Snack Classifier开发实验报告

学号	姓名
191220162	张乐简

Snack Classifier开发实验报告
- 概述
- 实验内容
  - Snack Classifier
    - 训练模型
    - 使用模型
  - HealthyOrNotHealthySnackClassifier
    - 构造数据
    - 使用模型
- 反思

概述

该实验利用CreateML和提供的零食数据集训练出一个零食分类器，同时在代码中使用它来判断给定照片的零食种类。另外，也要求改造数据集，再训练一个健康/非健康食品分类器。

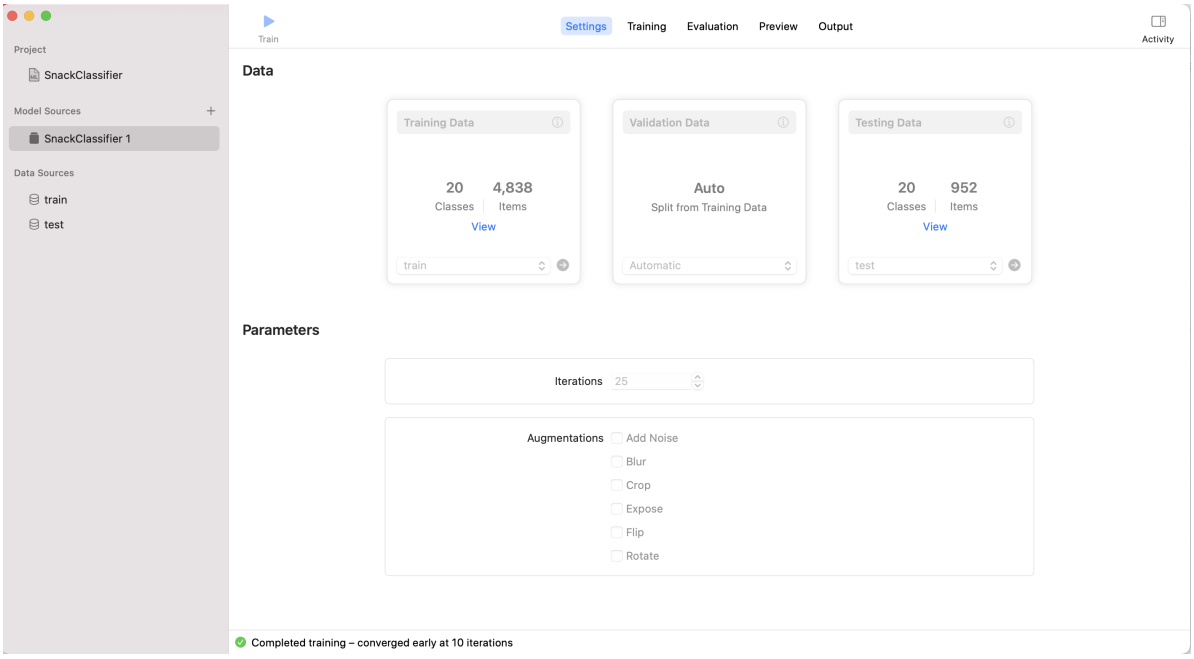
实验内容

Snack Classifier

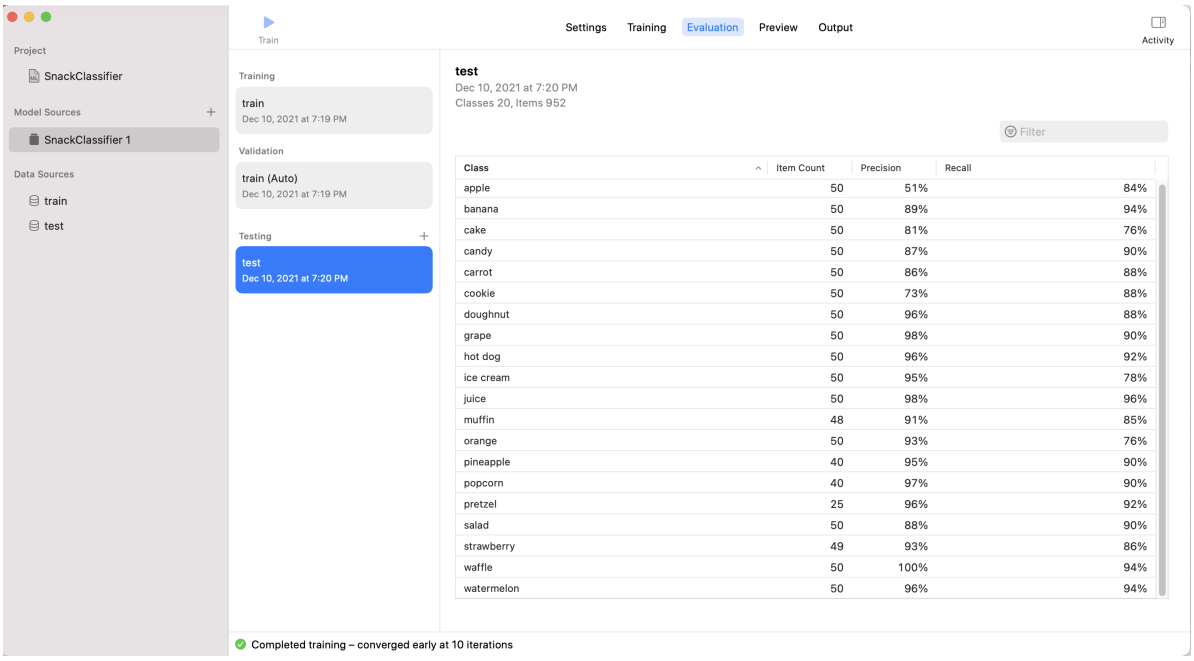
训练模型

简单地使用CreateML即可。

训练过程：



训练结果：



使用模型

首先，需要创建模型，并为模型提供输入。创建模型这一部分已经在框架代码中实现，不再赘述。在演示课件中，为模型提供的输入类型为CMSampleBuffer，而框架最后为我提供的参数类型为UIImage。开始时我在StackOverflow上寻得了将UIImage转换为CMSampleBuffer的方法并使用，但发现准确率很低，于是转而直接将输入参数的cgImage成员作为模型的输入。尽管后来发现准确率低似乎与CMSampleBuffer无关，但考虑到cgImage已经有不错的表现，便不再切换回去。

代码如下：

```
            DispatchQueue.main.async {
                let handler = VNImageRequestHandler(cgImage:
sampleBuffer.cgImage!,
orientation:CGImagePropertyOrientation(sampleBuffer.imageOrientation))
                do {
                    try handler.perform([self.classificationRequest])
                    try handler.perform([self.huhclassificationRequest])
                } catch {
                    print("Failed to perform classification: \(error)")
                }
                self.semphore.signal()
            }
```

得到模型输出的判断后，将该判断根据confidence进行筛选，决定显示它还是显示“我不确定”。代码如下。

```
    func processObservations(for request: VNRequest, error: Error?) {
        if let results = request.results as? [VNClassificationObservation] {
            if results.isEmpty {
                self.resultsLabel.text = "Nothing found\n"
            } else {
                let result = results[0].identifier
                let confidence = results[0].confidence
                if confidence<0.7{
                    self.resultsLabel.text="我不确定"+result+"\n"
```

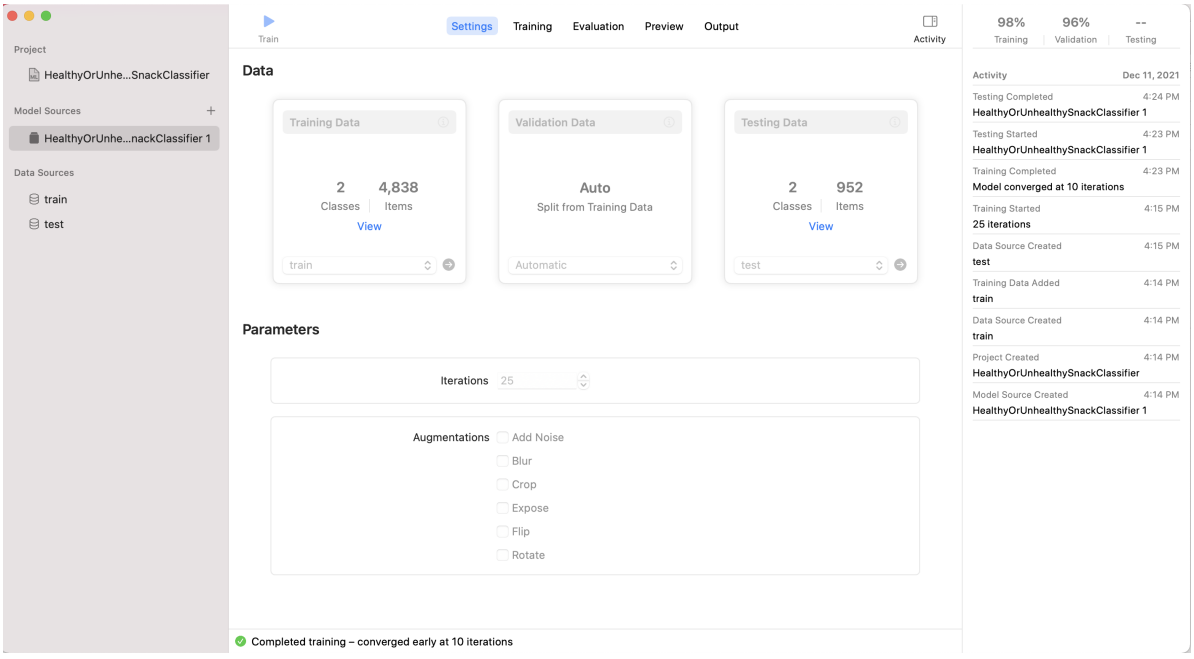
```
 self.resultsLabel.text=self.resultsLabel.text!+String(format: "%.1f%%",
confidence * 100)
                }else{
                    self.resultsLabel.text = result+"\n"

 self.resultsLabel.text=self.resultsLabel.text!+String(format: "%.1f%%",
confidence * 100)
                }
            }
        } else if let error = error {
            self.resultsLabel.text = "Error: \(error.localizedDescription)"
        } else {
            self.resultsLabel.text = "???"
        }
        showResultsView(delay: 0.5)
    }
```

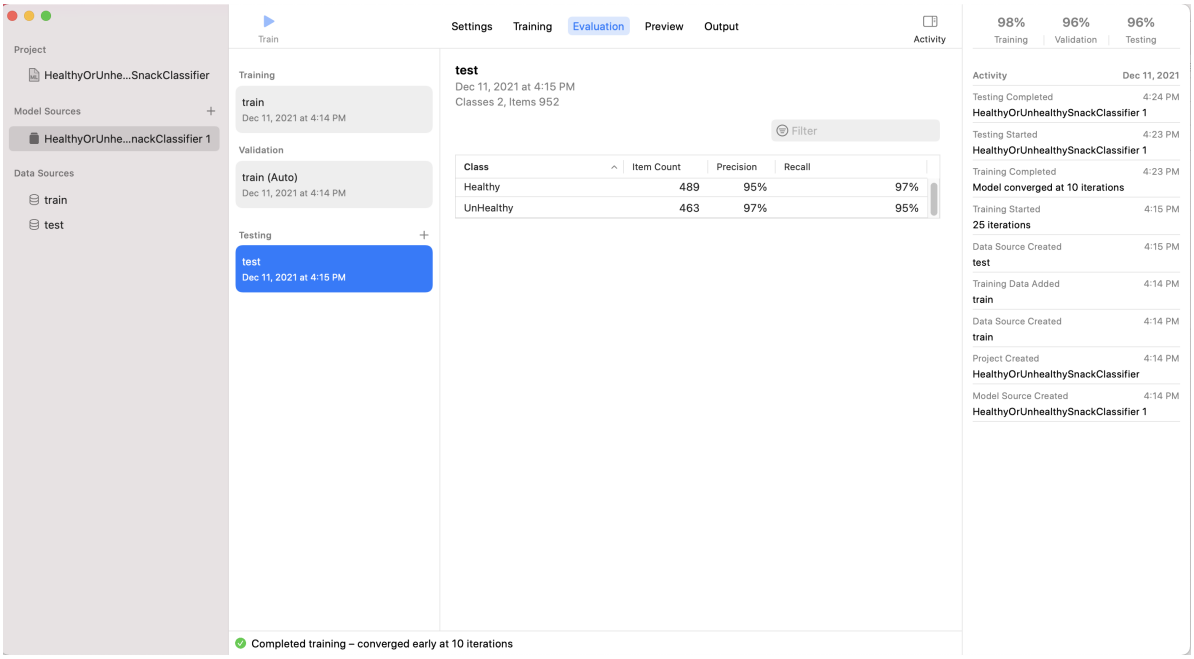
HealthyOrNotHealthySnackClassifier

构造数据

将apple,banana,carrot,grape,juice,orange,pineapple,salad,strawberry,watermelon归为Heathy，其余类别的食物归类为Unhealthy。将对应种类的照片从snack文件夹中对应类别下提取出来，整合成Healthy和Unhealthy两个大文件夹，用于训练；对测试集采用同样处理。这样得到训练数据与测试数据后，再次使用CreateML进行训练，得到模型。

训练过程与上一部分一致。





使用模型

使用模型的代码与SnackClassifier的使用方法基本一致，不再赘述。只是由于需要单独处理它的结果，另写一个processObservation函数来处理它的判断结果。

代码如下：

```
    func processObservations2(for request: VNRequest, error: Error?) {
        if let results = request.results as? [VNClassificationObservation] {
            self.resultsLabel.text=self.resultsLabel.text!+"\n"+"健康判断："
            if results.isEmpty {

            } else {
                let result = results[0].identifier
                let confidence = results[0].confidence
                if confidence<0.7{
                    self.resultsLabel.text=self.resultsLabel.text!+"我不确定\n"

 self.resultsLabel.text=self.resultsLabel.text!+String(format: "%.1f%%",
confidence * 100)
                }else{
                    self.resultsLabel.text = self.resultsLabel.text!+result+"\n"

 self.resultsLabel.text=self.resultsLabel.text!+String(format: "%.1f%%",
confidence * 100)
                }
            }
        } else if let error = error {
            self.resultsLabel.text = "Error: \(error.localizedDescription)"
        } else {
            self.resultsLabel.text = "???"
        }

    }
```

反思

在模型训练结束后，我立刻试图将其放入代码中使用，但无论是将图片转为CVPixelBuffer输入模型，还是直接使用cgImage作为参数传入模型，判断的结果都出现明显的错误。开始时，我以为是模型训练问题，但发现模型在Preview中表现很好；而且，模型在Preview中对一张图片的判断和模拟器中对同一张图片的判断完全不同，在这方面四处搜寻解决方案无果，花费大量时间。后来发现此系模拟器问题，在真机上运行模型则可以正常运行。以后需要更重视运行环境对代码的影响。