

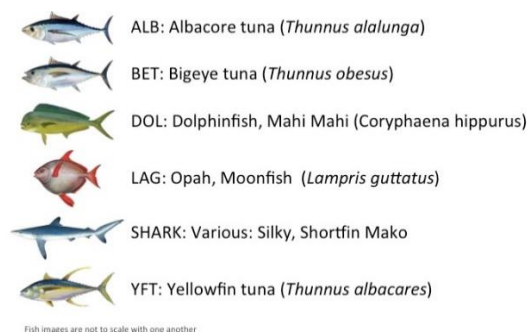
The Nature Conservancy Fisheries Monitoring

Description

为了保护和监控海洋环境及生态平衡，大自然保护协会（The Nature Conservancy）邀请 Kaggle 社区的参赛者们开发出能够胜任的机器学习算法，自动分类和识别远洋捕捞船上的摄像头拍摄到的图片中鱼类的品种，例如不同种类的吞拿鱼和鲨鱼。大自然保护协会一共提供了 3777 张标注的图片作为训练集，这些图片被分为了 8 类，其中 7 类是不同种类的海鱼，剩余 1 类则是不含有鱼图片，每张图片只属于 8 类中的某一类别。如下图给出了数据集中的几张图片样例，可以看到，有些图片中待识别的海鱼所占整张图片的一小部分，这就给识别带来了很大的挑战性。此外，为了衡量算法的有效性，还提供了额外的 1000 张图片作为测试集，参赛者们需要设计出一种图像识别的算法，尽可能地识别出这 1000 张测试图片属于 8 类中的哪一类别。Kaggle 平台为每一个竞赛都提供了一个榜单（Leaderboard），识别的准确率越高的竞赛者在榜单上的排名越靠前。











Eight target categories are available in this dataset: Albacore tuna, Bigeye tuna, Yellowfin tuna, Mahi Mahi, Opah, Sharks, Other (meaning that there are fish present but not in the above categories), and No Fish (meaning that no fish is in the picture). Each image has only one fish category, except that there are sometimes very small fish in the pictures that are used as bait. (在这个数据集中有八个目标类别：长鳍金枪鱼，大眼金枪鱼，黄鳍金枪鱼，Mahi Mahi，月鱼属，鲨鱼，其他（意味着鱼有存在但不在上述类别中），没有鱼（意思是图片中没有鱼）。每个图像只有一个鱼类，除了在用作诱饵的图片中有时非常小的鱼）



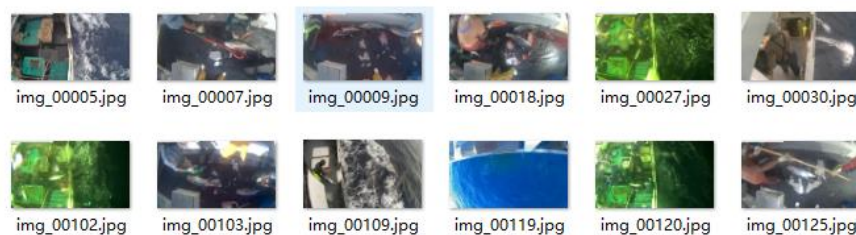
Data Set

- **train.zip** - zipped folder of all train images. The train folders are organized by fish species labels

带有鱼种类标签的训练集，训练集下面有八个文件夹（每个文件夹对应一个鱼的种类，各个文件夹下图片数目大约为 1700、200、117、67、465、299、176、734）

 ALB	2016/11/7 14:50	文件夹
 BET	2016/11/7 14:50	文件夹
 DOL	2016/11/7 14:50	文件夹
 LAG	2016/11/7 14:50	文件夹
 NoF	2016/11/7 14:50	文件夹
 OTHER	2016/11/7 14:50	文件夹
 SHARK	2016/11/7 14:50	文件夹
 YFT	2016/11/7 14:50	文件夹

- **test_stg1.zip** - zipped folder of all test images in stage 1（测试集 1-1000 张图片）
- **test_stg2.zip** - zipped folder of all test images in stage 2（测试集 2-12000 张图片）



- **sample_submission_stg1.csv** - a sample submission file in the correct format（提交格式样例）

	A	B	C	D	E	F	G	H	I
1	image	ALB	BET	DOL	LAG	NoF	OTHER	SHARK	YFT
2	img_00005	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
3	img_00007	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
4	img_00009	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
5	img_00018	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
6	img_00027	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
7	img_00030	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
8	img_00040	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
9	img_00046	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
10	img_00053	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
11	img_00071	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
12	img_00075	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
13	img_00102	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
14	img_00103	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
15	img_00109	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
16	img_00119	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
17	img_00120	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
18	img_00125	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
19	img_00128	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283
20	img_00129	0.455003	0.052938	0.030969	0.017734	0.123081	0.079142	0.046585	0.194283

Evaluation

Submissions are evaluated using the [multi-class logarithmic loss](#). Each image has been labeled with one true class. For each image, you must submit a set of predicted probabilities (one for every image). The formula is then,

$$\text{logloss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log(p_{ij}),$$

where N is the number of images in the test set, M is the number of image class labels, \log is the natural logarithm, y_{ij} is 1 if observation i belongs to class j and 0 otherwise, and p_{ij} is the predicted probability that observation i belongs to class j .

The submitted probabilities for a given image are not required to sum to one because they are rescaled prior to being scored (each row is divided by the row sum). In order to avoid the extremes of the log function, predicted probabilities are replaced with $\max(\min(p, 1 - 10^{-15}), 10^{-15})$.

对数损失函数

Multi Class Log Loss 是对数损失函数在多分类下的版本。对数损失函数的标准形式如下：

$$L(Y, P(Y|X)) = -\log P(Y|X)$$

$L(Y, P(Y|X))$ 这个对数损失函数的意思是指分类为 Y 的情况下，使 $P(Y|X)$ 达到最大。有一些模型是用最大概率的分类来做预测的，而 Y 是代表分类为正确的分类，而 $P(Y|X)$ 则是代表正确分类的概率，那对数取反就意味着 $P(Y|X)$ 越大，损失函数就越小。

更多概念以及推导过程参考 <https://www.zhihu.com/question/27126057>

Multi Class Log Loss

This is the multi-class version of the [Logarithmic Loss](#) metric. Each observation is in one class and for each observation, you submit a predicted probability for each class. The metric is negative the log likelihood of the model that says each test observation is chosen independently from a distribution that places the submitted probability mass on the corresponding class, for each observation.

$$\text{logloss} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \log(p_{i,j})$$

where N is the number of observations, M is the number of class labels, \log is the natural logarithm, $y_{i,j}$ is 1 if observation i is in class j and 0 otherwise, and $p_{i,j}$ is the predicted probability that observation i is in class j .

Both the solution file and the submission file are CSV's where each row corresponds to one observation, and each column corresponds to a class. The solution has 1's and 0's (exactly one "1" in each row), while the submission consists of predicted probabilities.

The submitted probabilities need not sum to 1, because they will be rescaled (each is divided by the sum) so that they do before evaluation.

(Note: the actual submitted predicted probabilities are replaced with $\max(\min(p, 1 - 10^{-15}), 10^{-15})$.)

```
id,col1,col2,Indicator
1,1,0,Public
2,1,0,Public
3,1,0,Public
4,0,1,Public
5,0,1,Public
6,0,1,Public
7,1,0,Private

id,col1,col2
1,0.5,0.5
2,0.1,0.9
3,0.01,0.99
4,0.9,0.1
5,0.75,0.25
6,0.001,0.999
7,1,0

Assert.AreEqual(1.881797068998267, actual: publicScore, delta: ErrorTolerance); language: php
```

Note: In this example 'publicScore' is the score for only the rows marked as 'Public'. The observation with ID of 7 is ignored in the calculation of 1.881797068998267 (and $N = 6$).

Submission File

You must submit a csv file with the image file name, and a probability for each class.

The 8 classes to predict are: 'ALB', 'BET', 'DOL', 'LAG', 'NoF', 'OTHER', 'SHARK', 'YFT'

The order of the rows does not matter. The file must have a header and should look like the following:

```
image,ALB,BET,DOL,LAG,NoF,OTHER,SHARK,YFT
img_00001.jpg,1,0,0,0,0,...,0
img_00002.jpg,0.3,0.1,0.6,0,...,0
...
```

Links

<https://www.kaggle.com/c/the-nature-conservancy-fisheries-monitoring>