Detection of calls of Minke whales using deep learning

CEE 696-007: Deep Learning in Civil and Environmental Engineering and Earth Science

Final Project

December 19th, 2019

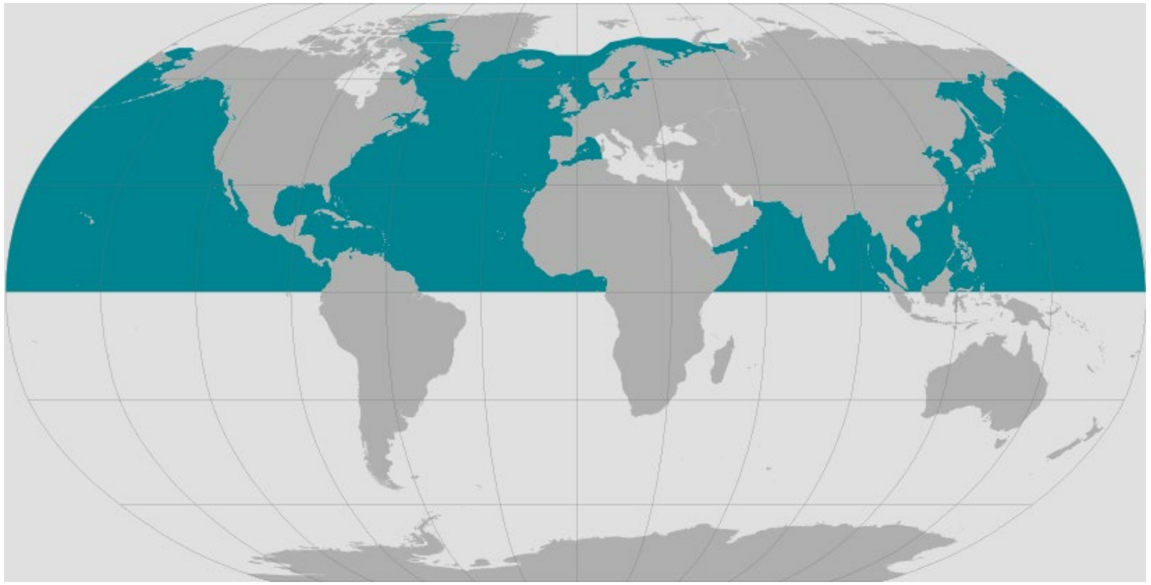
Kei Manabe

Introduction

WEIGHT: Up to 20,000 pounds

LIFESPAN: Up to 50 years

LENGTH: About 35 feet



approximate representation of the minke whale's range

NOAA Fisheries

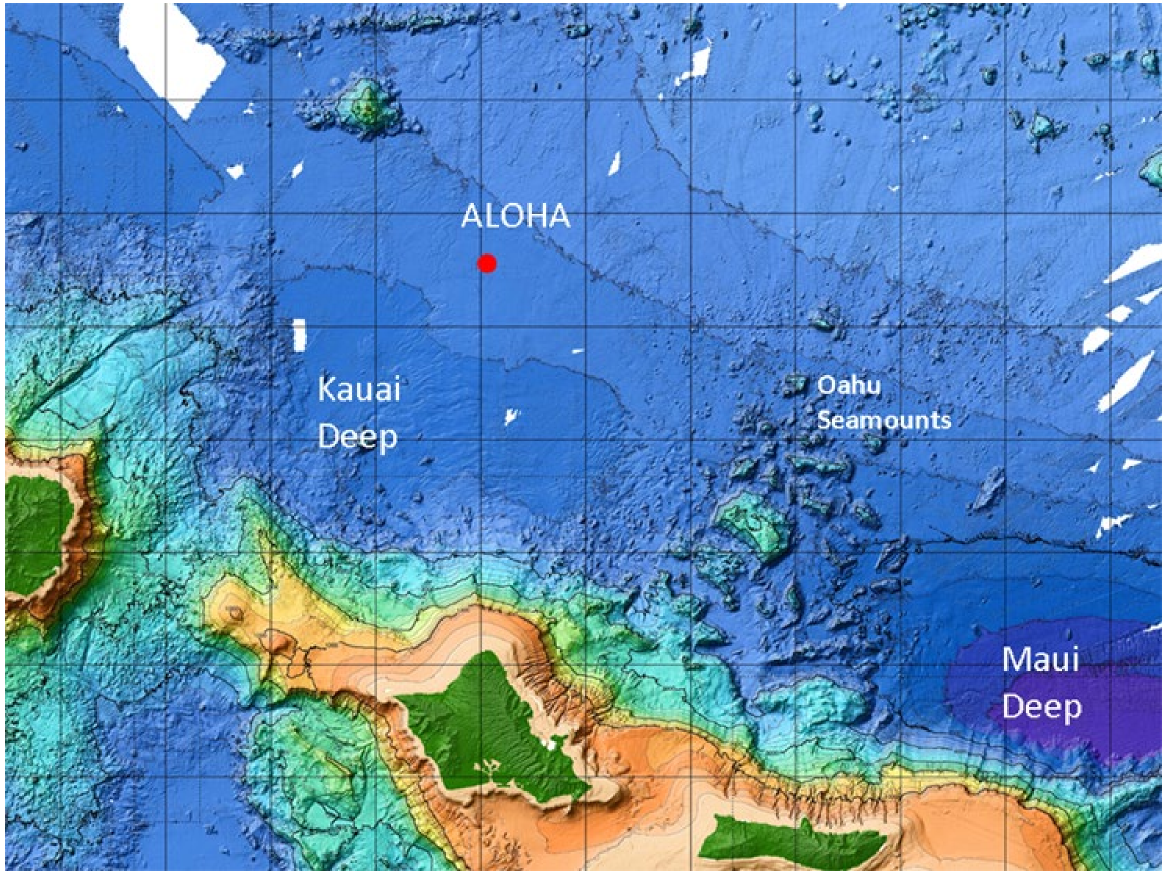
https://www.fisheries.noaa.gov/species/minke-whale

Motivation

- Detection of cetacean's sound is a theme in Ocean Engineering
- Characteristic of the sound mainly depends on specie
- Many algorithms exist to detect cetacean's sound in various situations
- Make a CNN model to determine if there is Minke whale or not

Data

- Sound data (wav format, $f_s = 4\text{kHz}$)



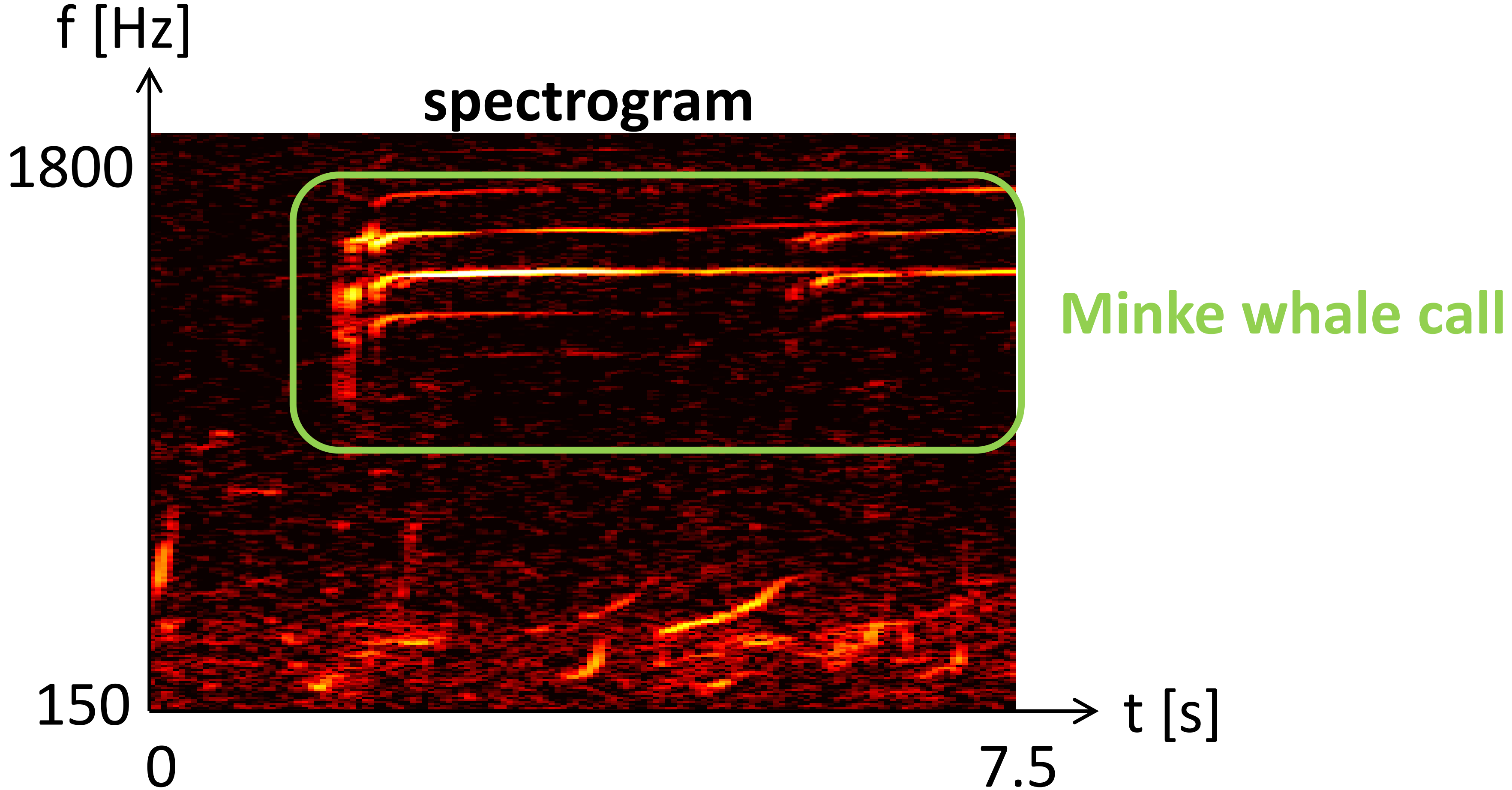
- Recorded Location:

  ALOHA Cabled Observatory

  (22.45N, 158W, 4728m depth)
- Recorded Date: January 9th, 2008
- Length: 60min
- Minke whales & Humpback whales are recorded

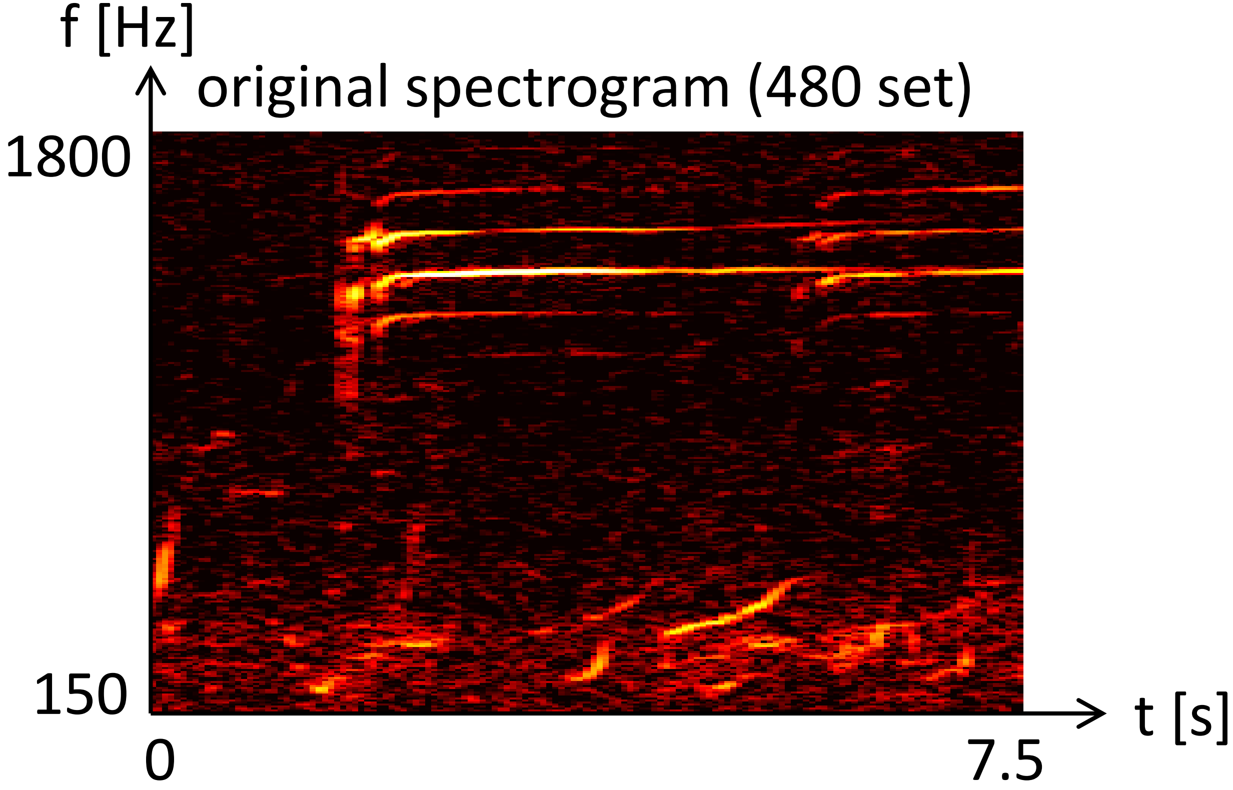
Data Preprocess 1

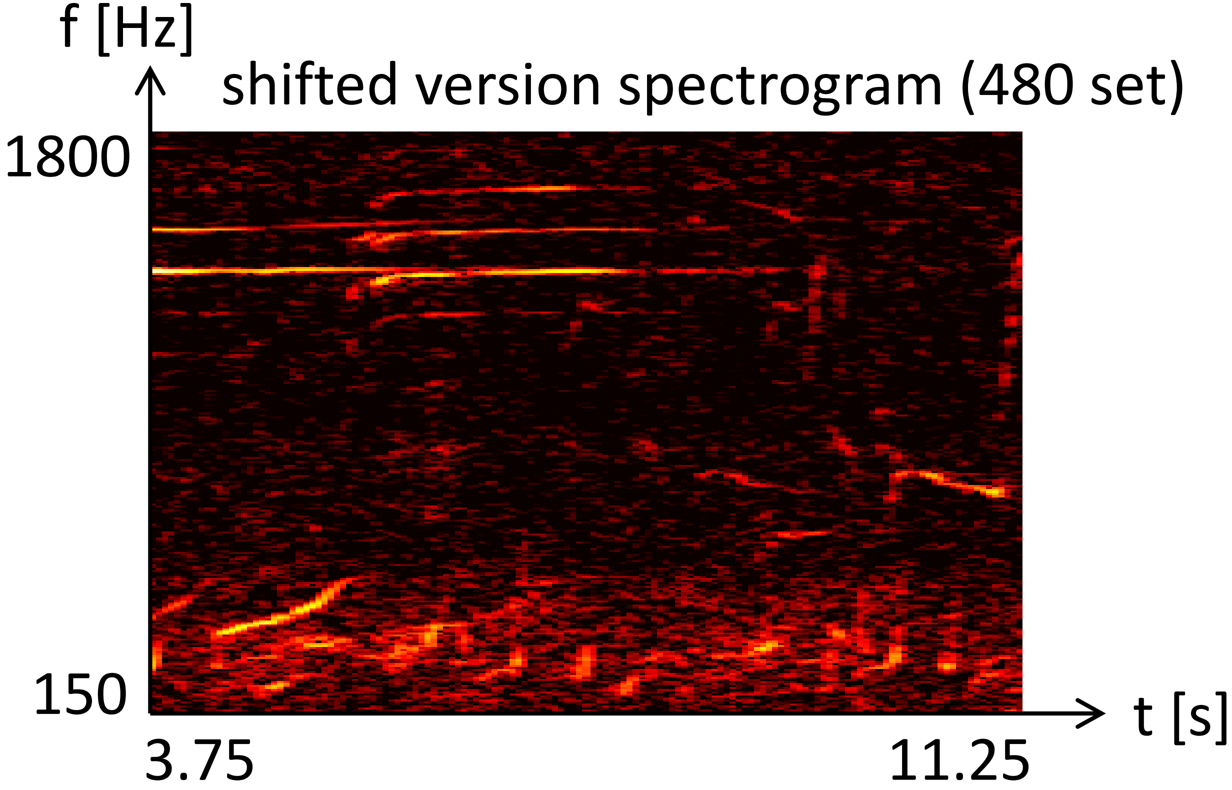
- Every 7.5s data is used to make a spectrogram



Data Preprocess 2

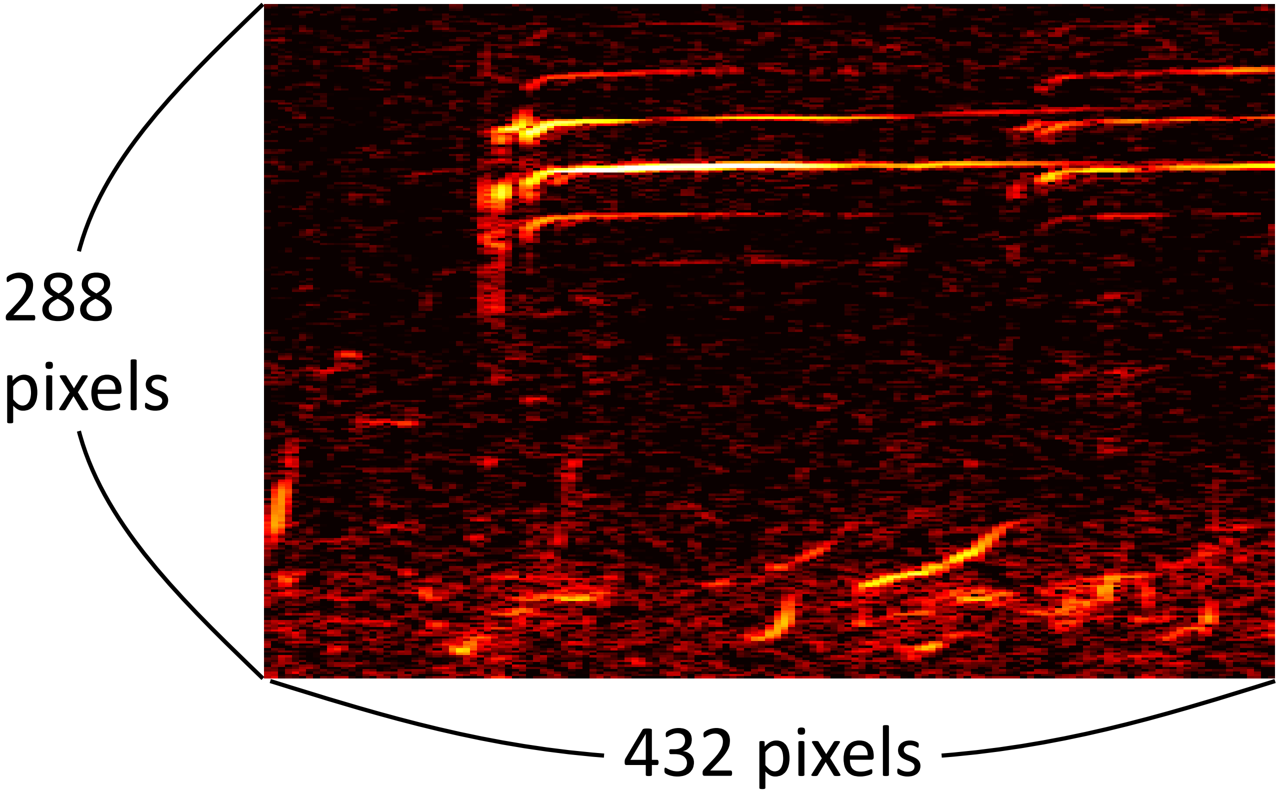
- 60min / 7.5s = 480 set of spectrograms
- More spectrograms needed ➡ 3.75s (= $\frac{7.5s}{2}$) shifted versions are also generated


original spectrogram (480 set)


shifted version spectrogram (480 set)

Data Preprocess 3

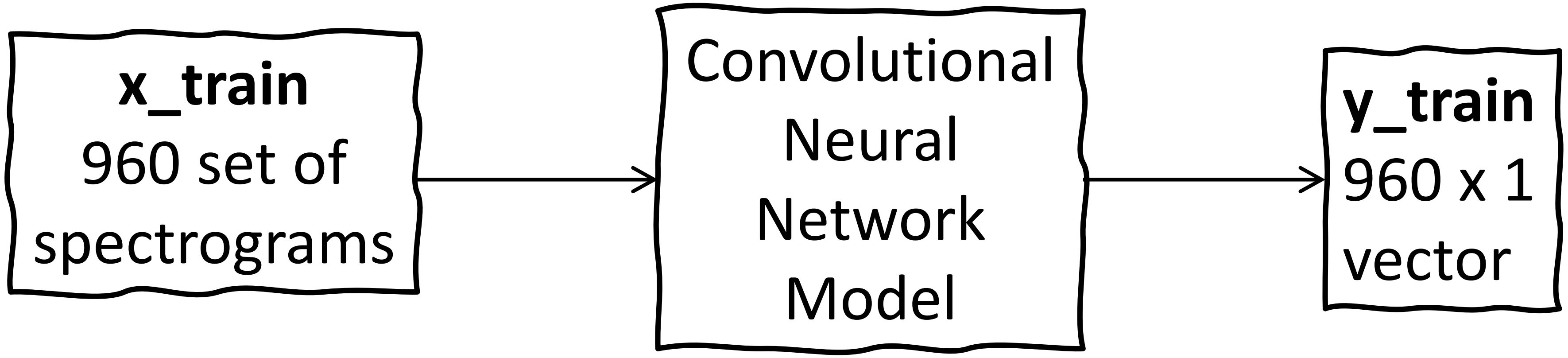
- Spectrograms are dealt as image data



PNG format

Color channel: 4

Overview



x_train
960 set of
spectrograms

Convolutional
Neural
Network
Model

y_train
960 x 1
vector

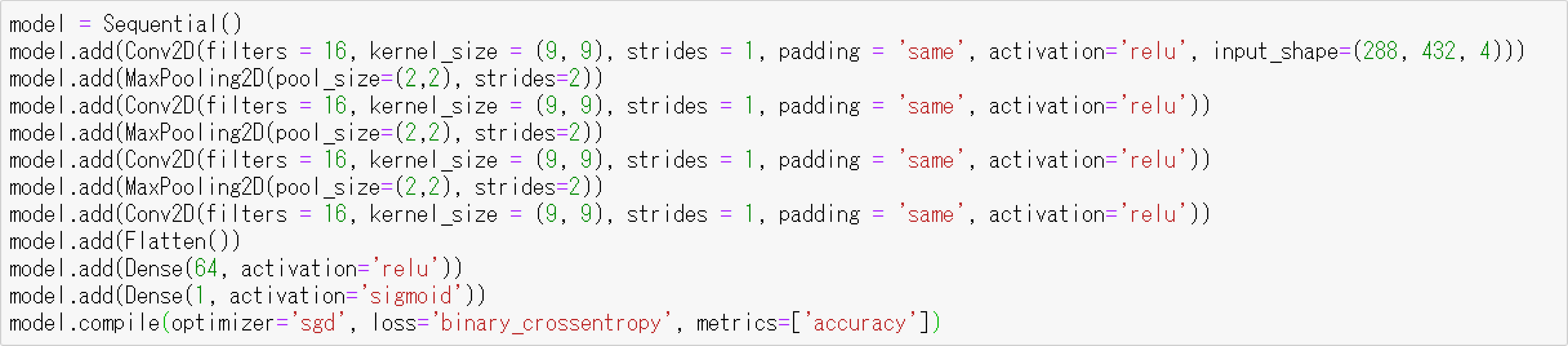
288 x 432 x 4 Tensor

Each element has 1 or 0
1: Minke whale exists
0: Minke whale does not exists

※Manually generated

Coding 1

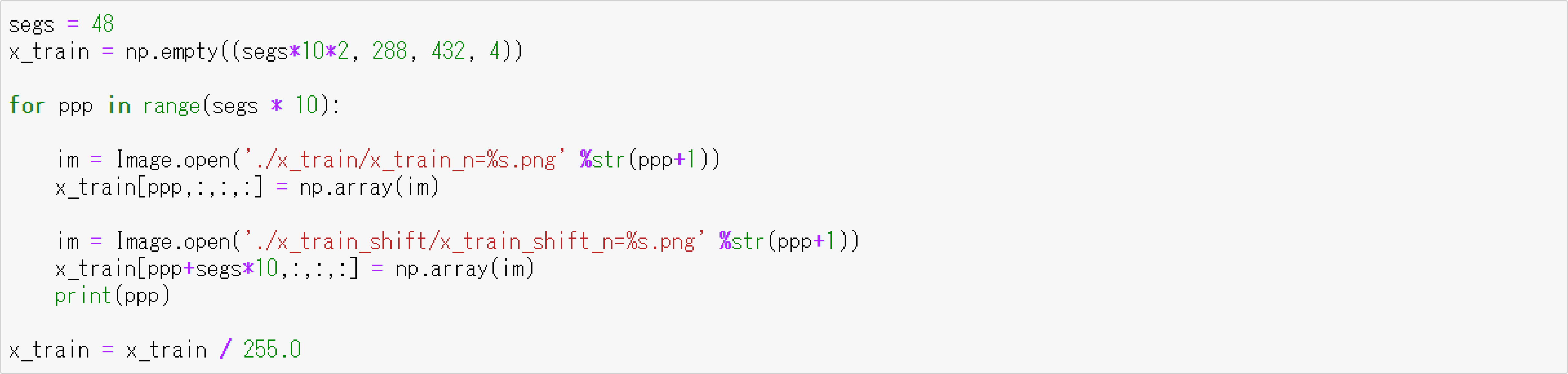
Model design part



```
model = Sequential()
model.add(Conv2D(filters = 16, kernel_size = (9, 9), strides = 1, padding = 'same', activation='relu', input_shape=(288, 432, 4)))
model.add(MaxPooling2D(pool_size=(2,2), strides=2))
model.add(Conv2D(filters = 16, kernel_size = (9, 9), strides = 1, padding = 'same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2), strides=2))
model.add(Conv2D(filters = 16, kernel_size = (9, 9), strides = 1, padding = 'same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2), strides=2))
model.add(Conv2D(filters = 16, kernel_size = (9, 9), strides = 1, padding = 'same', activation='relu'))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
```

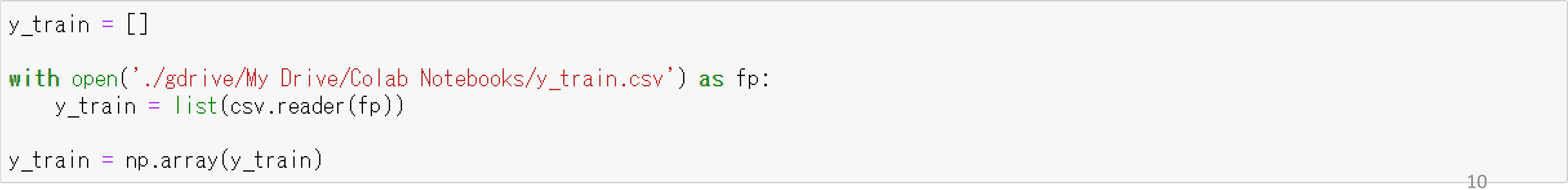
Coding 2

x_train data preparation part



```
segs = 48
x_train = np.empty((segs*10*2, 288, 432, 4))

for ppp in range(segs * 10):

    im = Image.open('./x_train/x_train_n=%s.png' %str(ppp+1))
    x_train[ppp,:,:,:] = np.array(im)

    im = Image.open('./x_train_shift/x_train_shift_n=%s.png' %str(ppp+1))
    x_train[ppp+segs*10,:,:,:] = np.array(im)
    print(ppp)

x_train = x_train / 255.0
```

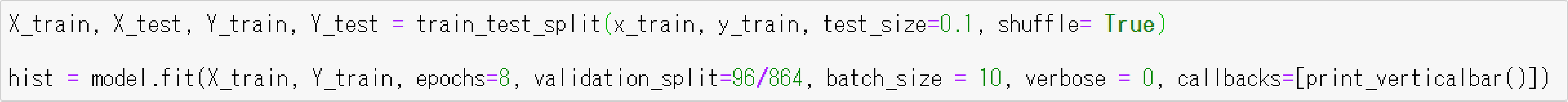
y_train data preparation part



```
y_train = []

with open('./gdrive/My Drive/Colab Notebooks/y_train.csv') as fp:
    y_train = list(csv.reader(fp))

y_train = np.array(y_train)
```

Coding 3

Model training part



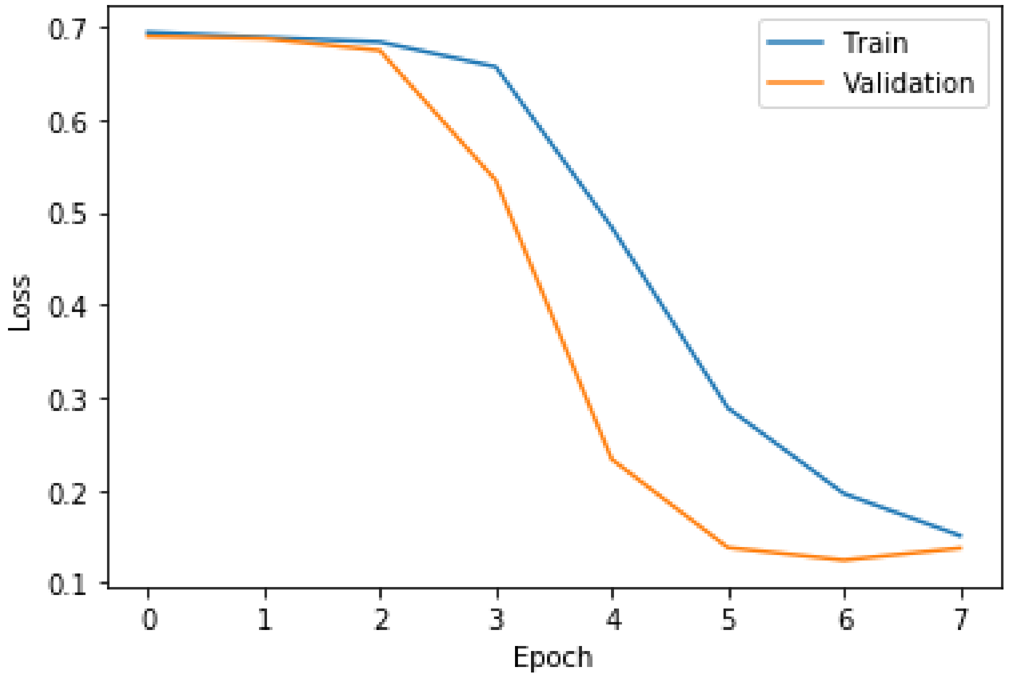
```
X_train, X_test, Y_train, Y_test = train_test_split(x_train, y_train, test_size=0.1, shuffle= True)

hist = model.fit(X_train, Y_train, epochs=8, validation_split=96/864, batch_size = 10, verbose = 0, callbacks=[print_verticalbar()])
```

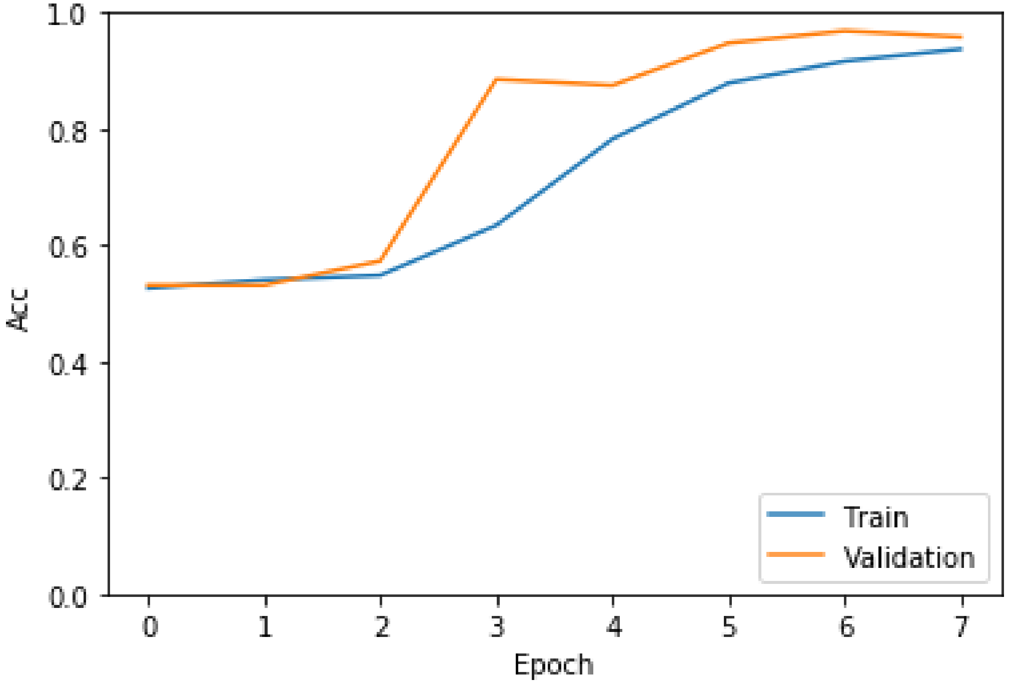
Training uses 8/10 data sets (768 sets)

Validation uses 1/10 data sets (96 sets)

Test uses 1/10 data sets (96 sets)

Result





Epoch is optimized as 8

94.79% model accuracy has been achieved with test data !!

Future Works

- Make another model using sound data directly instead of spectrogram

- Try different species' data

Thank you !!