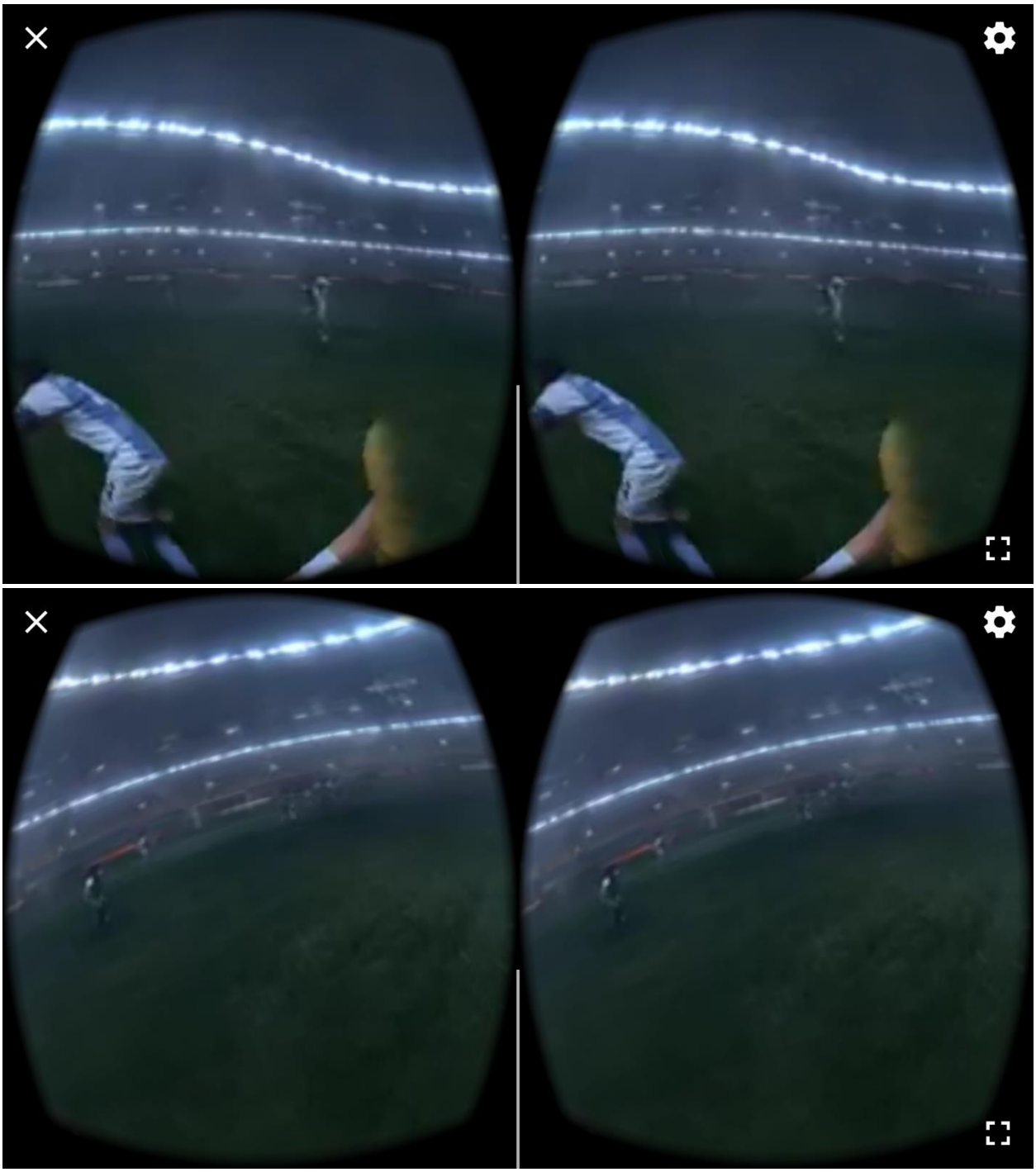
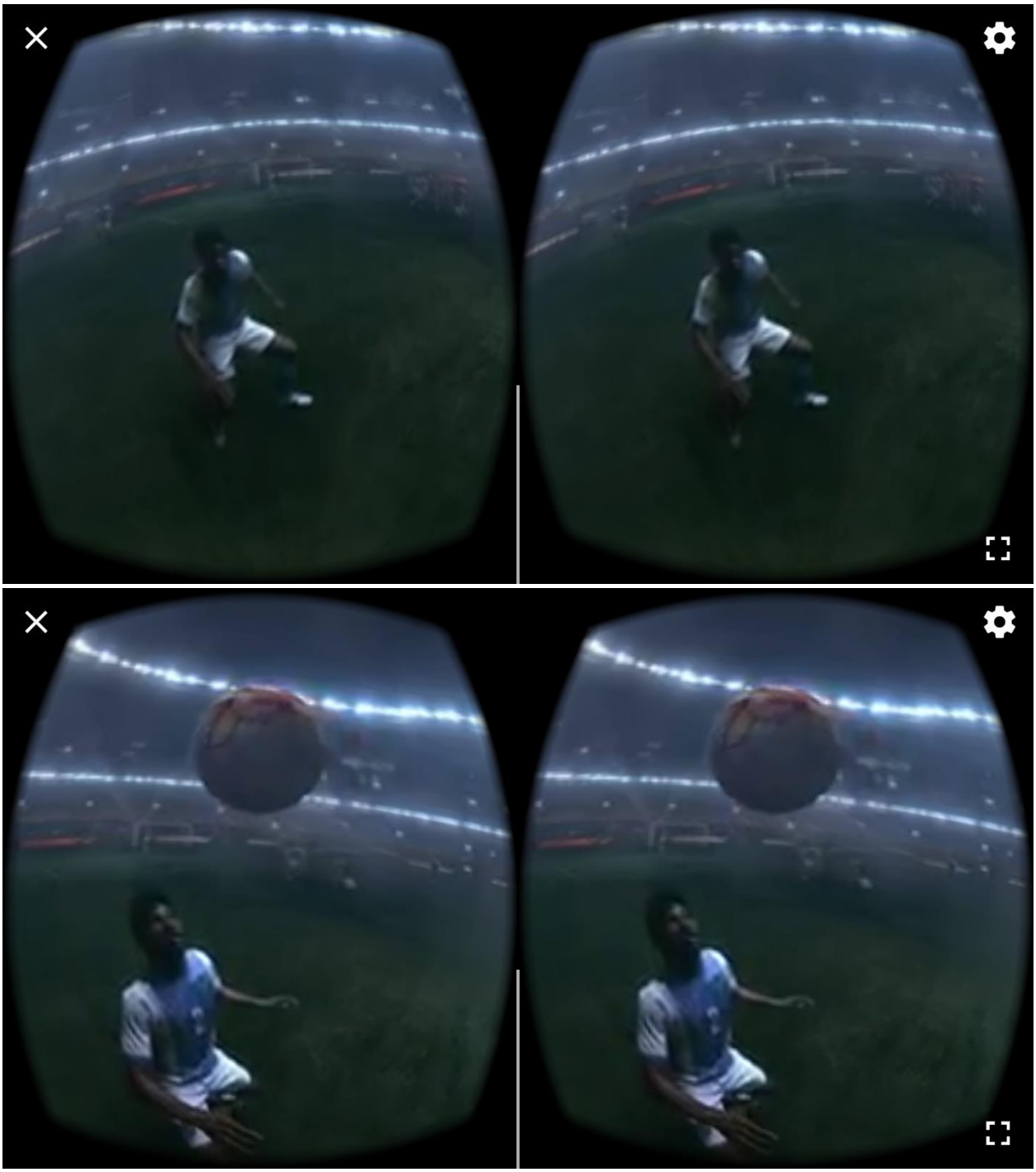
Lab 8 Report-CS 5542 Big Data Analytics and Apps

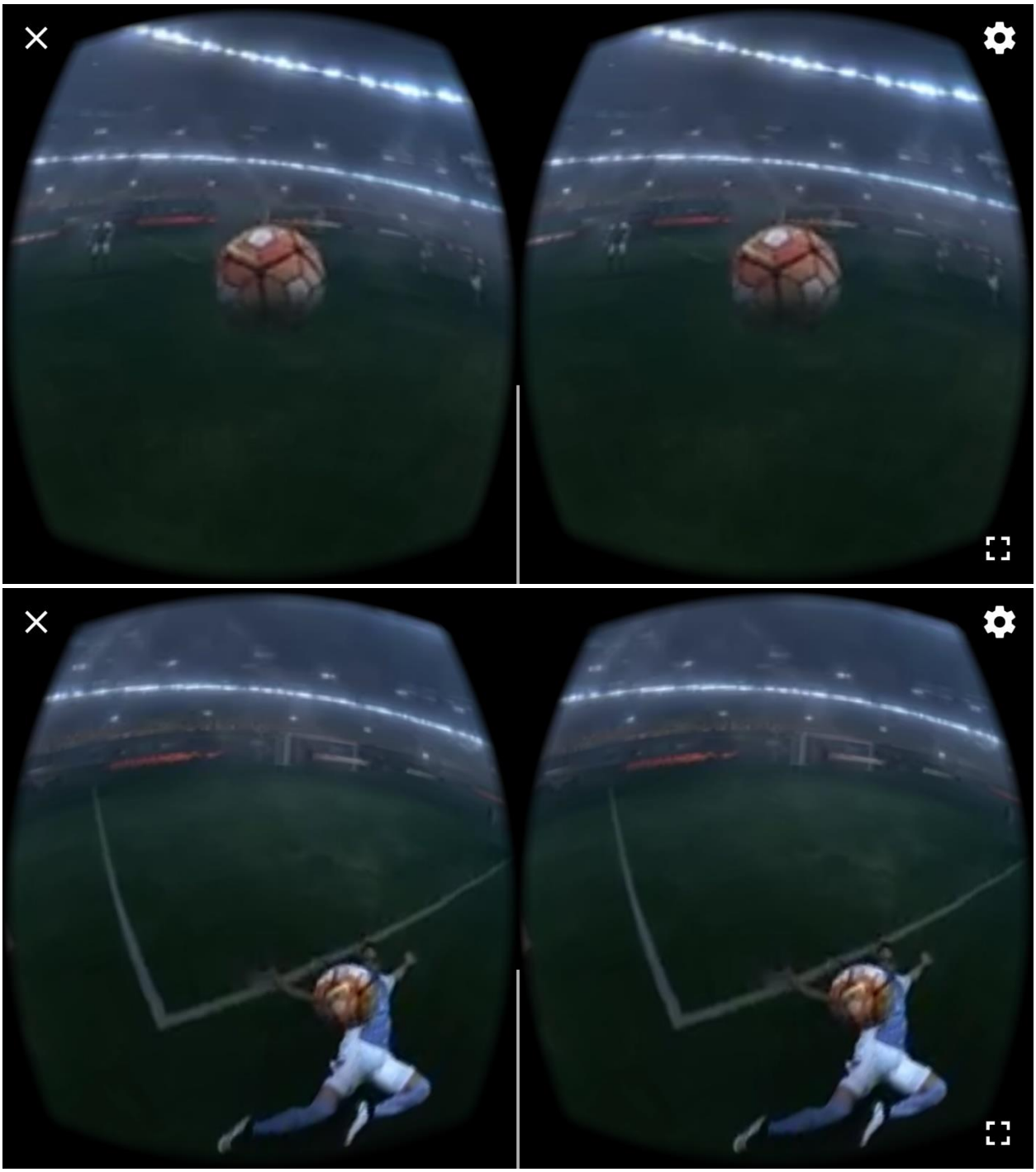
Android Studio CardBoard App

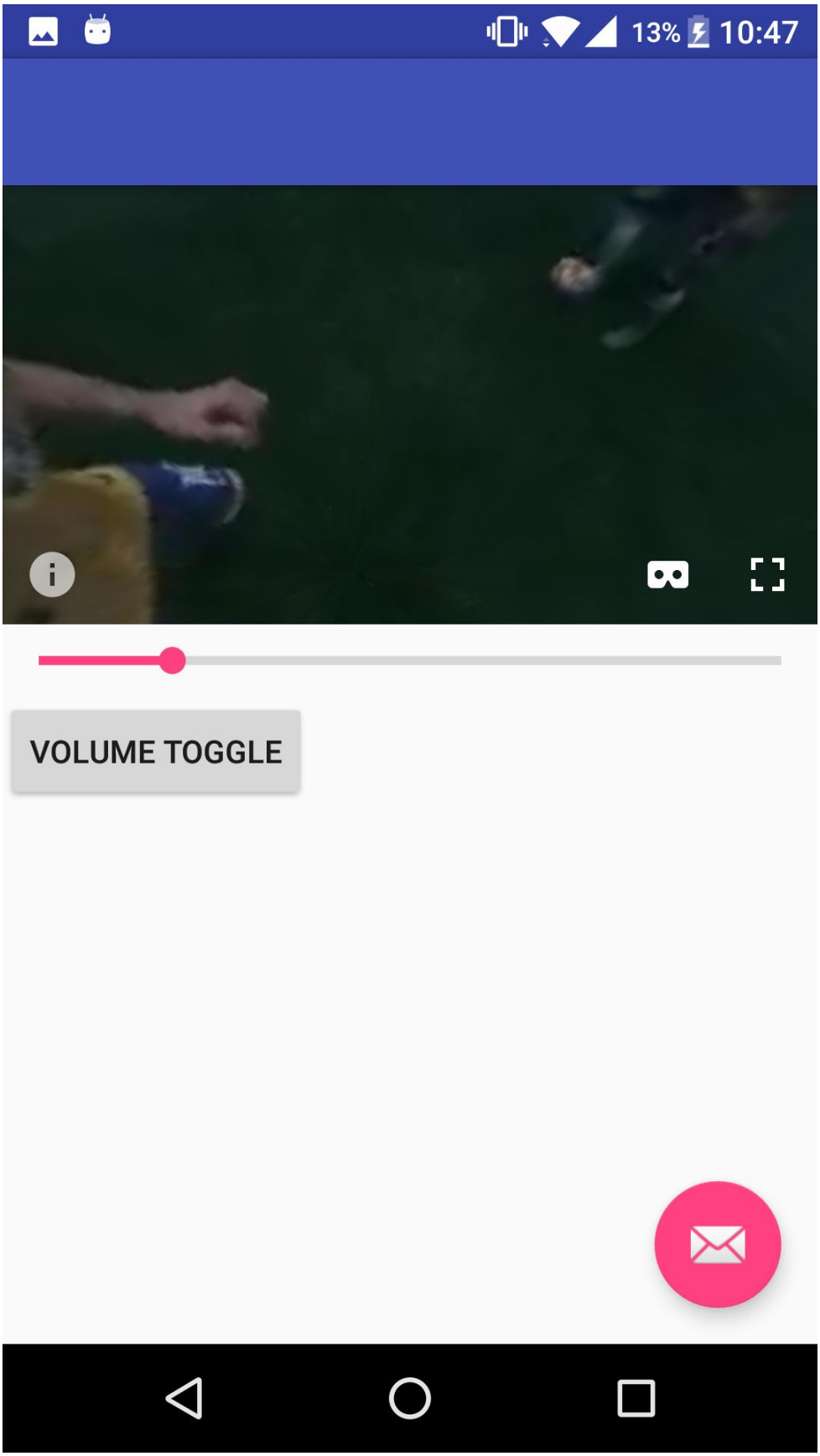
The second part of the program is google cardboard application program. For this part, two different programs are used one is the program given as source code for lab 8 and another is the default google program from the google github:https://github.com/googlevr/gvr-android-sdk

The first program, since it should be based on project. Sports video is choosen for this. The sports 360 degree video is taken as an input for this program. Here are some of the screen shots for the program:



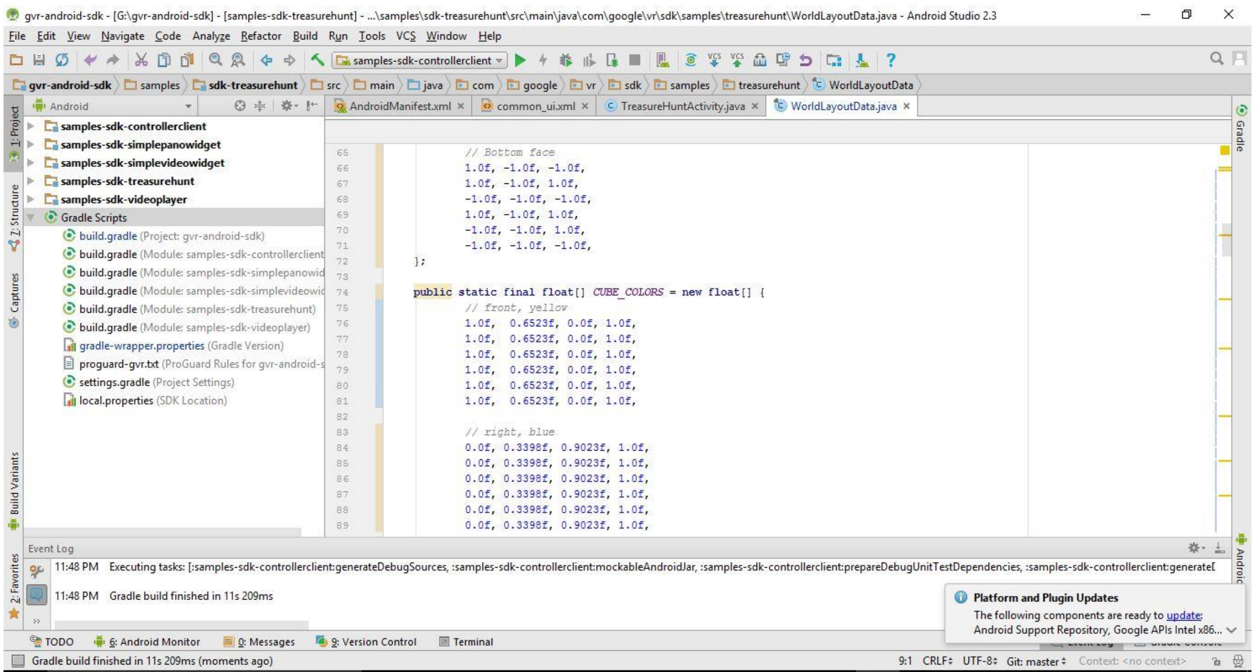


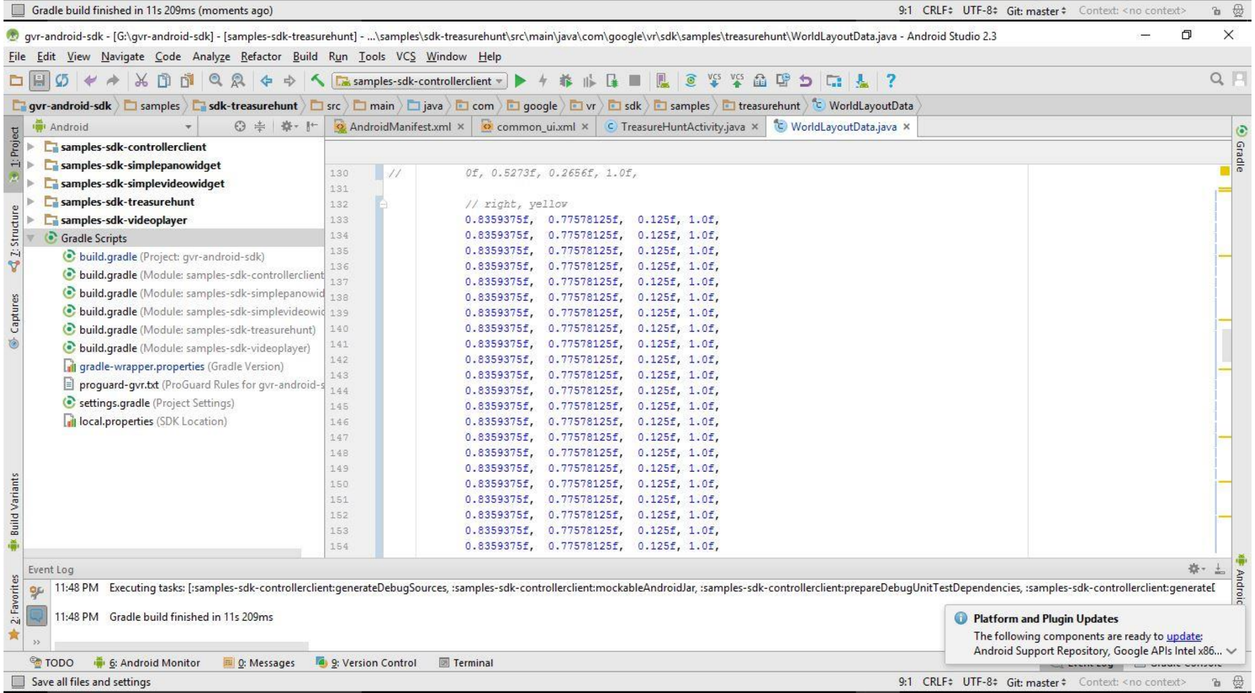


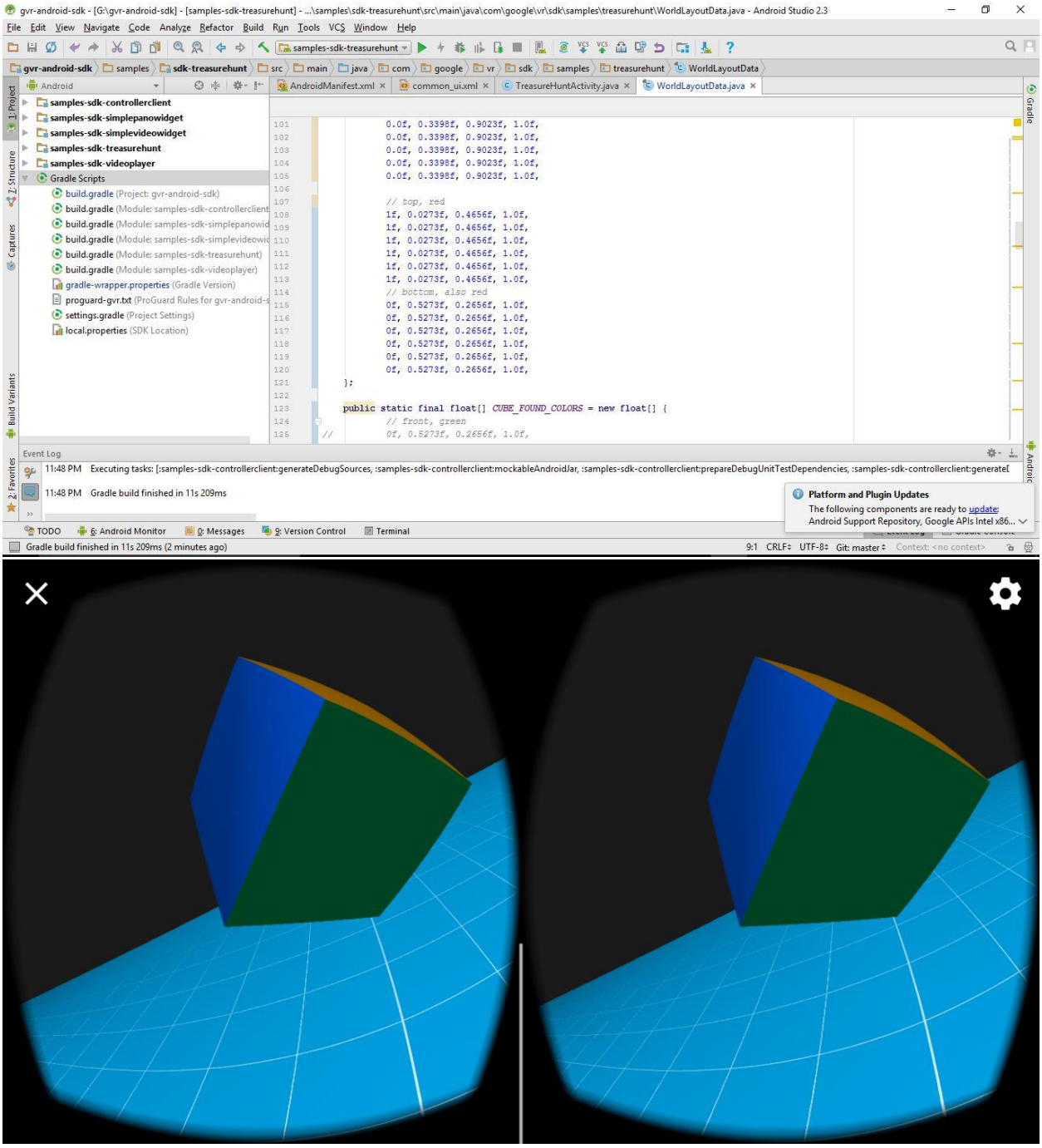

13%
10:47
VOLUME TOGGLE

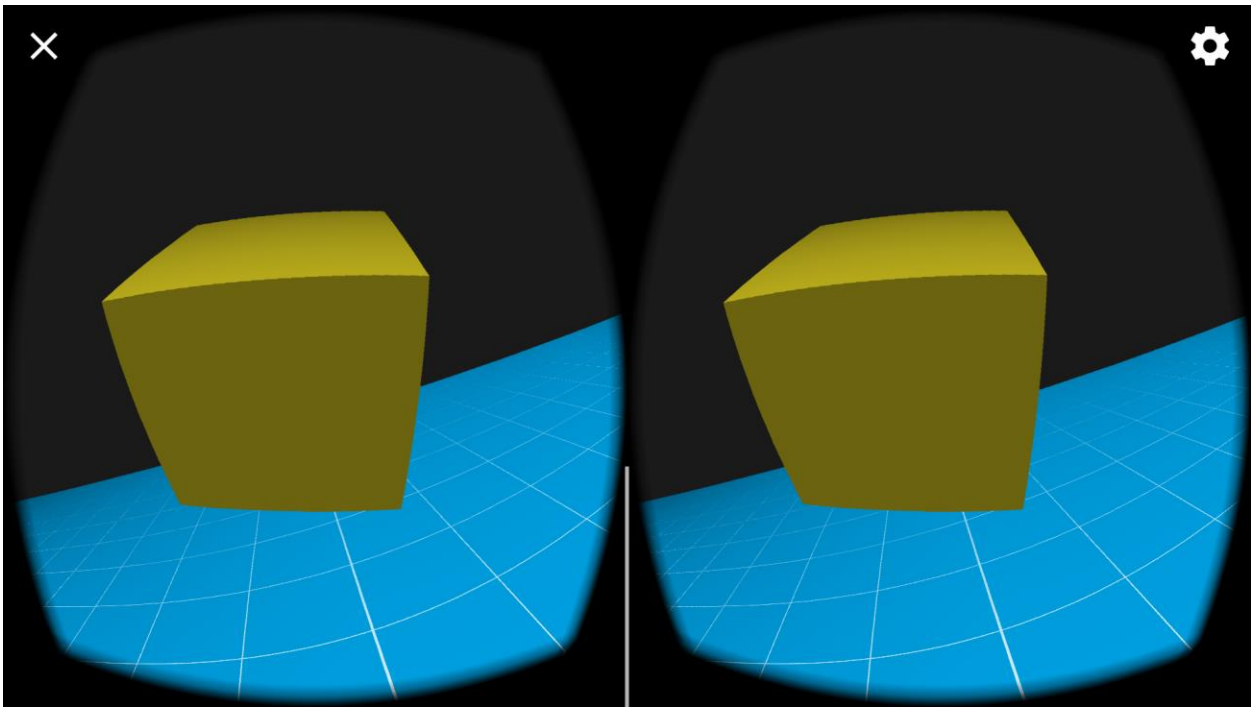
The second program is treasure hunt game which contains all the features such as client controller, head movements controller, spacial audio. The program is the exact program taken from the google github: https://github.com/googlevr/gvr-android-sdk

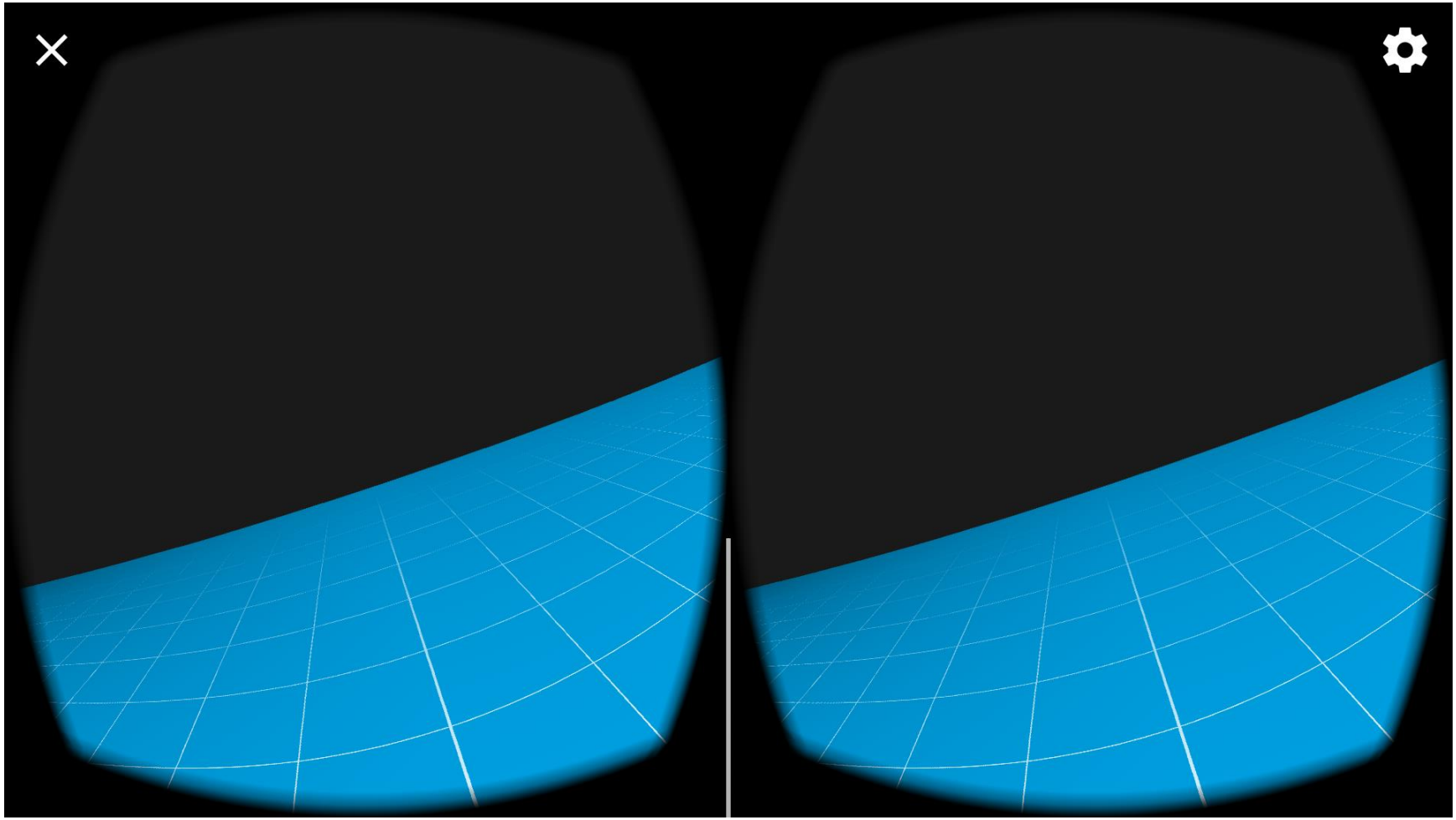
The changes made from the original game are evident in the screenshots. Here are the screenshots for the program:

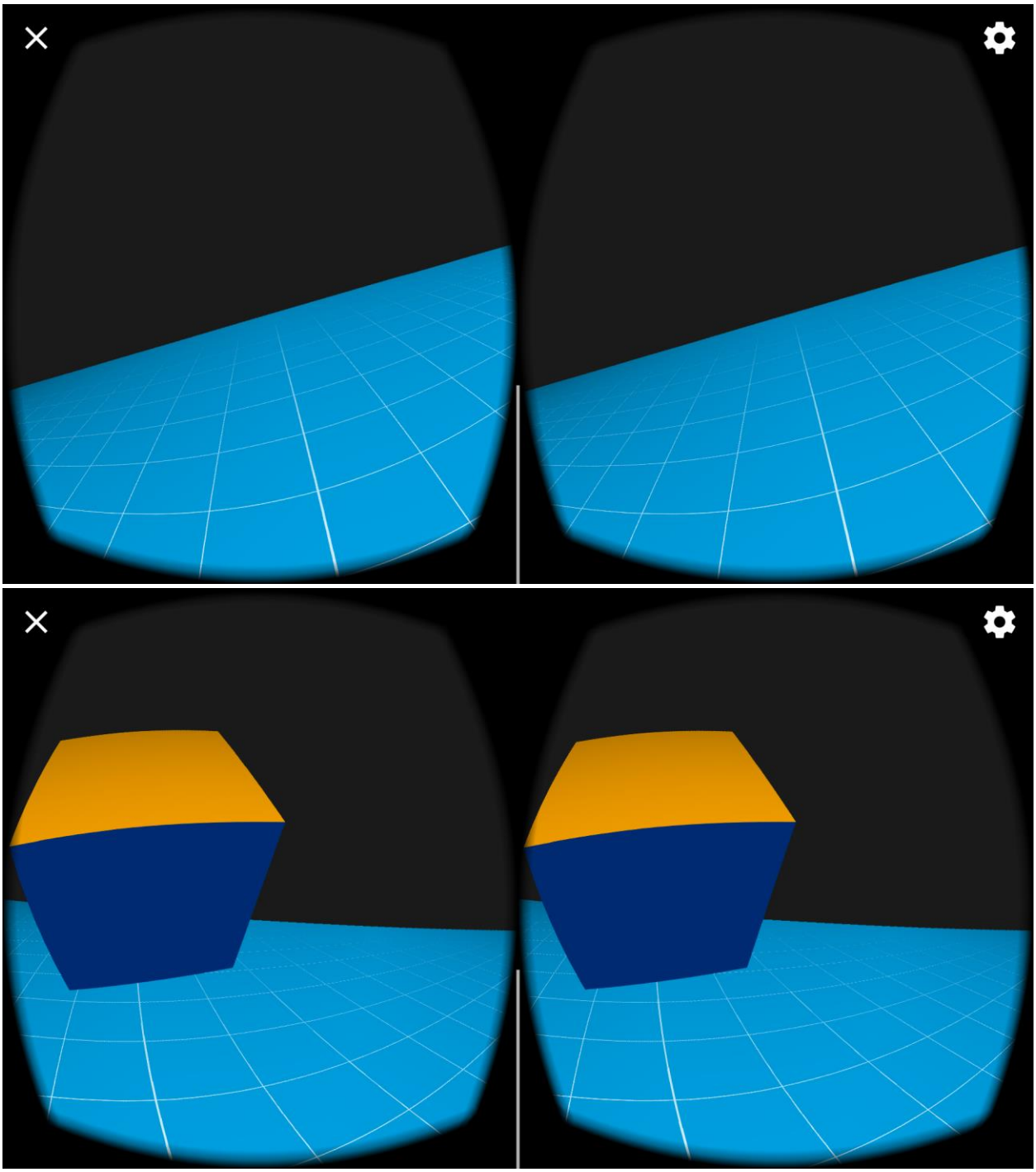


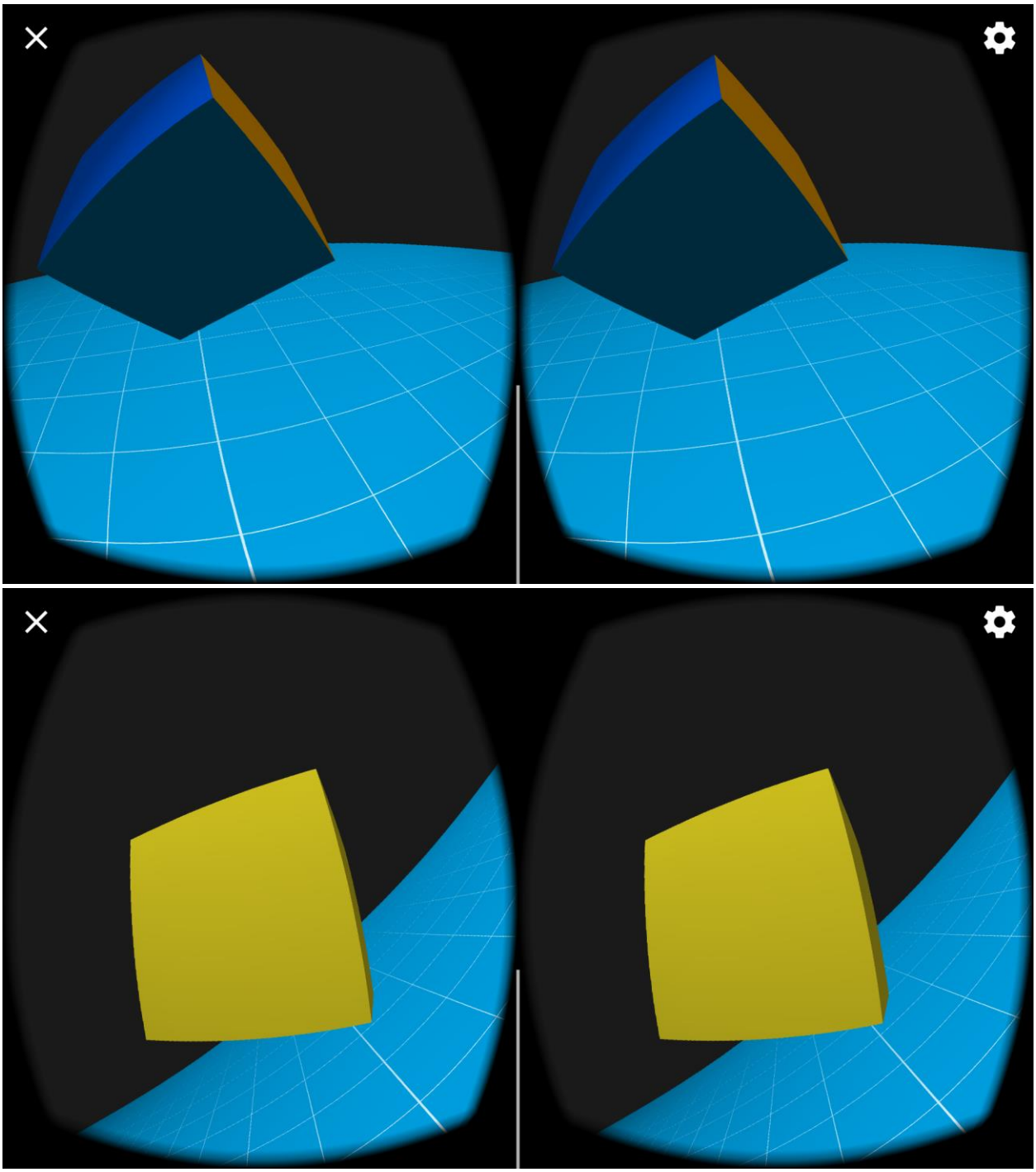


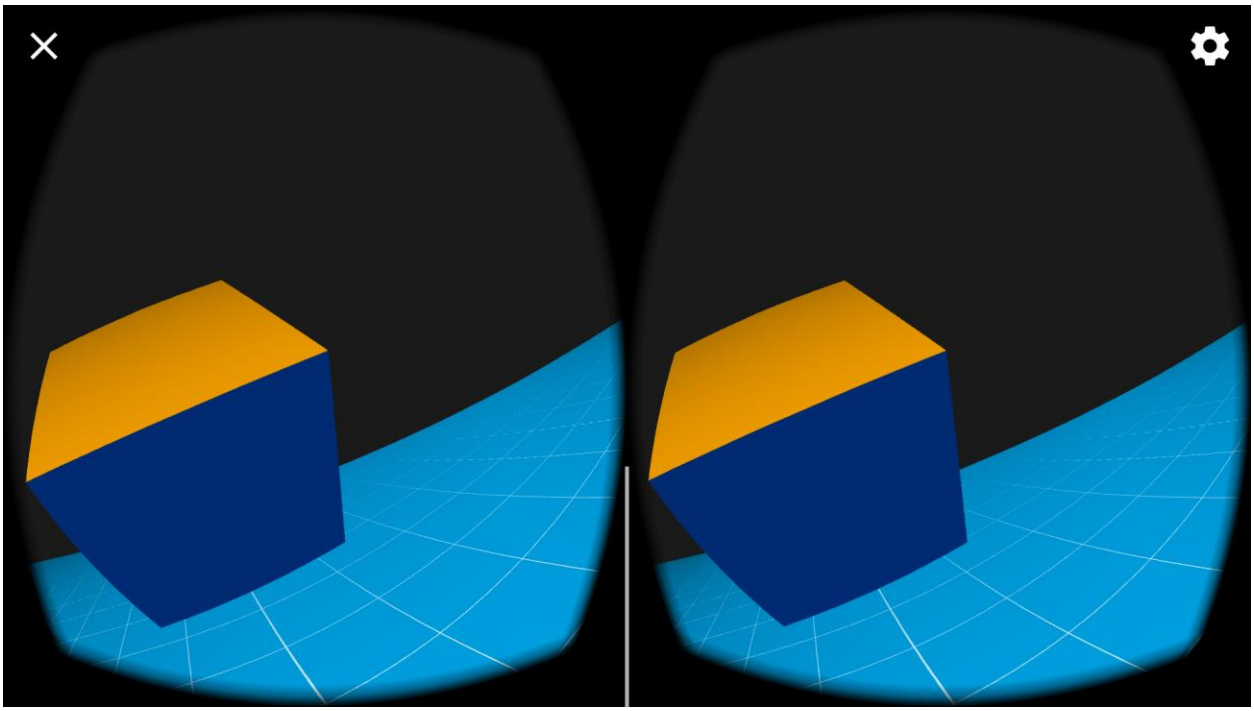

gvr-android-sdk - [G:\gvr-android-sdk] - [samples-sdk-treasurehunt] - ...\samples\sdk-treasurehunt\src\main\java\com\google\vr\sdk\samples\treasurehunt\WorldLayoutData.java - Android Studio 2.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
AndroidManifest.xml
common_ui.xml
TreasureHuntActivity.java
WorldLayoutData.java
samples-sdk-controllerclient
samples-sdk-simplepanowidget
samples-sdk-simplevideowidget
samples-sdk-treasurehunt
samples-sdk-videoplayer
Gradle Scripts
Event Log
11:48 PM Executing tasks: [:samples-sdk-controllerclient:generateDebugSources, :samples-sdk-controllerclient:mockableAndroidJar, :samples-sdk-controllerclient:prepareDebugUnitTestDependencies, :samples-sdk-controllerclient:generate
11:48 PM Gradle build finished in 11s 209ms
Platform and Plugin Updates
The following components are ready to update:
Android Support Repository, Google APIs Intel x86...
TODO
6: Android Monitor
0: Messages
9: Version Control
Terminal
Gradle build finished in 11s 209ms (2 minutes ago)
9:1 CRLF UTF-8 Git: master Context: <no context>











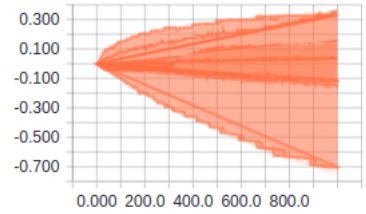
When the cube is placed in the centre, the color of the cube is changes to greenish tinge

Tensor flow program for Softmax Regression Classification:

For the tensor flow program same MNSIT data set is used since no image data set seems available and compatible. Here are the screen shots for the softmax regression program:

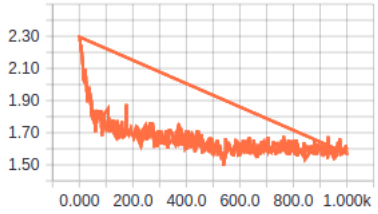
bias 1

bias
.

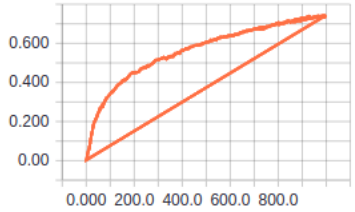


cross_hist 1

cross_hist
.

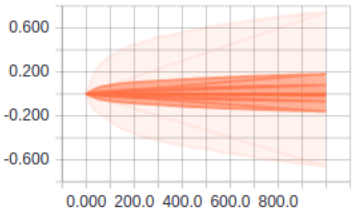


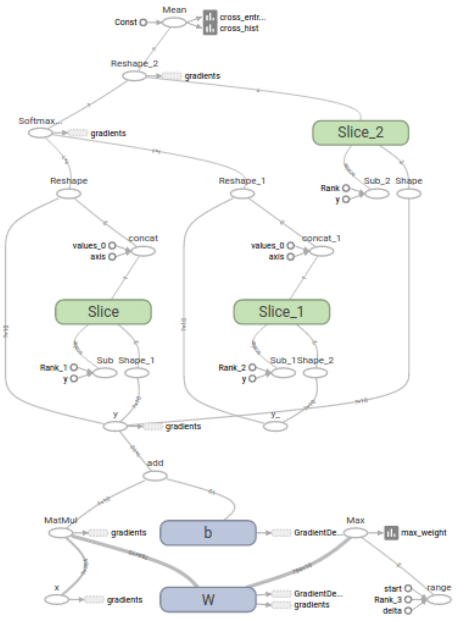
max_weight
.



weights 1

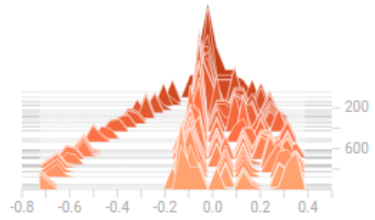
weights
.





bias

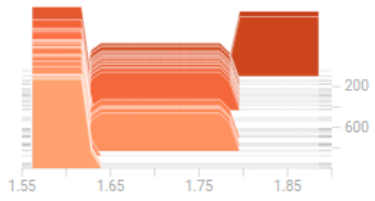
.



cross_hist

cross_hist

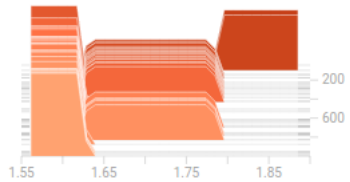
.



cross_hist 1

cross_hist

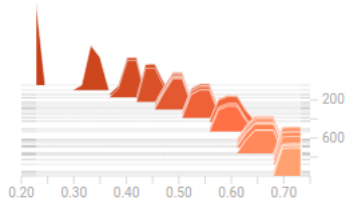
.



max_weight 1

max_weight

.



File Edit View Navigate Code Refactor Run Tools VCS Window Help

MNIST_SOFTMAX › mnist_test.py

mnist_test

Project

mnist_train.py × mnist_test.py ×

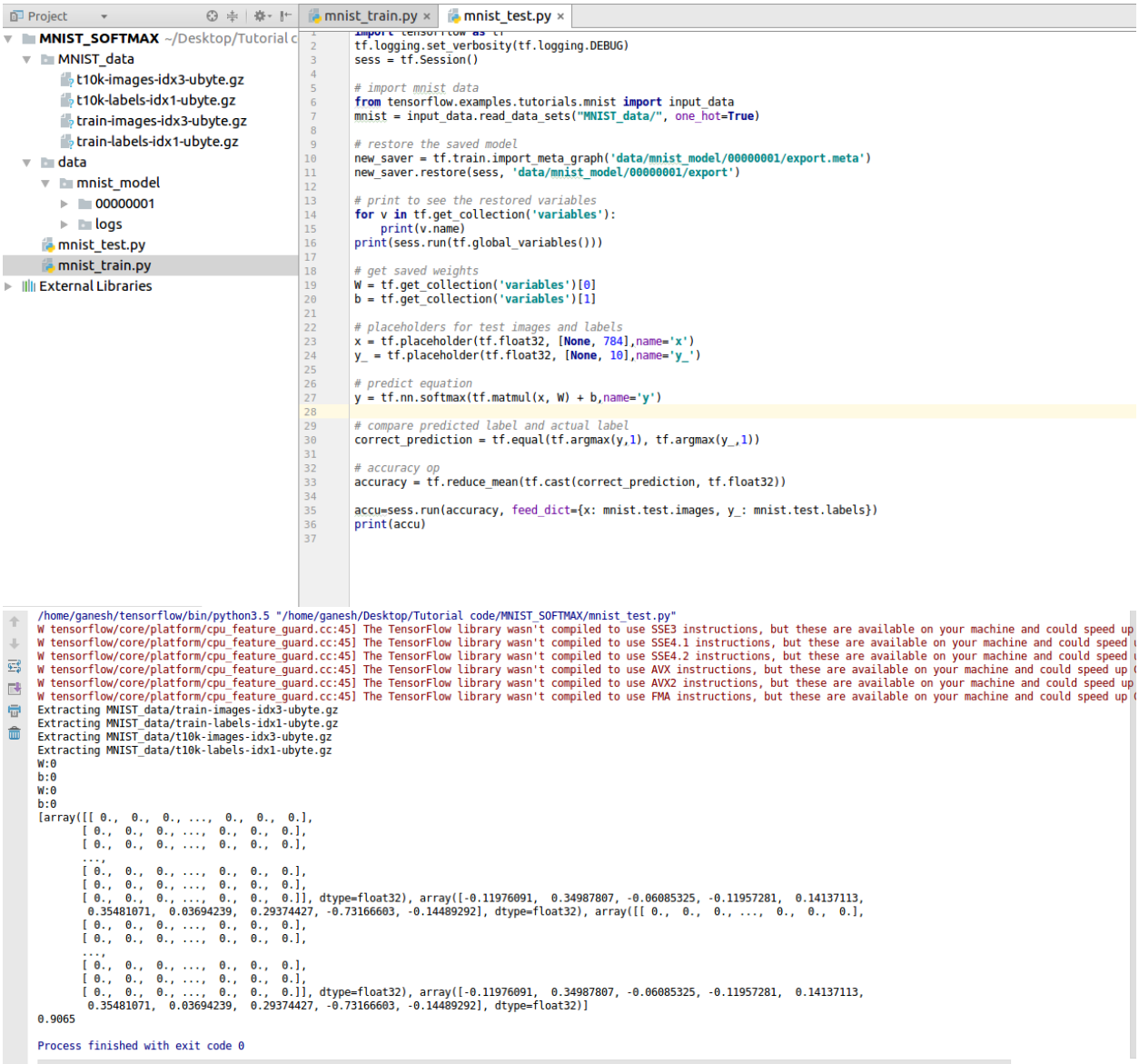
```
MNIST_SOFTMAX ~/Desktop/Tutorial c
  MNIST_data
    t10k-images-idx3-ubyte.gz
    t10k-labels-idx1-ubyte.gz
    train-images-idx3-ubyte.gz
    train-labels-idx1-ubyte.gz
  data
  mnist_test.py
  mnist_train.py
External Libraries
```

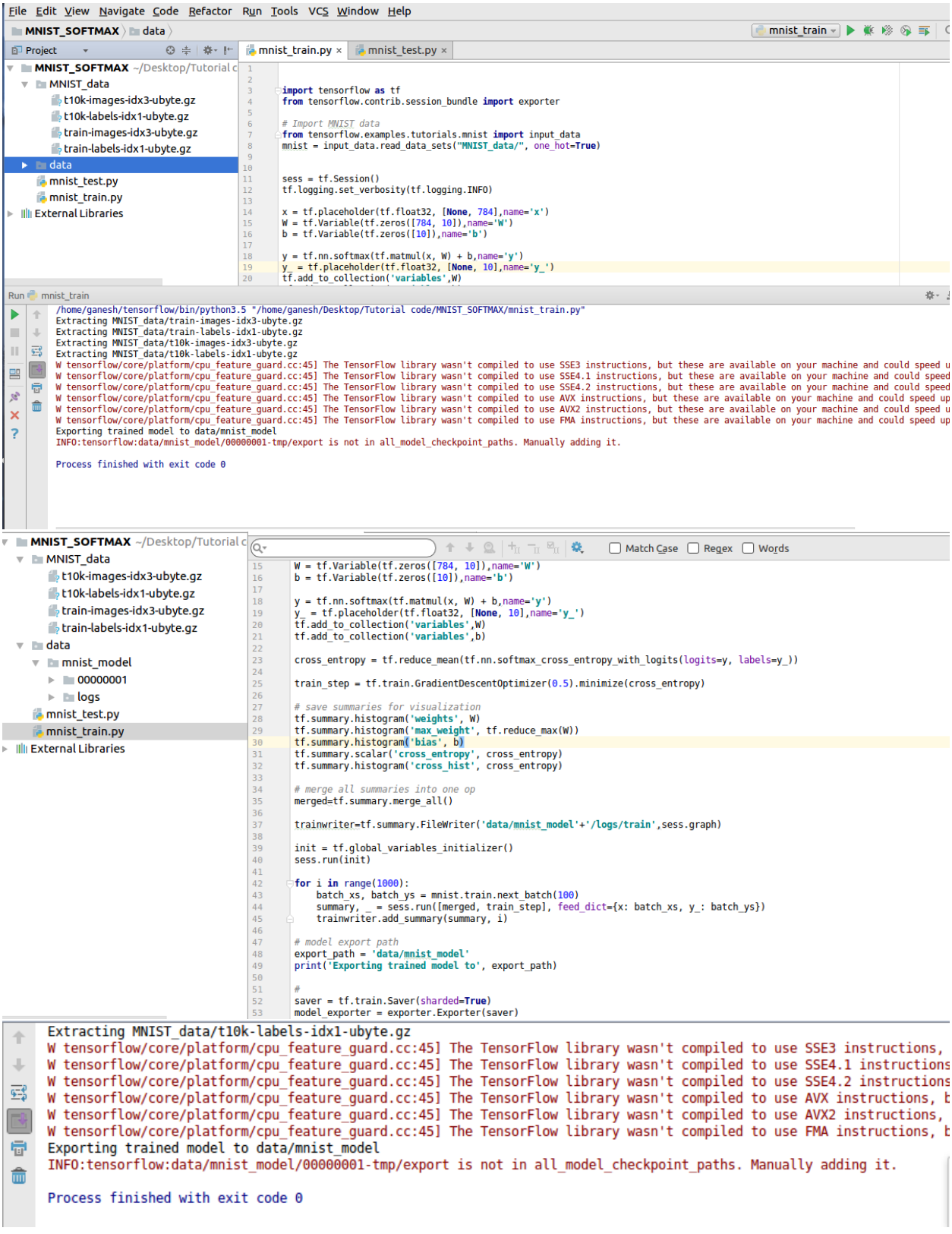
```
7  mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
8
9  # restore the saved model
10 new_saver = tf.train.import_meta_graph('data/mnist_model/00000001/export.meta')
11 new_saver.restore(sess, 'data/mnist_model/00000001/export')
12
13 # print to see the restored variables
14 for v in tf.get_collection('variables'):
15     print(v.name)
16 print(sess.run(tf.global_variables()))
17
18 # get saved weights
19 W = tf.get_collection('variables')[0]
20 b = tf.get_collection('variables')[1]
21
22 # placeholders for test images and labels
23 x = tf.placeholder(tf.float32, [None, 784],name='x')
24 y_ = tf.placeholder(tf.float32, [None, 10],name='y_')
25
26 # predict equation
```

Run mnist_test

```
b:0
[array([[ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       ...,
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.]], dtype=float32), array([-0.11976091,  0.34987807, -0.06085325, -0.11957281,  0.14137113,
        0.35481071,  0.03694239,  0.29374427, -0.73166603, -0.14489292], dtype=float32), array([[ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       ...,
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ...,  0.,  0.,  0.]], dtype=float32), array([-0.11976091,  0.34987807, -0.06085325, -0.11957281,  0.14137113,
        0.35481071,  0.03694239,  0.29374427, -0.73166603, -0.14489292], dtype=float32)]
0.9065

Process finished with exit code 0
```


Project
MNIST_SOFTMAX ~/Desktop/Tutorial c
MNIST_data
t10k-images-idx3-ubyte.gz
t10k-labels-idx1-ubyte.gz
train-images-idx3-ubyte.gz
train-labels-idx1-ubyte.gz
data
mnist_model
00000001
logs
mnist_test.py
mnist_train.py
External Libraries
mnist_train.py
mnist_test.py
import tensorflow as tf
tf.logging.set_verbosity(tf.logging.DEBUG)
sess = tf.Session()
# import mnist data
from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
# restore the saved model
new_saver = tf.train.import_meta_graph('data/mnist_model/00000001/export.meta')
new_saver.restore(sess, 'data/mnist_model/00000001/export')
# print to see the restored variables
for v in tf.get_collection('variables'):
print(v.name)
print(sess.run(tf.global_variables()))
# get saved weights
W = tf.get_collection('variables')[0]
b = tf.get_collection('variables')[1]
# placeholders for test images and labels
x = tf.placeholder(tf.float32, [None, 784],name='x')
y_ = tf.placeholder(tf.float32, [None, 10],name='y_')
# predict equation
y = tf.nn.softmax(tf.matmul(x, W) + b,name='y')
# compare predicted label and actual label
correct_prediction = tf.equal(tf.argmax(y,1), tf.argmax(y_,1))
# accuracy op
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
accu=sess.run(accuracy, feed_dict={x: mnist.test.images, y_: mnist.test.labels})
print(accu)
/home/ganesh/tensorflow/bin/python3.5 "/home/ganesh/Desktop/Tutorial code/MNIST_SOFTMAX/mnist_test.py"
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE3 instructions, but these are available on your machine and could speed up
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could speed
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine and could speed up
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions, but these are available on your machine and could speed up
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions, but these are available on your machine and could speed up
Extracting MNIST_data/train-images-idx3-ubyte.gz
Extracting MNIST_data/train-labels-idx1-ubyte.gz
Extracting MNIST_data/t10k-images-idx3-ubyte.gz
Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
W:0
b:0
W:0
b:0
[array([[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.],
...,
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.]], dtype=float32), array([-0.11976091, 0.34987807, -0.06085325, -0.11957281, 0.14137113,
0.35481071, 0.03694239, 0.29374427, -0.73166603, -0.14489292], dtype=float32), array([[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.],
...,
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.],
[0., 0., 0., ..., 0., 0., 0.]], dtype=float32), array([-0.11976091, 0.34987807, -0.06085325, -0.11957281, 0.14137113,
0.35481071, 0.03694239, 0.29374427, -0.73166603, -0.14489292], dtype=float32)]
0.9065
Process finished with exit code 0


File Edit View Navigate Code Refactor Run Tools VCS Window Help
MNIST_SOFTMAX data
mnist_train
Project
mnist_train.py
mnist_test.py
MNIST_SOFTMAX ~/Desktop/Tutorial c
MNIST_data
t10k-images-idx3-ubyte.gz
t10k-labels-idx1-ubyte.gz
train-images-idx3-ubyte.gz
train-labels-idx1-ubyte.gz
data
mnist_test.py
mnist_train.py
External Libraries
import tensorflow as tf
from tensorflow.contrib.session_bundle import exporter
# Import MNIST data
from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
sess = tf.Session()
tf.logging.set_verbosity(tf.logging.INFO)
x = tf.placeholder(tf.float32, [None, 784],name='x')
W = tf.Variable(tf.zeros([784, 10]),name='W')
b = tf.Variable(tf.zeros([10]),name='b')
y = tf.nn.softmax(tf.matmul(x, W) + b,name='y')
y_ = tf.placeholder(tf.float32, [None, 10],name='y_')
tf.add_to_collection('variables',W)
Run mnist_train
/home/ganesh/tensorflow/bin/python3.5 "/home/ganesh/Desktop/Tutorial code/MNIST_SOFTMAX/mnist_train.py"
Extracting MNIST_data/train-images-idx3-ubyte.gz
Extracting MNIST_data/train-labels-idx1-ubyte.gz
Extracting MNIST_data/t10k-images-idx3-ubyte.gz
Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE3 instructions, but these are available on your machine and could speed u
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could speed
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine and could speed up
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions, but these are available on your machine and could speed u
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions, but these are available on your machine and could speed up
Exporting trained model to data/mnist_model
INFO:tensorflow:data/mnist_model/00000001-tmp/export is not in all_model_checkpoint_paths. Manually adding it.
Process finished with exit code 0
Match Case
Regex
Words
mnist_model
00000001
logs
W = tf.Variable(tf.zeros([784, 10]),name='W')
b = tf.Variable(tf.zeros([10]),name='b')
y = tf.nn.softmax(tf.matmul(x, W) + b,name='y')
y_ = tf.placeholder(tf.float32, [None, 10],name='y_')
tf.add_to_collection('variables',W)
tf.add_to_collection('variables',b)
cross_entropy = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits=y, labels=y_))
train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)
# save summaries for visualization
tf.summary.histogram('weights', W)
tf.summary.histogram('max_weight', tf.reduce_max(W))
tf.summary.histogram('bias', b)
tf.summary.scalar('cross_entropy', cross_entropy)
tf.summary.histogram('cross_hist', cross_entropy)
# merge all summaries into one op
merged=tf.summary.merge_all()
trainwriter=tf.summary.FileWriter('data/mnist_model'+'/logs/train',sess.graph)
init = tf.global_variables_initializer()
sess.run(init)
for i in range(1000):
batch_xs, batch_ys = mnist.train.next_batch(100)
summary, _ = sess.run([merged, train_step], feed_dict={x: batch_xs, y_: batch_ys})
trainwriter.add_summary(summary, i)
# model export path
export_path = 'data/mnist_model'
print('Exporting trained model to', export_path)
#
saver = tf.train.Saver(sharded=True)
model_exporter = exporter.Exporter(saver)
Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE3 instructions,
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, b
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions,
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions, b
Exporting trained model to data/mnist_model
INFO:tensorflow:data/mnist_model/00000001-tmp/export is not in all_model_checkpoint_paths. Manually adding it.
Process finished with exit code 0

References:

https://github.com/googlevr/gvr-android-sdk