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

```
import tensorflow as tf
import os
import zipfile


DESIRED_ACCURACY = 0.999

!wget --no-check-certificate \
    "https://storage.googleapis.com/laurencemoroney-blog.appspot.com/happy-or-sad.zi
    -O "/tmp/happy-or-sad.zip"

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

class myCallback(tf.keras.callbacks.Callback):
  def on_epoch_end(self, epoch, logs={}):
    if(logs.get('acc') > DESIRED_ACCURACY):
      print("\nReached 99.9% accuracy so cancelling training!")
      self.model.stop_training = True

callbacks = myCallback()


# Define and Compile the Model

model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(64, (3,3), activation="relu", input_shape=(150,150,3)),
    tf.keras.layers.MaxPool2D(2, 2),
    tf.keras.layers.Conv2D(64, (3,3), activation="relu"),
    tf.keras.layers.MaxPool2D(2, 2),
    tf.keras.layers.Conv2D(64, (3,3), activation="relu"),
    tf.keras.layers.MaxPool2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation="relu"),
    tf.keras.layers.Dense(1, activation="sigmoid")
])

model.compile(loss="binary_crossentropy", optimizer="adam", metrics=['acc'])


# Creating an instance of an ImageDataGenerator called train_datagen
# And a train_generator by calling train_datagen.flow_from_directory

from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale=1/255)

train_generator = train_datagen.flow_from_directory(
        '/tmp/h-or-s', # this is source directory
        target_size=(150,150),
        batch_size=1,
        class_mode="binary")

# This code block should call model.fit_generator and train for
# a number of epochs.
history = model.fit_generator(
          train_generator,
          callbacks=[callbacks],
          steps_per_epoch=80,
          epochs=20,
          verbose=1,
          )
```

```
--2019-05-25 21:44:18--  https://storage.googleapis.com/laurencemoroney-blog.a
Resolving storage.googleapis.com (storage.googleapis.com)... 172.217.194.128,
Connecting to storage.googleapis.com (storage.googleapis.com)|172.217.194.128|
HTTP request sent, awaiting response... 200 OK
Length: 2670333 (2.5M) [application/zip]
Saving to: ‘/tmp/happy-or-sad.zip’

/tmp/happy-or-sad.z 100%[===================>]   2.55M  --.-KB/s    in 0.02s

2019-05-25 21:44:19 (163 MB/s) - ‘/tmp/happy-or-sad.zip’ saved [2670333/267033

Found 80 images belonging to 2 classes.
Epoch 1/20
80/80 [==============================] - 1s 14ms/step - loss: 0.8626 - acc: 0.
Epoch 2/20
80/80 [==============================] - 1s 13ms/step - loss: 0.4548 - acc: 0.
Epoch 3/20
80/80 [==============================] - 1s 13ms/step - loss: 0.3684 - acc: 0.
Epoch 4/20
80/80 [==============================] - 1s 13ms/step - loss: 0.1056 - acc: 0.
Epoch 5/20
80/80 [==============================] - 1s 13ms/step - loss: 0.0470 - acc: 0.
Epoch 6/20
79/80 [============================>.] - ETA: 0s - loss: 0.0053 - acc: 1.0000
Reached 99.9% accuracy so cancelling training!
80/80 [==============================] - 1s 13ms/step - loss: 0.0052 - acc: 1.
```