

Deploy and monitor a machine learning workflow for Image Classification

Setting up this notebook

Notes about the instance size and kernel setup: this notebook has been tested on

1. The `Python 3 (Data Science)` kernel
2. The `ml.t3.medium` Sagemaker notebook instance

Data Staging

We'll use a sample dataset called CIFAR to simulate the challenges Scones Unlimited are facing in Image Classification. In order to start working with CIFAR we'll need to:

1. Extract the data from a hosting service
2. Transform it into a usable shape and format
3. Load it into a production system

In other words, we're going to do some simple ETL!

1. Extract the data from the hosting service

In the cell below, define a function `extract_cifar_data` that extracts python version of the CIFAR-100 dataset. The CIFAR dataset is open source and generously hosted by the University of Toronto at: <https://www.cs.toronto.edu/~kriz/cifar-100-python.tar.gz>

```
In [1]: import requests

def extract_cifar_data(url, filename="cifar.tar.gz"):
    """A function for extracting the CIFAR-100 dataset and storing it as a gzipped file

    Arguments:
    url      -- the URL where the dataset is hosted
    filename -- the full path where the dataset will be written

    """

    # Todo: request the data from the data url
    # Hint: use `requests.get` method
    r = requests.get(url)

    with open(filename, "wb") as file_context:
        file_context.write(r.content)

    return
```

Let's test it out! Run the following cell and check whether a new file `cifar.tar.gz` is created in the file explorer.

```
In [3]: extract_cifar_data("https://www.cs.toronto.edu/~kriz/cifar-100-python.tar.gz")
```

2. Transform the data into a usable shape and format

Clearly, distributing the data as a gzipped archive makes sense for the hosting service! It saves on bandwidth, storage, and it's a widely-used archive format. In fact, it's so widely used that the Python community ships a utility for working with them, `tarfile`, as part of its Standard Library. Execute the following cell to decompress your extracted dataset:

```
In [4]: import tarfile

with tarfile.open("cifar.tar.gz", "r:gz") as tar:
    tar.extractall()
```

A new folder `cifar-100-python` should be created, containing `meta`, `test`, and `train` files. These files are `pickles` and the [CIFAR homepage](#) provides a simple script that can be used to load them. We've adapted the script below for you to run:

```
In [5]: import pickle

with open("./cifar-100-python/meta", "rb") as f:
    dataset_meta = pickle.load(f, encoding='bytes')

with open("./cifar-100-python/test", "rb") as f:
    dataset_test = pickle.load(f, encoding='bytes')

with open("./cifar-100-python/train", "rb") as f:
    dataset_train = pickle.load(f, encoding='bytes')
```

```
In [6]: # Feel free to explore the datasets

dataset_train.keys()
```

```
Out[6]: dict_keys([b'filenames', b'batch_label', b'fine_labels', b'coarse_labels', b'data'])
```

As documented on the homepage, `b'data'` contains rows of 3073 unsigned integers, representing three channels (red, green, and blue) for one 32x32 pixel image per row.

```
In [7]: 32*32*3
```

```
Out[7]: 3072
```

For a simple gut-check, let's transform one of our images. Each 1024 items in a row is a channel (red, green, then blue). Each 32 items in the channel are a row in the 32x32 image. Using python, we can stack these channels into a 32x32x3 array, and save it as a PNG file:

```
In [8]: import numpy as np
```

```

# Each 1024 in a row is a channel (red, green, then blue)
row = dataset_train[b'data'][0]
red, green, blue = row[0:1024], row[1024:2048], row[2048:]

# Each 32 items in the channel are a row in the 32x32 image
red = red.reshape(32,32)
green = green.reshape(32,32)
blue = blue.reshape(32,32)

# Combine the channels into a 32x32x3 image!
combined = np.dstack((red,green,blue))

```

For a more concise version, consider the following:

```

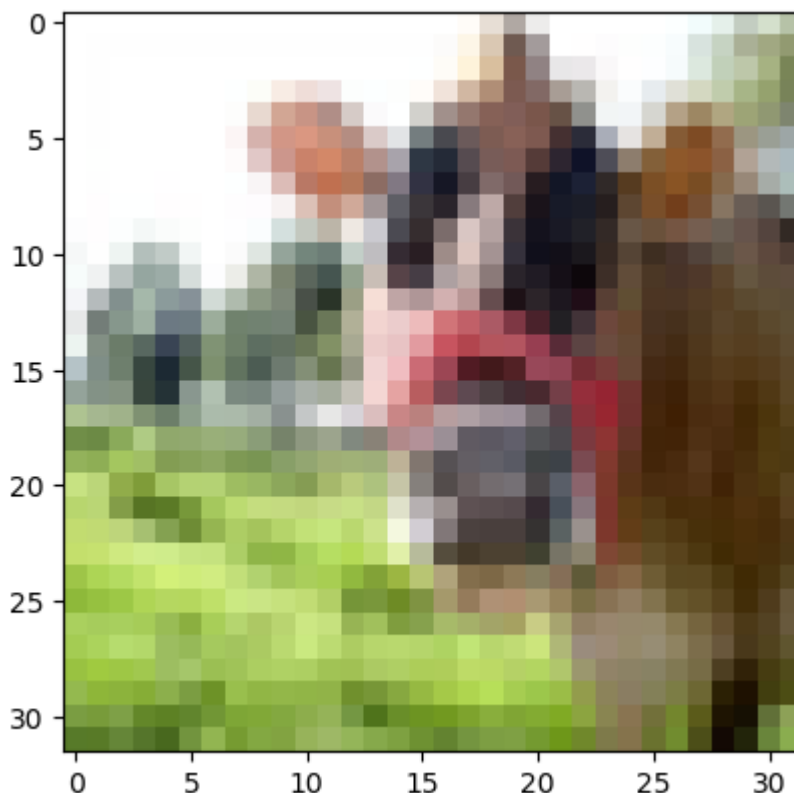
In [9]: # ALL in one:
test_image = np.dstack((
    row[0:1024].reshape(32,32),
    row[1024:2048].reshape(32,32),
    row[2048:].reshape(32,32)
))

```

```

In [10]: import matplotlib.pyplot as plt
plt.imshow(test_image);
plt.show()

```



Looks like a cow! Let's check the label. `dataset_meta` contains label names in order, and `dataset_train` has a list of labels for each row.

```

In [11]: dataset_train[b'fine_labels'][0]

```

```

Out[11]: 19

```

Our image has a label of 19, so let's see what the 19th item is in the list of label names.

```
In [12]: print(dataset_meta[b'fine_label_names'][19])  
b'cattle'
```

Ok! 'cattle' sounds about right. By the way, using the previous two lines we can do:

```
In [13]: n = 0  
print(dataset_meta[b'fine_label_names'][dataset_train[b'fine_labels'][n]])  
b'cattle'
```

Now we know how to check labels, is there a way that we can also check file names?

`dataset_train` also contains a `b'filenames'` key. Let's see what we have here:

```
In [14]: print(dataset_train[b'filenames'][0])  
b'bos_taurus_s_000507.png'
```

"Taurus" is the name of a subspecies of cattle, so this looks like a pretty reasonable filename. To save an image we can also do:

```
In [15]: plt.imshow(test_image)
```

Your new PNG file should now appear in the file explorer -- go ahead and pop it open to see!

Now that you know how to reshape the images, save them as files, and capture their filenames and labels, let's just capture all the bicycles and motorcycles and save them. Scones Unlimited can use a model that tells these apart to route delivery drivers automatically.

In the following cell, identify the label numbers for Bicycles and Motorcycles:

```
In [16]: import pandas as pd  
  
# TODO: Filter the dataset_train and dataset_meta objects to find the label numbers for  
# bicycles and motorcycles  
  
n=len(dataset_meta[b'fine_label_names'])  
d=dict(zip(dataset_meta[b'fine_label_names'],range(n)))  
{b'bicycle':d[b'bicycle'],b'motorcycle':d[b'motorcycle']}
```

```
Out[16]: {b'bicycle': 8, b'motorcycle': 48}
```

Good job! We only need objects with label 8 and 48 -- this drastically simplifies our handling of the data! Below we construct a dataframe for you, and you can safely drop the rows that don't contain observations about bicycles and motorcycles. Fill in the missing lines below to drop all other rows:

```
In [17]: #Construct the dataframe  
df_train = pd.DataFrame({  
    "filenames": dataset_train[b'filenames'],  
    "labels": dataset_train[b'fine_labels'],  
    "row": range(len(dataset_train[b'filenames']))  
})
```

```

# Drop all rows from df_train where label is not 8 or 48
df_train = df_train[df_train['labels'].isin([8,48])]

# Decode df_train.filenamees so they are regular strings
df_train["filenames"] = df_train["filenames"].apply(
    lambda x: x.decode("utf-8")
)

df_test = pd.DataFrame({
    "filenames": dataset_test[b'filenames'],
    "labels": dataset_test[b'fine_labels'],
    "row": range(len(dataset_test[b'filenames']))
})

# Drop all rows from df_test where label is not 8 or 48
df_test = df_test[df_test['labels'].isin([8,48])]

# Decode df_test.filenamees so they are regular strings
df_test["filenames"] = df_test["filenames"].apply(
    lambda x: x.decode("utf-8")
)

```

Now that the data is filtered for just our classes, we can save all our images.

In [18]:

```

!mkdir ./train
!mkdir ./test

```

```

mkdir: cannot create directory './train': File exists
mkdir: cannot create directory './test': File exists

```

In the previous sections we introduced you to several key snippets of code:

1. Grabbing the image data:

```
dataset_train[b'data'][0]
```

1. A simple idiom for stacking the image data into the right shape

```

import numpy as np
np.dstack((
    row[0:1024].reshape(32,32),
    row[1024:2048].reshape(32,32),
    row[2048:].reshape(32,32)
))

```

1. A simple `matplotlib` utility for saving images

```
plt.imsave(path+row['filenames'], target)
```

Compose these together into a function that saves all the images into the `./test` and `./train` directories. Use the comments in the body of the `save_images` function below to guide your construction of the function:

In [19]:

```

import os
def save_images(imgs_df, imgs_dataset, target_file):

```

```

# Grab the image data in row-major form
image_row_data = imgs_df['row']

imgs_file_names = imgs_df['filenames']

for row, image_name in zip(image_row_data, imgs_file_names):

    img = imgs_dataset[b'data'][row]

    # Consolidated stacking/reshaping from earlier
    target = np.dstack((
        img[0:1024].reshape(32,32,1),
        img[1024:2048].reshape(32,32,1),
        img[2048:].reshape(32,32,1)
    ))

    # Save the image
    save_img_filepath = os.path.join(target_file, image_name)
    try:
        plt.imsave(save_img_filepath, target)

        # For printing the saved image filepath in correct format
        raw_save_img_filepath = r"{}".format(save_img_filepath)
        nraw_save_img_filepath = os.path.normpath(raw_save_img_filepath)
        # print(f"Saved: {image_name} to {nraw_save_img_filepath}")

    except RuntimeError as e:
        return f"Error: {e} \nAn error encountered while saving {image_name} fil
print(f" Images saved in {target_file}")
return

## TODO: save ALL images using the save_images function

# Save all transformed train dataset images in "./train"
save_images(imgs_df= df_train,
            imgs_dataset= dataset_train,
            target_file= "./train")

# Save all transformed test dataset images in "./test"
save_images(imgs_df= df_test,
            imgs_dataset= dataset_test,
            target_file= "./test")

```

Images saved in ./train
Images saved in ./test

3. Load the data

Now we can load the data into S3.

Using the sagemaker SDK grab the current region, execution role, and bucket.

```

In [20]: import sagemaker

session = sagemaker.Session()

bucket= session.default_bucket()
print("Default Bucket: {}".format(bucket))

```

```
region = session.boto_region_name
print("AWS Region: {}".format(region))

role = sagemaker.get_execution_role()
print("RoleArn: {}".format(role))
```

Default Bucket: sagemaker-us-east-1-902062377305

AWS Region: us-east-1

RoleArn: arn:aws:iam::902062377305:role/service-role/AmazonSageMaker-ExecutionRole-20230917T211615

With this data we can easily sync your data up into S3!

In [21]:

```
import os

os.environ["DEFAULT_S3_BUCKET"] = bucket
!aws s3 sync ./train s3://{DEFAULT_S3_BUCKET}/train/
!aws s3 sync ./test s3://{DEFAULT_S3_BUCKET}/test/
```

upload: train/bicycle_s_000017.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000017.png
upload: train/bicycle_s_000038.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000038.png
upload: train/bicycle_s_000021.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000021.png
upload: train/bicycle_s_000039.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000039.png
upload: train/bicycle_s_000149.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000149.png
upload: train/bicycle_s_000035.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000035.png
upload: train/bicycle_s_000235.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000235.png
upload: train/bicycle_s_000243.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000243.png
upload: train/bicycle_s_000071.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000071.png
upload: train/bicycle_s_000180.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000180.png
upload: train/bicycle_s_000051.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000051.png
upload: train/bicycle_s_000282.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000282.png
upload: train/bicycle_s_000043.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000043.png
upload: train/bicycle_s_000231.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000231.png
upload: train/bicycle_s_000066.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000066.png
upload: train/bicycle_s_000124.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000124.png
upload: train/bicycle_s_000279.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000279.png
upload: train/bicycle_s_000369.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000369.png
upload: train/bicycle_s_000314.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000314.png
upload: train/bicycle_s_000159.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000159.png
upload: train/bicycle_s_000437.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000437.png
upload: train/bicycle_s_000371.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000371.png
upload: train/bicycle_s_000147.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000147.png
upload: train/bicycle_s_000137.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000137.png
upload: train/bicycle_s_000435.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000435.png
upload: train/bicycle_s_000399.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000399.png
upload: train/bicycle_s_000467.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000467.png
upload: train/bicycle_s_000463.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000463.png
upload: train/bicycle_s_000491.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000491.png
upload: train/bicycle_s_000408.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000408.png

upload: train/bicycle_s_000099.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000099.png
upload: train/bicycle_s_000522.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000522.png
upload: train/bicycle_s_000156.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000156.png
upload: train/bicycle_s_000396.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000396.png
upload: train/bicycle_s_000536.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000536.png
upload: train/bicycle_s_000537.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000537.png
upload: train/bicycle_s_000668.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000668.png
upload: train/bicycle_s_000723.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000723.png
upload: train/bicycle_s_000392.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000392.png
upload: train/bicycle_s_000759.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000759.png
upload: train/bicycle_s_000753.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000753.png
upload: train/bicycle_s_000569.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000569.png
upload: train/bicycle_s_000561.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000561.png
upload: train/bicycle_s_000774.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000774.png
upload: train/bicycle_s_000667.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000667.png
upload: train/bicycle_s_000546.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000546.png
upload: train/bicycle_s_000775.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000775.png
upload: train/bicycle_s_000778.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000778.png
upload: train/bicycle_s_000782.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000782.png
upload: train/bicycle_s_000781.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000781.png
upload: train/bicycle_s_000829.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000829.png
upload: train/bicycle_s_000785.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000785.png
upload: train/bicycle_s_000951.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000951.png
upload: train/bicycle_s_000822.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000822.png
upload: train/bicycle_s_001002.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001002.png
upload: train/bicycle_s_000986.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000986.png
upload: train/bicycle_s_001111.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001111.png
upload: train/bicycle_s_000861.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000861.png
upload: train/bicycle_s_001174.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001174.png
upload: train/bicycle_s_000978.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000978.png

upload: train/bicycle_s_001168.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001168.png
upload: train/bicycle_s_001203.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001203.png
upload: train/bicycle_s_001228.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001228.png
upload: train/bicycle_s_001205.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001205.png
upload: train/bicycle_s_001247.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001247.png
upload: train/bicycle_s_000996.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_000996.png
upload: train/bicycle_s_001245.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001245.png
upload: train/bicycle_s_001338.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001338.png
upload: train/bicycle_s_001402.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001402.png
upload: train/bicycle_s_001448.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001448.png
upload: train/bicycle_s_001388.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001388.png
upload: train/bicycle_s_001348.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001348.png
upload: train/bicycle_s_001409.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001409.png
upload: train/bicycle_s_001411.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001411.png
upload: train/bicycle_s_001452.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001452.png
upload: train/bicycle_s_001663.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001663.png
upload: train/bicycle_s_001554.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001554.png
upload: train/bicycle_s_001583.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001583.png
upload: train/bicycle_s_001679.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001679.png
upload: train/bicycle_s_001681.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001681.png
upload: train/bicycle_s_001569.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001569.png
upload: train/bicycle_s_001673.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001673.png
upload: train/bicycle_s_001642.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001642.png
upload: train/bicycle_s_001470.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001470.png
upload: train/bicycle_s_001687.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001687.png
upload: train/bicycle_s_001688.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001688.png
upload: train/bicycle_s_001757.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001757.png
upload: train/bicycle_s_001814.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001814.png
upload: train/bicycle_s_001748.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001748.png
upload: train/bicycle_s_002029.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002029.png

upload: train/bicycle_s_001928.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001928.png
upload: train/bicycle_s_002012.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002012.png
upload: train/bicycle_s_001693.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001693.png
upload: train/bicycle_s_002005.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002005.png
upload: train/bicycle_s_002049.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002049.png
upload: train/bicycle_s_001904.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001904.png
upload: train/bicycle_s_002100.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002100.png
upload: train/bicycle_s_001956.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_001956.png
upload: train/bicycle_s_002218.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002218.png
upload: train/bicycle_s_002129.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002129.png
upload: train/bicycle_s_002132.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002132.png
upload: train/bicycle_s_002237.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002237.png
upload: train/bicycle_s_002258.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002258.png
upload: train/bicycle_s_002299.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002299.png
upload: train/bicycle_s_002247.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002247.png
upload: train/bicycle_s_002373.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002373.png
upload: train/bicycle_s_002338.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002338.png
upload: train/bicycle_s_002374.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002374.png
upload: train/bicycle_s_002222.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002222.png
upload: train/bicycle_s_002410.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002410.png
upload: train/bicycle_s_002436.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002436.png
upload: train/bicycle_s_002400.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002400.png
upload: train/bicycle_s_002569.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002569.png
upload: train/bicycle_s_002475.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002475.png
upload: train/bicycle_s_002153.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002153.png
upload: train/bicycle_s_002661.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002661.png
upload: train/bicycle_s_002521.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002521.png
upload: train/bicycle_s_002624.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002624.png
upload: train/bicycle_s_002669.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002669.png
upload: train/bicycle_s_002448.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002448.png

upload: train/bicycle_s_002550.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002550.png
upload: train/bicycle_s_002711.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002711.png
upload: train/bicycle_s_002715.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002715.png
upload: train/bicycle_s_002762.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002762.png
upload: train/bicycle_s_002729.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002729.png
upload: train/bicycle_s_002759.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002759.png
upload: train/bike_s_000018.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000018.png
upload: train/bike_s_000015.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000015.png
upload: train/bike_s_000005.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000005.png
upload: train/bicycle_s_002728.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002728.png
upload: train/bicycle_s_002725.png to s3://sagemaker-us-east-1-902062377305/train/bicycle_s_002725.png
upload: train/bike_s_000001.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000001.png
upload: train/bike_s_000025.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000025.png
upload: train/bike_s_000021.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000021.png
upload: train/bike_s_000035.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000035.png
upload: train/bike_s_000040.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000040.png
upload: train/bike_s_000023.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000023.png
upload: train/bike_s_000034.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000034.png
upload: train/bike_s_000026.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000026.png
upload: train/bike_s_000003.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000003.png
upload: train/bike_s_000024.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000024.png
upload: train/bike_s_000164.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000164.png
upload: train/bike_s_000129.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000129.png
upload: train/bike_s_000154.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000154.png
upload: train/bike_s_000162.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000162.png
upload: train/bike_s_000121.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000121.png
upload: train/bike_s_000111.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000111.png
upload: train/bike_s_000127.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000127.png
upload: train/bike_s_000062.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000062.png
upload: train/bike_s_000204.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000204.png

upload: train/bike_s_000051.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000051.png
upload: train/bike_s_000256.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000256.png
upload: train/bike_s_000299.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000299.png
upload: train/bike_s_000390.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000390.png
upload: train/bike_s_000237.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000237.png
upload: train/bike_s_000336.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000336.png
upload: train/bike_s_000302.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000302.png
upload: train/bike_s_000304.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000304.png
upload: train/bike_s_000397.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000397.png
upload: train/bike_s_000364.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000364.png
upload: train/bike_s_000392.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000392.png
upload: train/bike_s_000544.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000544.png
upload: train/bike_s_000516.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000516.png
upload: train/bike_s_000522.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000522.png
upload: train/bike_s_000506.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000506.png
upload: train/bike_s_000474.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000474.png
upload: train/bike_s_000545.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000545.png
upload: train/bike_s_000628.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000628.png
upload: train/bike_s_000657.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000657.png
upload: train/bike_s_000593.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000593.png
upload: train/bike_s_000555.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000555.png
upload: train/bike_s_000934.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000934.png
upload: train/bike_s_000679.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000679.png
upload: train/bike_s_000682.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000682.png
upload: train/bike_s_000990.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000990.png
upload: train/bike_s_000722.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000722.png
upload: train/bike_s_001072.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001072.png
upload: train/bike_s_000926.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000926.png
upload: train/bike_s_001027.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001027.png
upload: train/bike_s_001078.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001078.png

upload: train/bike_s_000855.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_000855.png
upload: train/bike_s_001093.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001093.png
upload: train/bike_s_001113.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001113.png
upload: train/bike_s_001131.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001131.png
upload: train/bike_s_001200.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001200.png
upload: train/bike_s_001320.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001320.png
upload: train/bike_s_001260.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001260.png
upload: train/bike_s_001116.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001116.png
upload: train/bike_s_001418.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001418.png
upload: train/bike_s_001415.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001415.png
upload: train/bike_s_001380.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001380.png
upload: train/bike_s_001226.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001226.png
upload: train/bike_s_001375.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001375.png
upload: train/bike_s_001414.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001414.png
upload: train/bike_s_001519.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001519.png
upload: train/bike_s_001462.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001462.png
upload: train/bike_s_001683.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001683.png
upload: train/bike_s_001572.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001572.png
upload: train/bike_s_001679.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001679.png
upload: train/bike_s_001517.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001517.png
upload: train/bike_s_001761.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001761.png
upload: train/bike_s_001739.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001739.png
upload: train/bike_s_001613.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001613.png
upload: train/bike_s_001592.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001592.png
upload: train/bike_s_001839.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001839.png
upload: train/bike_s_001827.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001827.png
upload: train/bike_s_001767.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001767.png
upload: train/bike_s_001847.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001847.png
upload: train/bike_s_001876.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001876.png
upload: train/bike_s_001897.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001897.png

upload: train/bike_s_001877.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001877.png
upload: train/bike_s_001915.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001915.png
upload: train/bike_s_002090.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002090.png
upload: train/bike_s_001928.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001928.png
upload: train/bike_s_002024.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002024.png
upload: train/bike_s_002033.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002033.png
upload: train/bike_s_002042.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002042.png
upload: train/bike_s_001945.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001945.png
upload: train/bike_s_001980.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001980.png
upload: train/bike_s_001882.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_001882.png
upload: train/bike_s_002118.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002118.png
upload: train/bike_s_002098.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002098.png
upload: train/bike_s_002139.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002139.png
upload: train/bike_s_002277.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002277.png
upload: train/bike_s_002047.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002047.png
upload: train/bike_s_002164.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002164.png
upload: train/bike_s_002288.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002288.png
upload: train/bike_s_002222.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002222.png
upload: train/bike_s_002109.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002109.png
upload: train/bike_s_002215.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002215.png
upload: train/bike_s_002283.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002283.png
upload: train/bike_s_002292.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002292.png
upload: train/bike_s_002203.png to s3://sagemaker-us-east-1-902062377305/train/bike_s_002203.png
upload: train/cycle_s_000042.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_000042.png
upload: train/cycle_s_000492.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_000492.png
upload: train/cycle_s_000318.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_000318.png
upload: train/cycle_s_000463.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_000463.png
upload: train/cycle_s_000222.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_000222.png
upload: train/cycle_s_000639.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_000639.png
upload: train/cycle_s_000583.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_000583.png

upload: train/cycle_s_000871.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_000871.png
upload: train/cycle_s_000718.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_000718.png
upload: train/cycle_s_001309.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001309.png
upload: train/cycle_s_001286.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001286.png
upload: train/cycle_s_001412.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001412.png
upload: train/cycle_s_000666.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_000666.png
upload: train/cycle_s_000899.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_000899.png
upload: train/cycle_s_001439.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001439.png
upload: train/cycle_s_001413.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001413.png
upload: train/cycle_s_001474.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001474.png
upload: train/cycle_s_001640.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001640.png
upload: train/cycle_s_001472.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001472.png
upload: train/cycle_s_001374.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001374.png
upload: train/cycle_s_001976.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001976.png
upload: train/cycle_s_001735.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001735.png
upload: train/cycle_s_001745.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001745.png
upload: train/cycle_s_001875.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001875.png
upload: train/cycle_s_001477.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_001477.png
upload: train/cycle_s_002092.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002092.png
upload: train/cycle_s_002399.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002399.png
upload: train/cycle_s_002093.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002093.png
upload: train/cycle_s_002090.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002090.png
upload: train/cycle_s_002598.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002598.png
upload: train/cycle_s_002015.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002015.png
upload: train/cycle_s_002178.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002178.png
upload: train/cycle_s_002053.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002053.png
upload: train/cycle_s_002503.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002503.png
upload: train/cycle_s_002638.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002638.png
upload: train/cycle_s_002651.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002651.png
upload: train/cycle_s_002703.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002703.png

upload: train/cycle_s_002666.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002666.png
upload: train/cycle_s_002168.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002168.png
upload: train/cycle_s_002746.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002746.png
upload: train/cycle_s_002659.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002659.png
upload: train/cycle_s_002978.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002978.png
upload: train/cycle_s_002904.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002904.png
upload: train/cycle_s_003006.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_003006.png
upload: train/cycle_s_002844.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002844.png
upload: train/cycle_s_002882.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002882.png
upload: train/cycle_s_003043.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_003043.png
upload: train/cycle_s_002779.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_002779.png
upload: train/cycle_s_003008.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_003008.png
upload: train/cycle_s_003026.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_003026.png
upload: train/cycle_s_003147.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_003147.png
upload: train/cycle_s_003148.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_003148.png
upload: train/dirt_bike_s_000003.png to s3://sagemaker-us-east-1-902062377305/train/dirt_bike_s_000003.png
upload: train/cycle_s_003162.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_003162.png
upload: train/dirt_bike_s_000005.png to s3://sagemaker-us-east-1-902062377305/train/dirt_bike_s_000005.png
upload: train/cycle_s_003122.png to s3://sagemaker-us-east-1-902062377305/train/cycle_s_003122.png
upload: train/dirt_bike_s_000017.png to s3://sagemaker-us-east-1-902062377305/train/dirt_bike_s_000017.png
upload: train/dirt_bike_s_000030.png to s3://sagemaker-us-east-1-902062377305/train/dirt_bike_s_000030.png
upload: train/dirt_bike_s_000059.png to s3://sagemaker-us-east-1-902062377305/train/dirt_bike_s_000059.png
upload: train/dirt_bike_s_000119.png to s3://sagemaker-us-east-1-902062377305/train/dirt_bike_s_000119.png
upload: train/dirt_bike_s_000124.png to s3://sagemaker-us-east-1-902062377305/train/dirt_bike_s_000124.png
upload: train/dirt_bike_s_000134.png to s3://sagemaker-us-east-1-902062377305/train/dirt_bike_s_000134.png
upload: train/minibike_s_000010.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000010.png
upload: train/minibike_s_000020.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000020.png
upload: train/minibike_s_000011.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000011.png
upload: train/minibike_s_000036.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000036.png
upload: train/minibike_s_000022.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000022.png

[illegible]

[illegible]

upload: train/minibike_s_000709.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000709.png
upload: train/minibike_s_000802.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000802.png
upload: train/minibike_s_000741.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000741.png
upload: train/minibike_s_000738.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000738.png
upload: train/minibike_s_000824.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000824.png
upload: train/minibike_s_000813.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000813.png
upload: train/minibike_s_000820.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000820.png
upload: train/minibike_s_000830.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000830.png
upload: train/minibike_s_000831.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000831.png
upload: train/minibike_s_000855.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000855.png
upload: train/minibike_s_000885.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000885.png
upload: train/minibike_s_000886.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000886.png
upload: train/minibike_s_000507.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000507.png
upload: train/minibike_s_000906.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000906.png
upload: train/minibike_s_001016.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001016.png
upload: train/minibike_s_001079.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001079.png
upload: train/minibike_s_000968.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000968.png
upload: train/minibike_s_001072.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001072.png
upload: train/minibike_s_000914.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000914.png
upload: train/minibike_s_001094.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001094.png
upload: train/minibike_s_001017.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001017.png
upload: train/minibike_s_000967.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_000967.png
upload: train/minibike_s_001168.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001168.png
upload: train/minibike_s_001169.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001169.png
upload: train/minibike_s_001157.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001157.png
upload: train/minibike_s_001185.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001185.png
upload: train/minibike_s_001177.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001177.png
upload: train/minibike_s_001193.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001193.png
upload: train/minibike_s_001179.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001179.png
upload: train/minibike_s_001249.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001249.png

upload: train/minibike_s_001250.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001250.png
upload: train/minibike_s_001191.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001191.png
upload: train/minibike_s_001270.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001270.png
upload: train/minibike_s_001275.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001275.png
upload: train/minibike_s_001261.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001261.png
upload: train/minibike_s_001279.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001279.png
upload: train/minibike_s_001348.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001348.png
upload: train/minibike_s_001366.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001366.png
upload: train/minibike_s_001294.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001294.png
upload: train/minibike_s_001344.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001344.png
upload: train/minibike_s_001345.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001345.png
upload: train/minibike_s_001450.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001450.png
upload: train/minibike_s_001417.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001417.png
upload: train/minibike_s_001496.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001496.png
upload: train/minibike_s_001479.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001479.png
upload: train/minibike_s_001458.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001458.png
upload: train/minibike_s_001506.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001506.png
upload: train/minibike_s_001498.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001498.png
upload: train/minibike_s_001491.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001491.png
upload: train/minibike_s_001512.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001512.png
upload: train/minibike_s_001511.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001511.png
upload: train/minibike_s_001539.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001539.png
upload: train/minibike_s_001565.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001565.png
upload: train/minibike_s_001540.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001540.png
upload: train/minibike_s_001575.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001575.png
upload: train/minibike_s_001558.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001558.png
upload: train/minibike_s_001546.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001546.png
upload: train/minibike_s_001573.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001573.png
upload: train/minibike_s_001597.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001597.png
upload: train/minibike_s_001631.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001631.png

upload: train/minibike_s_001689.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001689.png
upload: train/minibike_s_001691.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001691.png
upload: train/minibike_s_001653.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001653.png
upload: train/minibike_s_001701.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001701.png
upload: train/minibike_s_001639.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001639.png
upload: train/minibike_s_001638.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001638.png
upload: train/minibike_s_001747.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001747.png
upload: train/minibike_s_001734.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001734.png
upload: train/minibike_s_001731.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001731.png
upload: train/minibike_s_001767.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001767.png
upload: train/minibike_s_001838.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001838.png
upload: train/minibike_s_001827.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001827.png
upload: train/minibike_s_001865.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001865.png
upload: train/minibike_s_001870.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001870.png
upload: train/minibike_s_001873.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001873.png
upload: train/minibike_s_001836.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001836.png
upload: train/minibike_s_001789.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001789.png
upload: train/minibike_s_001829.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001829.png
upload: train/minibike_s_001771.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001771.png
upload: train/minibike_s_001863.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001863.png
upload: train/minibike_s_001880.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001880.png
upload: train/minibike_s_001881.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001881.png
upload: train/minibike_s_001885.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001885.png
upload: train/minibike_s_001944.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001944.png
upload: train/minibike_s_001921.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001921.png
upload: train/minibike_s_001996.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001996.png
upload: train/minibike_s_001969.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001969.png
upload: train/minibike_s_002009.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002009.png
upload: train/minibike_s_001972.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001972.png
upload: train/minibike_s_002030.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002030.png

upload: train/minibike_s_002048.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002048.png
upload: train/minibike_s_001917.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_001917.png
upload: train/minibike_s_002046.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002046.png
upload: train/minibike_s_002038.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002038.png
upload: train/minibike_s_002130.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002130.png
upload: train/minibike_s_002124.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002124.png
upload: train/minibike_s_002146.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002146.png
upload: train/minibike_s_002115.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002115.png
upload: train/minibike_s_002084.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002084.png
upload: train/minibike_s_002163.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002163.png
upload: train/minibike_s_002176.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002176.png
upload: train/minibike_s_002182.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002182.png
upload: train/minibike_s_002186.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002186.png
upload: train/moped_s_000004.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000004.png
upload: train/minibike_s_002137.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002137.png
upload: train/minibike_s_002194.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002194.png
upload: train/minibike_s_002196.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002196.png
upload: train/minibike_s_002198.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002198.png
upload: train/minibike_s_002218.png to s3://sagemaker-us-east-1-902062377305/train/minibike_s_002218.png
upload: train/moped_s_000009.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000009.png
upload: train/moped_s_000021.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000021.png
upload: train/moped_s_000030.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000030.png
upload: train/moped_s_000034.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000034.png
upload: train/moped_s_000044.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000044.png
upload: train/moped_s_000035.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000035.png
upload: train/moped_s_000236.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000236.png
upload: train/moped_s_000065.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000065.png
upload: train/moped_s_000169.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000169.png
upload: train/moped_s_000119.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000119.png
upload: train/moped_s_000135.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000135.png

upload: train/moped_s_000071.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000071.png
upload: train/moped_s_000168.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000168.png
upload: train/moped_s_000124.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000124.png
upload: train/motorbike_s_000009.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000009.png
upload: train/motorbike_s_000021.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000021.png
upload: train/moped_s_000237.png to s3://sagemaker-us-east-1-902062377305/train/moped_s_000237.png
upload: train/motorbike_s_000035.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000035.png
upload: train/motorbike_s_000041.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000041.png
upload: train/motorbike_s_000022.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000022.png
upload: train/motorbike_s_000058.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000058.png
upload: train/motorbike_s_000124.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000124.png
upload: train/motorbike_s_000115.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000115.png
upload: train/motorbike_s_000062.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000062.png
upload: train/motorbike_s_000068.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000068.png
upload: train/motorbike_s_000117.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000117.png
upload: train/motorbike_s_000119.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000119.png
upload: train/motorbike_s_000106.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000106.png
upload: train/motorbike_s_000134.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000134.png
upload: train/motorbike_s_000060.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000060.png
upload: train/motorbike_s_000141.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000141.png
upload: train/motorbike_s_000223.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000223.png
upload: train/motorbike_s_000221.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000221.png
upload: train/motorbike_s_000254.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000254.png
upload: train/motorbike_s_000225.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000225.png
upload: train/motorbike_s_000308.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000308.png
upload: train/motorbike_s_000346.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000346.png
upload: train/motorbike_s_000362.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000362.png
upload: train/motorbike_s_000427.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000427.png
upload: train/motorbike_s_000361.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000361.png
upload: train/motorbike_s_000449.png to s3://sagemaker-us-east-1-902062377305/train/motorbike_s_000449.png

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

upload: train/motorcycle_s_002066.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002066.png
upload: train/motorcycle_s_002030.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002030.png
upload: train/motorcycle_s_001878.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_001878.png
upload: train/motorcycle_s_002153.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002153.png
upload: train/motorcycle_s_002031.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002031.png
upload: train/motorcycle_s_002067.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002067.png
upload: train/motorcycle_s_002105.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002105.png
upload: train/motorcycle_s_002121.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002121.png
upload: train/motorcycle_s_002143.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002143.png
upload: train/motorcycle_s_002140.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002140.png
upload: train/motorcycle_s_002197.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002197.png
upload: train/motorcycle_s_002192.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002192.png
upload: train/motorcycle_s_002162.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002162.png
upload: train/motorcycle_s_002183.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002183.png
upload: train/motorcycle_s_002214.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002214.png
upload: train/motorcycle_s_002193.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002193.png
upload: train/motorcycle_s_002215.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002215.png
upload: train/motorcycle_s_002222.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002222.png
upload: train/motorcycle_s_002237.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002237.png
upload: train/motorcycle_s_002225.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002225.png
upload: train/motorcycle_s_002236.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002236.png
upload: train/motorcycle_s_002221.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002221.png
upload: train/motorcycle_s_002291.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002291.png
upload: train/motorcycle_s_002234.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002234.png
upload: train/motorcycle_s_002271.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002271.png
upload: train/motorcycle_s_002254.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002254.png
upload: train/motorcycle_s_002275.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002275.png
upload: train/motorcycle_s_002298.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002298.png
upload: train/motorcycle_s_002316.png to s3://sagemaker-us-east-1-902062377305/train/motorcycle_s_002316.png
upload: train/ordinary_bicycle_s_000022.png to s3://sagemaker-us-east-1-902062377305/train/ordinary_bicycle_s_000022.png

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

upload: train/safety_bike_s_000989.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_000989.png
upload: train/safety_bike_s_001289.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001289.png
upload: train/safety_bike_s_001465.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001465.png
upload: train/safety_bike_s_001257.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001257.png
upload: train/safety_bike_s_001371.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001371.png
upload: train/safety_bike_s_001012.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001012.png
upload: train/safety_bike_s_001474.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001474.png
upload: train/safety_bike_s_001560.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001560.png
upload: train/safety_bike_s_001715.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001715.png
upload: train/safety_bike_s_001640.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001640.png
upload: train/safety_bike_s_001659.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001659.png
upload: train/safety_bike_s_001472.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001472.png
upload: train/safety_bike_s_001552.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001552.png
upload: train/safety_bike_s_001608.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001608.png
upload: train/safety_bike_s_001590.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001590.png
upload: train/trail_bike_s_000016.png to s3://sagemaker-us-east-1-902062377305/train/trail_bike_s_000016.png
upload: train/safety_bike_s_001784.png to s3://sagemaker-us-east-1-902062377305/train/safety_bike_s_001784.png
upload: train/velocipede_s_000010.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000010.png
upload: train/velocipede_s_000012.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000012.png
upload: train/velocipede_s_000049.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000049.png
upload: train/velocipede_s_000265.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000265.png
upload: train/velocipede_s_000021.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000021.png
upload: train/velocipede_s_000023.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000023.png
upload: train/velocipede_s_000139.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000139.png
upload: train/velocipede_s_000282.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000282.png
upload: train/velocipede_s_000430.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000430.png
upload: train/velocipede_s_000358.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000358.png
upload: train/velocipede_s_000807.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000807.png
upload: train/velocipede_s_000702.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000702.png
upload: train/velocipede_s_000485.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000485.png

upload: train/velocipede_s_000659.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000659.png
upload: train/velocipede_s_000586.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000586.png
upload: train/velocipede_s_000825.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000825.png
upload: train/velocipede_s_000931.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000931.png
upload: train/velocipede_s_000670.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000670.png
upload: train/velocipede_s_001142.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001142.png
upload: train/velocipede_s_000330.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000330.png
upload: train/velocipede_s_000949.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000949.png
upload: train/velocipede_s_001141.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001141.png
upload: train/velocipede_s_001056.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001056.png
upload: train/velocipede_s_001158.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001158.png
upload: train/velocipede_s_000989.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_000989.png
upload: train/velocipede_s_001210.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001210.png
upload: train/velocipede_s_001200.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001200.png
upload: train/velocipede_s_001222.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001222.png
upload: train/velocipede_s_001166.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001166.png
upload: train/velocipede_s_001225.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001225.png
upload: train/velocipede_s_001298.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001298.png
upload: train/velocipede_s_001299.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001299.png
upload: train/velocipede_s_001278.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001278.png
upload: train/velocipede_s_001338.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001338.png
upload: train/velocipede_s_001422.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001422.png
upload: train/velocipede_s_001358.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001358.png
upload: train/velocipede_s_001514.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001514.png
upload: train/velocipede_s_001361.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001361.png
upload: train/velocipede_s_001637.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001637.png
upload: train/velocipede_s_001585.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001585.png
upload: train/velocipede_s_001244.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001244.png
upload: train/velocipede_s_001872.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001872.png
upload: train/velocipede_s_001279.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001279.png

upload: train/velocipede_s_001489.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001489.png
upload: train/velocipede_s_001883.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001883.png
upload: train/velocipede_s_001882.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001882.png
upload: train/velocipede_s_001880.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001880.png
upload: train/velocipede_s_001870.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001870.png
upload: train/velocipede_s_001920.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001920.png
upload: train/velocipede_s_001907.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001907.png
upload: train/velocipede_s_001874.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001874.png
upload: train/velocipede_s_001935.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001935.png
upload: train/velocipede_s_001958.png to s3://sagemaker-us-east-1-902062377305/train/velocipede_s_001958.png
upload: test/bicycle_s_000030.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_000030.png
upload: test/bicycle_s_000777.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_000777.png
upload: test/bicycle_s_000059.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_000059.png
upload: test/bicycle_s_000479.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_000479.png
upload: test/bicycle_s_000031.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_000031.png
upload: test/bicycle_s_000977.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_000977.png
upload: test/bicycle_s_001789.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_001789.png
upload: test/bicycle_s_001804.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_001804.png
upload: test/bicycle_s_001047.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_001047.png
upload: test/bicycle_s_000513.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_000513.png
upload: test/bicycle_s_001107.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_001107.png
upload: test/bicycle_s_000779.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_000779.png
upload: test/bicycle_s_002735.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_002735.png
upload: test/bicycle_s_002458.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_002458.png
upload: test/bicycle_s_001218.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_001218.png
upload: test/bicycle_s_001102.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_001102.png
upload: test/bicycle_s_000776.png to s3://sagemaker-us-east-1-902062377305/test/bicycle_s_000776.png
upload: test/bike_s_000041.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000041.png
upload: test/bike_s_000131.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000131.png
upload: test/bike_s_000457.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000457.png

upload: test/bike_s_000643.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000643.png
upload: test/bike_s_000330.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000330.png
upload: test/bike_s_000487.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000487.png
upload: test/bike_s_000071.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000071.png
upload: test/bike_s_000658.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000658.png
upload: test/bike_s_000694.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000694.png
upload: test/bike_s_000163.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000163.png
upload: test/bike_s_000941.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000941.png
upload: test/bike_s_001073.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_001073.png
upload: test/bike_s_001068.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_001068.png
upload: test/bike_s_001342.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_001342.png
upload: test/bike_s_001159.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_001159.png
upload: test/bike_s_001738.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_001738.png
upload: test/bike_s_002116.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_002116.png
upload: test/bike_s_001216.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_001216.png
upload: test/bike_s_000801.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_000801.png
upload: test/bike_s_001852.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_001852.png
upload: test/bike_s_001784.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_001784.png
upload: test/bike_s_002208.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_002208.png
upload: test/cycle_s_001044.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_001044.png
upload: test/cycle_s_000068.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_000068.png
upload: test/cycle_s_000970.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_000970.png
upload: test/cycle_s_000010.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_000010.png
upload: test/cycle_s_001297.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_001297.png
upload: test/cycle_s_001214.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_001214.png
upload: test/cycle_s_002305.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_002305.png
upload: test/cycle_s_002661.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_002661.png
upload: test/cycle_s_001306.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_001306.png
upload: test/cycle_s_001648.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_001648.png
upload: test/bike_s_002009.png to s3://sagemaker-us-east-1-902062377305/test/bike_s_002009.png

upload: test/cycle_s_001915.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_001915.png
upload: test/cycle_s_002964.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_002964.png
upload: test/minibike_s_000288.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000288.png
upload: test/minibike_s_000075.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000075.png
upload: test/cycle_s_002613.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_002613.png
upload: test/dirt_bike_s_000001.png to s3://sagemaker-us-east-1-902062377305/test/dirt_bike_s_000001.png
upload: test/minibike_s_000254.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000254.png
upload: test/cycle_s_001953.png to s3://sagemaker-us-east-1-902062377305/test/cycle_s_001953.png
upload: test/minibike_s_000052.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000052.png
upload: test/minibike_s_000055.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000055.png
upload: test/minibike_s_000290.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000290.png
upload: test/minibike_s_000309.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000309.png
upload: test/minibike_s_000398.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000398.png
upload: test/minibike_s_000497.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000497.png
upload: test/minibike_s_000801.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000801.png
upload: test/minibike_s_000880.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000880.png
upload: test/minibike_s_000505.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000505.png
upload: test/minibike_s_000947.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000947.png
upload: test/minibike_s_000792.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000792.png
upload: test/minibike_s_000573.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000573.png
upload: test/minibike_s_000913.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000913.png
upload: test/minibike_s_001441.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_001441.png
upload: test/minibike_s_001089.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_001089.png
upload: test/minibike_s_000828.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000828.png
upload: test/minibike_s_001473.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_001473.png
upload: test/minibike_s_000960.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_000960.png
upload: test/minibike_s_001651.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_001651.png
upload: test/minibike_s_001605.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_001605.png
upload: test/minibike_s_001732.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_001732.png
upload: test/minibike_s_001867.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_001867.png

upload: test/minibike_s_001547.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_001547.png
upload: test/minibike_s_002227.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_002227.png
upload: test/minibike_s_002051.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_002051.png
upload: test/minibike_s_001893.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_001893.png
upload: test/moped_s_000007.png to s3://sagemaker-us-east-1-902062377305/test/moped_s_000007.png
upload: test/moped_s_000033.png to s3://sagemaker-us-east-1-902062377305/test/moped_s_000033.png
upload: test/moped_s_000064.png to s3://sagemaker-us-east-1-902062377305/test/moped_s_000064.png
upload: test/motorbike_s_000121.png to s3://sagemaker-us-east-1-902062377305/test/motorbike_s_000121.png
upload: test/minibike_s_002173.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_002173.png
upload: test/motorbike_s_000005.png to s3://sagemaker-us-east-1-902062377305/test/motorbike_s_000005.png
upload: test/motorbike_s_000324.png to s3://sagemaker-us-east-1-902062377305/test/motorbike_s_000324.png
upload: test/motorbike_s_000433.png to s3://sagemaker-us-east-1-902062377305/test/motorbike_s_000433.png
upload: test/moped_s_000306.png to s3://sagemaker-us-east-1-902062377305/test/moped_s_000306.png
upload: test/motorbike_s_000333.png to s3://sagemaker-us-east-1-902062377305/test/motorbike_s_000333.png
upload: test/motorbike_s_000629.png to s3://sagemaker-us-east-1-902062377305/test/motorbike_s_000629.png
upload: test/motorbike_s_000465.png to s3://sagemaker-us-east-1-902062377305/test/motorbike_s_000465.png
upload: test/motorbike_s_000135.png to s3://sagemaker-us-east-1-902062377305/test/motorbike_s_000135.png
upload: test/motorcycle_s_000007.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000007.png
upload: test/minibike_s_002230.png to s3://sagemaker-us-east-1-902062377305/test/minibike_s_002230.png
upload: test/motorcycle_s_000139.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000139.png
upload: test/motorcycle_s_000040.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000040.png
upload: test/motorcycle_s_000060.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000060.png
upload: test/motorcycle_s_000049.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000049.png
upload: test/motorbike_s_000126.png to s3://sagemaker-us-east-1-902062377305/test/motorbike_s_000126.png
upload: test/motorcycle_s_000042.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000042.png
upload: test/motorcycle_s_000141.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000141.png
upload: test/motorcycle_s_000063.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000063.png
upload: test/motorcycle_s_000222.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000222.png
upload: test/motorcycle_s_000323.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000323.png
upload: test/motorcycle_s_000352.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000352.png

upload: test/motorcycle_s_000427.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000427.png
upload: test/motorcycle_s_000446.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000446.png
upload: test/motorcycle_s_000211.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000211.png
upload: test/motorcycle_s_000450.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000450.png
upload: test/motorcycle_s_000494.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000494.png
upload: test/motorcycle_s_000485.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000485.png
upload: test/motorcycle_s_000171.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000171.png
upload: test/motorcycle_s_000512.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000512.png
upload: test/motorcycle_s_000615.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000615.png
upload: test/motorcycle_s_000679.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000679.png
upload: test/motorcycle_s_000530.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000530.png
upload: test/motorcycle_s_000739.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000739.png
upload: test/motorcycle_s_000825.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000825.png
upload: test/motorcycle_s_000797.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000797.png
upload: test/motorcycle_s_000866.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000866.png
upload: test/motorcycle_s_000685.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000685.png
upload: test/motorcycle_s_000602.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000602.png
upload: test/motorcycle_s_001249.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001249.png
upload: test/motorcycle_s_000878.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000878.png
upload: test/motorcycle_s_000876.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000876.png
upload: test/motorcycle_s_001261.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001261.png
upload: test/motorcycle_s_000606.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000606.png
upload: test/motorcycle_s_000963.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_000963.png
upload: test/motorcycle_s_001519.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001519.png
upload: test/motorcycle_s_001385.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001385.png
upload: test/motorcycle_s_001164.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001164.png
upload: test/motorcycle_s_001269.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001269.png
upload: test/motorcycle_s_001348.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001348.png
upload: test/motorcycle_s_001782.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001782.png
upload: test/motorcycle_s_001892.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001892.png

upload: test/motorcycle_s_001508.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001508.png
upload: test/motorcycle_s_001679.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001679.png
upload: test/motorcycle_s_001936.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001936.png
upload: test/motorcycle_s_001687.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001687.png
upload: test/motorcycle_s_001971.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001971.png
upload: test/motorcycle_s_002026.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_002026.png
upload: test/motorcycle_s_001955.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001955.png
upload: test/motorcycle_s_001960.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001960.png
upload: test/ordinary_bicycle_s_000105.png to s3://sagemaker-us-east-1-902062377305/test/ordinary_bicycle_s_000105.png
upload: test/motorcycle_s_002126.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_002126.png
upload: test/motorcycle_s_002112.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_002112.png
upload: test/ordinary_bicycle_s_000113.png to s3://sagemaker-us-east-1-902062377305/test/ordinary_bicycle_s_000113.png
upload: test/ordinary_bicycle_s_000169.png to s3://sagemaker-us-east-1-902062377305/test/ordinary_bicycle_s_000169.png
upload: test/motorcycle_s_002177.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_002177.png
upload: test/motorcycle_s_001906.png to s3://sagemaker-us-east-1-902062377305/test/motorcycle_s_001906.png
upload: test/ordinary_bicycle_s_000437.png to s3://sagemaker-us-east-1-902062377305/test/ordinary_bicycle_s_000437.png
upload: test/safety_bicycle_s_000315.png to s3://sagemaker-us-east-1-902062377305/test/safety_bicycle_s_000315.png
upload: test/ordinary_bicycle_s_000284.png to s3://sagemaker-us-east-1-902062377305/test/ordinary_bicycle_s_000284.png
upload: test/ordinary_bicycle_s_000431.png to s3://sagemaker-us-east-1-902062377305/test/ordinary_bicycle_s_000431.png
upload: test/safety_bicycle_s_000280.png to s3://sagemaker-us-east-1-902062377305/test/safety_bicycle_s_000280.png
upload: test/ordinary_bicycle_s_000350.png to s3://sagemaker-us-east-1-902062377305/test/ordinary_bicycle_s_000350.png
upload: test/safety_bicycle_s_001322.png to s3://sagemaker-us-east-1-902062377305/test/safety_bicycle_s_001322.png
upload: test/safety_bicycle_s_001153.png to s3://sagemaker-us-east-1-902062377305/test/safety_bicycle_s_001153.png
upload: test/safety_bike_s_000390.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_000390.png
upload: test/ordinary_bicycle_s_000158.png to s3://sagemaker-us-east-1-902062377305/test/ordinary_bicycle_s_000158.png
upload: test/safety_bike_s_000438.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_000438.png
upload: test/safety_bicycle_s_001286.png to s3://sagemaker-us-east-1-902062377305/test/safety_bicycle_s_001286.png
upload: test/safety_bicycle_s_000235.png to s3://sagemaker-us-east-1-902062377305/test/safety_bicycle_s_000235.png
upload: test/safety_bicycle_s_001575.png to s3://sagemaker-us-east-1-902062377305/test/safety_bicycle_s_001575.png
upload: test/safety_bike_s_000488.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_000488.png

upload: test/safety_bike_s_001065.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_001065.png
upload: test/safety_bike_s_000791.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_000791.png
upload: test/safety_bike_s_001132.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_001132.png
upload: test/safety_bike_s_000699.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_000699.png
upload: test/safety_bike_s_001253.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_001253.png
upload: test/safety_bike_s_000540.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_000540.png
upload: test/safety_bike_s_000192.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_000192.png
upload: test/safety_bike_s_001481.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_001481.png
upload: test/safety_bike_s_001088.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_001088.png
upload: test/safety_bike_s_000541.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_000541.png
upload: test/safety_bike_s_001087.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_001087.png
upload: test/velocipede_s_000106.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_000106.png
upload: test/velocipede_s_000292.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_000292.png
upload: test/safety_bike_s_001690.png to s3://sagemaker-us-east-1-902062377305/test/safety_bike_s_001690.png
upload: test/velocipede_s_000041.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_000041.png
upload: test/trail_bike_s_000073.png to s3://sagemaker-us-east-1-902062377305/test/trail_bike_s_000073.png
upload: test/velocipede_s_000001.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_000001.png
upload: test/velocipede_s_001201.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_001201.png
upload: test/velocipede_s_000863.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_000863.png
upload: test/velocipede_s_001335.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_001335.png
upload: test/velocipede_s_001379.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_001379.png
upload: test/velocipede_s_000369.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_000369.png
upload: test/velocipede_s_001633.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_001633.png
upload: test/velocipede_s_001277.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_001277.png
upload: test/velocipede_s_001466.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_001466.png
upload: test/velocipede_s_001744.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_001744.png
upload: test/velocipede_s_001232.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_001232.png
upload: test/velocipede_s_001699.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_001699.png
upload: test/velocipede_s_001790.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_001790.png
upload: test/velocipede_s_001355.png to s3://sagemaker-us-east-1-902062377305/test/velocipede_s_001355.png

And that's it! You can check the bucket and verify that the items were uploaded.

Model Training

For Image Classification, Sagemaker [also expects metadata](#) e.g. in the form of TSV files with labels and filepaths. We can generate these using our Pandas DataFrames from earlier:

```
In [22]: def to_metadata_file(df, prefix):
    df["s3_path"] = df["filenames"]
    df["labels"] = df["labels"].apply(lambda x: 0 if x==8 else 1)
    return df[["row", "labels", "s3_path"]].to_csv(
        f"{prefix}.lst", sep="\t", index=False, header=False
    )

to_metadata_file(df_train.copy(), "train")
to_metadata_file(df_test.copy(), "test")
```

We can also upload our manifest files:

```
In [23]: import boto3

# Upload files
boto3.Session().resource('s3').Bucket(
    bucket).Object('train.lst').upload_file('./train.lst')
boto3.Session().resource('s3').Bucket(
    bucket).Object('test.lst').upload_file('./test.lst')
```

Using the `bucket` and `region` info we can get the latest prebuilt container to run our training job, and define an output location on our s3 bucket for the model. Use the `image_uris` function from the SageMaker SDK to retrieve the latest `image-classification` image below:

```
In [24]: # Use the image_uris function to retrieve the latest 'image-classification' image
algo_image = sagemaker.image_uris.retrieve('image-classification', region=region, v
s3_output_location = f"s3://{bucket}/models/image_model"
```

We're ready to create an estimator! Create an estimator `img_classifier_model` that uses one instance of `m1.p3.2xlarge`. Ensure that you use the output location we defined above - we'll be referring to that later!

```
In [25]: img_classifier_model=sagemaker.estimator.Estimator(
    image_uri=algo_image,
    role=role,
    instance_count=1,
    instance_type='m1.p3.2xlarge',
    # volume_size=5,
    output_path=s3_output_location,
    sagemaker_session=sagemaker.Session()
)
```



```
/opt/conda/lib/python3.7/site-packages/boto3/compat.py:82: PythonDeprecationWarning:
Boto3 will no longer support Python 3.7 starting December 13, 2023. To continue recei
ving service updates, bug fixes, and security updates please upgrade to Python 3.8 or
later. More information can be found here: https://aws.amazon.com/blogs/developer/pyt
hon-support-policy-updates-for-aws-sdks-and-tools/
warnings.warn(warning, PythonDeprecationWarning)
```

We can also set a few key hyperparameters and define the inputs for our model:

```
In [26]: img_classifier_model.set_hyperparameters(
        image_shape= '3,32,32', # TODO: Fill in
        num_classes=2, # TODO: Fill in
        num_training_samples=len(df_train),
        epochs = 25,
        early_stopping_min_epochs=15,

        # TODO: fill in
    )
```

The `image-classification` image uses four input channels with very specific input parameters. For convenience, we've provided them below:

```
In [27]: from sagemaker.debugger import Rule, rule_configs
        from sagemaker.session import TrainingInput
        model_inputs = {
            "train": sagemaker.inputs.TrainingInput(
                s3_data=f"s3://{bucket}/train/",
                content_type="application/x-image"
            ),
            "validation": sagemaker.inputs.TrainingInput(
                s3_data=f"s3://{bucket}/test/",
                content_type="application/x-image"
            ),
            "train_lst": sagemaker.inputs.TrainingInput(
                s3_data=f"s3://{bucket}/train.lst",
                content_type="application/x-image"
            ),
            "validation_lst": sagemaker.inputs.TrainingInput(
                s3_data=f"s3://{bucket}/test.lst",
                content_type="application/x-image"
            )
        }
```

Great, now we can train the model using the `model_inputs`. In the cell below, call the `fit` method on our model,:

```
In [28]: img_classifier_model.fit(model_inputs)
```

```
INFO:sagemaker:Creating training-job with name: image-classification-2023-11-09-19-00
-19-122
```

```
2023-11-09 19:00:19 Starting - Starting the training job...
```

```
2023-11-09 19:00:45 Starting - Preparing the instances for training..
```

```

-----
KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-28-a6fc8f1c7043> in <module>
----> 1 img_classifier_model.fit(model_inputs)

/opt/conda/lib/python3.7/site-packages/sagemaker/workflow/pipeline_context.py in wrapper(*args, **kwargs)
    309         return _StepArguments(retrieve_caller_name(self_instance), run_func, *args, **kwargs)
    310
--> 311         return run_func(*args, **kwargs)
    312
    313     return wrapper

/opt/conda/lib/python3.7/site-packages/sagemaker/estimator.py in fit(self, inputs, wait, logs, job_name, experiment_config)
    1308         self.jobs.append(self.latest_training_job)
    1309         if wait:
-> 1310             self.latest_training_job.wait(logs=logs)
    1311
    1312     def _compilation_job_name(self):

/opt/conda/lib/python3.7/site-packages/sagemaker/estimator.py in wait(self, logs)
    2578         # If logs are requested, call logs_for_jobs.
    2579         if logs != "None":
-> 2580             self.sagemaker_session.logs_for_job(self.job_name, wait=True, log_type=logs)
    2581         else:
    2582             self.sagemaker_session.wait_for_job(self.job_name)

/opt/conda/lib/python3.7/site-packages/sagemaker/session.py in logs_for_job(self, job_name, wait, poll, log_type, timeout)
    4847         exceptions.UnexpectedStatusException: If waiting and the training job fails.
    4848         """
-> 4849         _logs_for_job(self.boto_session, job_name, wait, poll, log_type, timeout)
    4850
    4851     def logs_for_processing_job(self, job_name, wait=False, poll=10):

/opt/conda/lib/python3.7/site-packages/sagemaker/session.py in _logs_for_job(boto_session, job_name, wait, poll, log_type, timeout)
    6708         break
    6709
-> 6710         time.sleep(poll)
    6711
    6712         if state == LogState.JOB_COMPLETE:

KeyboardInterrupt:

```

If all goes well, you'll end up with a model topping out above `.8` validation accuracy. With only 1000 training samples in the CIFAR dataset, that's pretty good. We could definitely pursue data augmentation & gathering more samples to help us improve further, but for now let's proceed to deploy our model.

Getting ready to deploy

To begin with, let's configure Model Monitor to track our deployment. We'll define a `DataCaptureConfig` below:

```
In [30]: from sagemaker.model_monitor import DataCaptureConfig

data_capture_config = DataCaptureConfig(
    ## TODO: Set config options
    enable_capture=True,
    sampling_percentage=100,
    destination_s3_uri=f"s3://{bucket}/data_capture"
)
```

Note the `destination_s3_uri` parameter: At the end of the project, we can explore the `data_capture` directory in S3 to find crucial data about the inputs and outputs Model Monitor has observed on our model endpoint over time.

With that done, deploy your model on a single `m1.m5.xlarge` instance with the data capture config attached:

```
In [31]: deployment = img_classifier_model.deploy(
    ## TODO: fill in deployment options
    initial_instance_count=1, instance_type='m1.m5.xlarge',
    data_capture_config=data_capture_config
)

endpoint = deployment.endpoint_name
print(endpoint)
```

```

-----
KeyError                                Traceback (most recent call last)
<ipython-input-31-f0966b29919a> in <module>
      2     ## TODO: fill in deployment options
      3     initial_instance_count=1, instance_type='ml.m5.xlarge',
----> 4     data_capture_config=data_capture_config
      5 )
      6

/opt/conda/lib/python3.7/site-packages/sagemaker/estimator.py in deploy(self, initial
_instance_count, instance_type, serializer, deserializer, accelerator_type, endpoint_
name, use_compiled_model, wait, model_name, kms_key, data_capture_config, tags, serve
rless_inference_config, async_inference_config, volume_size, model_data_download_time
out, container_startup_health_check_timeout, inference_recommendation_id, explainer_c
onfig, **kwargs)
    1632         else:
    1633             kwargs["model_kms_key"] = self.output_kms_key
-> 1634             model = self.create_model(**kwargs)
    1635
    1636             model.name = model_name

/opt/conda/lib/python3.7/site-packages/sagemaker/estimator.py in create_model(self, r
ole, image_uri, predictor_cls, vpc_config_override, **kwargs)
    3161         return Model(
    3162             image_uri or self.training_image_uri(),
-> 3163             self.model_data,
    3164             role,
    3165             vpc_config=self.get_vpc_config(vpc_config_override),

/opt/conda/lib/python3.7/site-packages/sagemaker/estimator.py in model_data(self)
    1784             TrainingJobName=self.latest_training_job.name
    1785         )
-> 1786         model_uri = job_details["ModelArtifacts"]["S3ModelArtifacts"]
    1787         compression_type = job_details.get("OutputDataConfig", {}).get(
    1788             "CompressionType", "GZIP"

KeyError: 'ModelArtifacts'

```

Note the endpoint name for later as well.

Next, instantiate a Predictor:

```

In [ ]: from sagemaker.predictor import Predictor
        predictor = Predictor(endpoint_name = endpoint)## TODO: fill in

```

In the code snippet below we are going to prepare one of your saved images for prediction. Use the predictor to process the `payload`.

```

In [ ]: from sagemaker.serializers import IdentitySerializer
        import base64

        predictor.serializer = IdentitySerializer("image/png")
        with open("./test/bicycle_s_001789.png", "rb") as f:
            payload = f.read()

        inference = predictor.predict(payload)## TODO: Process the payload with your predictor

```

Your `inference` object is an array of two values, the predicted probability value for each of your classes (bicycle and motorcycle respectively.) So, for example, a value of `b'[0.91, 0.09]'` indicates the probability of being a bike is 91% and being a motorcycle is 9%.

```
In [ ]: print(inference)
```

Draft Lambdas and Step Function Workflow

Your operations team uses Step Functions to orchestrate serverless workflows. One of the nice things about Step Functions is that [workflows can call other workflows](#), so the team can easily plug your workflow into the broader production architecture for Scones Unlimited.

In this next stage you're going to write and deploy three Lambda functions, and then use the Step Functions visual editor to chain them together! Our functions are going to work with a simple data object:

```
{
  "inferences": [], # Output of predictor.predict
  "s3_key": "", # Source data S3 key
  "s3_bucket": "", # Source data S3 bucket
  "image_data": "" # base64 encoded string containing the image data
}
```

A good test object that you can use for Lambda tests and Step Function executions, throughout the next section, might look like this:

```
{
  "image_data": "",
  "s3_bucket": MY_BUCKET_NAME, # Fill in with your bucket
  "s3_key": "test/bicycle_s_000513.png"
}
```

Using these fields, your functions can read and write the necessary data to execute your workflow. Let's start with the first function. Your first Lambda function will copy an object from S3, base64 encode it, and then return it to the step function as `image_data` in an event.

Go to the Lambda dashboard and create a new Lambda function with a descriptive name like "serializeImageData" and select the 'Python 3.8' runtime. Add the same permissions as the SageMaker role you created earlier. (Reminder: you do this in the Configuration tab under "Permissions"). Once you're ready, use the starter code below to craft your Lambda handler:

```
import json
import boto3
import base64

s3 = boto3.client('s3')

def lambda_handler(event, context):
    """A function to serialize target data from S3"""

    # Get the s3 address from the Step Function event input
```

```

key = ## TODO: fill in
bucket = ## TODO: fill in

# Download the data from s3 to /tmp/image.png
## TODO: fill in

# We read the data from a file
with open("/tmp/image.png", "rb") as f:
    image_data = base64.b64encode(f.read())

# Pass the data back to the Step Function
print("Event:", event.keys())
return {
    'statusCode': 200,
    'body': {
        "image_data": image_data,
        "s3_bucket": bucket,
        "s3_key": key,
        "inferences": []
    }
}

```

The next function is responsible for the classification part - we're going to take the image output from the previous function, decode it, and then pass inferences back to the the Step Function.

Because this Lambda will have runtime dependencies (i.e. the SageMaker SDK) you'll need to package them in your function. *Key reading:*

<https://docs.aws.amazon.com/lambda/latest/dg/python-package-create.html#python-package-create-with-dependency>

Create a new Lambda function with the same rights and a descriptive name, then fill in the starter code below for your classifier Lambda.

```

import json
import sagemaker
import base64
from sagemaker.serializers import IdentitySerializer

# Fill this in with the name of your deployed model
ENDPOINT = ## TODO: fill in

def lambda_handler(event, context):

    # Decode the image data
    image = base64.b64decode(## TODO: fill in)

    # Instantiate a Predictor
    predictor = ## TODO: fill in

    # For this model the IdentitySerializer needs to be "image/png"
    predictor.serializer = IdentitySerializer("image/png")

```

```

# Make a prediction:
inferences = ## TODO: fill in

# We return the data back to the Step Function
event["inferences"] = inferences.decode('utf-8')
return {
    'statusCode': 200,
    'body': json.dumps(event)
}

```

Finally, we need to filter low-confidence inferences. Define a threshold between 1.00 and 0.000 for your model: what is reasonable for you? If the model predicts at `.70` for it's highest confidence label, do we want to pass that inference along to downstream systems? Make one last Lambda function and tee up the same permissions:

```

import json

THRESHOLD = .93

def lambda_handler(event, context):

    # Grab the inferences from the event
    inferences = ## TODO: fill in

    # Check if any values in our inferences are above THRESHOLD
    meets_threshold = ## TODO: fill in

    # If our threshold is met, pass our data back out of the
    # Step Function, else, end the Step Function with an error
    if meets_threshold:
        pass
    else:
        raise("THRESHOLD_CONFIDENCE_NOT_MET")

    return {
        'statusCode': 200,
        'body': json.dumps(event)
    }

```

Once you have tested the lambda functions, save the code for each lambda function in a python script called 'lambda.py'.

With your lambdas in place, you can use the Step Functions visual editor to construct a workflow that chains them together. In the Step Functions console you'll have the option to author a Standard step function *Visually*.

When the visual editor opens, you'll have many options to add transitions in your workflow. We're going to keep it simple and have just one: to invoke Lambda functions. Add three of them chained together. For each one, you'll be able to select the Lambda functions you just created in the proper order, filter inputs and outputs, and give them descriptive names.

Make sure that you:

1. Are properly filtering the inputs and outputs of your invocations (e.g. `$.body`)
2. Take care to remove the error handling from the last function - it's supposed to "fail loudly" for your operations colleagues!

Take a screenshot of your working step function in action and export the step function as JSON for your submission package.

Great! Now you can use the files in `./test` as test files for our workflow. Depending on our threshold, our workflow should reliably pass predictions about images from `./test` on to downstream systems, while erroring out for inferences below our confidence threshold!

Testing and Evaluation

Do several step function invocations using data from the `./test` folder. This process should give you confidence that the workflow both *succeeds* AND *fails* as expected. In addition, SageMaker Model Monitor will generate recordings of your data and inferences which we can visualize.

Here's a function that can help you generate test inputs for your invocations:

In [60]:

```
import random
import boto3
import json

def generate_test_case():
    # Setup s3 in boto3
    s3 = boto3.resource('s3')
    prefix = "test"
    # Randomly pick from sfn or test folders in our bucket
    objects = s3.Bucket(bucket).objects.filter(Prefix=prefix)

    # Grab any random object key from that folder!
    obj = random.choice([x.key for x in objects])

    return json.dumps({
        "image_data": "",
        "s3_bucket": bucket,
        "s3_key": obj
    })

generate_test_case()
```

Out[60]:

```
'{"image_data": "", "s3_bucket": "sagemaker-us-east-1-902062377305", "s3_key": "test/minibike_s_000913.png}"'
```

In the Step Function dashboard for your new function, you can create new executions and copy in the generated test cases. Do several executions so that you can generate data you can evaluate and visualize.

Once you've done several executions, let's visualize the record of our inferences. Pull in the JSONLines data from your inferences like so:

```
In [61]: from sagemaker.s3 import S3Downloader

# In S3 your data will be saved to a datetime-aware path
# Find a path related to a datetime you're interested in
data_path = 's3://sagemaker-us-east-1-902062377305/data_capture/image-classification-2

S3Downloader.download(data_path, "captured_data")

# Feel free to repeat this multiple times and pull in more data
```

```
Out[61]: ['captured_data/08/08/53-35-388-bf6bf931-4085-4487-a6ca-7b8859989eda.jsonl',
'captured_data/09/09/35-12-235-855455d2-579c-47e3-9e4b-51798af7bcfe.jsonl',
'captured_data/09/09/41-19-972-bd7ade6b-5410-4ea8-a94e-1b0df4a200a2.jsonl',
'captured_data/09/09/42-20-984-0dd5cf71-368f-4df8-8e4e-21849c80f953.jsonl',
'captured_data/09/09/48-37-869-dc0b60be-3942-4588-8ecb-228e100cbf1b.jsonl',
'captured_data/09/09/52-22-199-bdc6ac5d-1e82-4adb-a12c-594d23f95215.jsonl',
'captured_data/09/10/05-30-803-6961d14a-d835-457e-a5b9-fd323af3c68a.jsonl',
'captured_data/09/10/10-03-981-1a1722fe-a9da-463a-b735-876fa8b645f4.jsonl',
'captured_data/09/10/15-39-405-4bed8c9e-ee04-4306-9eee-474fd33cbc54.jsonl',
'captured_data/09/10/26-09-770-775d6b1b-3264-4e66-95e1-2a208b949f98.jsonl',
'captured_data/09/10/28-08-081-12be304e-fd43-42ac-a150-4b1ce82b940e.jsonl',
'captured_data/09/19/03-58-536-dec3d8b8-4d7a-4e1c-87e8-ea5d76c82921.jsonl',
'captured_data/09/19/05-45-271-97eccc92-29f2-4325-ad94-a78e852040c2.jsonl']
```

The data are in JSONLines format, where multiple valid JSON objects are stacked on top of each other in a single `jsonl` file. We'll import an open-source library, `jsonlines` that was purpose built for parsing this format.

```
In [62]: !pip install jsonlines
import jsonlines

Requirement already satisfied: jsonlines in /opt/conda/lib/python3.7/site-packages (3.1.0)
Requirement already satisfied: attrs>=19.2.0 in /opt/conda/lib/python3.7/site-packages (from jsonlines) (23.1.0)
Requirement already satisfied: typing-extensions in /opt/conda/lib/python3.7/site-packages (from jsonlines) (4.7.1)
Requirement already satisfied: importlib-metadata in /opt/conda/lib/python3.7/site-packages (from attrs>=19.2.0->jsonlines) (6.7.0)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.7/site-packages (from importlib-metadata->attrs>=19.2.0->jsonlines) (2.2.0)
DEPRECATION: pyodbc 4.0.0-unsupported has a non-standard version number. pip 23.3 will enforce this behaviour change. A possible replacement is to upgrade to a newer version of pyodbc or contact the author to suggest that they release a version with a conforming version number. Discussion can be found at https://github.com/pypa/pip/issues/12063
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: pip install --upgrade pip
```

Now we can extract the data from each of the source files:

```
In [63]: import os

# List the file names we downloaded
file_handles = os.listdir("./captured_data/08/08")

# Dump all the data into an array
json_data = []
for jsonl in file_handles:
    with jsonlines.open(f"./captured_data/08/08/{jsonl}") as f:
        json_data.append(f.read())
```

The data should now be a list of dictionaries, with significant nesting. We'll give you an example of some code that grabs data out of the objects and visualizes it:

```
In [64]: # Define how we'll get our data
def simple_getter(obj):
    inferences = obj["captureData"]["endpointOutput"]["data"]
    timestamp = obj["eventMetadata"]["inferenceTime"]
    return json.loads(inferences), timestamp

simple_getter(json_data[0])
```

```
Out[64]: ([0.9342197775840759, 0.06578023731708527], '2023-11-08T08:53:35Z')
```

Finally, here's an example of a visualization you can build with this data. In this last part, you will take some time and build your own - the captured data has the input images, the resulting inferences, and the timestamps.

```
In [65]: # Populate the data for the x and y axis
x = []
y = []
for obj in json_data:
    inference, timestamp = simple_getter(obj)

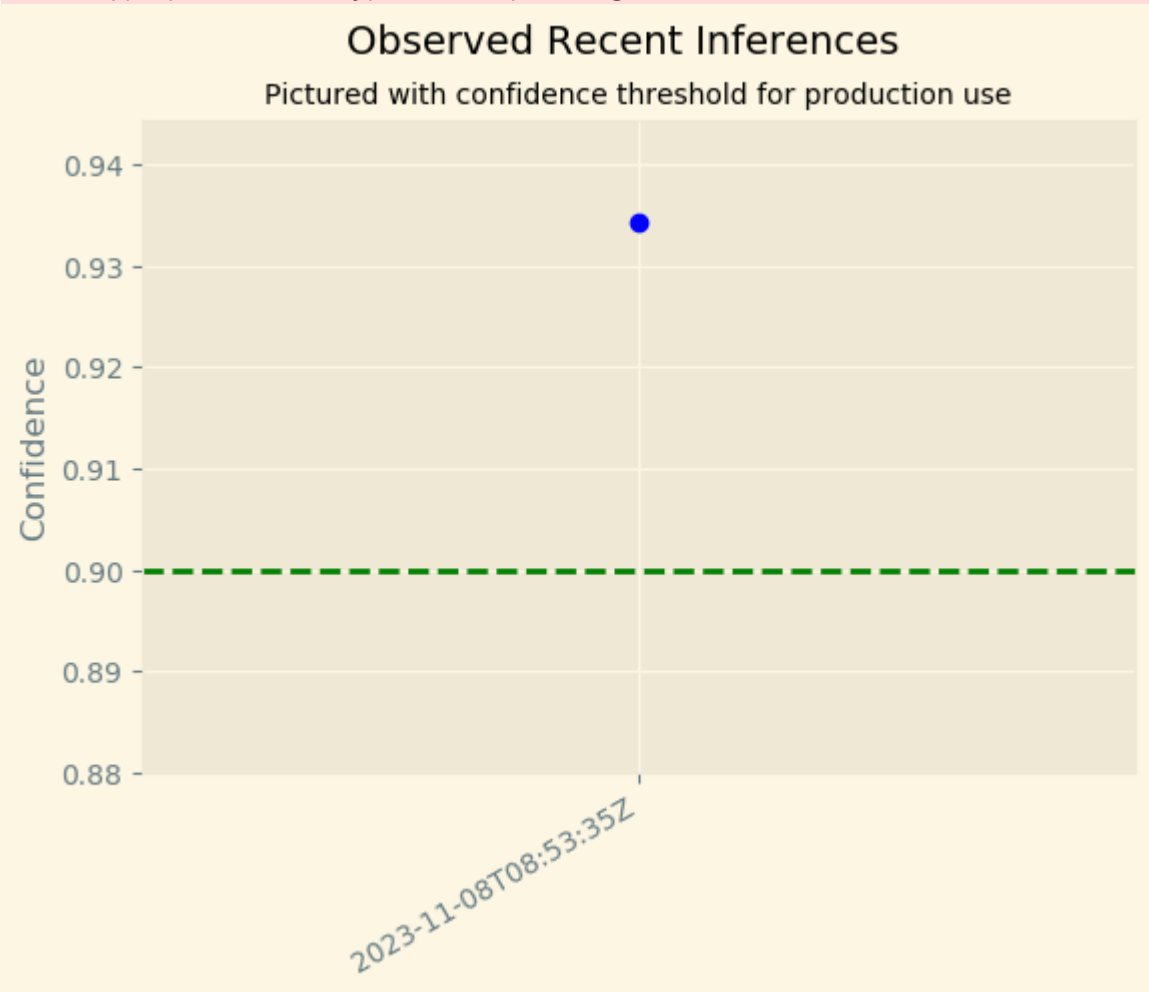
    y.append(max(inference))
    x.append(timestamp)

# Todo: here is an visualization example, take some time to build another visual that
# Plot the data
plt.scatter(x, y, c=['r' if k<.90 else 'b' for k in y ])
plt.axhline(y=0.90, color='g', linestyle='--')
plt.ylim(bottom=.88)

# Add Labels
plt.ylabel("Confidence")
plt.suptitle("Observed Recent Inferences", size=14)
plt.title("Pictured with confidence threshold for production use", size=10)

# Give it some pizzaz!
plt.style.use("Solarize_Light2")
plt.gcf().autofmt_xdate()
plt.show()
```

INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as floats or dates. If these strings should be plotted as numbers, cast to the appropriate data type before plotting.
INFO:matplotlib.category:Using categorical units to plot a list of strings that are all parsable as floats or dates. If these strings should be plotted as numbers, cast to the appropriate data type before plotting.



Todo: build your own visualization

```
In [66]: import matplotlib.pyplot as plt

def visualize_data_bar_chart(json_data):
    # Define a threshold for visualization
    THRESHOLD = 0.80
    # Initialize lists to store data for plotting
    timestamps = []
    confidences = []

    # Extract relevant data for plotting
    for obj in json_data:
        inference, timestamp = simple_getter(obj)
        timestamps.append(timestamp)
        confidences.append(max(inference))
    print(confidences)
    # Create a bar chart
    plt.figure(figsize=(12, 6))
    plt.bar(timestamps, confidences, color=['r' if k < THRESHOLD else 'b' for k in confidences])
    plt.axhline(y=THRESHOLD, color='g', linestyle='--')
```

```

plt.ylim(bottom=0.60)

# Rotate x-axis labels for better readability
plt.xticks(rotation=45, fontsize=10)

# Add Labels and titles
plt.xlabel("Timestamp")
plt.ylabel("Confidence")
plt.title("Confidence Scores Over Time", size=14)

# Show the plot
plt.tight_layout()
plt.show()

```

```

In [67]: import boto3
import time

import random

def generate_test_case():
    # Setup s3 in boto3
    s3 = boto3.resource('s3')
    prefix = "test"
    # Randomly pick from sfn or test folders in our bucket
    objects = s3.Bucket(bucket).objects.filter(Prefix=prefix)
    # print([x.key for x in objects])
    # Grab any random object key from that folder!
    obj = random.choice([x.key for x in objects])
    print(obj)
    return {
        "image_data": "",
        "s3_bucket": bucket,
        "s3_key": obj
    }

```

```

In [68]: def list_execution_timestamps(step_function_arn, num_executions):
    client = boto3.client("stepfunctions")
    executions = client.list_executions(
        stateMachineArn=step_function_arn,
        maxResults=num_executions
    )['executions']
    # print(executions)
    timestamps = [e['stopDate'] for e in executions]
    return timestamps

```

```

In [69]: import shutil

# Function to clear the local capture data folder
# You can call this function to clear the local capture data folder before reinstalling
def clear_local_capture_data():
    capture_data_folder = "./captured_data"

    try:
        shutil.rmtree(capture_data_folder)
        print("Local capture data cleared.")
    except FileNotFoundError:
        print("Local capture data folder does not exist, nothing to clear.")
    except Exception as e:
        print(f"An error occurred while clearing local capture data: {str(e)}")

```

```

# Function to create the "captured_data" directory if it doesn't exist
def create_captured_data_directory():
    if not os.path.exists("captured_data"):
        os.makedirs("captured_data")

```

```

In [70]: from sagemaker.s3 import S3Downloader

def download_data_for_timestamp(data_path, destination_directory):
    # Download the data to the destination directory
    # data_path = "s3://sagemaker-us-east-1-974801369137/data_capture/image-classifica
    S3Downloader.download(data_path, destination_directory)

```

```

In [71]: # Function to run Step Function multiple times with different test cases
def run_step_function_multiple_times(bucket, num_runs, step_function_arn):
    # Set up AWS Step Functions client
    sfn = boto3.client('stepfunctions')

    # Generate and run test cases
    for run in range(1, num_runs + 1):
        print(f"Running Step Function - Test Case {run}...")
        test_case = generate_test_case() # Generate a test case
        execution_name = f"test_case_{run}_{int(time.time())}" # Unique execution name

        # Start execution of the Step Function
        response = sfn.start_execution(
            stateMachineArn=step_function_arn,
            name=execution_name,
            input=json.dumps(test_case)
        )

        print(f"Step Function execution started with ID: {response['executionArn']}")
        # print(response)

    print(f"Step Function invoked {num_runs} times with different test cases.")

```

```

In [72]: # This function orchestrates the entire workflow for capturing, processing, and visual
# It performs the following steps:
# 1. Clears and reinstalls the capture data from the specified S3 bucket.
# 2. Dynamically generates datetime paths based on Step Function execution timestamps.
# 3. Runs the Step Function multiple times with different test cases.
# 4. Downloads and processes captured data from S3.
# 5. Visualizes the inference data.
#
# Parameters:
# - bucket: The name of the S3 bucket where data will be stored and captured.
# - num_runs: The number of times to run the Step Function with different test cases.
# - step_function_arn: The ARN (Amazon Resource Name) of the Step Function to invoke.
# - num_executions: The number of Step Function executions to consider for data retrieval
#
# Note: Before running this function, ensure that AWS credentials are properly configured
def run_inference_workflow(bucket, num_runs, step_function_arn, num_executions):
    # ... (function implementation)
    clear_local_capture_data()

    # Generate timestamps from Step Function executions
    timestamps = list_execution_timestamps(step_function_arn, num_executions)

```

```

print(timestamps)

# Generate datetime_paths from timestamps
bucket_prefix="s3://sagemaker-us-east-1-902062377305/data_capture/image-classifica
datetime_paths = []
for timestamp in timestamps:
    formatted_date = timestamp.strftime("%Y/%m/%d/%H")
    datetime_paths.append(f"{bucket_prefix}{formatted_date}")

run_step_function_multiple_times(bucket, num_runs, step_function_arn)

create_captured_data_directory()

# Download data from the current time path
current_time_path = datetime_paths # Use the last timestamp
print(f"Downloading result from : {current_time_path}")

# as step function take some time to execute successfully
# Delay for 30 sec (adjust the duration as needed)
delay_seconds = 30 # 30 sec

# Sleep for the specified duration
time.sleep(delay_seconds)

# Now, call the download_data_for_timestamp function
for i in current_time_path:
    download_data_for_timestamp(i, "captured_data")

# List the file names we downloaded
file_handles = os.listdir("./captured_data")

# Dump all the data into an array
json_data = []
for jsonl in file_handles:
    with jsonlines.open(f"./captured_data/{jsonl}") as f:
        json_data.append(f.read())

# Visualize the data
# visualize_data(json_data)
visualize_data_bar_chart(json_data)

```

```

In [73]: step_function_arn = "arn:aws:states:us-east-1:902062377305:stateMachine:MyStateMachine
# num_executions = 4 # Adjust the number of executions as needed
num_runs=5
bucket="sagemaker-us-east-1-902062377305"
run_inference_workflow(bucket, num_runs, step_function_arn, num_executions=num_runs)

```

Local capture data cleared.

```

[datetime.datetime(2023, 11, 9, 19, 5, 46, 926000, tzinfo=tzlocal()), datetime.datetime(2023, 11, 9, 19, 5, 46, 522000, tzinfo=tzlocal()), datetime.datetime(2023, 11, 9, 19, 5, 46, 448000, tzinfo=tzlocal()), datetime.datetime(2023, 11, 9, 19, 5, 46, 199000, tzinfo=tzlocal()), datetime.datetime(2023, 11, 9, 19, 5, 45, 913000, tzinfo=tzlocal())]

```

Running Step Function - Test Case 1...

```
/opt/conda/lib/python3.7/site-packages/boto3/compat.py:82: PythonDeprecationWarning:
Boto3 will no longer support Python 3.7 starting December 13, 2023. To continue recei
ving service updates, bug fixes, and security updates please upgrade to Python 3.8 or
later. More information can be found here: https://aws.amazon.com/blogs/developer/pyt
hon-support-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
```

```
test/bicycle_s_001047.png
```

```
Step Function execution started with ID: arn:aws:states:us-east-1:902062377305:execut
ion:MyStateMachine-un5auv9gn:test_case_1_1699556829
```

```
Running Step Function - Test Case 2...
```

```
test/cycle_s_001214.png
```

```
Step Function execution started with ID: arn:aws:states:us-east-1:902062377305:execut
ion:MyStateMachine-un5auv9gn:test_case_2_1699556829
```

```
Running Step Function - Test Case 3...
```

```
test/cycle_s_001044.png
```

```
Step Function execution started with ID: arn:aws:states:us-east-1:902062377305:execut
ion:MyStateMachine-un5auv9gn:test_case_3_1699556829
```

```
Running Step Function - Test Case 4...
```

```
test/ordinary_bicycle_s_000113.png
```

```
Step Function execution started with ID: arn:aws:states:us-east-1:902062377305:execut
ion:MyStateMachine-un5auv9gn:test_case_4_1699556829
```

```
Running Step Function - Test Case 5...
```

```
test/minibike_s_001867.png
```

```
Step Function execution started with ID: arn:aws:states:us-east-1:902062377305:execut
ion:MyStateMachine-un5auv9gn:test_case_5_1699556829
```

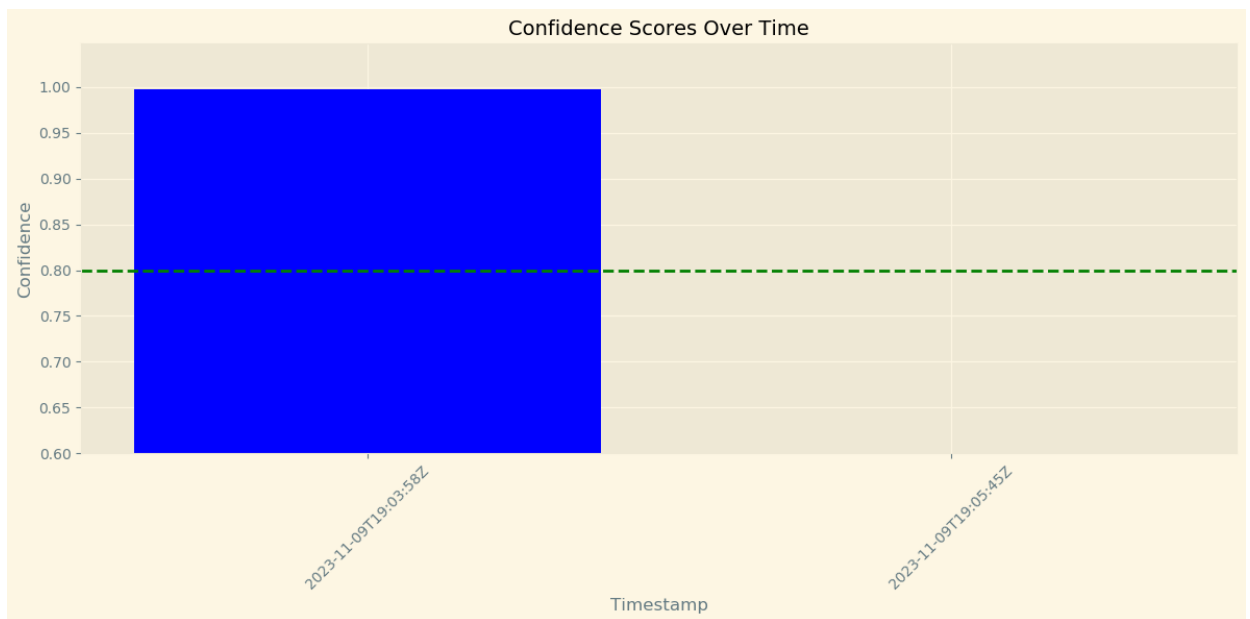
```
Step Function invoked 5 times with different test cases.
```

```
Downloading result from : ['s3://sagemaker-us-east-1-902062377305/data_capture/image-
classification-2023-11-08-08-47-02-224/AllTraffic/2023/11/09/19', 's3://sagemaker-us-
east-1-902062377305/data_capture/image-classification-2023-11-08-08-47-02-224/AllTraf
fic/2023/11/09/19', 's3://sagemaker-us-east-1-902062377305/data_capture/image-classif
ication-2023-11-08-08-47-02-224/AllTraffic/2023/11/09/19', 's3://sagemaker-us-east-1-
902062377305/data_capture/image-classification-2023-11-08-08-47-02-224/AllTraffic/202
3/11/09/19', 's3://sagemaker-us-east-1-902062377305/data_capture/image-classification
-2023-11-08-08-47-02-224/AllTraffic/2023/11/09/19']
```

```
INFO:matplotlib.category:Using categorical units to plot a list of strings that are a
ll parsable as floats or dates. If these strings should be plotted as numbers, cast t
o the appropriate data type before plotting.
```

```
INFO:matplotlib.category:Using categorical units to plot a list of strings that are a
ll parsable as floats or dates. If these strings should be plotted as numbers, cast t
o the appropriate data type before plotting.
```

```
[0.997179388999939, 0.5962597727775574]
```



In [74]: `import os`

```
# List the file names we downloaded
file_handles = os.listdir("./captured_data")

# Dump all the data into an array
json_data = []
for jsonl in file_handles:
    with jsonlines.open(f"./captured_data/{jsonl}") as f:
        json_data.append(f.read())
```

In [76]: `# My Simple Getter:`

```
def my_simple_getter(obj):
    inferences_str = obj["captureData"]["endpointOutput"]["data"]
    inferences = [float(value) for value in inferences_str.strip('[]').split(', ')] #
    timestamp = obj["eventMetadata"]["inferenceTime"]
    images = obj["captureData"]["endpointInput"]["data"]

    # Determine prediction based on inference values
    prediction = "bicycle" if inferences[0] > inferences[1] else "motorcycle"

    return inferences, timestamp, images, prediction

# Example: first object retrieved
my_simple_getter(json_data[1])
```



```
Out[76]: ([0.403740257024765, 0.5962597727775574],
          '2023-11-09T19:05:45Z',
          'iVBORw0KGgoAAAANSUHEUgAAACAAAAAgCAYAAABzenr0AAAABHNCsvQICAgIfAhkiAAAAAlwSFlzAAAPYQAAD2EBQD+naQAACKVJREFUWlU9l0mPHtd1hp9z762qb+rh67k5NieJJZJuxHNgyhUBibEcX4kUgWIA3+RPxL8HPyDqLIGsvAmiVIHFsxbQm0wIlUqJEUhrHdbPZc39jVd3hZPEX2tWm6p73vafeQf71n/9JFUhAIpKI5A1L0cyovUckB6N47WMBgk8eNZGscFijmByQiEkG6yyIYIXgjEFFEAXgESbPIgaAlBjiHE5aBiSiqUzJdSZipn004dGqJGQI4ipyKYANfY0IjiXQUqgkdxk0ApUhaQRayFppPIRRbDWogo+1lhnUVU0KULEhWKAzCGZSOYEK+W4pkVFyTUDhKgJm7UoXAdRooJVAjicZmhlTWxtommAESURAieorDEGPEHAAPGgEhKERWl9gE3vQJqLCSKGCWmiJeAMx1ERTWBgLUZYiCGCiHhrEVQiCwL1khW4auSFAP05QgGazJQ00RJKAamkkgoig+e4XiII0tEjYQUCFUNyZi5g7g2hStAKmKqsQoaa2IYE01ApYFFUBNQEcpxSeXHkwFCjnU51Am1JZpamCDUboQa0Di5grzpcCFGFBA15K5A1NDIF2i5i4x7PUB1A1pTEa+eihqrCYzDWCAZ6qgYSViTkbs06iIiCbGJFMDtFzQGLUpq00YQ61EUfQUBV6YRhctQnwgYjG1QJjU/z9NmY/t4jFo73ScUinKNSMOMMB/sV3ZmM6ZmZMHJCSqgKMUBqHAgYNfRfBC5snuE1t8I9s80jxT0SQ8AjZgLCbwztMDczS2YsogV+PMX9zx9T+W3Wjs2zsvwq3vY46inSA39UcOfLR0wvZ7xyfon2dKJfP6YxVaKpgpSTQgtfOtIjw7nGPN1Qs/l8mxfNIQvHM4ImcIoRwV2c/zEA1hlGI8+XN3c42B3xxlvn6M6dZ6q9wu0v7nD371NaVWLaQitboddTPv34gMQhpvOcyz+YhzSNrxr0DpDt5yPe1EU6ec1vvvqA/z76mguXL6LZHATFWcGKwb1x9m0q70mifH7/Lkf697myfo7VuTM83Djg4eMDXmzt8vjhA4ZHfarDfaKD6Cw2RgaDIA987xLtYoGy9FRJt39QkpPRmDb85433+W351KU3TzCz2GFYVhgSqoBLOLG0wLisc4QqcPz4DJ10m7t3nvDpF18ztbDEsw0rZHSaPlz3JcN+D2ehKHKcWMBjkv7RmBcbY6L3CIk8yxgc7vHvd29x4tQif/fLX9FZ7TAaH+DrEdaBaoBacD56fPA0pEHYFaPhgE9u3mZnt2J/75Dp4ZB2s+AnP32b7twCtz+7yeHOFrEuqWJNnudkxqIXEFKqKocMjvY53Nvme6+e48ovfs76979PZsCPDjBEEAjBk6LBxRRRnQjQ2toZ/u03v+OTW/dYWD5DZowNwx8IvqLbneWvi5fotFuU/UPq8ZBe/5BGA4qd/T7PHj8ghjGpHnHm5Cp/e/WnnF+/zNz8IsP+gMwI3U6HRp4RYiQlXRbHFV7E2MQQ97s8LO/+QVB0swvHmdn8znBWxs8vPcl+wf7LK+uIgrtVoPX/uIKi/OzPPjmIS7fZXUIMtM0vLZ+meXFBZWIC0vLdLtdxAlBEtYY6pSIRMSCikX+9P5/qQK1D7zYG9Lv19y9d5ebNz91Y30batJdMEREHEF00tPMLa3Q6rQ5e2KV9UvruNwxM9vg9PFVPv0Mz768AYhRE6vneLNqz/i7NpxsBFn28QQ8aHC2IR1ARfQEsKk7j/d5vcf3mb3xRELm46NjcfEsubUbJetso9J0Myam0jZfbFNa9BitL/NqRMnWV9fZ3tng/95700uX38fMHSmpnjweJPD/nV+cu2HrC41MZLhXIuUEhDIcnD05Nx98IT3rt9ka+eITsdwcFAyGo45027zD3/9czYzqNstzpxaIlZD/uXffo01y/++tesnbmAtZZWp8X7H/yZvGjhnEVEyL0co96I0189YG52HwsqQpx4QUg1Rcxg1Hgxo3bbD7ZZDw6YHmuiU9d8sYMRzHQ6ih//oxBu0BzbTB/17FVD6mNbXC90z8xIhqj80Lzp4/x7NvvyUEi/ee0G6R502eb+4xHibm56aIIESEGRcEWt15+pynT7dIvqbTNKivef1HP8S6aT75+EP+8N519m5FKj0geNgYBvyB5a/e/jFootc/IIbAw4ePOH3qNNeuXaM/GLC3u8P+3i7j8ZiBE/r9IYtzyAGJSEKIoLZ09hnOCppFJbjJ1ZxZokTJ04iJtBuZmwNh/zx5iY3bgb+/AXceuCJ2RxPvn3O//7+d4wHh3Rnmirf8+6777K1tcXy0hJXr151ZWVWGCmHVKRu4+ta0a7xdU2Kkaoc41QDJA900PfqRXpHJXfu3KEa95mabuIbGfd3noMzpFQsXGDGev700R8oxwMeP7rLW9fe4vzZC3yx3OWjj//I06cTng73D4gpUhQ5M+00pI1TTnJihssMbnIhcHA040E327TyB18//YZmq0U776BqWT15mcOjPs1WA3GWfq9HqilTzS5f3XrIi2eHvPPO08y3F0ilUg4qdjd2GfQGaKo5fmyRVqOFqiIGkiqZK7BZh10b0+12GdeJF59v0ZEGUimpEVhgreJshm1njOKIpisY1yXMnKTIHEWlgwVfPpsgyeIwom4Su+bI/ayMdo0nLy0wJXLa4QYiaoURQ5AXVeYlHBCYna6Tf/gC0sheU9mBCdCZ6qDEUsoI93ZLlW/nGQ966iqC1/VoBnT09PudY0Vx8VXL7G9/xzRwPLaElfe0Et3pgMi5H1BSpEQAiEEnCputuLYw+1iY8Xh0ZDKszqdKU6fOU3RLqiqmrp0100a/c2Sme4sg3GPOK4gz6hrJeUDpNvCqCECreUZWoXjzK1VpqbbpCBgI+NyBKoT74kRFcGdPbnK6mKX1//yB4zKPgTPzNQ0jXaLFANV7a195Kg/5JarmVuYY2rq0M1XBih0DPs9VlZniWz0VUIRi3kztKZdZjM4GyBWCH6CXprJ0IVfJjkgc7UDNZZZmc7k0yH4ENaseSZJXOKtY5za6fYfL7Fwc6Lye8lQmYtF8+uceHsaaIqURURAcA5g7G0uqqIPpHnDfJ8sg0j0WiSCcVYjLWEiUZ0ZNQIYiwAqpA5R9YouHipyf1XTuHrmhgTMSp5ltNptbDGQAxISlhrSSmRXmqAqiIISWMSvhryf0cqBEnIoQQUFwctWRZPu11GQTvXxoHGDFYV9DIDdISYgwYEoKhqjw+TZCLTA76/0VTBdXJUDFEvAaMMVhrYVJw7HeUiYCPCaNMphbBWetSnZTXMKmxIkKMSCFgRKhjIqaIneCc/e5wwZB1DoAYIzG1776bUkIFHDppqi1NKhgvh7HWYoxMWLFmsjQxUpY1xlhwEMIElSbQpDhr8N4TY8QYS4qKKjjnsNbhYyC19NItDcbmuKIo8MNA+g41L9lQUcWJic8czjqcm1BZ1zUhBEAmdukyfvGvARiEWNxYrDWEHxADFjjcG7SN8WAIvwf31Sz5eVQO+cAAAAASUVORK5CYII=',
          'motorcycle')
```

```
In [79]: # Visualizing images and their inferences in comparison to the assumed threshold
%matplotlib inline
import io
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PIL import Image
import base64
THRESHOLD = 0.7
x = []
y = []

for idx, obj in enumerate(json_data):
    inference, timestamp, image, prediction = my_simple_getter(obj)
```

```

img = base64.b64decode(image)    # decode string using base64
img = io.BytesIO(img)
img = Image.open(img)            # Open image

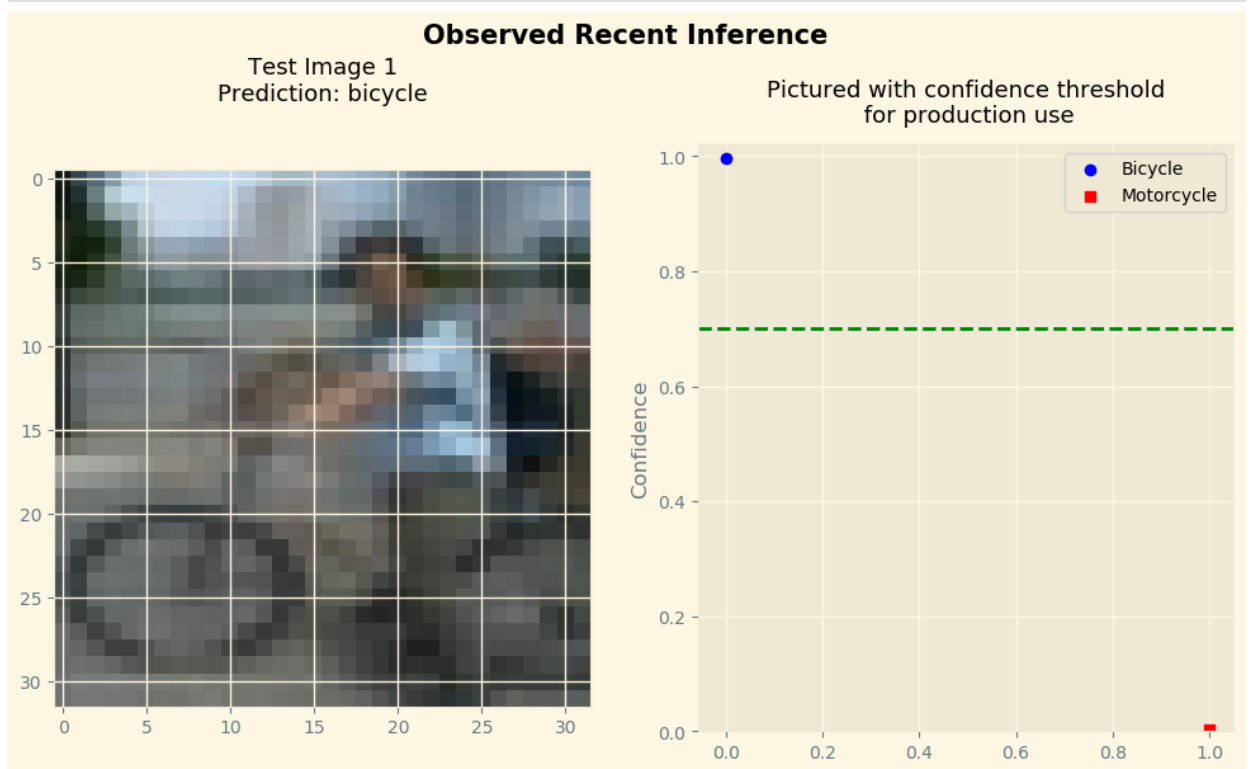
# Plotting subplots to display Images along with their performances
f = plt.figure(figsize=(12, 6))
f.suptitle("Observed Recent Inference", size=15, y=1.04, fontweight='bold')

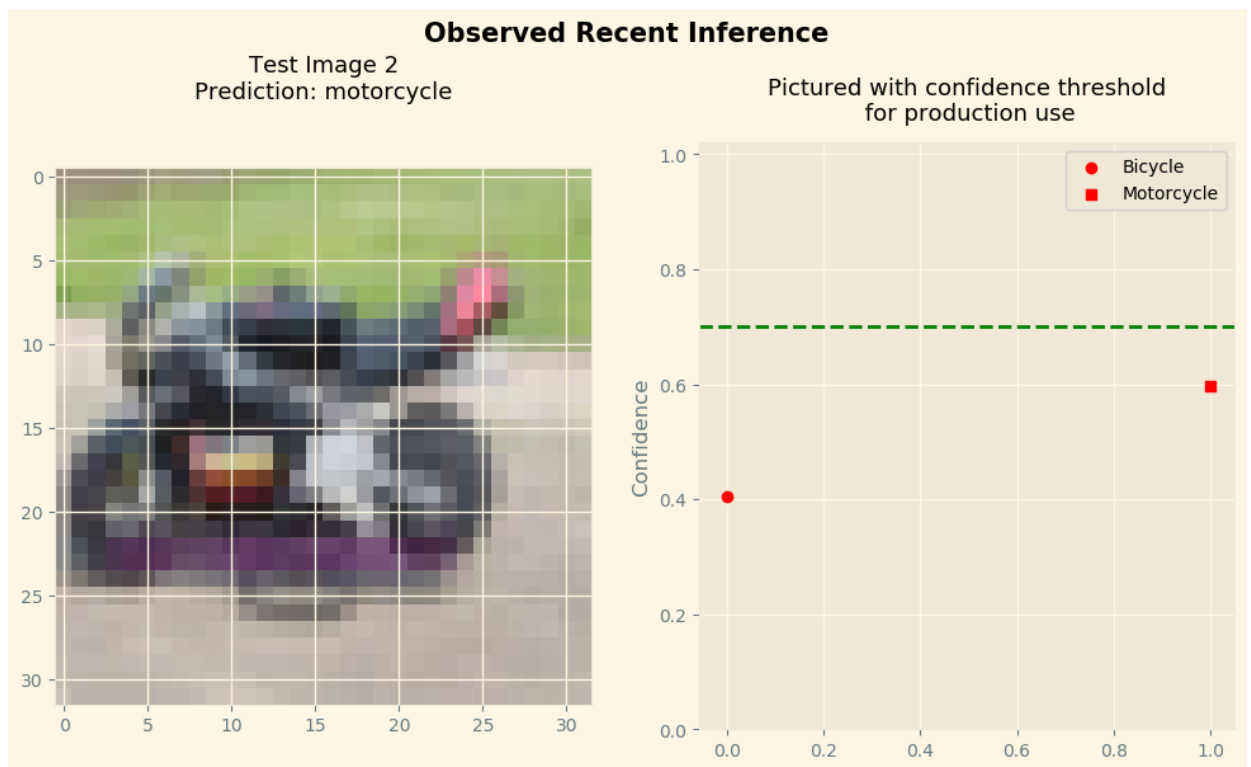
# Plot Test Image
ax1 = f.add_subplot(121)
ax1.set_title(f"Test Image {idx+1}\nPrediction: {prediction}", size=13, y=1.11) #
ax1.imshow(img, interpolation='nearest')

# Plot Inference and Time Data
ax2 = f.add_subplot(122)
for i, conf in enumerate(inference):
    shape = 'o' if i == 0 else 's' # Circle for the first inference (bicycle), Square for motorcycle
    color = 'r' if conf < THRESHOLD else 'b'
    label = 'Bicycle' if i == 0 else 'Motorcycle'
    ax2.scatter(i, conf, c=color, marker=shape, label=label)
ax2.axhline(y=THRESHOLD, color='g', linestyle='--')
ax2.set_ylim(bottom=0, top=1.02)
ax2.legend() # Show legend to distinguish between bicycle and motorcycle
# Add labels
ax2.set_ylabel("Confidence")
ax2.set_title("Pictured with confidence threshold\n for production use", size=13,

plt.show()

```





In []:

In []:

Congratulations!

You've reached the end of the project. In this project you created an event-driven ML workflow that can be incorporated into the Scones Unlimited production architecture. You used the SageMaker Estimator API to deploy your SageMaker Model and Endpoint, and you used AWS Lambda and Step Functions to orchestrate your ML workflow. Using SageMaker Model Monitor, you instrumented and observed your Endpoint, and at the end of the project you built a visualization to help stakeholders understand the performance of the Endpoint over time. If you're up for it, you can even go further with these stretch goals:

- Extend your workflow to incorporate more classes: the CIFAR dataset includes other vehicles that Scones Unlimited can identify with this model.
- Modify your event driven workflow: can you rewrite your Lambda functions so that the workflow can process multiple image inputs in parallel? Can the Step Function "fan out" to accomodate this new workflow?
- Consider the test data generator we provided for you. Can we use it to create a "dummy data" generator, to simulate a continuous stream of input data? Or a big paralell load of data?
- What if we want to get notified every time our step function errors out? Can we use the Step Functions visual editor in conjunction with a service like SNS to accomplish this? Try it out!