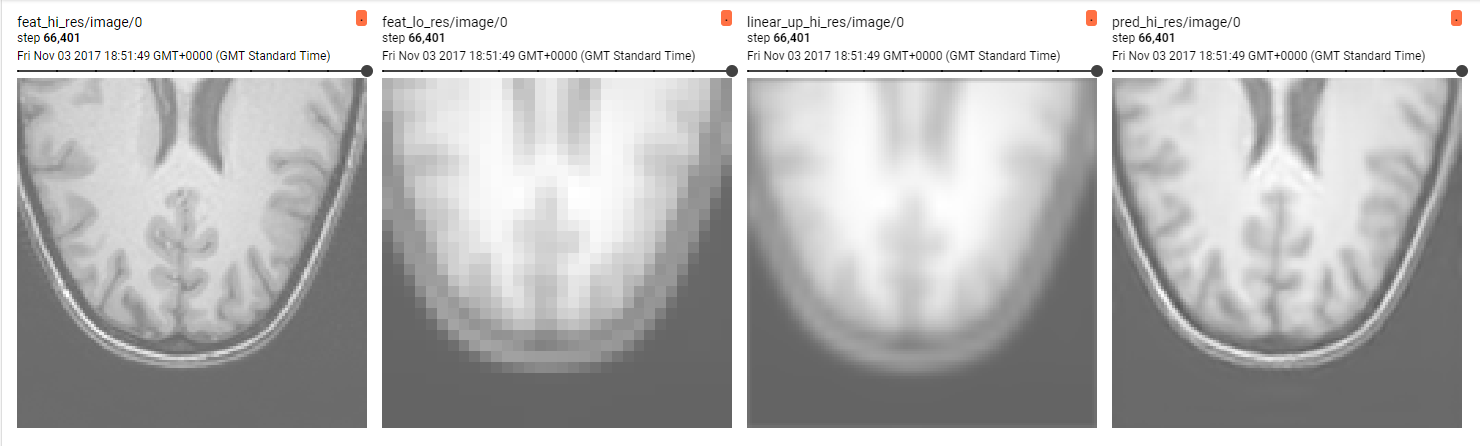
#### 3T多通道脑MR图像的表示学习



在这里，我们演示了一种深卷积自动编码器结构的使用，这是一种用于表示学习的强大工具：该网络以多序列MR图像为输入，旨在重建它们。通过这样做，它将整个训练数据库的信息压缩到其潜在变量中。训练后的权值也可用于传递学习或信息压缩。请注意，重建的图像非常平滑：这可能是因为此应用程序使用二级丢失功能或网络太小，无法正确编码详细信息。

代码和说明可以找到。

#### T1w脑MR图像的简单图像超分辨率

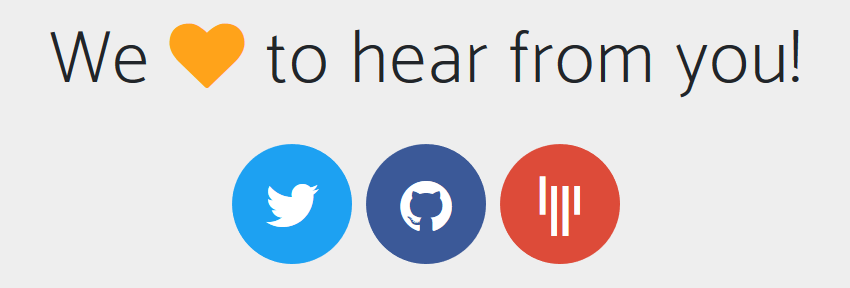


单图像超分辨率的目的是学习如何从低分辨率的输入中对高分辨率图像进行上采样和重建。这个简单的实现创建了图像的低分辨率版本，超级分辨率网络学习将图像提升到其原始分辨率（这里的提升采样因子是[4,4,4]）。此外，我们还计算了线性上采样版本，以显示与重建图像的差异。

代码和说明可以找到。

### 最后…

我们希望本教程能帮助您轻松进入生物医学图像的深入学习主题。如果您觉得它有帮助，我们感谢您分享它并关注DLTK。如果您有类似的问题需要帮助，请到我们这里来。也许有一天我们可以在DLTK中托管您的应用程序。谢谢你的阅读！



### Resources

[Tutorial code](https://github.com/DLTK/DLTK/tree/master/examples/tutorials), [example applications](https://github.com/DLTK/DLTK/tree/master/examples/applications), [DLTK source](https://github.com/DLTK/DLTK)