

# fish model

May 9, 2018

```
In [1]: import os, cv2, random
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import log_loss
from sklearn.preprocessing import LabelEncoder

import matplotlib.pyplot as plt
from matplotlib import ticker
import seaborn as sns
%matplotlib inline

from keras.models import Sequential
from keras.layers import Dropout, Flatten, Convolution2D, MaxPooling2D, ZeroPadding2D, Dense
from keras.optimizers import RMSprop, Adam
from keras.callbacks import EarlyStopping
from keras.utils import np_utils
from keras import backend as K

TRAIN_DIR = 'input/train/'
TEST_DIR = 'input/test_stg1/'
FISH_CLASSES = ['ALB', 'BET', 'DOL', 'LAG', 'NoF', 'OTHER', 'SHARK', 'YFT']
ROWS = 90 #720
COLS = 160 #1280
CHANNELS = 3

/home/nikhil/anaconda3/lib/python3.6/site-packages/h5py/__init__.py:36: FutureWarning: Conversion
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.

In [2]: def get_images(fish):
        """Load files from train folder"""
        fish_dir = TRAIN_DIR+'{}'.format(fish)
        images = [fish+'/'+im for im in os.listdir(fish_dir)]
        return images

def read_image(src):
```

```

        """Read and resize individual images"""
        im = cv2.imread(src, cv2.IMREAD_COLOR)
        im = cv2.resize(im, (COLS, ROWS), interpolation=cv2.INTER_CUBIC)
        return im

files = []
y_all = []

for fish in FISH_CLASSES:
    fish_files = get_images(fish)
    files.extend(fish_files)

    y_fish = np.tile(fish, len(fish_files))
    y_all.extend(y_fish)
    print("{0} photos of {1}".format(len(fish_files), fish))

y_all = np.array(y_all)

1719 photos of ALB
200 photos of BET
117 photos of DOL
67 photos of LAG
465 photos of NoF
299 photos of OTHER
176 photos of SHARK
734 photos of YFT

In [3]: X_all = np.ndarray((len(files), ROWS, COLS, CHANNELS), dtype=np.uint8)

        for i, im in enumerate(files):
            X_all[i] = read_image(TRAIN_DIR+im)
            if i%1000 == 0: print('Processed {} of {}'.format(i, len(files)))

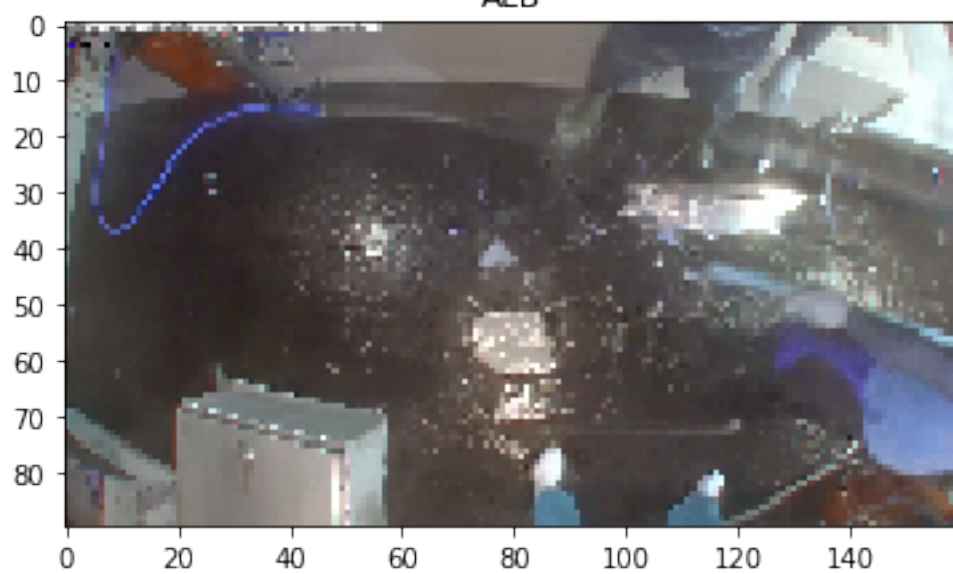
        print(X_all.shape)

Processed 0 of 3777
Processed 1000 of 3777
Processed 2000 of 3777
Processed 3000 of 3777
(3777, 90, 160, 3)

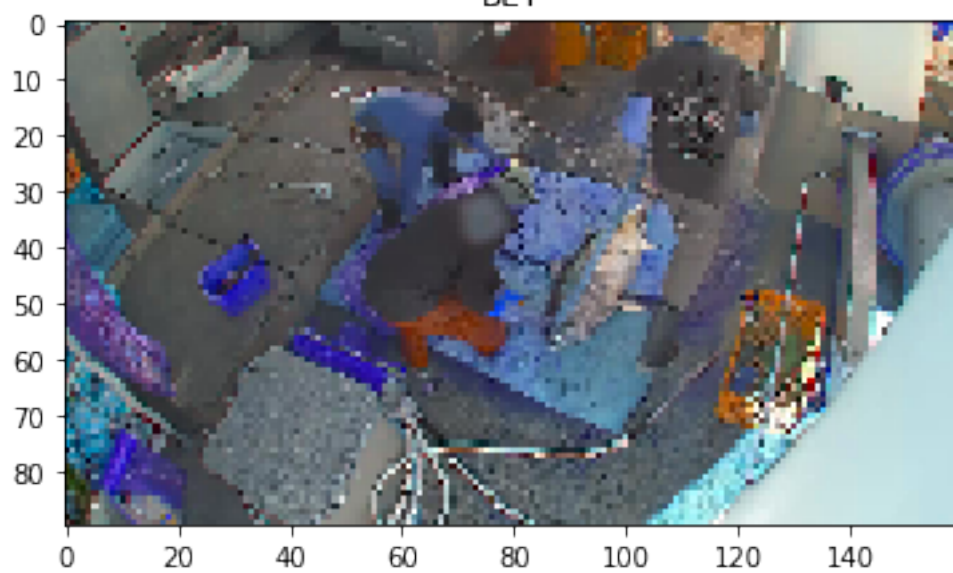
In [4]: uniq = np.unique(y_all, return_index=True)
        for f, i in zip(uniq[0], uniq[1]):
            plt.imshow(X_all[i])
            plt.title(f)
            plt.show()

```

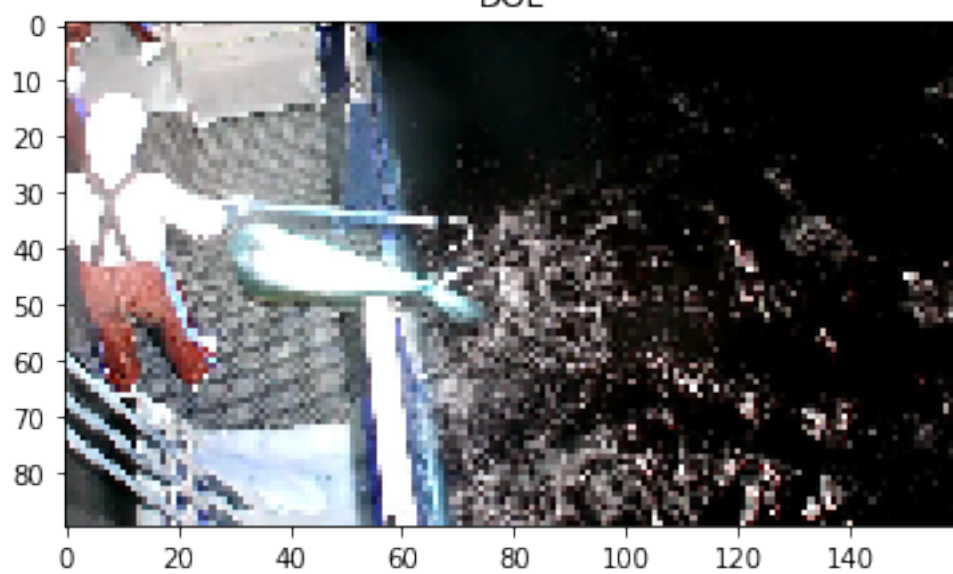
ALB



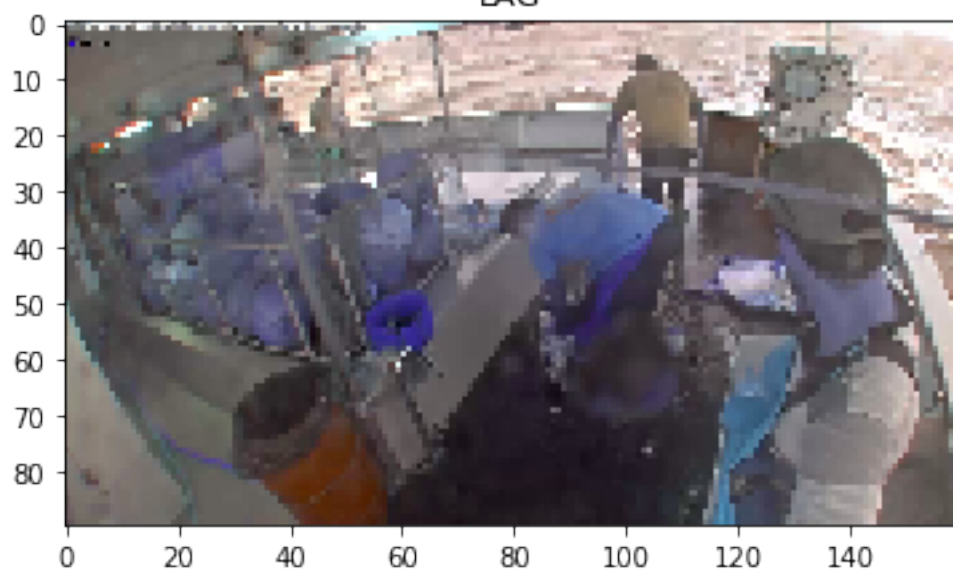
BET

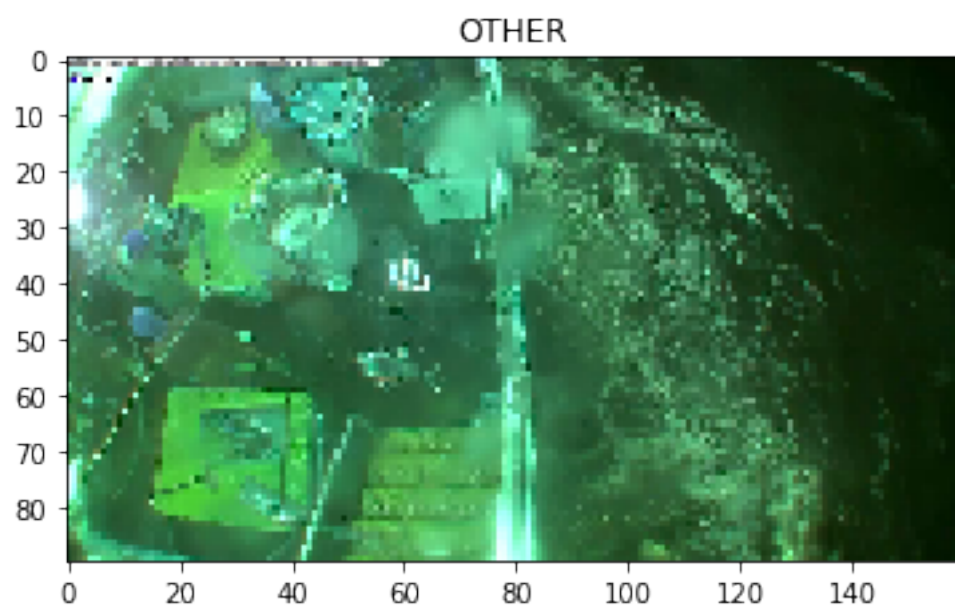
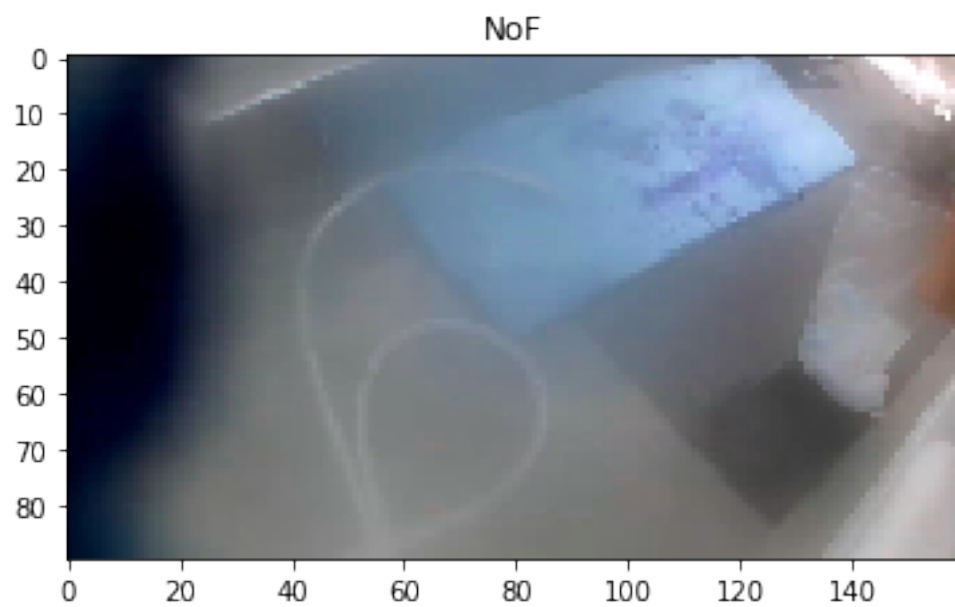


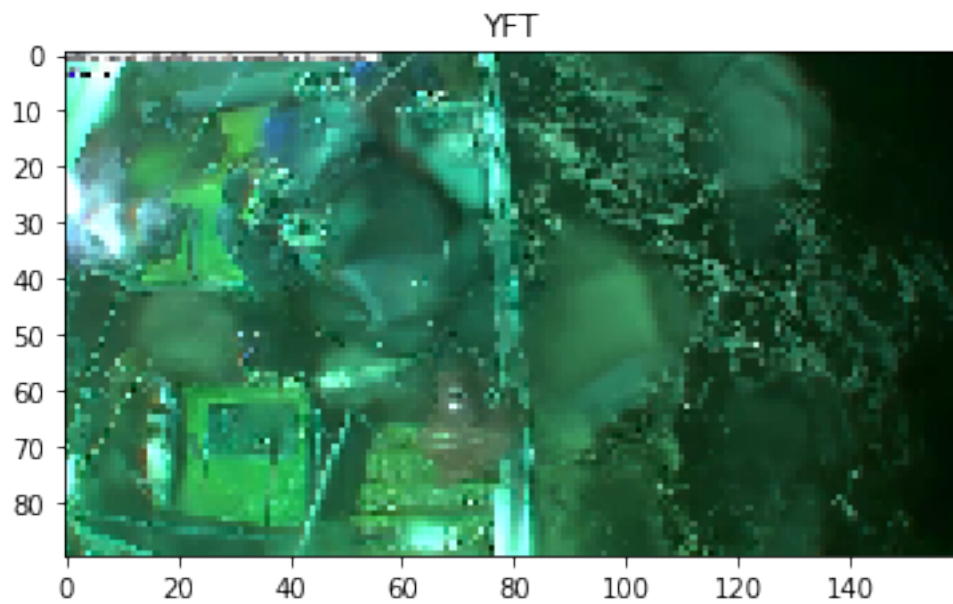
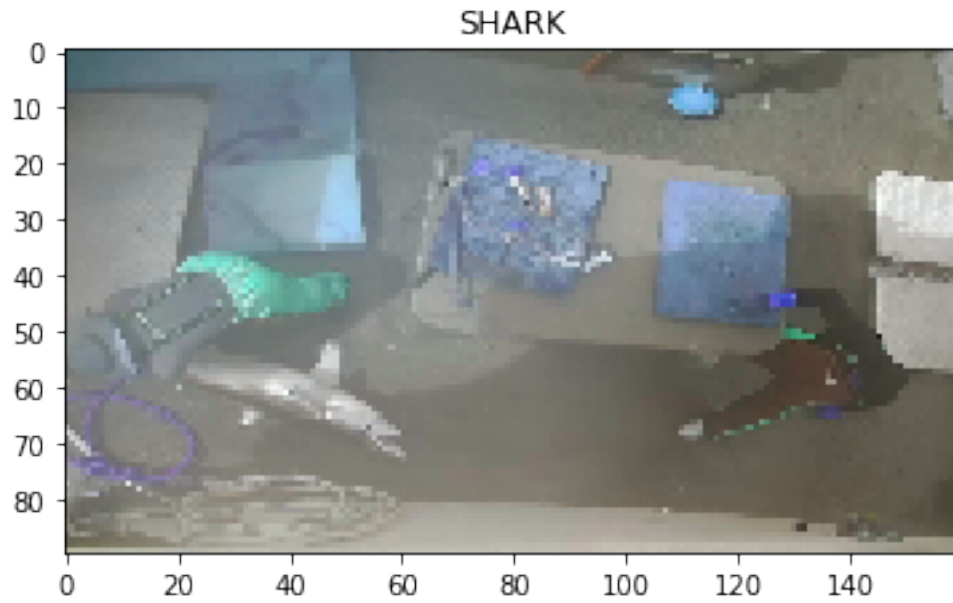
DOL



LAG







```
In [5]: y_all = LabelEncoder().fit_transform(y_all)
        y_all = np_utils.to_categorical(y_all)

        X_train, X_valid, y_train, y_valid = train_test_split(X_all, y_all,
                                                                test_size=0.2, random_state=23,
                                                                stratify=y_all)
```

```

In [6]: optimizer = RMSprop(lr=1e-4)
        objective = 'categorical_crossentropy'

        def center_normalize(x):
            return (x - K.mean(x)) / K.std(x)

        model = Sequential()

        model.add(Activation(activation=center_normalize, input_shape=(ROWS, COLS, CHANNELS)))

        model.add(Convolution2D(32, 5, 5, border_mode='same', activation='relu', dim_ordering='t')
        model.add(Convolution2D(32, 5, 5, border_mode='same', activation='relu', dim_ordering='t')
        model.add(MaxPooling2D(pool_size=(2, 2), dim_ordering='tf'))

        model.add(Convolution2D(64, 3, 3, border_mode='same', activation='relu', dim_ordering='t')
        model.add(Convolution2D(64, 3, 3, border_mode='same', activation='relu', dim_ordering='t')
        model.add(MaxPooling2D(pool_size=(2, 2), dim_ordering='tf'))

        model.add(Convolution2D(128, 3, 3, border_mode='same', activation='relu', dim_ordering='t')
        model.add(Convolution2D(128, 3, 3, border_mode='same', activation='relu', dim_ordering='t')
        model.add(MaxPooling2D(pool_size=(2, 2), dim_ordering='tf'))

        model.add(Convolution2D(256, 3, 3, border_mode='same', activation='relu', dim_ordering='t')
        model.add(Convolution2D(256, 3, 3, border_mode='same', activation='relu', dim_ordering='t')
        model.add(MaxPooling2D(pool_size=(2, 2), dim_ordering='tf'))

        model.add(Flatten())
        model.add(Dense(256, activation='relu'))
        model.add(Dropout(0.5))

        model.add(Dense(64, activation='relu'))
        model.add(Dropout(0.5))

        model.add(Dense(len(FISH_CLASSES)))
        model.add(Activation('sigmoid'))

        model.compile(loss=objective, optimizer=optimizer)

/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:11: UserWarning: Update
# This is added back by InteractiveShellApp.init_path()
/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:12: UserWarning: Update
if sys.path[0] == '':
/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:13: UserWarning: Update
del sys.path[0]
/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:15: UserWarning: Update
from ipykernel import kernelapp as app
/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:16: UserWarning: Update

```

```

app.launch_new_instance()
/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:17: UserWarning: Update
/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:19: UserWarning: Update
/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:20: UserWarning: Update
/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:21: UserWarning: Update
/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:23: UserWarning: Update
/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:24: UserWarning: Update
/home/nikhil/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:25: UserWarning: Update

```

```

In [7]: early_stopping = EarlyStopping(monitor='val_loss', patience=4, verbose=1, mode='auto')

        model.fit(X_train, y_train, batch_size=64, nb_epoch=1,
                  validation_split=0.2, verbose=1, shuffle=True, callbacks=[early_stopping])

/home/nikhil/anaconda3/lib/python3.6/site-packages/keras/models.py:942: UserWarning: The `nb_epoch`
warnings.warn('The `nb_epoch` argument in `fit` ')

```

Train on 2416 samples, validate on 605 samples

Epoch 1/1

2416/2416 [=====] - 377s 156ms/step - loss: 1.8468 - val\_loss: 1.7208

Out[7]: <keras.callbacks.History at 0x7f441218d400>

```

In [8]: preds = model.predict(X_valid, verbose=1)
        print("Validation Log Loss: {}".format(log_loss(y_valid, preds)))

```

756/756 [=====] - 36s 47ms/step

Validation Log Loss: 1.711710183078019

```

In [9]: test_files = [im for im in os.listdir(TEST_DIR)]
        test = np.ndarray((len(test_files), ROWS, COLS, CHANNELS), dtype=np.uint8)

```

```

        for i, im in enumerate(test_files):
            test[i] = read_image(TEST_DIR+im)

```

```

        test_preds = model.predict(test, verbose=1)

```

1000/1000 [=====] - 47s 47ms/step

```

In [11]: submission = pd.DataFrame(test_preds, columns=FISH_CLASSES)
        submission.insert(0, 'image', test_files)
        submission.head()

```



```

Out[11]:
      image      ALB      BET      DOL      LAG      NoF      OTHER \
0  img_05139.jpg  0.916198  0.155555  0.423464  0.091762  0.612399  0.269948
1  img_05657.jpg  0.916966  0.200932  0.354845  0.119914  0.566853  0.362618
2  img_02974.jpg  0.887490  0.213137  0.360376  0.144954  0.591034  0.346267
3  img_02837.jpg  0.885008  0.220824  0.360276  0.135349  0.594016  0.326693
4  img_00432.jpg  0.825517  0.252544  0.433282  0.207467  0.589070  0.389228

      SHARK      YFT
0  0.189734  0.766048
1  0.207011  0.672743
2  0.269630  0.692588
3  0.261343  0.704006
4  0.320735  0.651753

```