Report

**IMAGE SEARCH FOR E-COMMERCE**

Prepared by

Sharan Raj

**Table of Contents**

**Table of Contents ...........................................................................................................................ii**

**1. Introduction ................................................................................................................................1**

1.1 Purpose ............................................................................................................................................. 1

1.2 Objective…………………………………………………………………………………………....1

**2. Overall Description ....................................................................................................................1**

2.1 Product Perspective.....................................................................................................................…...1

2.2 Product Functions ............................................................................................................................ 1

2.3 User Classes and Characteristic........................................................................................................ 1

2.4 Operating Environment .................................................................................................................... 2

**3. External Interface Requirements .............................................................................................2**

3.1 User Interfaces.................................................................................................................................. 2

3.2 Hardware Interfaces ......................................................................................................................... 2

**4. Functional Requirement.........................................................................................................…2**

**5. System architecture description ............................................................................................…3**

**6. Reuse and relationships to other products …………………………..…………….……….…..4**

**7. Design decisions and trade offs……………………………………………………………..…...4**

**8. Prerequisite steps in order to build the system …………………………………………..…..4**

**9.Code………………………………………………………………………………………………6**

**10. Sample Output …..................................................................................................................….9**

# Introduction

When it comes to E-commerce user experience plays a vital role in the success of the business, reducing the burden of the user by understanding user’s requirements and providing the list of products he/she is looking for will enrich the experience of the e-commerce service provider. Describing a product by words can lead to multiple results as many products will fit into this description, whereas an image of the product will give a complete description of the product which in turn will provide a convenient method for product description and search for the user

## Purpose

This document gives a detailed description of the software requirements and the hardware requirements of the project, a step by step procedure is given in order to build the web app from bottom up.

## Objective

The main objective of this software is to provide an alternative approach to text based search for querying, as a text based querying can lead to numerous number of hits(results) whereas an image based search can narrow it down to the fewest hits since in E-commerce the greatest selling point is the fewest number of clicks to get the product the customer wants this alternative approach greatly improves ease of shopping thus improving the E-commerce sales

# Overall Description

## Product Perspective

This project is made as a web-app which the E-commerce service provider can integrate in their systems and connect to their database which contains the required information. For the purpose of development I have used only two products for identification

## Product Functions

The only function of this project is image based search (reverse image search)

## User Classes and Characteristic

There are two User classes:

1. Customers: Customers can use this web-app to search for the product they want to purchase.
2. Admin: Admin is responsible for the error detection and correction, also to augment new products to the system.

## Operating Environment

All back-end processing will run on a Linux server. The results will be shown on the website using Php and HTML.(For the purpose this project a local server is hosted in a Linux virtual machine).

# External Interface Requirements

## User Interfaces

An E-commerce system is usually either a website or a mobile app this system can be integrated to both in this project the user interface is a website running on a local-host Apache server

**3.2 Hardware Interface**

* Processor: Intel Xeon
* Ram: 1 GB Ram
* Clock speed: 2.4GHz
* Hard disk: 20Gb

# Functional Requirements

* An interface for the user for uploading the image of the required products to be searched
* The main function of this software is to search for the product from the company’s database
* If the image is of a product in E-commerce data base then display the product description

**5. System architecture description**

This system is designed with scalability in mind . Many of the anticipated changes to this system in future phases will only require augmenting the new product data and to train and build the classifier

**Back-end** : For searching the product the machine should understand what the product is for that a classifier has to be built in order to classify the image as a particular product since an image is provided to the machine, machine learning techniques such as Convolutional neural network (CNN) is used in order to teach the machine about the detailed description of the product

A CNN is feed forward artificial neural network, which works similar to how we process an image seen with our eyes, the CNN consist of multiple layers of neural network first a data set of images of a particular product is fed into the neural network to train the CNN that these images belong to the same class, now the CNN is trained to classify the class of image (similar images) it was fed prior to. Now when a new image is fed into the CNN it processes portions of the inputs and the finally concludes by classifying it



for building the CNN open source library such as Tensorflow and other machine learning library should be incorporated

First a large data set of each product containing images of each product should be imported then an api is called for importing a CNN such as inception v3 (Google’s trained CNN for image classification) and the imported data is used as the training data then the querryed image is inputted to the CNN using an api once the classifier gives the output this is used to choose the hits that should be displayed for that a query has to be passed to the company’s database to choose the products that matches the description of the output class and retrieve these hits

Front-end : The E-commerce website has an image submit button in order to upload the image and once the search button is clicked the website will display the detailed description of the product such as price, name etc the technology stack for this will be Html, Css and PHP

**6****.0 Reuse and relationship to other products**

As mentioned earlier the back-end of this project works by calling Google’s open source api for machine learning **TENSOR FLOW** which reduces the time to manually input attributes for each product as tensorflow finds attributes for each product by using the pixel relationship of the training data provided for each product to classify it, also to build the classifier Google’s neural network **INCEPTION V3** is used thus reducing the time to implement the neural network and also reduces the training time as we are using **transfer training principle** were we only train the final layer of the neural network in order to build our own custom classifier

**7. Design decisions and trade off**

### **7.1 space constraints**

### To build a robust search engine for a E-commerce service provider requires at least 100 images of each product thus requiring a huge and fast server to store the data set

**7.2 computational and time constraints**

To process, build and classify these 1000’s of products require workstation with high computational capability and GPU support and weeks of processing time

### thus, only 2 products was used as mock trial to implement this system

**8. Prerequisite steps in order to build the system (In terminal or command line)**

1. Instal pip : sudo apt-get install python-pip python-dev
2. upgrade pip : sudo pip install –upgrade pip
3. install tensorflow: sudo pip install tensorflow
4. install github : sudo apt-get install git
5. clone tensorflow library : sudo git clone <https://github.com/tensorflow/tensorflow>
6. install JDK 8 : sudo add-apt-repository ppa:webupd8team/java && sudo apt-get update && sudo apt-get update
7. adding bazel uri package: echo "deb [arch=amd64] <http://storage.googleapis.com/bazel-apt> stable jdk1.8" | sudo tee/etc/apt/sources.list.d/bazel.list

curl [https://bazel.build/bazel-release.pub...](https://bazel.build/bazel-release.pub.gpg)| sudo apt-key add –

1. Install and update Bazel : sudo apt-get update && sudo apt-get install bazel && sudo apt-get upgrade bazel
2. Install numpy : sudo apt-get install python-numpy python-dev python-wheel
3. sudo pip install backports.weakref﻿
4. get into tensorflow directory : cd tensorflow
5. configure tensorflow : sudo ./configure
6. Configure question answers

Python location: Default-enter, Jemalloc: Yes ,Google Cloud: No, Hadoop: No, XLA: No, Python library: Default-enter, OpenCL: No ,Cuda: No

1. Run the Inception-v3 retrain model: sudo bazel-bin/tensorflow/examples/image\_retraining/retrain –image\_dir [directory were training data is stored]
2. Build the label\_image example:sudo bazel build tensorflow/examples/label\_image:label\_image
3. The output of model build is graph and labels which is stored in temporary folder to store it permenantly copy the outputs to desktop
4. Install LAMP server I have used apache server for localhost
5. copy the attached html file “i.html” and “test.php” to var/www/
6. create an upload folder in var/www/html where the image will get uploaded
7. copy the attached shell script “t.sh” to home folder also create a textfile called “r.txt” in home folder

Note: give necessary file permisions for all the files also i modified sudo command to not to ask password this can be done by

sudo vi sudo

go to the bottom and type the following

[user] ALL=(ALL) NOPASSWD:ALL

**9. Code**

**9.1 Shell script**

#!/bin/sh

sudo bazel-bin/tensorflow/examples/label\_image/label\_image –graph=/home/cse/Desktop/output\_graph.pb--labels=/home/cse/Desktop/output\_labels.txt --output\_layer=final\_result --image=/var/www/html/$1 --input\_layer=Mul 2> /home/cse/r.txt

Note: 2> /home/cse/r.txt saves the output in the text file created r.txt this is done because by default the output of the api comes out in warning format

**9.2 PHP**

<?php

$target\_dir = "uploads/";

$target\_file = $target\_dir . basename($\_FILES["fileToUpload"]["name"]);

$uploadOk = 1;

$imageFileType = pathinfo($target\_file,PATHINFO\_EXTENSION);

// Check if image file is a actual image or fake image

if(isset($\_POST["submit"]))

{

$check = getimagesize($\_FILES["fileToUpload"]["tmp\_name"]);

if($check !== false)

{

//echo "File is an image - " . $check["mime"] . ".";

$uploadOk = 1;

}

else {

echo "File is not an image.";

$uploadOk = 0;

}

}

// Check if $uploadOk is set to 0 by an error

if ($uploadOk == 0)

{

echo "Sorry, your file was not uploaded.";

// if everything is ok, try to upload file

}

else

{

if (move\_uploaded\_file($\_FILES["fileToUpload"]["tmp\_name"], $target\_file))

{

}

else {

echo "Sorry, there was an error uploading your file.";

}

}

$r = shell\_exec("sudo /home/cse/t.sh $target\_file 2>&1");

$t = file\_get\_contents("/home/cse/r.txt");

preg\_match("/(ga.\*|mo.\*)/",$t,$matches);/\* a regualr expression is used to output only the required data\*/

echo $matches[0];

?>

**9.3 Html**

<!DOCTYPE html>

<html>

<head>

<title>SEARCH</title>

<style>

input[type=submit]

{

width: 300px;

background-color: green;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

border-radius: 4px;

cursor: pointer;

}

</style>

<body>

<form action="test.php" method="post" enctype="multipart/form-data">

Select image to upload:

<input type="file" name="fileToUpload" id="fileToUpload"></br>

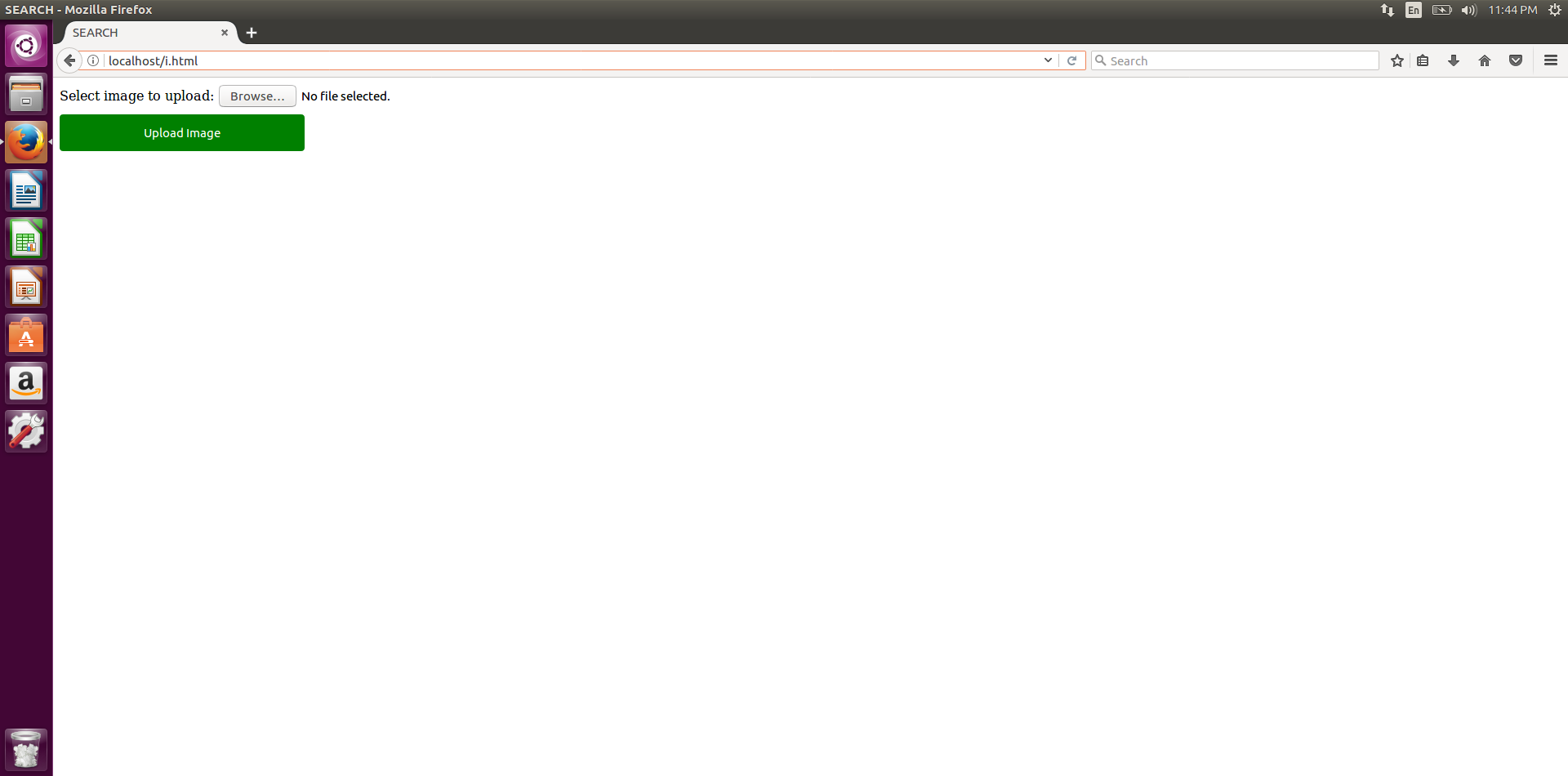
<input type="submit" value="Upload Image" name="submit">

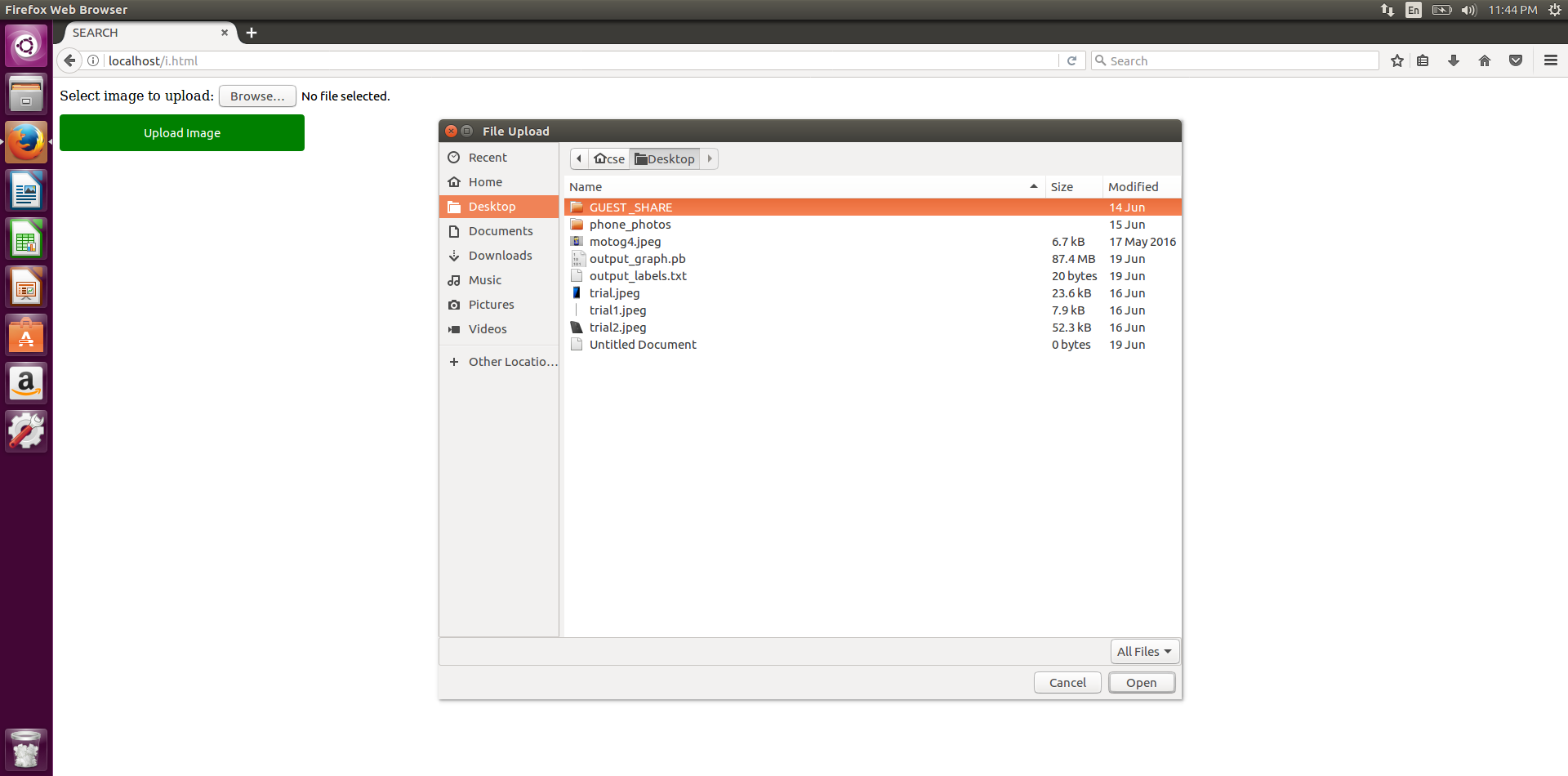
</form>

</body>

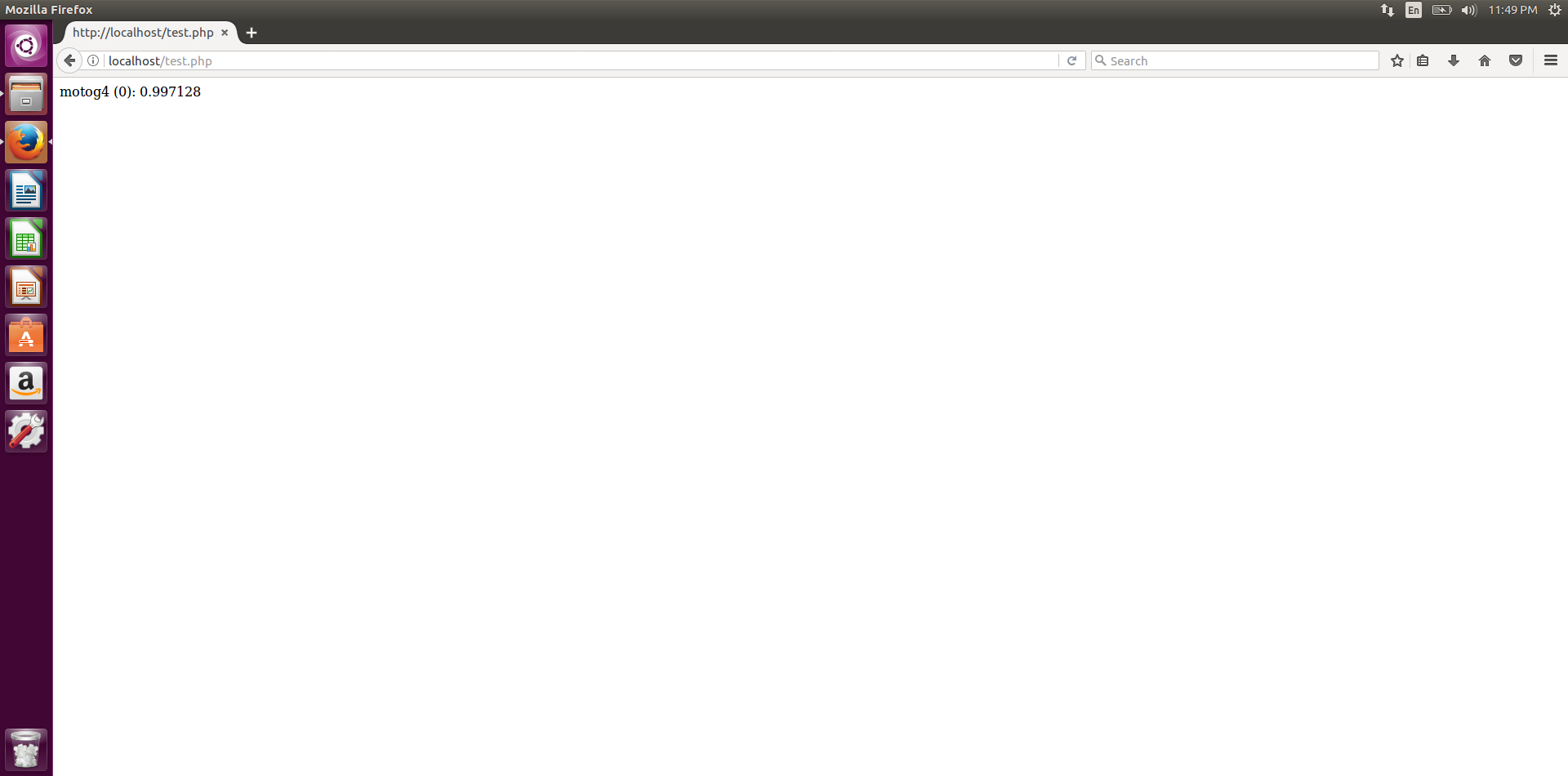
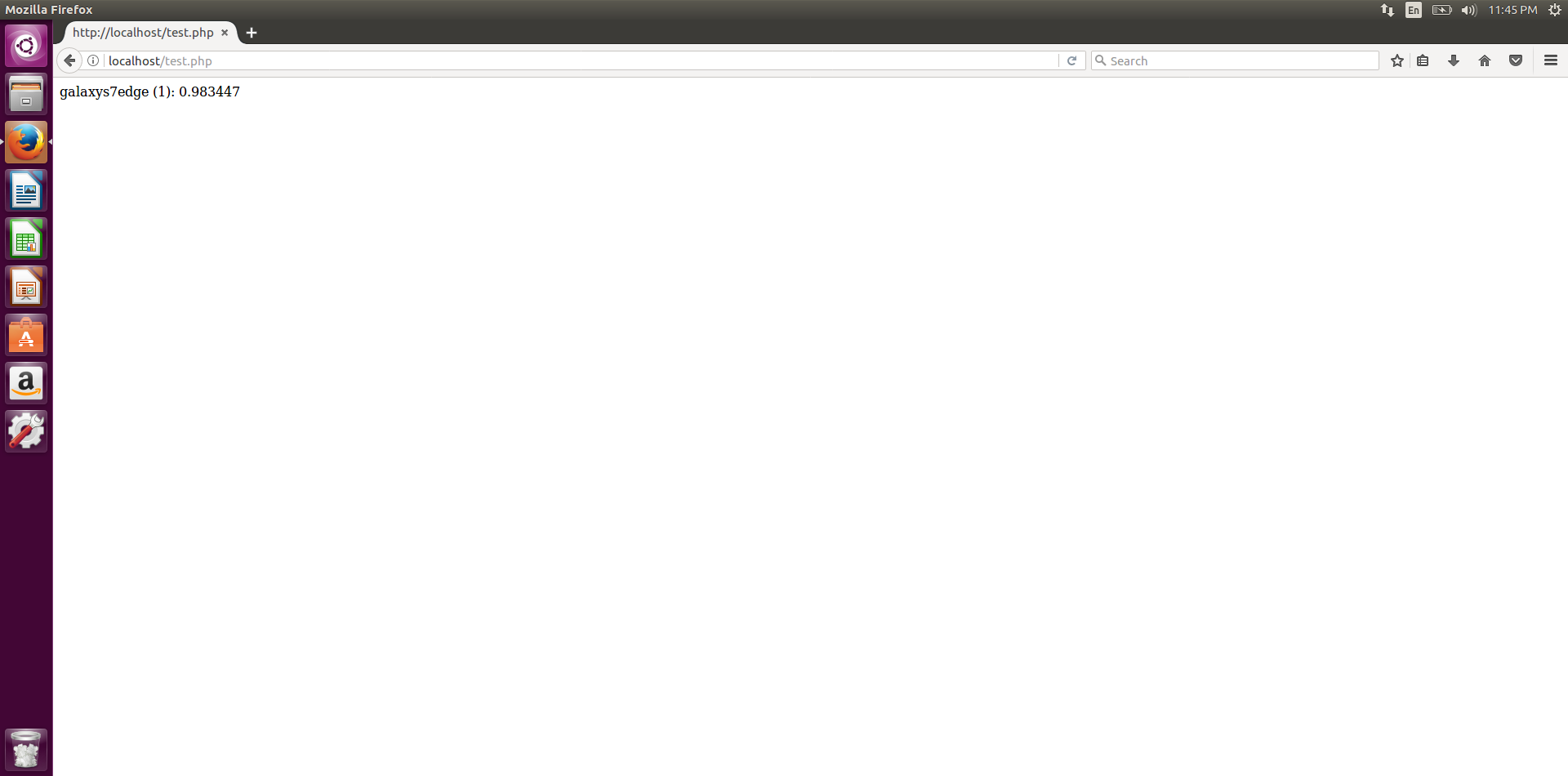
</html>

**10. Sample Output**





when image of galaxy s7 edge is given as input



when image of moto g4 is given as input

actual output of the tensorflow api which is stored in r.txt