

Below is code with a link to a happy or sad dataset which contains 80 images, 40 happy and 40 sad. Create a convolutional neural network that trains to 100% accuracy on these images, which cancels training upon hitting training accuracy of $>.999$

Hint -- it will work best with 3 convolutional layers.

```
import tensorflow as tf
import os
import zipfile
from os import path, getcwd, chdir

# DO NOT CHANGE THE LINE BELOW. If you are developing in a local
# environment, then grab happy-or-sad.zip from the Coursera Jupyter Notebook
# and place it inside a local folder and edit the path to that location
path = f"{getcwd()}/../tmp2/happy-or-sad.zip"

zip_ref = zipfile.ZipFile(path, 'r')
zip_ref.extractall("/tmp/h-or-s")
zip_ref.close()

# GRADED FUNCTION: train_happy_sad_model
def train_happy_sad_model():
    # Please write your code only where you are indicated.
    # please do not remove # model fitting inline comments.

    DESIRED_ACCURACY = 0.999

    class myCallback(# your code):
        # Your Code

    callbacks = myCallback()

    # This Code Block should Define and Compile the Model. Please assume the image
    model = tf.keras.models.Sequential([
        # Your Code Here
    ])

    from tensorflow.keras.optimizers import RMSprop

    model.compile(# Your Code Here #)

    # This code block should create an instance of an ImageDataGenerator called tra
    # And a train_generator by calling train_datagen.flow_from_directory

    from tensorflow.keras.preprocessing.image import ImageDataGenerator

    train_datagen = # Your Code Here

    # Please use a target_size of 150 X 150.
    train_generator = train_datagen.flow_from_directory(
        # Your Code Here
```

```

# Expected output: 'Found 80 images belonging to 2 classes'

# This code block should call model.fit_generator and train for
# a number of epochs.
# model fitting
history = model.fit_generator(
    # Your Code Here)
# model fitting
return history.history['acc'][-1]

```



File "<ipython-input-2-7b9511ef9ca6>", line 14
 model = tf.keras.models.Sequential([

^

SyntaxError: invalid syntax

SEARCH STACK OVERFLOW

```

# The Expected output: "Reached 99.9% accuracy so cancelling training!"
train_happy_sad_model()

```



NameError Traceback (most recent call last)

<ipython-input-3-a7dc121ed15a> in <module>

```

1 # The Expected output: "Reached 99.9% accuracy so cancelling training!"
----> 2 train_happy_sad_model()

```

NameError: name 'train_happy_sad_model' is not defined

SEARCH STACK OVERFLOW

```

# Now click the 'Submit Assignment' button above.
# Once that is complete, please run the following two cells to save your work and c

```

```

%%javascript
<!-- Save the notebook -->
IPython.notebook.save_checkpoint();

```

```

%%javascript
IPython.notebook.session.delete();
window.onbeforeunload = null
setTimeout(function() { window.close(); }, 1000);

```

