

MICROSOFT 人工智能 | 机器学习与乳腺癌检测

原创：IDZ英特尔开发人员专区1周前

本文展示了如何利用现有的深度学习技术来训练人工智能 (AI)，以支持在未标记的组织学图像中检测浸润性导管癌 (IDC) (乳腺癌)。 具体来说，我们将展示如何使用 TensorFlow*和迁移学习，借助包含阴性和阳性组织学图像的数据集来训练卷积神经网络。除了展示如何利用人工智能检测 IDC 外，我还将展示如何利用物联网 (IoT) 和人工智能创建应用于医疗行业的自动化系统。

乳腺癌一直是公众关注的问题，也是女性中发病率最高的癌症之一。预计 2018 年，美国将出现 266,120 例新增病例。人工智能的使用可大幅降低医务人员手动检查乳房造影涂片的需求，不仅能节省时间和资金，还能挽救生命。在本文中，我将展示如何利用英特尔技术创建能够检测 IDC 的深度学习神经网络。

IDC 分类器介绍

为了创建 IDC 分类器，我利用英特尔® AI DevCloud 来训练神经网络，利用英特尔® Movidius™ 产品在边缘执行推理，并利用 UP Squared*设备处理经过训练的模型，使其可通过 API 访问，我还利用 Raspberry Pi*设备创建了一个物联网互联报警系统，展示了通过 IoT JumpWay*和人工智能使用物联网的潜力，以创建智能的自动医疗系统。

该项目从一个名为 TASS 的计算机视觉项目发展而来，至今已经经历了数年的时间。TASS 是一个使用多种不同技术、框架和软件开发套件 (SDK) 实施的开源面部识别项目。

浸润性导管癌 (IDC)

IDC 是乳腺癌最常见的一类，癌症起源于乳房的乳腺管，并入侵周围组织。美国癌症协会的调查结果显示，此类癌症约占全部乳腺癌病例的 80%，每年有超过 180,000 位美国女性被诊断为 IDC。

卷积神经网络

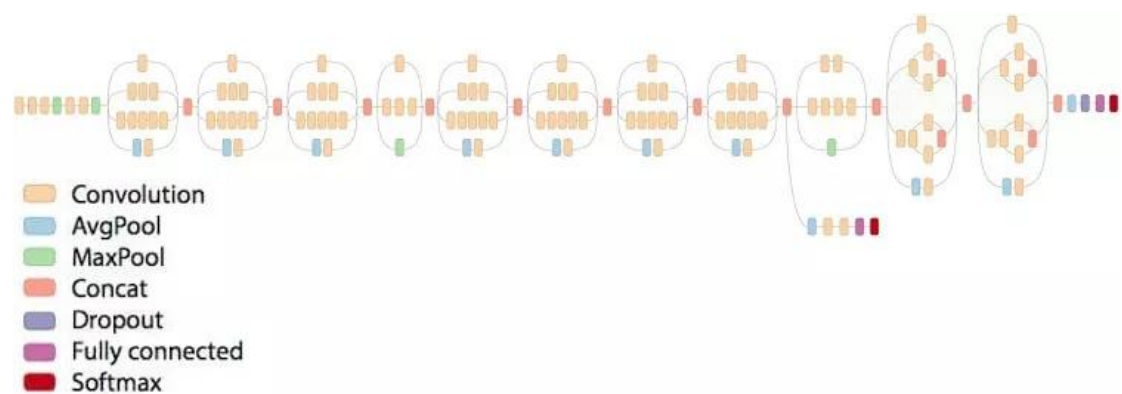


图 1. Inception v3 架构

卷积神经网络是一种深度学习神经网络。过去几年，这类神经网络广泛应用于计算机视觉，并推动了计算机视觉的发展，其性能远远超过之前推出的传统神经网络，但研究显示，该网络可能会牺牲训练时间和准确性。

迁移学习

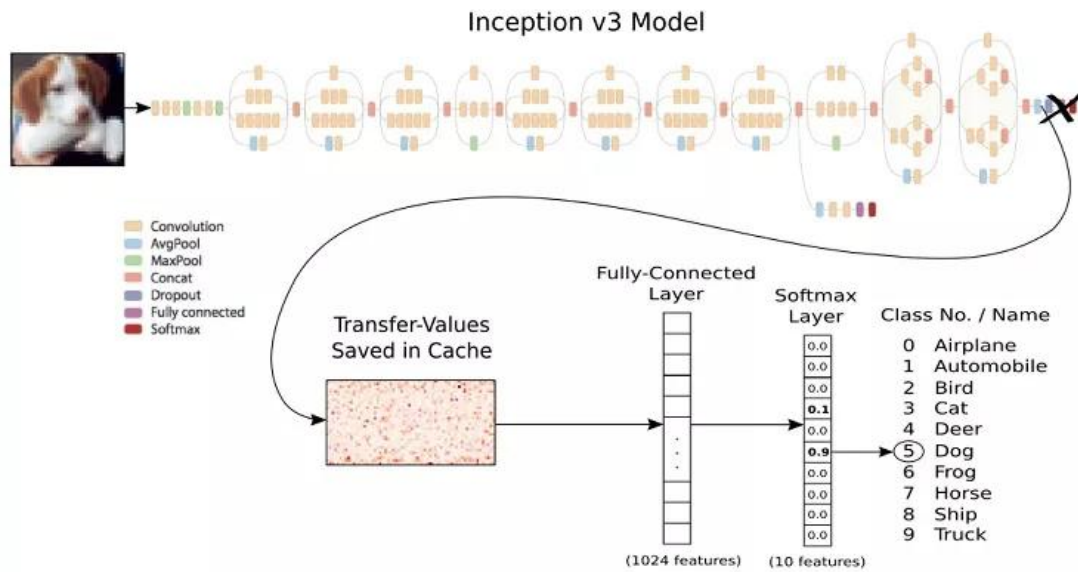


图 2. Inception V3 迁移学习

迁移学习支持您重新训练现有模型的最后一层，能够大幅减少训练时间和所需的数据集尺寸。谷歌* 创建的 Inception V3 模型是进行迁移学习的最著名的模型之一。在极其强大的设备上使用来自 1,001 个类别的数千张图像对该模型进行训练。能够重新训练最后一层意味着您可以保留模型在最初的训练中学习到的知识，并将其应用于更小的数据集，以执行高度准确的分类，无需大量的训练和计算能力。在某个 TASS 版本中，我使用迁移学习在 Raspberry Pi 3 设备上重新训练了 Inception V3 模型，以获取关于迁移学习功能的相关信息。

英特尔® AI DevCloud

英特尔 AI DevCloud 是一个用于训练机器学习与深度学习模型的平台。该平台由使用英特尔® 至强® 可扩展处理器的服务器集群构成，并且免费提供一系列框架和工具，包括 TensorFlow、Caffe*、Keras*、Theano* 和英特尔® Python* 分发包。英特尔 AI DevCloud 是入门级

用户的理想之选，可帮助他们学习如何训练机器学习与深度学习模型，因为图形处理单元 (GPU) 非常昂贵，而 DevCloud 可免费访问。

在本项目中，我使用英特尔 AI DevCloud 对数据进行分类，训练并评估模型。本文还附有我编写的完整教程，为您提供了复制整个项目所需的所有代码；阅读完整教程并访问源代码。

英特尔® Movidius™ 神经计算棒

英特尔® Movidius™ 神经计算棒是一款新型硬件，用于改进低功耗边缘设备上计算机视觉模型的推理流程。英特尔 Movidius 产品是一款 USB 设备，可插入 Raspberry Pi、UP Squared 等设备，将设备的处理能力全部集中在英特尔 Movidius 品牌芯片上，显著提升了分类流程的速度。开发人员可利用现有的 TensorFlow 和 Caffe 脚本训练模型，并通过在开发设备上安装英特尔 Movidius 神经计算棒 SDK 来编译兼容英特尔 Movidius 产品的图形。在功耗较低的设备上安装更轻便的 API，以支持通过英特尔 Movidius 产品执行推理。

开始编码

现在，您可能已经迫不及待地想要学习这项技术，以创建自己的计算机视觉程序，对阴性和阳性乳腺癌细胞进行分类，下面我们来详细讨论。接下来我将逐步介绍如何训练与编译英特尔 Movidius 产品的图形。如欲获取完整详解，包括物联网互联设备，请访问 [GitHub*](#) 存储库。继续阅读本教程前，请执行存储库中的以下步骤设置您的 IoT JumpWay 设备，这是开始分类测试前所需的。

在开发设备上安装 SDK

首先，您需要在您的开发设备上安装 英特尔 Movidius 神经计算棒 SDK，用于将经过训练的模型转换为兼容英特尔 Movidius 产品的格式。

```
$ mkdir -p ~/workspace
```

```
$ cd ~/workspace
```

```
$ git clone https://github.com/movidius/ncsdk.git
```

```
$ cd ~/workspace/ncsdk
```

```
$ make install
```

接下来，将英特尔 Movidius 产品插入您的设备，并发布以下命令：

```
$ cd ~/workspace/ncsdk
```

```
$ make examples
```

在推理设备上安装 SDK

接下来，您需要在您的 Raspberry Pi 3/UP Squared 设备上安装 英特尔 Movidius 神经计算棒 SDK，分类器用它对本地图像或通过我们创建的 API 接收的图像执行推理。请确保英特尔 Movidius 产品已插入。

```
$ mkdir -p ~/workspace
```

```
$ cd ~/workspace
```

```
$ git clone https://github.com/movidius/ncsdk.git
```

```
$ cd ~/workspace/ncsdk/api/src
```

```
$ make
```

```
$ sudo make install
```

```
$ cd ~/workspace
```

```
$ git clone https://github.com/movidius/ncappzoo
```

```
$ cd ncappzoo/apps/hello_ncs_py
```

```
$ python3 hello_ncs.py
```

准备训练数据

在本教程中，我使用来自 Kaggle*（预测乳腺癌组织学图像中的 IDC）的数据集，但是您可以免费使用您喜欢的任何数据集。我已经上传了我使用的阴性与阳性图片集，它们位于 model/train 目录中。一旦确定了数据集，您需要将数据整理至 model/train 目录中。应使用整数命名每个子目录；我使用 0 和 1 来表示阳性与阴性。在测试中，我使用 4400 个阳性和 4400 个阴性示例，整体训练精确度为 0.8596（参阅以下训练结果），正确识别的平均置信度为 0.96。提供的数据为 50px x 50px；由于训练 Inception V3 的图像大小为 299px x 299px，将图像重新调整为 299px x 299px。在理想情况下，图像就应该是该尺寸，您可能想尝试不同的数据集，以查看结果的变化。

微调参数

您可以通过编辑 model/confs.json 文件中的分类器设置，随时微调网络设置。

```
"ClassifierSettings":{  
  "dataset_dir":"model/train/",  
  "log_dir":"model/_logs",  
  "log_eval":"model/_logs_eval",  
  "classes":"model/classes.txt",
```

```
"labels":"labels.txt",
"labels_file":"model/train/labels.txt",
"validation_size":0.3,
"num_shards":2,
"random_seed":50,
"tfrecord_filename":"200label",
"file_pattern":"200label_%s_*.tfrecord",
"image_size":299,
"num_classes":2,
"num_epochs":60,
"dev_cloud_epochs":60,
"test_num_epochs":1,
"batch_size":10,
"test_batch_size":36,
"initial_learning_rate":0.0001,
"learning_rate_decay_factor":0.96,
"num_epochs_before_decay":10,
"NetworkPath":"",
"InceptionImagePath":"model/test/",
"InceptionThreshold": 0.54,
"InceptionGraph":"igraph"
}
```

开始训练

现在，您已经准备好将下文中的文件和文件夹上传至英特尔 AI DevCloud。

model

tools

DevCloudTrainer.ipynb

DevCloudTrainer.py

Eval.py

上传完成后，执行 DevCloudTrainer.ipynb 中的指令，该 notebook 将帮助您分类数据、训练与评估模型。

训练结果



图 3.训练精确度 Tensorboard

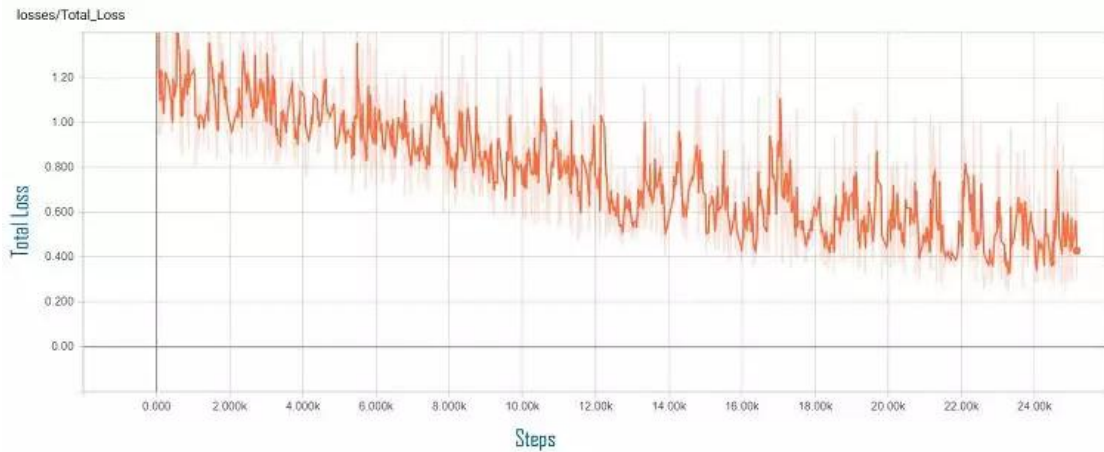


图 4.训练总损失

评估模型

在英特尔 AI DevCloud 上完成训练后，通过运行评估任务来完成 notebook。

评估结果

INFO:tensorflow:Global Step 1: Streaming Accuracy: 0.0000 (2.03 sec/step)

INFO:tensorflow:Global Step 2: Streaming Accuracy: 0.8889 (0.59 sec/step)

INFO:tensorflow:Global Step 3: Streaming Accuracy: 0.8750 (0.67 sec/step)

INFO:tensorflow:Global Step 4: Streaming Accuracy: 0.8981 (0.65 sec/step)

INFO:tensorflow:Global Step 5: Streaming Accuracy: 0.8681 (0.76 sec/step)

INFO:tensorflow:Global Step 6: Streaming Accuracy: 0.8722 (0.64 sec/step)

INFO:tensorflow:Global Step 7: Streaming Accuracy: 0.8843 (0.64 sec/step)

--

INFO:tensorflow:Global Step 68: Streaming Accuracy: 0.8922 (0.81 sec/step)

INFO:tensorflow:Global Step 69: Streaming Accuracy: 0.8926 (0.70 sec/step)

INFO:tensorflow:Global Step 70: Streaming Accuracy: 0.8921 (0.63 sec/step)

INFO:tensorflow:Global Step 71: Streaming Accuracy: 0.8929 (0.84 sec/step)

INFO:tensorflow:Global Step 72: Streaming Accuracy: 0.8932 (0.75 sec/step)

INFO:tensorflow:Global Step 73: Streaming Accuracy: 0.8935 (0.61 sec/step)

INFO:tensorflow:Global Step 74: Streaming Accuracy: 0.8942 (0.67 sec/step)

INFO:tensorflow:Final Streaming Accuracy: 0.8941

在这里我们可以看到评估的最终流传输精确度为 0.8941。

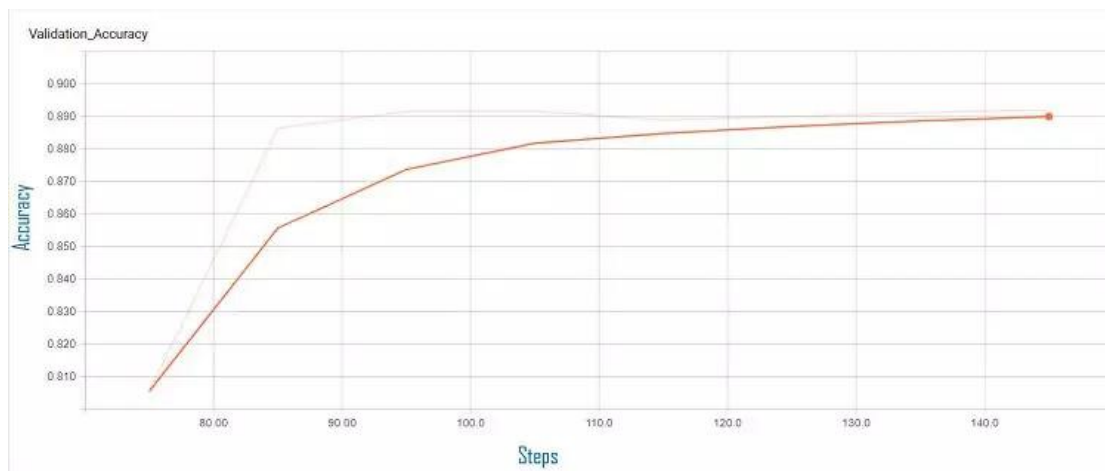


图 5.评估精确度

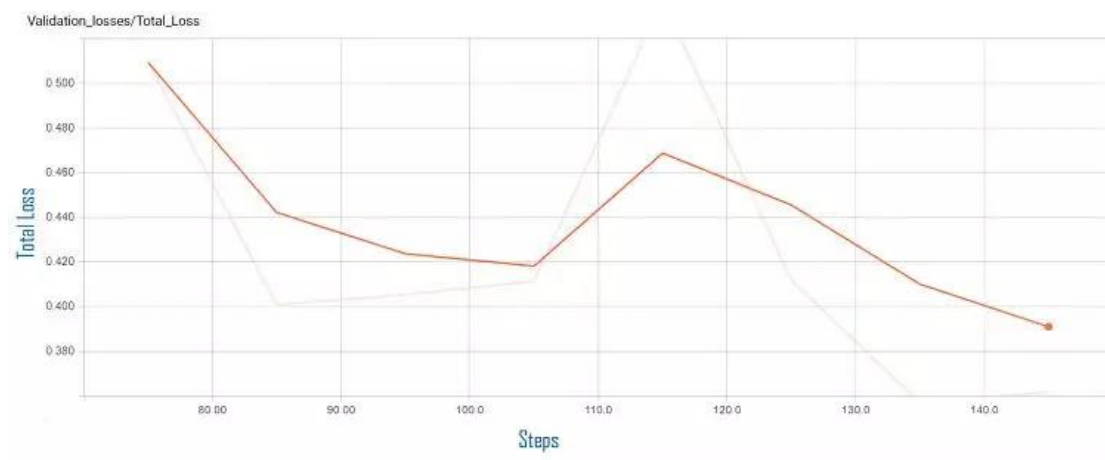


图 6.评估总损失

下载模型

训练完成后，您需要将 model/DevCloudIDC.pb and model/classes.txt 下载到开发设备的 model 目录。请确保英特尔 Movidius 产品已设置并连接，然后在开发设备上运行以下命令：

```
$ cd
~/IoT-JumpWay-Intel-Examples/master/Intel-Movidius/IDC-Classification
```

```
$ ./DevCloudTrainer.sh
```

DevCloudTrainer.sh 的内容如下所示：

```
#IDC Classification Trainer
mvNCCompile model/DevCloudIDC.pb -in=input
-on=InceptionV3/Predictions/Softmax -o igrph
python3.5 Classifier.py InceptionTest
```

1.

编译面向英特尔 Movidius 产品的模型

2.

3.

测试

4.

测试未知图像

shell 脚本完成测试后将启动程序。在本示例中，有 0 和 1 (IDC 阴性和 IDC 阳性) 两类，分类 0 表示人工智能认为图像不是 IDC 阳性，分类 1 表示阳性。

```
-- Loaded Test Image model/test/negative.png
```

```
-- DETECTION STARTING
```

```
-- STARTED: : 2018-04-24 14:14:26.780554
```

```
-- DETECTION ENDING
```

-- ENDED: 2018-04-24 14:14:28.691870

-- TIME: 1.9114031791687012

inception-v3 on NCS

0 0 0.9873

1 1 0.01238

-- Loaded Test Image model/test/positive.png

-- DETECTION STARTING

-- STARTED: : 2018-04-24 14:14:28.699254

-- DETECTION ENDING

-- ENDED: 2018-04-24 14:14:30.577683

-- TIME: 1.8784320354461678

TASS Identified IDC with a confidence of 0.945

-- Published to Device Sensors Channel

```
*****
*****

inception-v3 on NCS

*****

*****

1 1 0.945

0 0 0.05542

*****

*****

-- INCEPTION V3 TEST MODE ENDING
-- ENDED: 2018-04-24 14:14:30.579247
-- TESTED: 2
-- IDENTIFIED: 1
-- TIME(secs): 3.984593152999878
```

因此，在开发设备上，您应该看到类似上面的结果。您会看到程序已成功将结果分为阴性和阳性。接下来我们将在边缘上测试。

边缘上的推理

既然已经完成了所有训练与测试，接下来设置用于 API 的服务器。为此，我提供了 Server.py 和 Client.py。

以下指令将帮助您设置服务器并测试阳性与阴性预测：

1.如果您采用了 预测乳腺癌组织学图像中的 IDC 数据集，可利用 positive.png 和 negative.png，因为它们也来自该数据集；如若不然，您应从测试集中选择阳性与阴性示例，并替换这些图像。

2.服务器目前设置为在本地主机上启动。如果您想修改该设置，需要编辑 Server.py 的第 281 行和 Client.py 的第 38 行，以匹配您的主机。设置完成后，如果您想继续运行并从外界访问它，应利用 Let's Encrypt* 或类似工具来确保它的安全。

3.将以下文件和文件夹上传至服务器使用的 UP Squared 或 Raspberry Pi 3 设备。

model/test/

model/classes.txt

model/confs.json

tools

igraph

Server.py

4.打开一个终端，导航至包含 Server.py 的文件夹，然后发布以下命令。

这将启动服务器，并等待接收图像进行分类。

5.\$ python3.5 Server.py

如果您执行了上述所有步骤，现在可以使用以下命令打开开发设备上的客户端：

\$ python3.5 Client.py

它将向 Raspberry Pi 3 或 UP Squared 设备发送阳性与阴性组织学涂片，设备将返回预测结果。

!! Welcome to IDC Classification Client, please wait while the program initiates !!

-- Running on Python 3.5.2 (default, Nov 23 2017, 16:37:01)

[GCC 5.4.0 20160609]

-- Imported Required Modules

-- IDC Classification Client Initiated

```
{'Response': 'OK', 'ResponseMessage': 'IDC Detected!', 'Results': 1}
```

```
{'Response': 'OK', 'ResponseMessage': 'IDC Not Detected!', 'Results': 0}
```

* Running on <http://0.0.0.0:7455/> (Press CTRL+C to quit)

-- IDC CLASSIFIER LIVE INFERENCE STARTING

-- STARTED: : 2018-04-24 14:25:36.465183

-- Loading Sample

-- Loaded Sample

-- DETECTION STARTING

-- STARTED: : 2018-04-24 14:25:36.476371

-- DETECTION ENDING

-- ENDED: 2018-04-24 14:25:38.386121

-- TIME: 1.9097554683685303

TASS Identified IDC with a confidence of 0.945

-- Published: 2

-- Published to Device Warnings Channel

-- Published: 3

-- Published to Device Sensors Channel

inception-v3 on NCS

1 1 0.945

0 0 0.05542

-- IDC CLASSIFIER LIVE INFERENCE ENDING

-- ENDED: 2018-04-24 14:25:38.389217

-- TESTED: 1

-- IDENTIFIED: 1

-- TIME(secs): 1.9240257740020752

192.168.1.40 - - [24/Apr/2018 14:25:38] "POST /api/infer
HTTP/1.1" 200 -

-- IDC CLASSIFIER LIVE INFERENCE STARTING

-- STARTED: : 2018-04-24 14:25:43.422319

-- Loading Sample

-- Loaded Sample

-- DETECTION STARTING

-- STARTED: : 2018-04-24 14:25:43.432647

-- DETECTION ENDING

-- ENDED: 2018-04-24 14:25:45.310354

-- TIME: 1.877711534500122

-- Published: 4

-- Published to Device Warnings Channel

-- Published: 5

-- Published to Device Sensors Channel

inception-v3 on NCS

0 0 0.9873

1 1 0.01238

-- IDC CLASSIFIER LIVE INFERENCE ENDING

-- ENDED: 2018-04-24 14:25:45.313174

-- TESTED: 1

-- IDENTIFIED: 0

-- TIME(secs): 1.89084792137146

192.168.1.40 - - [24/Apr/2018 14:25:45] "POST /api/infer
HTTP/1.1" 200 -

在这里我们可以看到，在 UP Squared 设备和开发设备上使用英特尔 Movidius 产品时，两者的分类准确性没有区别；本示例采用了搭载 NVIDIA* GTX 750ti 的 Linux* 设备，只有在完成分类流程的时间方面存在细微差别。有趣的是，上述结果实际上比在 GPU 上训练模型更准确。

物联网连接

为了设置物联网连接，建议您完成 GitHub 存储库上的教程，但是我将详细介绍这一部分项目的作用，以及如何将本文提供的概念验证应用于其他医疗应用。

我们创建的设备是一台基于 Raspberry Pi 设备的物联网互联报警系统。设置完成后，捕捉发送至服务器的图像分类结果，将在与 Raspberry Pi 设备通信的物联网上触发操作。在本示例中，操作为：如果检测出癌症，打开红色 LED 和蜂鸣器，如果分类结果显示没有检测到癌症，打开蓝色 LED。显然，这是一个非常简单的概念验证，但是它表明我们有可能开发强大的应用，以节约医务人员的宝贵时间，而且如果利用得当，它就有希望提供精确的早期检测，挽救更多生命。