

# **CLASSIFICATION OF BRAIN CANCER IMAGES USING BIGDL**

**A PROJECT REPORT**

Submitted by

DHRUBANKA DUTTA(17BCE1019)

ARNAV TRIPATHY(17BCE1026)

AMAN SHAH(17BCE1221)

SHIKAR BHARADWAJ(17BCE1250)

**B.Tech**

Computer Science and Engineering



**VIT<sup>®</sup>**

**Vellore Institute of Technology**

(Deemed to be University under section 3 of UGC Act, 1956)

**School of Computing Science and Engineering**

# ACKNOWLEDGEMENT

We would like to express my special thanks of gratitude to my professor Prof. Ramesh Ragala, who gave me this opportunity to do this project of Large Scale Data Processing on “CLASSIFICATION OF BRAIN CANCER IMAGES USING BIGDL”, who also helped me in the completion of my project. I came to know about many new things and I’m really thankful to them.

I would also like to thank my friends who helped me a lot in finalizing this project within the limited time frame.

Dhrubanka Dutta  
17BCE1019

# CONTENTS

Sr.no	Topic	Page Number
1	Acknowledgement	2
2	Introduction	4
3	Overview	5
4	Required Modules	6
5	Implementation of modules	19
6	Result	28
7	Outputs	29
8	Conclusion	34

# INTRODUCTION

**Classification of brain cancer images** is important in the medical field for quick evaluation and fast reports. The identification, segmentation and detection of infecting area in brain tumor MRI images are a tedious and time-consuming task. This becomes a problem when there are large number of scanned images to go through. Here a solution is provided in form of BigDL which is a platform for processing large amount of data in deep learning.

**The dataset** consists of around 4700 brain cancer images that will be classified as brain having tumor or not.

The Cancer Genome Atlas Sarcoma (TCGA-SARC) data collection is part of a larger effort to build a research community focused on connecting cancer phenotypes to genotypes by providing clinical images matched to subjects from The Cancer Genome Atlas (TCGA). Clinical, genetic, and pathological data resides in the Genomic Data Commons (GDC) Data Portal while the radiological data is stored on The Cancer Imaging Archive (TCIA).

Matched TCGA patient identifiers allow researchers to explore the TCGA/TCIA databases for correlations between tissue genotype, radiological phenotype and patient outcomes. Tissues for TCGA were collected from many sites all over the world in order to reach their accrual targets, usually around 500 specimens per cancer type. For this reason the image data sets are also extremely heterogeneous in terms of scanner modalities, manufacturers and acquisition protocols. In most cases the images were acquired as part of routine care and not as part of a controlled research study or clinical trial.

**Hadoop** is an open-source software utility that facilitates using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce

**Apache Spark** is an open-source distributed general-purpose cluster-computing framework. Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. Apache Spark has as its architectural foundation the resilient distributed dataset (RDD), a read-only multiset of data items distributed over a cluster of machines, that is maintained in a fault-tolerant way. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework.

**PySpark** is a python API for spark released by Apache Spark community to support python with Spark. Using PySpark, one can easily integrate and work with RDD in python programming language too. There are numerous features that make PySpark such an amazing framework when it comes to working with huge datasets.

**BigDL** is a distributed deep learning framework for Apache Spark, created by Jason Dai at Intel. BigDL is a distributed deep learning library for Apache Spark; with BigDL, users can write their deep learning applications as standard Spark programs, which can directly run on top of existing Spark or Hadoop clusters

**Tensorflow Inception v3** is a widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batchnorm is used extensively throughout the model and applied to activation inputs. Loss is computed via Softmax. Here this model is retrained using a separate dataset to be able to classify

**OpenCV** is a library of programming functions mainly aimed at real-time computer vision. This module will be mainly used to support image transformations necessary for executing tensorflow code.

## OVERVIEW

The main aim of this projects is to work through a large dataset and perform deep learning on the dataset. The dataset consists of more than four thousand images. These are dicom images that contain a lot of valuable information layered on top of the actual image. These underlying images are extracted and uploaded to the hadoop file system. These are uploaded in bytes fomat to preserve the content of the image. Other formats such as jpeg and png in hdfs leads to data loss and therefore corruption of dataset.

A classifier model is trained using google inception v3 model to classify brain scans as brain having a tumor or not having a tumor. The datatset used to train this model is a separate dataset acquires from kaggle which had divided images into yes or no. The model loads the trained weights and forms a graph from it and passes the images through the graph. The graph then extracts features of the image and classifies the image as yes or no.

# **REQUIRED MODULES**

## **Hadoop -**

Apache Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model.

## **Installation -**

To check, use command - \$java -version

```
aman@ubuntu:~$ java -version
openjdk version "11.0.4" 2019-07-16
OpenJDK Runtime Environment (build 11.0.4+11-post-Ubuntu-1u
buntu218.04.3)
OpenJDK 64-Bit Server VM (build 11.0.4+11-post-Ubuntu-1ubun
tu218.04.3, mixed mode, sharing)
aman@ubuntu:~$
```

- 1) Download Hadoop file using terminal or from website
- 2) Use command \$ tar xzf hadoop-2.7.3.tar.gz to extract file

```
aman@ubuntu:~$ su
Password:
root@ubuntu:/home/aman# tar xzf hadoop-2.7.3.tar.gz
root@ubuntu:/home/aman# mv hadoop-2.7.3 hadoop/
root@ubuntu:/home/aman# exit
exit
```

- 3) Install Hadoop in Pseudo Distributed Mode
- 4) Move Hadoop folder to usr/local/hadoop
- 5) Set Hadoop Environment variables by adding commands to ~/.bashrc file
- 6) Use \$ gedit ~/.bashrc to edit the file

```
aman@ubuntu:~$ gedit ~/.bashrc
```

- 7) Add the following commands

```
export HADOOP_HOME=/usr/local/hadoop  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export YARN_HOME=$HADOOP_HOME  
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native  
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin  
export HADOOP_INSTALL=$HADOOP_HOME
```

8) Apply changes using \$ source ~/.bashrc

9) Check Hadoop Installation

```
aman@ubuntu:/usr/local/jdk-12.0.2$ hadoop version  
Hadoop 2.7.3  
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb  
92be5982de4719c1c8af91ccff  
Compiled by root on 2016-08-18T01:41Z  
Compiled with protoc 2.5.0  
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4  
This command was run using /usr/local/hadoop-2.7.3/share/hadoop/common/hadoop-c  
ommon-2.7.3.jar
```

10) Use \$ cd \$HADOOP\_HOME/etc/hadoop

11) Edit the Hadoop-env.sh file and replace JAVA\_HOME value by export

12) JAVA\_HOME=/usr/local/jdk12.0.2

13) Edit the core-site.xml

The screenshot shows a text editor window titled "core-site.xml" located at "/usr/local/hadoop-2.7.3/etc/hadoop". The file contains the Apache License text and a configuration section:

```
Unless required by applicable law or agreed to in
writing, software
distributed under the License is distributed on an "AS
IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied.
See the License for the specific language governing
permissions and
limitations under the License. See accompanying LICENSE
file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

Below the code, the status bar shows "XML" and "Tab Width: 8".

14) Edit hdfs-site.xml

The screenshot shows a text editor window titled "hdfs-site.xml" located at "/usr/local/hadoop-2.7.3/etc/hadoop". The file contains the following configuration:

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>

  <property>
    <name>dfs.name.dir</name>
    <value>file:///home/aman/hadoopinfra/hdfs/namenode </
value>
  </property>

  <property>
    <name>dfs.data.dir</name>
    <value>file:///home/aman/hadoopinfra/hdfs/datanode </
value>
  </property>
</configuration>
```

Below the code, the status bar shows "XML" and "Tab Width: 8".

15) Edit yarn-site.xml

The screenshot shows a text editor window with the title bar "yarn-site.xml" and the path "/usr/local/hadoop-2.7.3/etc/hadoop". The content of the file is as follows:

```
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in
writing, software
distributed under the License is distributed on an "AS
IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied.
See the License for the specific language governing
permissions and
limitations under the License. See accompanying LICENSE
file.
-->
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
</configuration>
```

At the bottom of the editor, there are status indicators: "XML ▾ Tab Width: 8 ▾ Ln 19, Col 15 ▾ INS".

## 16) Edit Mapred-site.xml

The screenshot shows a text editor window with the title bar "mapred-site.xml [Read-Only]" and the path "/usr/local/hadoop-2.7.3/etc/hadoop". The content of the file is as follows:

```
Unless required by applicable law or agreed to in
writing, software
distributed under the License is distributed on an "AS
IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied.
See the License for the specific language governing
permissions and
limitations under the License. See accompanying LICENSE
file.
-->

<!-- Put site-specific property overrides in this file. --&gt;

&lt;configuration&gt;
  &lt;property&gt;
    &lt;name&gt;mapreduce.framework.name&lt;/name&gt;
    &lt;value&gt;yarn&lt;/value&gt;
  &lt;/property&gt;
&lt;/configuration&gt;</pre>

At the bottom of the editor, there are status indicators: "XML ▾ Tab Width: 8 ▾ Ln 9, Col 1 ▾ INS".


```

17) Setup namenode using \$ hdfs namenode -format

```
aman@ubuntu:/usr/local/hadoop-2.7.3/etc/hadoop$ cd ~
aman@ubuntu:~$ hdfs namenode -format
19/09/26 11:07:58 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = ubuntu/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.7.3
*****
```

```
19/09/26 11:08:03 INFO common.Storage: Storage directory /home/aman/hadoopinfra
/hdfs/namenode has been successfully formatted.
19/09/26 11:08:03 INFO namenode.FSImageFormatProtobuf: Saving image file /home/
aman/hadoopinfra/hdfs/namenode/current/fsimage.ckpt_00000000000000000000 using n
o compression
19/09/26 11:08:03 INFO namenode.FSImageFormatProtobuf: Image file /home/aman/ha
doopinfra/hdfs/namenode/current/fsimage.ckpt_00000000000000000000 of size 351 by
tes saved in 0 seconds.
19/09/26 11:08:03 INFO namenode.NNStorageRetentionManager: Going to retain 1 im
ages with txid >= 0
19/09/26 11:08:03 INFO util.ExitUtil: Exiting with status 0
19/09/26 11:08:03 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at ubuntu/127.0.1.1
*****
```

## Apache Spark –

Apache Spark is an open-source distributed general-purpose cluster-computing framework. Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. Apache Spark has as its architectural foundation the resilient distributed dataset (RDD), a read-only multiset of data items distributed over a cluster of machines, that is maintained in a fault-tolerant way. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework.

## Installation -

In manjaro arch-linux, apache spark is directly available as a package in the package manager.

All    Installed    Pending    Updates

Sort by: Relevance ▾

Search: apache Spark

Installed	<b>apache-spark</b> fast and general engine for large-scale data processing	2.4.3-1	<b>Build</b> ➔
Repositories	<b>apache-zipperlin</b> Data analytics and visualization notebook with backends of Spark, Hadoop, SQL and more	0.8.1-1	<b>Build</b> ➔
AUR	<b>apache-spark-git</b> fast and general engine for large-scale data processing	2.4.0.SNAPSHOT.20180817.22533-1	<b>Build</b> ➔

1 pending operation

Cancel    Apply ➔

```

< 
Building apache-spark...
==> Making package: apache-spark 2.4.3-1 (Wednesday 25 September 2019 04:16:35 PM IST)
==> Checking runtime dependencies...
==> Checking buildtime dependencies...
==> Retrieving sources...
-> Found spark-2.4.3-bin-hadoop2.7.tgz
-> Found apache-spark-master.service
-> Found apache-spark-slave@.service
-> Found spark-env.sh
-> Found spark-daemon-run.sh
-> Found run-master.sh
-> Found run-slave.sh
==> Validating source files with shalsums...
spark-2.4.3-bin-hadoop2.7.tgz ... Passed
apache-spark-master.service ... Passed
apache-spark-slave@.service ... Passed
spark-env.sh ... Passed
spark-daemon-run.sh ... Passed
run-master.sh ... Passed
run-slave.sh ... Passed
==> Removing existing $srcdir/ directory...
==> Extracting sources...
-> Extracting spark-2.4.3-bin-hadoop2.7.tgz with bsdtar
==> Starting prepare()...
==> Entering fakeroot environment...
==> Starting package()...
==> Pre-linking installed packages...
-> Removing libtool files...
-> Purging unwanted files...
-> Removing static library files...
-> Stripping unneeded symbols from binaries and libraries...
-> Compressing man and info pages...
==> Checking for packaging issues...
==> Creating package "apache-spark"...
-> Generating _PKGINFO file...
-> Generating _BUILDINFO file...
-> Adding install file...
-> Generating _MTREE file...
-> Compressing package...
==> Leaving fakeroot environment.
==> Finished making: apache-spark 2.4.3-1 (Wednesday 25 September 2019 04:19:33 PM IST)
==> Cleaning up...

Resolving dependencies...
Checking inter-conflicts...
Running post-transaction hooks...
Checking keyring...
Checking integrity...
Loading packages files...
Checking file conflicts...
Checking available disk space...
Installing apache-spark (2.4.3-1)...
Running post-transaction hooks...
Reloading system manager configuration...

```

Cancel    Apply ➔

```

dhrubanka@dhrubanka-pc:~$ spark-shell
/usr/local/hadoop/bin/hadoop
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/apache-spark/jars/slf4j-log4j12-1.7.16.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.16.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
19/09/25 16:28:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkK, use setLogLevel(newLevel).
Spark context Web UI available at http://localhost:4040
spark context available as 'sc' (master = local[*], app id = local-1569408662285).
spark session available as 'spark'.
Welcome to

    / \ \
   /   \ \
  /     \ \
 /       \ \
/         \ \
version 2.4.3

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_222)
Type in expressions to have them evaluated.
Type :help for more information.

scala> 
```

## PySpark -

PySpark is a python API for spark released by Apache Spark community to support python with Spark. Using PySpark, one can easily integrate and work with RDD in python programming language too. There are numerous features that make PySpark such an amazing framework when it comes to working with huge datasets.

## Installation -

The package pyspark can be installed using command, pip install pyspark –user.

```
dhrubanka@dhrubanka-pc ~]$ pip install pyspark
Collecting pyspark
  Requirement already satisfied: py4j==0.10.7 in ./local/lib/python3.7/site-packages (from pyspark) (0.10.7)
Installing collected packages: pyspark
Successfully installed pyspark-2.4.4
dhrubanka@dhrubanka-pc ~]$ python
Python 3.7.4 (default, Jul 16 2019, 07:12:58)
[GCC 9.1.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyspark
>>> pyspark.__version__
'2.4.4'
>>>
```

### **BigDL -**

BigDL is a distributed deep learning framework for Apache Spark, created by Jason Dai at Intel. igDL is a distributed deep learning library for Apache Spark; with BigDL, users can write their deep learning applications as standard Spark programs, which can directly run on top of existing Spark or Hadoop clusters

```
dhrubanka@dhrubanka:~
```

```
Collecting BigDL
  Using cached https://files.pythonhosted.org/packages/40/ca/aa8071309d68e88879cd45520f8a18a577341fc30e2ef0760854fb5c6b/BigDL-0.9.0-py2.py3-none-manylinux1_x86_64.whl
Requirement already satisfied: numpy>=1.7 in /usr/lib/python3.7/site-packages (from BigDL) (1.17.0)
Requirement already satisfied: six>=1.10.0 in /usr/lib/python3.7/site-packages (from BigDL) (1.12.0)
Requirement already satisfied: pyspark>=2.2 in ./local/lib/python3.7/site-packages (from BigDL) (2.4.4)
Requirement already satisfied: py4j==0.10.7 in ./local/lib/python3.7/site-packages (from pyspark>=2.2>BigDL) (0.10.7)
Installing collected packages: BigDL
Successfully installed BigDL-0.9.0
dhrubanka@dhrubanka:~
```

```
python 3.7.4 (default, Jul 16 2019, 07:12:58)
[GCC 9.1.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import bigdl
/home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/util/engine.py:41: UserWarning: Find both SPARK_HOME and pyspark. You may need to check whether they match with each other. SPARK_HOME environment variable is set to: /opt/apache-spark, and pyspark is found in /home/dhrubanka/.local/lib/python3.7/site-packages/pyspark/_init__.py. If they are unmatched, please use one source only to avoid conflict. For example, you can unset SPARK_HOME and use pyspark only.
  warnings.warn(warning_msg)
Prepending /home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/share/conf/spark-bigdl.conf to sys.path
>>> bigdl
<module 'bigdl' from '/home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/_init__.py'>
>>>
```

Installation - The package bigdl can be installed using command, pip install bigdl –user.

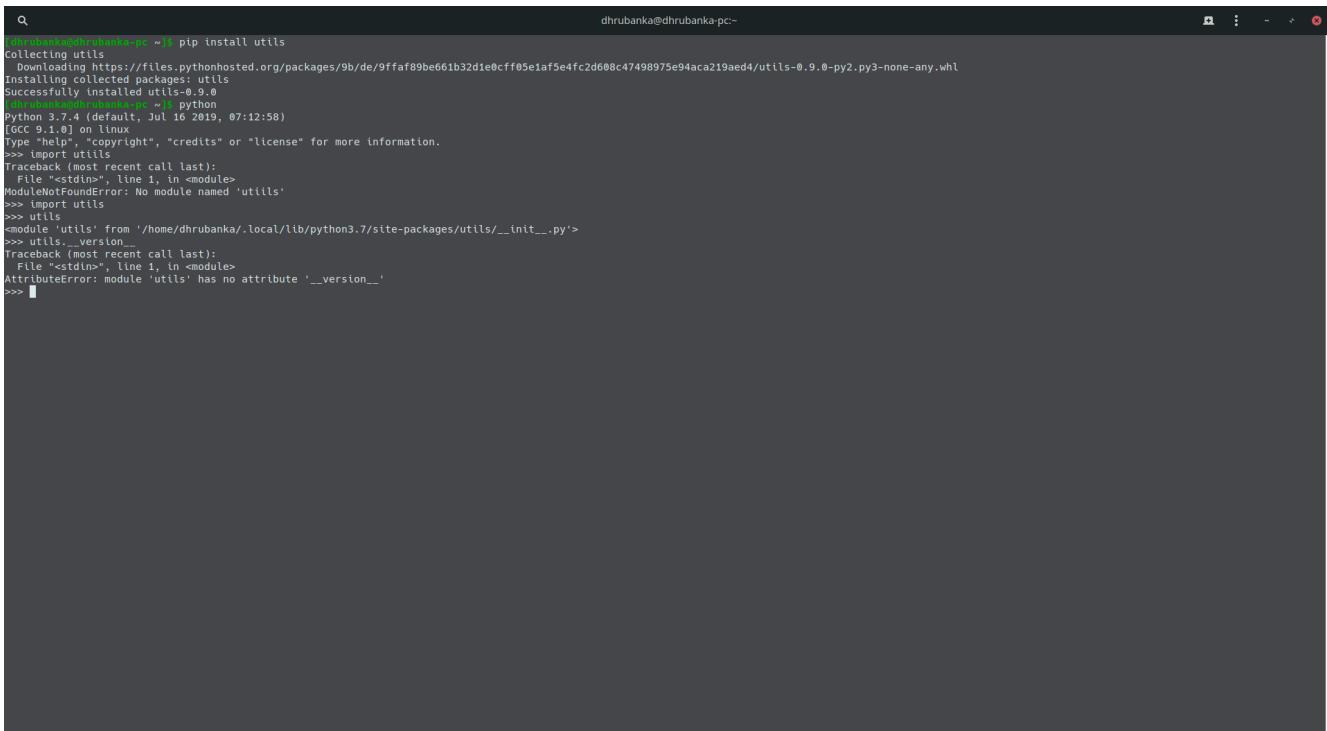
### Utils -

#### Installation -

The package utils can be installed using command, pip install utils –user.

```
dhrubanka@dhrubanka:~
```

```
Collecting git+https://github.com/micahhauser/python3-utils.git --user
  Cloning git://github.com/micahhauser/python3-utils.git to /tmp/pip-req-build-wkdtaffu
  Running command git clone -q git://github.com/micahhauser/python3-utils.git /tmp/pip-req-build-wkdtaffu
Requirement already satisfied: six==1.8.0 in /usr/lib/python3.7/site-packages (from python3-utils==0.4.0) (1.12.0)
Building wheels for collected packages: python3-utils
  Building wheel for python3-utils (setup.py) ... done
  Created wheel for python3-utils: filename=python3-utils-0.4.0-py2.py3-none-any.whl size=5146 sha256=bfb599258b71aa546a68d631406291a2bda6606b0925859f880503b4aa01b6716
  Stored in directory: /tmp/pip-ephem-wheel-cache-7ricwyl/wheels/85/63/29/1184b994a921546121132296aa0961f1f7384ea9bafc9b97f
Successfully built python3-utils
Installing collected packages: python3-utils
Successfully installed python3-utils-0.4.0
>>> import python3_utils
Python 3.7.4 (default, Jul 16 2019, 07:12:58)
[GCC 9.1.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import python3_utils
>>> python3_utils.__version__
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
AttributeError: module 'python3_utils' has no attribute '__version__'
>>>
```

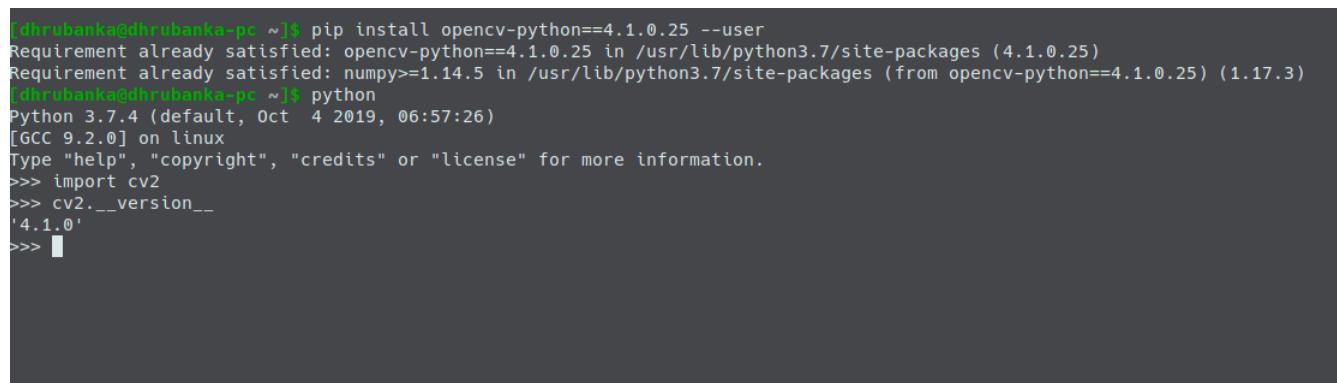


```
dhrubanka@dhrubanka-pc:~$ pip install utils
Collecting utils
  Downloading https://files.pythonhosted.org/packages/9b/de/9ffaf89be661b32d1e0cff05e1af5e4fc2d608c47498975e94aca219aed4/utils-0.9.0-py2.py3-none-any.whl
Successfully installed utils-0.9.0
dhrubanka@dhrubanka-pc:~$ python
Python 3.7.4 (default, Jul 16 2019, 07:12:58)
[GCC 9.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import utils
>>> utils
<module 'utils' from '/home/dhrubanka/.local/lib/python3.7/site-packages/utils/__init__.py'>
>>> utils.__version__
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'utils'
>>> import utils
>>> utils
<module 'utils' from '/home/dhrubanka/.local/lib/python3.7/site-packages/utils/__init__.py'>
>>> utils.__version__
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: module 'utils' has no attribute '__version__'
>>> 
```

## OpenCV:

OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source BSD license.

It can be installed using `pip install opencv-python=="version" --user`



```
dhrubanka@dhrubanka-pc:~$ pip install opencv-python==4.1.0.25 --user
Requirement already satisfied: opencv-python==4.1.0.25 in /usr/lib/python3.7/site-packages (4.1.0.25)
Requirement already satisfied: numpy>=1.14.5 in /usr/lib/python3.7/site-packages (from opencv-python==4.1.0.25) (1.17.3)
dhrubanka@dhrubanka-pc:~$ python
Python 3.7.4 (default, Oct  4 2019, 06:57:26)
[GCC 9.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.1.0'
>>> 
```

## Tensorflow:

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.

It can be installed using pip install tensorflow=="version" --user

```
[dhrubanka@localhost ~]$ pip install tensorflow==1.14.0
Requirement already satisfied: tensorflow==1.14.0 in ./local/lib/python3.7/site-packages (1.14.0)
Requirement already satisfied: grpcio>=1.8.6 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (1.24.3)
Requirement already satisfied: gast>=0.2.0 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (0.3.2)
Requirement already satisfied: wheel>=0.26 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (0.33.6)
Requirement already satisfied: astor>=0.6.0 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (0.8.0)
Requirement already satisfied: numpy<2.0,>=1.14.5 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (1.17.3)
Requirement already satisfied: wrapt>=1.11.1 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (1.11.2)
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (1.1.0)
Requirement already satisfied: google-pasta>=0.1.0 in ./local/lib/python3.7/site-packages (from tensorflow==1.14.0) (0.1.7)
Requirement already satisfied: tensorboard<1.15.0,>=1.14.0 in ./local/lib/python3.7/site-packages (from tensorflow==1.14.0) (1.14.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (1.1.0)
Requirement already satisfied: protobuf>=3.6.1 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (3.10.0)
Requirement already satisfied: absl-py>=0.7.0 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (0.8.1)
Requirement already satisfied: six>=1.10.0 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (1.12.0)
Requirement already satisfied: keras-applications>=1.0.6 in /usr/lib/python3.7/site-packages (from tensorflow==1.14.0) (1.0.8)
Requirement already satisfied: tensorflow-estimator<1.15.0rc0,>=1.14.0rc0 in ./local/lib/python3.7/site-packages (from tensorflow==1.14.0) (1.14.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/lib/python3.7/site-packages (from tensorboard<1.15.0,>=1.14.0->tensorflow==1.14.0) (3.1.1)
Requirement already satisfied: werkzeug>=0.11.15 in /usr/lib/python3.7/site-packages (from tensorboard<1.15.0,>=1.14.0->tensorflow==1.14.0) (0.16.0)
Requirement already satisfied: setuptools>=41.0.0 in /usr/lib/python3.7/site-packages (from tensorboard<1.15.0,>=1.14.0->tensorflow==1.14.0) (41.2.0)
Requirement already satisfied: h5py in /usr/lib/python3.7/site-packages (from keras-applications>=1.0.6->tensorflow==1.14.0) (2.9.0)
```

```
[dhrubanka@localhost ~]$ python
Python 3.7.4 (default, Oct  4 2019, 06:57:26)
[GCC 9.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:516: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint8 = np.dtype([('qint8', np.int8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:517: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_quint8 = np.dtype([('quint8', np.uint8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:518: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint16 = np.dtype([('qint16', np.int16, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:519: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_quint16 = np.dtype([('quint16', np.uint16, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:520: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint32 = np.dtype([('qint32', np.int32, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:525: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_resource = np.dtype([('resource', np.ubyte, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:541: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint8 = np.dtype([('qint8', np.int8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:542: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_quint8 = np.dtype([('quint8', np.uint8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:543: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint16 = np.dtype([('qint16', np.int16, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:544: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_quint16 = np.dtype([('quint16', np.uint16, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:545: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_qint32 = np.dtype([('qint32', np.int32, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:550: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
    _np_resource = np.dtype([('resource', np.ubyte, 1)])
>>> tf._version_
'1.14.0'
>>> █
```

## Example Code :

This is an example code to run linear regression on the BigDL platform.

The screenshot shows a Jupyter notebook interface with the title "Linear Regression". The notebook content includes:

```
In [1]: import matplotlib
matplotlib.use('Agg')
%pylab inline
import pandas
import datetime as dt

from bigdl.nn.layer import *
from bigdl.nn.criterion import *
from bigdl.optim.optimizer import *
from bigdl.util.common import *
from bigdl.dataset.sample import Sample
import matplotlib.pyplot as plt
from bigdl.dataset.transformer import *
from matplotlib.pyplot import imshow
from pyspark import SparkContext
sc=SparkContext.getOrCreate(conf=create_spark_conf().setMaster("local[4]").set("spark.driver.memory","2g"))

init_engine()

Populating the interactive namespace from numpy and matplotlib

/home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/util/engine.py:41: UserWarning: Find both SPARK_HOME and
pyspark. You may need to check whether they match with each other. SPARK_HOME environment variable is set to: /opt
/apache-spark, and pyspark is found in: /home/dhrubanka/.local/lib/python3.7/site-packages/pyspark/_init_.py. If
they are unmatched, please use one source only to avoid conflict. For example, you can unset SPARK_HOME and use
pyspark only.
warnings.warn(warning_msg)

Prepending /home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/share/conf/spark-bigdl.conf to sys.path
```

**1. Generate random training dataset**  
Generate a random dataset for training which consists of 100 examples of dimensions [2x1]

```
In [2]: FEATURES_DIM = 2
data_len = 100

def gen_rand_sample():
    features = np.random.uniform(0, 1, (FEATURES_DIM))
    label = (2 * features).sum() + 0.4
    return Sample.from_ndarray(features, label)
```

The screenshot shows a Jupyter notebook interface with the title "1. Generate random training dataset". The notebook content includes:

```
In [2]: FEATURES_DIM = 2
data_len = 100

def gen_rand_sample():
    features = np.random.uniform(0, 1, (FEATURES_DIM))
    label = (2 * features).sum() + 0.4
    return Sample.from_ndarray(features, label)

rdd_train = sc.parallelize(range(0, data_len)).map( lambda i: gen_rand_sample() )
```

**2. Hyperparameter Setup**  
Specify the necessary parameters and construct a linear regression model using BigDL. batch\_size should be divisible by number of cores being used.

```
In [3]: Parameters
learning_rate = 0.2
training_epochs = 5
batch_size = 4
n_input = FEATURES_DIM
n_output = 1

def linear_regression(n_input, n_output):
    # Initialize a sequential container
    model = Sequential()
    # Add a linear layer
    model.add(Linear(n_input, n_output))

    return model

model = linear_regression(n_input, n_output)
creating: createSequential
creating: createLinear
```

**3. Optimizer setup and training**  
Here we construct the optimizer to optimize the linear regression problem by using Stochastic gradient descent(SGD) to update the model weights. You can specify your own learning rate in SGD() method. Also, you can replace the SGD() with other optimizer such like Adam().

```
In [4]: # Create an Optimizer
```

**3. Optimizer setup and training**

Here we construct the optimizer to optimize the linear regression problem by using Stochastic gradient descent(SGD) to update the model weights. You can specify your own learning rate in SGD() method. Also, you can replace the SGD() with other optimizer such like Adam.

```
In [4]: # Create an Optimizer
optimizer = Optimizer(
    mode="train",
    training=rdd_train,
    criterion=MSECriterion(),
    optim_method=SGD(learningrate=learning_rate),
    end_trigger=MaxEpoch(training_epochs),
    batch_size=batch_size)

creating: createMSECriterion
creating: createDefault
creating: createSGD
creating: createMaxEpoch
creating: createDistributedOptimizer
```

```
In [5]: # Start to train
trained_model = optimizer.optimize()
```

**4. Prediction on training data**

```
In [6]: # Print the first five predicted results of training data.
predict_result = trained_model.predict(rdd_train)
p = predict_result.take(5)

print("predict predict: \n")
for i in p:
    print(str(i) + "\n")
```

```
predict predict:
[3.369568]
[4.1339116]
[2.540867]
[3.773341]
[1.9410255]
```

```
p = predict_result.take(5)

print("predict predict: \n")
for i in p:
    print(str(i) + "\n")
```

**5. Model evaluation on random test data**

```
In [10]: def test_predict(trained_model):
    np.random.seed(100)
    total_length = 1000
    features = np.random.uniform(0, 1, (total_length, 2))
    label = (features.sum(axis=1) > 0.4)
    predict_data = sc.parallelize(range(0, total_length)).map(
        lambda i: Sample.from_ndarray(features[i], label))

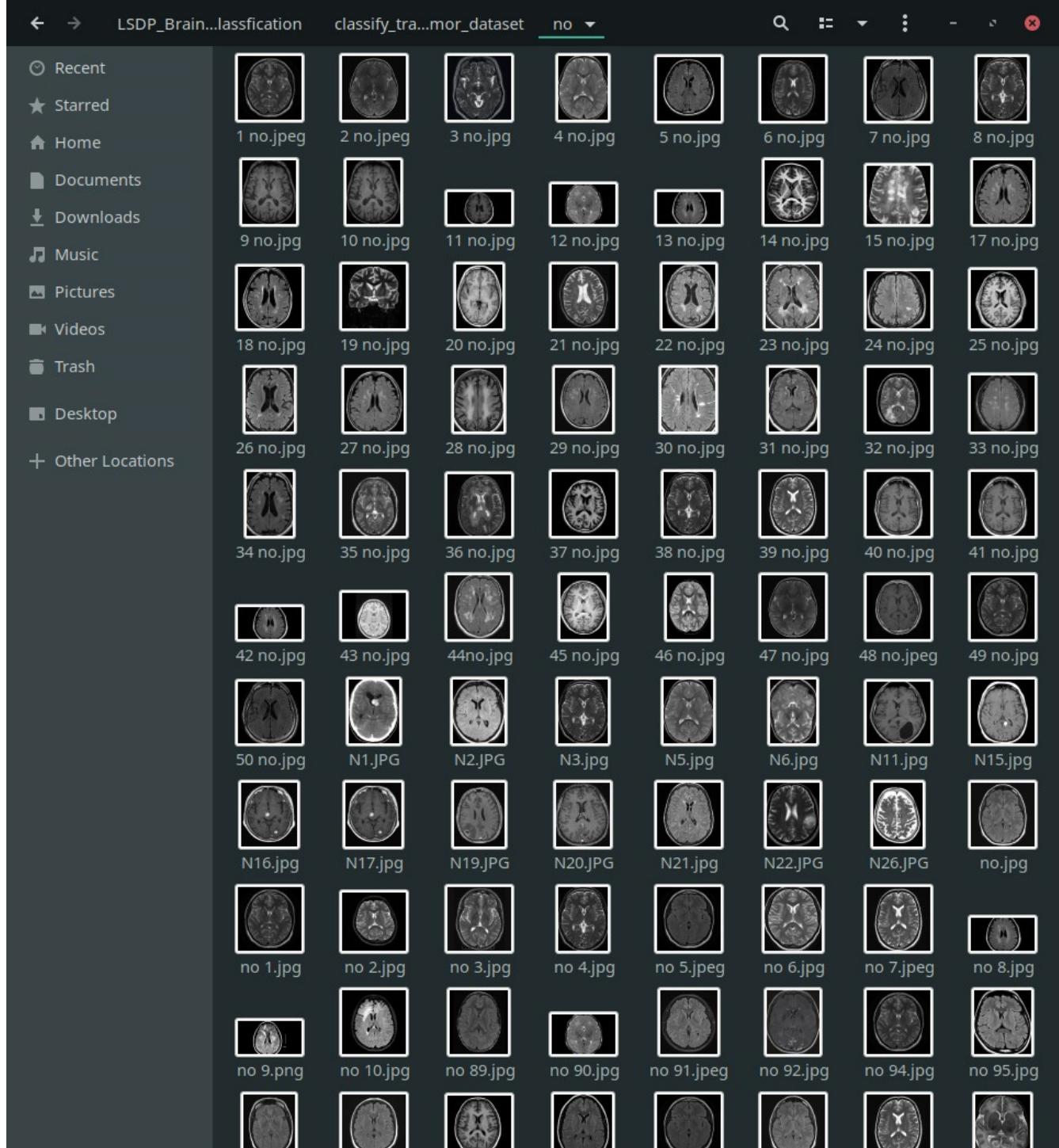
    predict_result = trained_model.predict(predict_data)
    p = predict_result.take(6)
    ground_label = np.array([-0.47596836, [-0.37598032, [+0.00492062,
        [-0.5986958, [-0.12307882, [-0.77907401]], dtype="float32"])

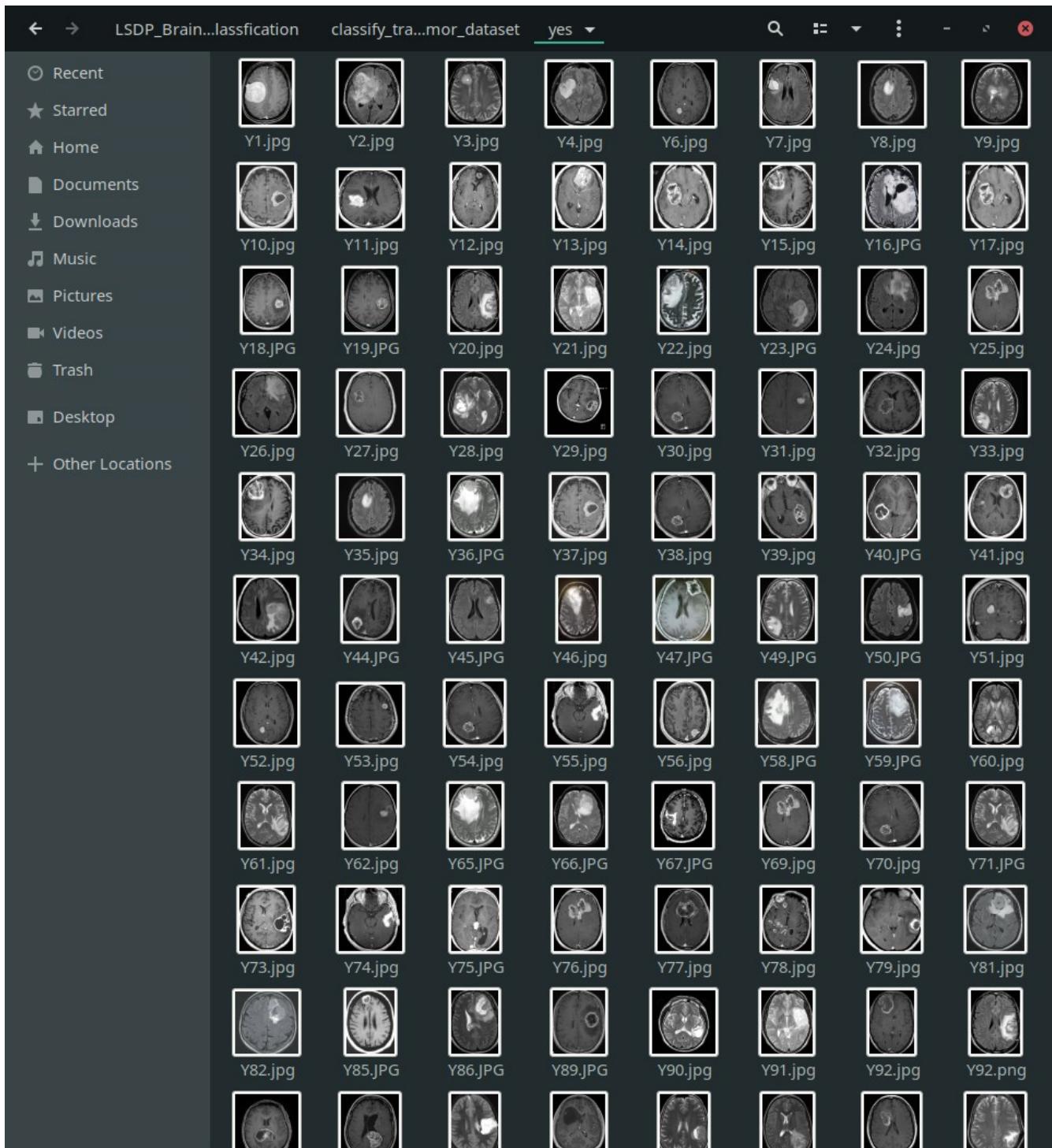
    mse = ((p - ground_label) ** 2).mean()
    print(mse)

test_predict(trained_model)
8.158284
```

# IMPLEMENTATION

Form dataset for classifier :





## The classifier is trained using inception v3

```
dhurbanka@dhrubanka-pc:~/Image-classification-transfer-learning
  Image-classification-transfer-learning] python retrnain.py --image_dir brain_tumor_dataset/
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:516: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_int8 = np.dtype([(“int8”, np.int8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:517: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_uint8 = np.dtype([(“uint8”, np.uint8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:518: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_int16 = np.dtype([(“int16”, np.int16, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:519: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_uint16 = np.dtype([(“uint16”, np.uint16, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:520: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_int32 = np.dtype([(“int32”, np.int32, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:525: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_int64 = np.dtype([(“int64”, np.int64, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:541: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_resource = np.dtype([(“resource”, np.ubyte, 1)])
  _np_int8 = np.dtype([(“int8”, np.int8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:542: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_int16 = np.dtype([(“int16”, np.int16, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:543: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_int32 = np.dtype([(“int32”, np.int32, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:550: FutureWarning: Passing (type, 1) or 'ittype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_int64 = np.dtype([(“int64”, np.int64, 1)])
WARNING:tensorflow:From retrnain.py:1328: The name tf.app.run is deprecated. Please use tf.compat.v1.app.run instead.
WARNING:tensorflow:From retrnain.py:972: The name tf.logging.set_verbosity is deprecated. Please use tf.compat.v1.logging.set_verbosity instead.
W1107 20:44:56.705673 139958439134912 deprecation_wrapper.py:119] From retrnain.py:972: The name tf.logging.set_verbosity is deprecated. Please use tf.compat.v1.logging.set_verbosity instead.
WARNING:tensorflow:From retrnain.py:972: The name tf.logging.INFO is deprecated. Please use tf.compat.v1.logging.INFO instead.
W1107 20:44:56.705880 139958439134912 deprecation_wrapper.py:119] From retrnain.py:972: The name tf.logging.INFO is deprecated. Please use tf.compat.v1.logging.INFO instead.
WARNING:tensorflow:From retrnain.py:837: The name tf.gfile.Exists is deprecated. Please use tf.io.gfile.exists instead.
W1107 20:44:56.706059 139958439134912 deprecation_wrapper.py:119] From retrnain.py:837: The name tf.gfile.Exists is deprecated. Please use tf.io.gfile.exists instead.
WARNING:tensorflow:From retrnain.py:839: The name tf.gfile.MakeDirs is deprecated. Please use tf.io.gfile.makedirs instead.
W1107 20:44:56.706226 139958439134912 deprecation_wrapper.py:119] From retrnain.py:839: The name tf.gfile.MakeDirs is deprecated. Please use tf.io.gfile.makedirs instead.
>>> Downloading inception-2015-12-05.tgz 10.9%
```

```
dhurbanka@dhrubanka-pc:~/Image-classification-transfer-learning
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/0 no_94.jpg_inception_v3.txt
I1107 20:48:15.715972 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/0 no_94.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/0 no_2.jpg_inception_v3.txt
I1107 20:48:15.861912 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/0 no_2.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/3 no_no.jpg_inception_v3.txt
I1107 20:48:15.997758 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/32 no_no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/N16.jpg_inception_v3.txt
I1107 20:48:16.000208 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/N16.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/30 no_no.jpg_inception_v3.txt
I1107 20:48:16.351277 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/30 no_no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/6 no_no.jpg_inception_v3.txt
I1107 20:48:16.497386 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/6 no_no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/44no.jpg_inception_v3.txt
I1107 20:48:16.648896 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/44no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/N21.jpg_inception_v3.txt
I1107 20:48:16.848642 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/N21.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/100.jpg_inception_v3.txt
I1107 20:48:17.000208 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/100.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/34 no_no.jpg_inception_v3.txt
I1107 20:48:17.157682 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/34 no_no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/14 no_no.jpg_inception_v3.txt
I1107 20:48:17.299916 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/14 no_no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/100_no.jpg_inception_v3.txt
I1107 20:48:17.447886 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/100_no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/10_no.jpg_inception_v3.txt
I1107 20:48:17.500000 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/10_no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/36 no_no.jpg_inception_v3.txt
I1107 20:48:17.733574 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/36 no_no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/47 no_no.jpg_inception_v3.txt
I1107 20:48:17.874113 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/47 no_no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/no_96.jpg_inception_v3.txt
I1107 20:48:18.013321 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/no_96.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/no_92.jpg_inception_v3.txt
I1107 20:48:18.04458 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/no_92.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/13_no.jpg_inception_v3.txt
I1107 20:48:18.050000 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/13_no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/18_no_no.jpg_inception_v3.txt
I1107 20:48:18.559763 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/18_no_no.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/N021.jpg_inception_v3.txt
I1107 20:48:18.791092 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/N021.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/no_99.jpg_inception_v3.txt
I1107 20:48:20.040588 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/no_99.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/N011.jpg_inception_v3.txt
I1107 20:48:20.250871 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/N011.jpg_inception_v3.txt
INFO:tensorflow:Creating bottleneck at /tmp/bottleneck/no/N014.jpg_inception_v3.txt
I1107 20:48:20.391077 139958439134912 retrnain.py:363] Creating bottleneck at /tmp/bottleneck/no/N014.jpg_inception_v3.txt
```



```
[INFO]:tensorFlow:2019-11-07 20:59:24.434115: Step 7990: Validation accuracy = 96.0% (N=100)
I1107 20:59:24.434119 139958439134912 retrain.py:1102] 2019-11-07 20:59:24.434115: Step 7990: Validation accuracy = 96.0% (N=100)
INFO:tensorflow:2019-11-07 20:59:25.136510: Step 7990: Train accuracy = 100.0%
I1107 20:59:25.136511 139958439134912 retrain.py:1084] 2019-11-07 20:59:25.136510: Step 7990: Train accuracy = 100.0%
INFO:tensorflow:2019-11-07 20:59:25.136717: Step 7990: Cross entropy = 0.012224
I1107 20:59:25.136734 139958439134912 retrain.py:1086] 2019-11-07 20:59:25.136717: Step 7990: Cross entropy = 0.012224
INFO:tensorflow:2019-11-07 20:59:25.211638: Step 7990: Validation accuracy = 96.0% (N=100)
I1107 20:59:25.211701 139958439134912 retrain.py:1102] 2019-11-07 20:59:25.211638: Step 7990: Validation accuracy = 96.0% (N=100)
INFO:tensorflow:2019-11-07 20:59:25.211702: Step 7990: Train accuracy = 100.0%
I1107 20:59:25.211703 139958439134912 retrain.py:1084] 2019-11-07 20:59:25.211702: Step 7990: Train accuracy = 100.0%
INFO:tensorflow:2019-11-07 20:59:25.865462: Step 7999: Cross entropy = 0.010826
I1107 20:59:25.865482 139958439134912 retrain.py:1086] 2019-11-07 20:59:25.865462: Step 7999: Cross entropy = 0.010826
INFO:tensorflow:2019-11-07 20:59:25.940358: Step 7999: Validation accuracy = 98.0% (N=100)
I1107 20:59:25.940423 139958439134912 retrain.py:1102] 2019-11-07 20:59:25.940358: Step 7999: Validation accuracy = 98.0% (N=100)
INFO:tensorflow:Final test accuracy = 91.7% (N=24)
I1107 20:59:26.916956 139958439134912 retrain.py:1120] Final test accuracy = 91.7% (N=24)
WARNING:tensorflow:From retrain.py:829: convert_variables_to_constants (from tensorflow.python.framework.graph_util_impl) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.compat.v1.graph_util.convert_variables_to_constants.
INFO:tensorflow:2019-11-07 20:59:27.025857 139958439134912 deprecation.py:323] From retrain.py:829: convert_variables_to_constants (from tensorflow.python.framework.graph_util_impl) is deprecated and will be removed in a future version.
I1107 20:59:27.025857 139958439134912 deprecation.py:323] From /home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/graph_util_impl.py:270: extract_sub_graph (from tensorflow.python.framework.graph_util_impl) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.compat.v1.graph_util.extract_sub_graph.
INFO:tensorflow:2019-11-07 20:59:27.126499 139958439134912 graph_util_impl.py:311] Froze 2 variables.
INFO:tensorflow:Converted 2 variables to const ops.
I1107 20:59:27.166627 139958439134912 graph_util_impl.py:364] Converted 2 variables to const ops.
    Image-classification-transfer-learning]
```

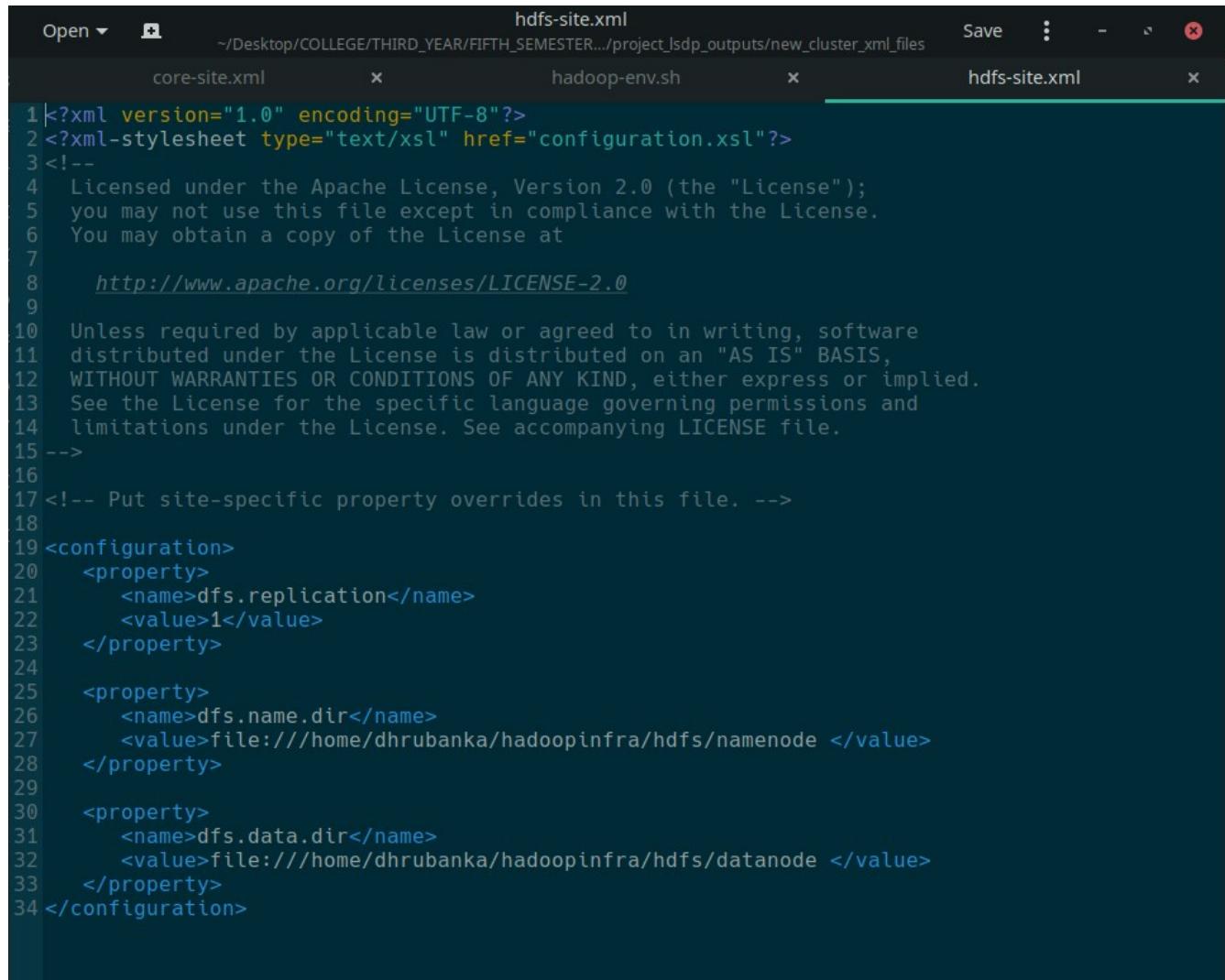
## Forming a cluster:

Add the ips of the machines to be put together in the cluster in the `/etc/hosts` file.

```
GNU nano 4.5                               /etc/hosts                         Modified
127.0.0.1      localhost
127.0.1.1      dhrubanka-pc
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters

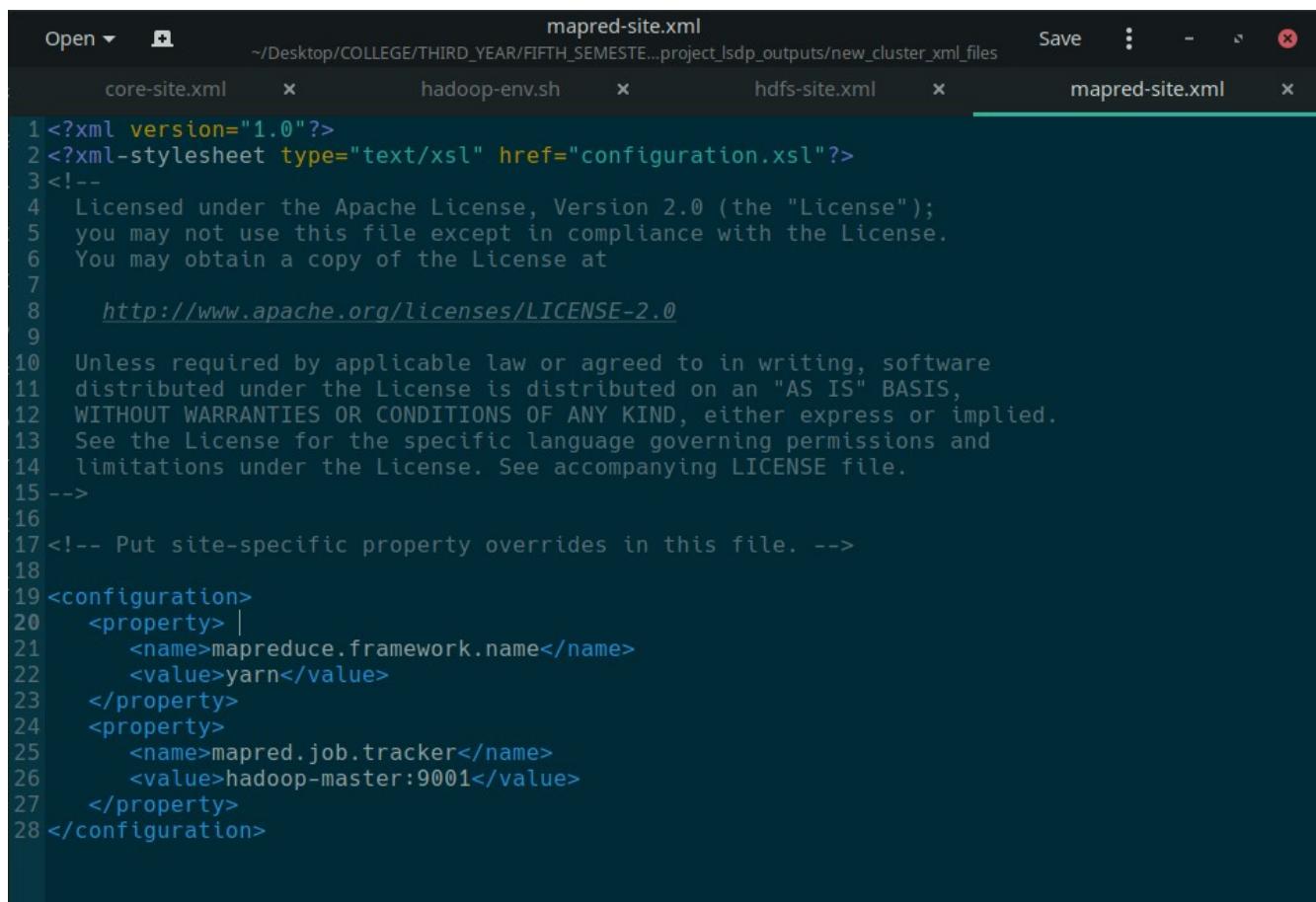
#hadoop cluster
192.168.43.91 hadoop-master
192.168.43.151 hadoop-slave-1
```

Edit hdfs-site.xml as follows:



```
1<?xml version="1.0" encoding="UTF-8"?>
2<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3<!--
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8   http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15-->
16
17<!-- Put site-specific property overrides in this file. -->
18
19<configuration>
20  <property>
21    <name>dfs.replication</name>
22    <value>1</value>
23  </property>
24
25  <property>
26    <name>dfs.name.dir</name>
27    <value>file:///home/dhrubanka/hadoopinfra/hdfs/namenode </value>
28  </property>
29
30  <property>
31    <name>dfs.data.dir</name>
32    <value>file:///home/dhrubanka/hadoopinfra/hdfs/datanode </value>
33  </property>
34</configuration>
```

Edit mapred-site.xml as follows:

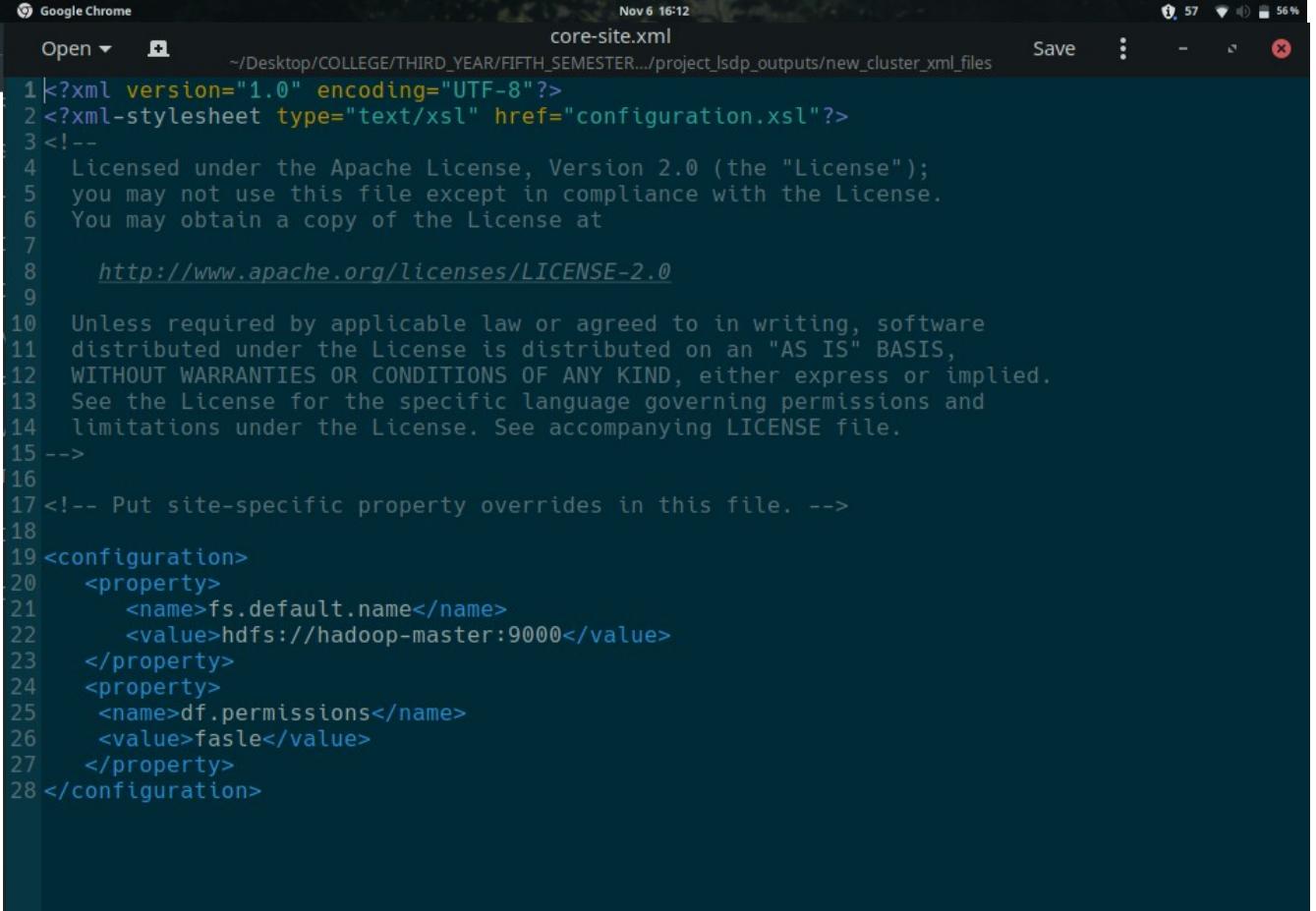


The screenshot shows a terminal window with the following details:

- File: mapred-site.xml
- Path: ~/Desktop/COLLEGE/THIRD\_YEAR/FIFTH\_SEMESTER...project\_lsdp\_outputs/new\_cluster\_xml\_files
- Tab Bar: core-site.xml, hadoop-env.sh, hdfs-site.xml, mapred-site.xml
- Content (Lines 1-28):

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3 <!--
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8     http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15 -->
16
17 <!-- Put site-specific property overrides in this file. -->
18
19 <configuration>
20   <property> |
21     <name>mapreduce.framework.name</name>
22     <value>yarn</value>
23   </property>
24   <property>
25     <name>mapred.job.tracker</name>
26     <value>hadoop-master:9001</value>
27   </property>
28 </configuration>
```

Edit core-site.xml as follows:



The screenshot shows a Google Chrome window with the title bar "core-site.xml" and the URL "Nov 6 16:12 ~/Desktop/COLLEGE/THIRD\_YEAR/FIFTH\_SEMESTER.../project\_lsdp\_outputs/new\_cluster\_xml\_files". The main content area displays the XML code for core-site.xml. The code includes the Apache License 2.0 header and specific configuration properties for HDFS and DFS permissions.

```
1<?xml version="1.0" encoding="UTF-8"?>
2<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
3<!--
4 Licensed under the Apache License, Version 2.0 (the "License");
5 you may not use this file except in compliance with the License.
6 You may obtain a copy of the License at
7
8 http://www.apache.org/licenses/LICENSE-2.0
9
10 Unless required by applicable law or agreed to in writing, software
11 distributed under the License is distributed on an "AS IS" BASIS,
12 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 See the License for the specific language governing permissions and
14 limitations under the License. See accompanying LICENSE file.
15-->
16
17<!-- Put site-specific property overrides in this file. -->
18
19<configuration>
20  <property>
21    <name>fs.default.name</name>
22    <value>hdfs://hadoop-master:9000</value>
23  </property>
24  <property>
25    <name>df.permissions</name>
26    <value>fasle</value>
27  </property>
28</configuration>
```

Then run **\$ bin/hadoop namenode –format** in the hadoop bin folder

Google Chrome Nov 6 16:12

Namenode information

Datanode usage histogram

In operation

Show	25	entries	Search:			
Node	Http Address	Last contact	Capacity	Blocks	Block pool used	Version
dhrubanka-pc:50010 (192.168.43.191:50010)	http://dhrubanka-pc:50075	1s	758.14 GB	0	3.47 GB (0.46%)	2.8.5
shreyansh-pc:50010 (192.168.43.115:50010)	http://shreyansh-pc:50075	3s	881.5 GB	0	3.47 GB (0.46%)	2.8.5

Showing 1 to 2 of 2 entries

Previous 1 Next

Nov 6 4:12 PM

Namenode information

localhost:50070/dfshealth.html#tab-datanode

Datanode Information

Datanode usage histogram

In operation

Show	25	entries	Search:			
Node	Http Address	Last contact	Capacity	Blocks	Block pool used	Version
dhrubanka-pc:50010 (192.168.43.191:50010)	http://dhrubanka-pc:50075	1s	758.14 GB	0	3.47 GB (0.46%)	2.8.5
shreyansh-pc:50010 (192.168.43.115:50010)	http://shreyansh-pc:50075	3s	881.5 GB	0	3.47 GB (0.46%)	2.8.5

Showing 1 to 2 of 2 entries

Decommissioning

Jupyter hdfs\_classify\_final (unstaged changes)

```

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 O
252779806152571225 000027
Prediction for Tumor: yes
File: hdfs://localhost:9000/lssp project/brain cancer dataset/2.8Gb D
ignosis for Cancer DOT R 006 1.3.6.1.4.1.14519.5.2.1.4320.5030.2030593
46048546067166621241946 1.3.6.1.4.1.14519.5.2.1.4320.5030.113686129632
252779806152571225 000029
Prediction for Tumor: yes
File: hdfs://localhost:9000/lssp project/brain cancer dataset/2.8Gb D
ignosis for Cancer DOT R 006 1.3.6.1.4.1.14519.5.2.1.4320.5030.2030593
46048546067166621241946 1.3.6.1.4.1.14519.5.2.1.4320.5030.113686129632
252779806152571225 000030
Prediction for Tumor: yes
File: hdfs://localhost:9000/lssp project/brain cancer dataset/2.8Gb D
ignosis for Cancer DOT R 006 1.3.6.1.4.1.14519.5.2.1.4320.5030.2030593
46048546067166621241946 1.3.6.1.4.1.14519.5.2.1.4320.5030.113686129632
252779806152571225 000031
Prediction for Tumor: yes
In [10]: import pickle
with open('classification_results.pkl', 'wb') as f:
    pickle.dump(file_predictions, f)
In [11]: count
Out[11]: 100
In [1]: 
```

# **RESULT**

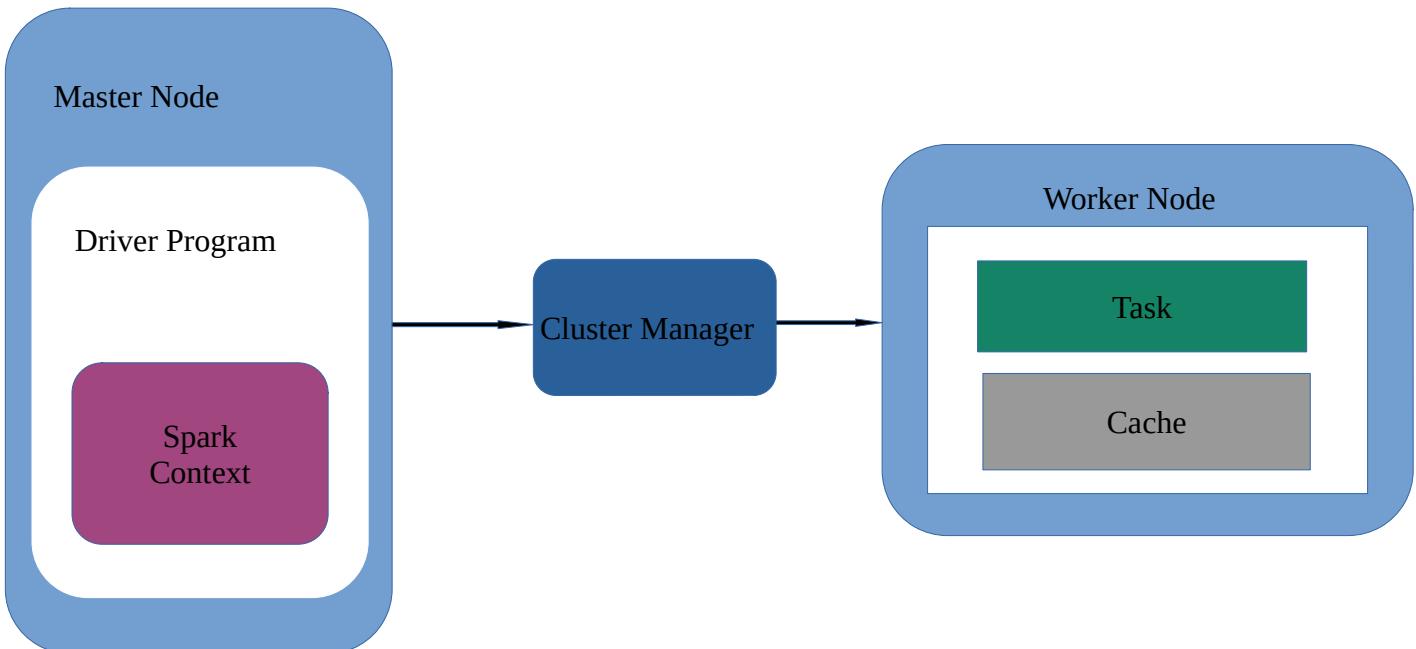
The classifier ran on the instance of PySpark and classified the input images from the dataset one by one. The images were stored in the HDFS in form of binary files. These were then extracted to the PySpark instance using SparkContext. The tumor prediction of the images were given as output by the classifier. The script classified the images into Yes or No categories using the trained classifier. This data was available in the hdfs and can be accessed by all nodes of the cluster.

Model	Training Time	Training accuracy	Testing accuracy
Google Incption v3	20 minutes	97%	96%

The following hardware setup was used in the cluster. It led to fast computation with 200 images classified in 3.5 minutes.

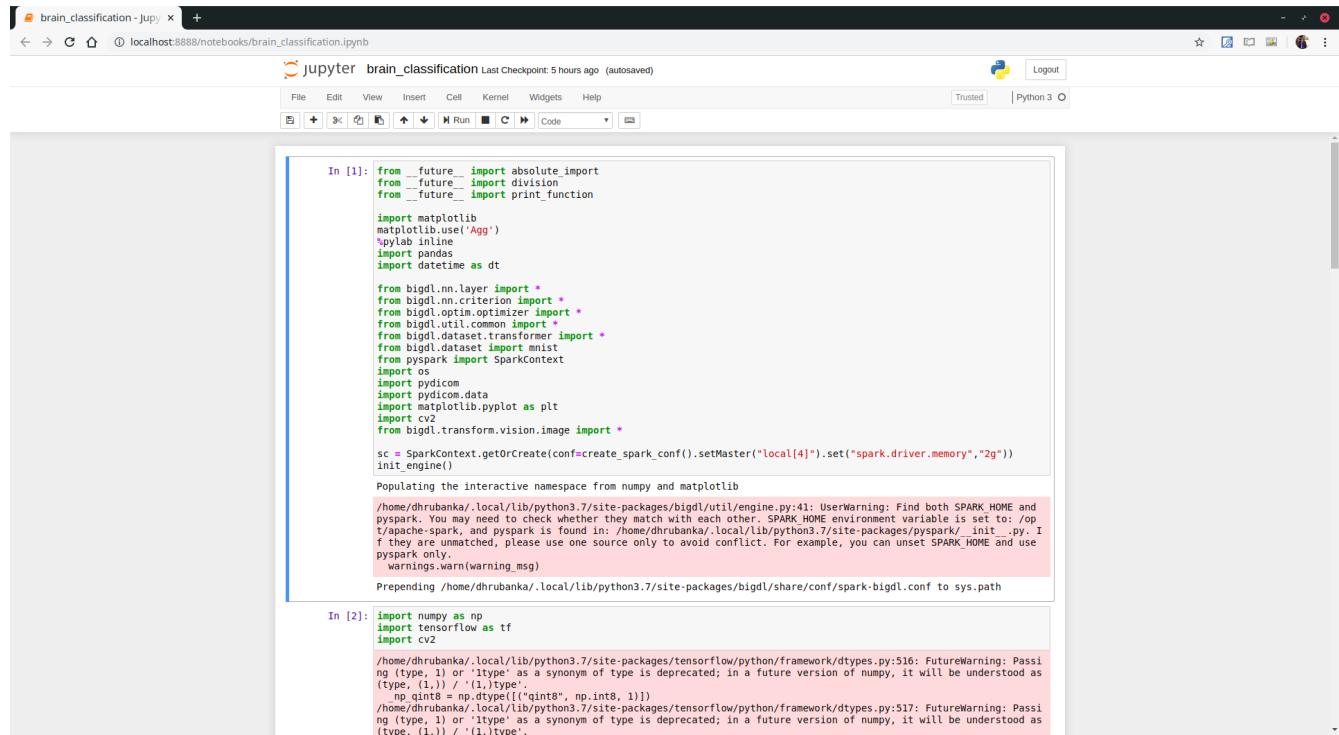
Machine	Processor	GPU	RAM	OS	HARD DISK
Master	Intel i5	AMD Radeon 530	8 GB	Manjaro Arch-Linux	1 TB
Slave1	Intel i7	Nvidia GTX 1050	12 GB	Ubuntu 18.04	1 TB

A cluster architecture was established as follows :



# OUTPUTS

For single image :



The screenshot shows a Jupyter Notebook interface running on a local host. The title bar indicates the notebook is titled "brain\_classification" and was last checkpointed 5 hours ago. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar below the menu bar includes icons for New, Open, Save, Run, Cell, Kernel, Help, and Code. The main area is divided into two code cells:

**In [1]:**

```
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import matplotlib
matplotlib.use('Agg')
%pylab inline
import pandas
import datetime as dt

from bigdl.nn.layer import *
from bigdl.nn.criterion import *
from bigdl.optim.optimizer import *
from bigdl.optim.optimizer import *
from bigdl.dataset.dataset import *
from bigdl.dataset.transformer import *
from bigdl.dataset import mnist
from pyspark import SparkContext
import os
import pydicom
import pydicom.data
import matplotlib.pyplot as plt
import cv2
from bigdl.transform.vision.image import *

sc = SparkContext.getOrCreate(conf=create_spark_conf().setMaster("local[4]").set("spark.driver.memory","2g"))
init_engine()

Populating the interactive namespace from numpy and matplotlib
/home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/util/engine.py:41: UserWarning: Find both SPARK_HOME and
pyspark. You may need to check whether they match with each other. SPARK_HOME environment variable is set to: /op
t/apache-spark, and pyspark is found in: /home/dhrubanka/.local/lib/python3.7/site-packages/pyspark/_init_.py I
f they are unmatched, please use one source only to avoid conflict. For example, you can unset SPARK_HOME and use
pyspark only.
warnings.warn(warning_msg)

Prepending /home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/share/conf/spark-bigdl.conf to sys.path
```

**In [2]:**

```
import numpy as np
import tensorflow as tf
import cv2

/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:516: FutureWarning: Passi
ng (type, 1) or 'dtype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as
(type, (1,)) / ((1,),)
    np qint8 = np.dtype([('qint8', np.int8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:517: FutureWarning: Passi
ng (type, 1) or 'dtype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as
(tvoe, (1,)) / ((1,),tvoe').
```

```

In [2]: import numpy as np
import tensorflow as tf
import cv2

/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:516: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,) / (1,).type)
  np.int8 = np.dtype([('int8', np.int8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:517: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,) / (1,).type)
  np.uint8 = np.dtype([('uint8', np.uint8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:518: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,) / (1,).type)
  np.int16 = np.dtype([('int16', np.int16, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:519: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,) / (1,).type)
  np.uint16 = np.dtype([('uint16', np.uint16, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:520: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,) / (1,).type)
  np.int32 = np.dtype([('int32', np.int32, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/compat/tensorflow_stub/dtypes.py:525: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,) / (1,).type)
  np.int8 = np.dtype([('int8', np.int8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:541: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,) / (1,).type)
  np.uint8 = np.dtype([('uint8', np.uint8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:542: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,) / (1,).type)
  np.int16 = np.dtype([('int16', np.int16, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:543: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,) / (1,).type)
  np.uint16 = np.dtype([('uint16', np.uint16, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:544: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,) / (1,).type)
  np.int32 = np.dtype([('int32', np.int32, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:550: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,) / (1,).type')

```

```

In [*]: path_to_folder = "/home/dhrubanka/Desktop/COLLEGE/THIRD_YEAR/FIFTH_SEMESTER/CSE_3025_LARGE_SCALE_DATA_PROCESSING/LS"
image_file = "000000.dcm"
filenames = pydicom.data.data_manager.get_files(path_to_folder,image_file)[0]
ds = pydicom.dcmread(filenames)
# print(ds)
brain_image = ds.pixel_array
brain_image = brain_image.astype('uint8')
brain_image = cv2.cvtColor(brain_image,cv2.COLOR_GRAY2BGR)
cv2.imwrite("train_image_after",brain_image)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

```

In [11]: def load_graph(model_file):
    graph = tf.Graph()
    graph_def = tf.GraphDef()

    with open(model_file, "rb") as f:
        graph_def.ParseFromString(f.read())
    with graph.as_default():
        tf.import_graph_def(graph_def)

    return graph

def read_tensor_from_image_file(image,
                                input_height=299,
                                input_width=299,
                                input_mean=0,
                                input_std=128/255):
    # input_image = cv2.imread(file_name)
    img2= cv2.resize(image,dsize=(input_height,input_width), interpolation = cv2.INTER_CUBIC)
    #Numpy array
    np_image = np.asarray(img2)
    #Note: Insert your conversion here - see edit remark
    np_final = np.expand_dims(np_image, axis=0)
    normalized = tf.divide(tf.subtract(np_final, [input_mean]), [input_std])
    sess = tf.Session()
    result = sess.run(normalized)

    return result

def load_labels(label_file):
    label = []
    proto_as_ascii_lines = tf.gfile.GFile(label_file).readlines()
    for l in proto_as_ascii_lines:
        label.append(l.rstrip())
    return label

```

```
In [7]: def load_graph(model_file):
    graph = tf.Graph()
    graph_def = tf.GraphDef()

    with open(model_file, "rb") as f:
        graph_def.ParseFromString(f.read())
    with graph.as_default():
        tf.import_graph_def(graph_def)

    return graph

def read_tensor_from_image_file(image,
                                input_height=299,
                                input_width=299,
                                input_mean=0,
                                input_std=255):

    # input_image = cv2.imread(file_name)
    img2=cv2.resize(image,dsize=(input_height,input_width), interpolation = cv2.INTER_CUBIC)
    #Numpy array
    np_image_data = np.asarray(img2)
    #maybe insert float conversion here - see edit remark!
    np_final = np.expand_dims(np_image_data, axis=0)
    normalized = tf.divide(tf.subtract(np_final, [input_mean]), [input_std])
    sess = tf.Session()
    result = sess.run(normalized)

    return result

def load_labels(label_file):
    label = []
    proto_as_ascii_lines = tf.gfile.GFile(label_file).readlines()
    for l in proto_as_ascii_lines:
        label.append(l.rstrip())
    return label

def predict(image):
    model_file = "brain_tumor_weights/output_graph.pb"
    label_file = "brain_tumor_weights/output_labels.txt"
    input_height = 299
    input_width = 299
    input_mean = 0
    input_std = 255
    input_layer = "Mul"
    output_layer = "final_result"

    graph = load_graph(model_file)
```

```
    label.append(l.rstrip())
    return label

def predict(image):
    model_file = "brain_tumor_weights/output_graph.pb"
    label_file = "brain_tumor_weights/output_labels.txt"
    input_height = 299
    input_width = 299
    input_mean = 0
    input_std = 255
    input_layer = "Mul"
    output_layer = "final_result"

    graph = load_graph(model_file)
    t = read_tensor_from_image_file(
        image,
        input_height=input_height,
        input_width=input_width,
        input_mean=input_mean,
        input_std=input_std)

    input_name = "import/" + input_layer
    output_name = "import/" + output_layer
    input_operation = graph.get_operation_by_name(input_name)
    output_operation = graph.get_operation_by_name(output_name)

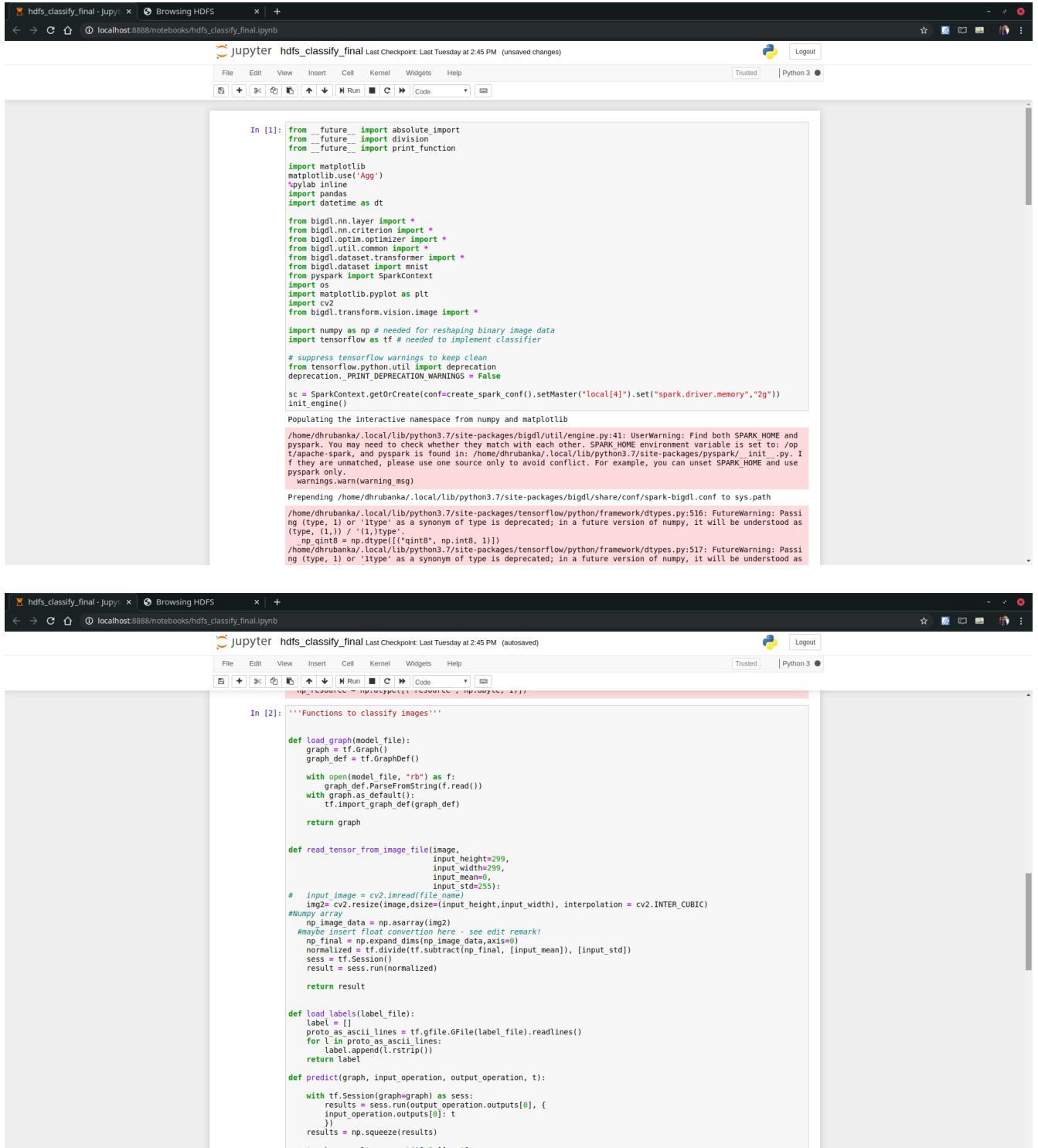
    with tf.Session(graph=graph) as sess:
        results = sess.run(output_operation.outputs[0], {
            input_operation.inputs[0]: t
        })
    results = np.squeeze(results)

    top_k = results.argsort()[-5:][::-1]
    labels = load_labels(label_file)
    return [top_k, labels, results]

if __name__ == "__main__":
    top_k, labels, results = predict(brain_image)
    for i in top_k:
        print(labels[i], results[i])

no 0.8919933
yes 0.10800669
```

## For multiple images :



The screenshot shows two Jupyter Notebook sessions. The top session (In [1]) contains code for setting up the environment, including importing necessary libraries like numpy, tensorflow, and matplotlib, and initializing a SparkContext. It also includes a warning message about deprecated TensorFlow code related to NumPy types. The bottom session (In [2]) contains a more complex set of functions for image classification, including loading a graph from a file, reading tensors from an image file, normalizing the input image, loading labels, and performing predictions using a TensorFlow session.

```

In [1]: from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import matplotlib
matplotlib.use('Agg')
%pylab inline
import pandas
import datetime as dt

from bigdl.nn.layer import *
from bigdl.nn.criterion import *
from bigdl.optim.optimizer import *
from bigdl.util.common import *
from bigdl.dataset.transformer import *
from bigdl.dataset import mnist
from spark import SparkContext
import os
import matplotlib.pyplot as plt
import cv2
from bigdl.transform.vision.image import *

import numpy as np # needed for reshaping binary image data
import tensorflow as tf # needed to implement classifier

# suppress tensorflow warnings to keep clean
from tensorflow.python.util import deprecation
deprecation._PRINT_DEPRECATION_WARNINGS = False

sc = SparkContext.getOrCreate(conf=create_spark_conf().setMaster("local[4]").set("spark.driver.memory","2g"))
init_engine()

Populating the interactive namespace from numpy and matplotlib
/home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/util/engine.py:41: UserWarning: Find both SPARK_HOME and pyspark. You may need to check whether they match with each other. SPARK_HOME environment variable is set to: /opt/apache-spark, and pyspark is found in: /home/dhrubanka/.local/lib/python3.7/site-packages/pyspark/_init_.py. If they are unmatched, please use one source only to avoid conflict. For example, you can unset SPARK_HOME and use pyspark only.
warnings.warn(warning_msg)

PendingDeprecationWarning: /home/dhrubanka/.local/lib/python3.7/site-packages/bigdl/share/conf/spark-bigdl.conf to sys.path
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:516: FutureWarning: Passing (type, 1) or 'Itype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, 1,) / '(1,)type'.
np qint8 = np.dtype([('qint8', np.int8, 1)])
/home/dhrubanka/.local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:517: FutureWarning: Passing (type, 1) or 'Itype' as a synonym of type is deprecated; in a future version of numpy, it will be understood as

In [2]: """Functions to classify images"""

def load_graph(model_file):
    graph = tf.Graph()
    graph_def = tf.GraphDef()

    with open(model_file, "rb") as f:
        graph_def.ParseFromString(f.read())
        with graph.as_default():
            tf.import_graph_def(graph_def)

    return graph

def read_tensor_from_image_file(image,
                                input_height=299,
                                input_width=299,
                                input_mean=0,
                                input_std=255):
    # input image = cv2.imread(file_name)
    img = cv2.resize(image, (input_height, input_width), interpolation = cv2.INTER_CUBIC)
    # Numpy array
    np_image_data = np.asarray(img)
    #maybe insert float conversion here - see edit remark!
    np_final = np.expand_dims(np_image_data, axis=0)
    normalized = np.divide(tf.subtract(np_final, [input_mean]), [input_std])
    sess = tf.Session()
    result = sess.run(normalized)

    return result

def load_labels(label_file):
    label = []
    proto_as_ascii_lines = tf.gfile.GFile(label_file).readlines()
    for l in proto_as_ascii_lines:
        label.append(l.rstrip())
    return label

def predict(graph, input_operation, output_operation, t):
    with tf.Session(graph=graph) as sess:
        results = sess.run(output_operation.outputs[0], {
            input_operation.inputs[0]: t
        })
    results = np.squeeze(results)
    top_k = results.argsort()[-5:][::-1]

```

```

File Edit View Insert Cell Kernel Widgets Help
[+] Run C Code Trusted Python 3
results = sess.run(operation.outputs[0], {
    input_operation.inputs[0]: t
})
results = np.squeeze(results)
top_k = results.argsort()[-5:][::-1]
labels = load_labels(label_file)
return [top_k, labels, results]

# Defining model parameters
model_file = "brain_tumor_weights/output_graph.pb"
label_file = "brain_tumor_weights/output_labels.txt"
input_height = 299
input_width = 299
input_mean = 0
input_std = 255
input_layer = "Mul"
output_layer = "final_result"

# Loading graph beforehand
graph = load_graph(model_file)

# Setting name and input parameters beforehand
input_name = "import/" + input_layer
output_name = "import/" + output_layer
input_operation = graph.get_operation_by_name(input_name)
output_operation = g.graph.get_operation_by_name(output_name)

```

In [\*]: # load the hadoop filesystem into pyspark context

```

hadoop = sc._jvm.org.apache.hadoop
fs = hadoop.fs.FileSystem
conf = hadoop.conf.Configuration()

# give dataset path to pyspark context
path = hadoop.fs.Path('/lsdp_project/brain_cancer_dataset')

# initialize empty list
file_predictions = []

# count to check if all files accessed : around 4682
count = 0

for f in fs.get(conf).listStatus(path):
    # get the path of each binary file
    image_file_path = f.getPath()
    image_file_path = f.getPath()
    # print(image_file_path)

    # Load binary file into variable

```

```

File Edit View Insert Cell Kernel Widgets Help
[+] Run C Code Trusted Python 3
t = read_tensor_from_image_file(
    recovered_image,
    input_height=input_height,
    input_width=input_width,
    input_mean=input_mean,
    input_std=input_std)

# get predictions from the model
top_k, labels, results = predict(graph, input_operation, output_operation, t)
prediction_label = labels[top_k[0]]

# append file,prediction to list
file_predictions.append((str(image_file_path), prediction_label))
count += 1

print('Image: ', str(image_file_path), ' Tumor Prediction: ', prediction_label)

if count==10:
    break

```

Image: hdfs://localhost:9000/lsdp\_project/brain\_cancer\_dataset/2.8Gb Dignosis for Cancer DOT R 004 1.3.6.1.4.1.14  
5 000000 Tumor Prediction: no  
Image: hdfs://localhost:9000/lsdp\_project/brain\_cancer\_dataset/2.8Gb Dignosis for Cancer DOT R 004 1.3.6.1.4.1.14  
5 19.5.2.1.4320.5030.248552508121514040263344871813 1.3.6.1.4.1.14519.5.2.1.4320.5030.96635407587648204223576192929  
5 000001 Tumor Prediction: yes  
Image: hdfs://localhost:9000/lsdp\_project/brain\_cancer\_dataset/2.8Gb Dignosis for Cancer DOT R 004 1.3.6.1.4.1.14  
5 19.5.2.1.4320.5030.248552508121514040263344871813 1.3.6.1.4.1.14519.5.2.1.4320.5030.96635407587648204223576192929  
5 000002 Tumor Prediction: yes  
Image: hdfs://localhost:9000/lsdp\_project/brain\_cancer\_dataset/2.8Gb Dignosis for Cancer DOT R 004 1.3.6.1.4.1.14  
5 19.5.2.1.4320.5030.248552508121514040263344871813 1.3.6.1.4.1.14519.5.2.1.4320.5030.96635407587648204223576192929  
5 000003 Tumor Prediction: yes  
Image: hdfs://localhost:9000/lsdp\_project/brain\_cancer\_dataset/2.8Gb Dignosis for Cancer DOT R 004 1.3.6.1.4.1.14  
5 19.5.2.1.4320.5030.248552508121514040263344871813 1.3.6.1.4.1.14519.5.2.1.4320.5030.96635407587648204223576192929  
5 000004 Tumor Prediction: yes  
Image: hdfs://localhost:9000/lsdp\_project/brain\_cancer\_dataset/2.8Gb Dignosis for Cancer DOT R 004 1.3.6.1.4.1.14  
5 19.5.2.1.4320.5030.248552508121514040263344871813 1.3.6.1.4.1.14519.5.2.1.4320.5030.96635407587648204223576192929  
5 000005 Tumor Prediction: yes  
Image: hdfs://localhost:9000/lsdp\_project/brain\_cancer\_dataset/2.8Gb Dignosis for Cancer DOT R 004 1.3.6.1.4.1.14  
5 19.5.2.1.4320.5030.248552508121514040263344871813 1.3.6.1.4.1.14519.5.2.1.4320.5030.96635407587648204223576192929  
5 000006 Tumor Prediction: yes  
Image: hdfs://localhost:9000/lsdp\_project/brain\_cancer\_dataset/2.8Gb Dignosis for Cancer DOT R 004 1.3.6.1.4.1.14  
5 19.5.2.1.4320.5030.248552508121514040263344871813 1.3.6.1.4.1.14519.5.2.1.4320.5030.96635407587648204223576192929  
5 000007 Tumor Prediction: yes  
Image: hdfs://localhost:9000/lsdp\_project/brain\_cancer\_dataset/2.8Gb Dignosis for Cancer DOT R 004 1.3.6.1.4.1.14  
5 19.5.2.1.4320.5030.248552508121514040263344871813 1.3.6.1.4.1.14519.5.2.1.4320.5030.96635407587648204223576192929  
5 000008 Tumor Prediction: yes  
Image: hdfs://localhost:9000/lsdp\_project/brain\_cancer\_dataset/2.8Gb Dignosis for Cancer DOT R 004 1.3.6.1.4.1.14  
5 19.5.2.1.4320.5030.248552508121514040263344871813 1.3.6.1.4.1.14519.5.2.1.4320.5030.96635407587648204223576192929  
5 000009 Tumor Prediction: yes

## **CONCLUSION**

1. Hadoop and PySpark were setup in the machines.
  2. BigDL was setup in all the machines.
  3. The dataset was uploaded to the HDFS.
  4. An instance of PySpark was started and integrated with BigDL.
  5. The image classifier was trained using inception with validation metric of .
  6. The classifier model was loaded into the PySpark instance using BigDL.
  7. The dataset was iterated through and each image was passed to the classifier.
  8. The classifier classified the images and the image path along with prediction were listed.
  9. The image paths along with their predcition were written to a pickle file.
  10. A hadoop cluster was setup.
  11. The script was run on the cluster and speed improvement was noticed.