Started on	Friday, 20 November 2020, 9:01 AM
State	Finished
Completed on	Friday, 20 November 2020, 11:45 AM
Time taken	2 hours 44 mins
Grade	Not yet graded

Question **1**

Complete

Marked out of 10.00

How non-functional requirements' (NFR) are elicitated, defined, and verified in your software project? Do you have made any design modification to incorporate this NFR? Explain the above questions with a detailed diagram and example from your project?

Non Functional requirements of any system of project include the following attributes

1. Portability

2.Security

3.Maintainability

4.Reliability

5.Scalability

6.Perfomance

7.Reusability

8.Flexibility

a) Portability:

- The MalJPEG code that we have written can be run on any system independent of the operating system specifications.
- And hence is portable across windows, linux and macOS systems for example.
- Also the only prerequisite for being able to run the MalJPEG code is having the python interpreter installed in the system and also to have the necessary libraries installed.

b)Security:

- In its current state the MalJPEG application doesn't need any security. It hasn't been incorporated into an application yet. But when creating an application we can add some authentication so that only verified users can submit the malicious images.

c)Maintainability:

- Proper commenting of code was done to ensure that anyone reading the code could understand what was happening.
- We gave proper names to the variables so that editing the code and understanding is easy.
- In the future as more code might be added we can modularize the code to make it easier to work with the modules. So far since the codebase isnt big enough we haven't divided the code into separate modules.

d)Reliability:

- The MalJPEG code works on any image dataset. It scans a directory for images that are JPEG only and runs the machine learning algorithms.
- If you run the MalJPEG code over and over again it gives the same results for the same dataset.
- The code isn't expected to fail unless the system running the code fails due to some reason.
- Normally Reliability is expressed in percentage which describes how likely the system is expected to run in a given time period but we haven't undertaken any such measurement.

e)Scalability:

- The dataset used by us contained over 30000 benign images and over 700 malicious images.
- The code has the ability to run on any size of the dataset. The feature extractor has no limitation over the number of images that it can run on and the speed of its execution is also very good.
- While running the feature extractor for example on a small dataset and a large dataset we didn't find a lot of time difference especially for bengin images. So it's performance as well as the performance of the models was good for our dataset.

f)Performance:

- Our code can process a directory of a few thousand images in a matter of seconds consistently and output the result in the same way.
- Even if it is hosted online it is expected to perform similarly.

g)Reusability:

- The feature extractor can be used to parse a JPEG file for any other requirement.

- Only minimal changes have to be made in the code for parsing the JPEG file in different ways.

h)Flexibility:

- MalJPEG feature extractor currently works on only JPEG images as it parses the internal structure of a JPEG image.
- The same logic can be applies to accomodate any change in the functional requirement for example if we have to parse the internal structure of a PNG file to determine if it's malicious or benign.
- Also the machine learning models are independent of the feature extractor so the required model can be trained as and when required for the new set of features.

Question **2**

Complete

Marked out of 10.00

Elaborate on the testing done on your project with the help of test cases?

Our project had two phases
1. Feature extraction
2 .Model training

For testing the feature extractor we ran the feature extractor on a dataset of single image and then on a dataset of 100 images.
Only after getting the accurate results did we run the feature extractor on the final MalJPEG dataset.
While developing the feature extractor we initially wrote a simple JPEG Parser to detect the markers.
We then used an open Source software called JPEGSnoop to compare the results of the parser that we wrote. JPEG snoop can also be used to check if a given JPEG image is altered or not.
Once we were satisfied that the results are accurate and is validated by the JPEG snoop software we wrote the code for creating the csv files which hold the accumulated feature vectors of the images.

For the model training phase we used a similar approach, first using a small dataset to check if the code works as it is expected to and then we checked if the metrics are printed the way they are meant to and then we expanded it to accomodate the bigger dataset.

Unit Testing
1. Tested feature extractor which was initially a simlple JPEG parser for its accuracy in detecting the markers of a JPEG image
Used JPEG snoop software for this

2. Tested Dataset creator program to check if proper files with proper extension is being created.
we found out that proper files train.csv that contained features for training the model and test.csv that contained features for testing were created.

3.Tested Model training programs to check if they detected the test and train csv files and checked if the logic works fine.

Integration Testing
1.Linked the datasets created by the dataset creator program to the model training programs.

System Testing
1. We ran the code in two different operating systems. Windows 10 and Ubuntu 20.04 and got same output.

Stress Testing.
1.We added files with non JPEG extension to the directory and checked if the code ran without any issues.
Expected result: work without issues
Result: worked without any issues

Performance Testing.
1. We compared the code with datasets of different sizes 100 images, 600 images, 30000 images.

test cases.jpeg

tcs2.jpeg

Question **3**

Complete

Marked out of 10.00

Explain design patterns with the help of an example from your software engineering project?

Design Pattern is a problem, solution pair. For a problem that is recurring one can use the same approach to solve it. It is like a template or description on how to solve the problem that can be used in varying problems .
Reusing design patterns helps to prevent subtle issues that can cause major problems and improves code readability for coders and architects familiar with the patterns.There are three kinds of design patterns Creational , Structural and Behavioural.
The creational design patterns include.

1. Abstract Factory.

Abstract Factory patterns work around a super-factory which creates other factories. This factory is also called as factory of factories.
This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object.
In Abstract Factory pattern an interface is responsible for creating a factory of related objects without explicitly specifying their classes.
Each generated factory can give the objects as per the Factory pattern.

2.Builder

Builder pattern builds a complex object using simple objects and using a step by step approach.
This pattern provides one of the best ways to create an object.
A Builder class builds the final object step by step. This builder is independent of other objects.

3.Factory method

In Factory pattern, we create object without exposing the creation logic to the client and refer to newly created object using a common interface.

4.Singleton etc

In Singleton pattern we only allow a single instance of a class globally. This is done using static methods and instances.
It is advised not to use the Singleton patterns because it is possible that one might need more than one instances of the classes in the future.

The structural design patterns include
1. Adapter

Use interface of one class to interface into another class. For example to use a class that gives data in JSON format with a class that used XML format.

2.Bridge

Question **4**

Complete

Marked out of 10.00

Draw the detailed design diagrams (which is suitable to the paradigm chosen) of your project?

design_MALJPEG_Feature extracor.jpeg

activity diagram.jpeg

Question **5**

Complete

Marked out of 10.00

Explain the software configuration management policies and tools used in your project?

The need of software configuration management is to keep track of make changes for the changing software requirements.
Configuration management helps keep track of changes made to each version of the system under developement.
There are four main activites involved in configuration management
1. Version management
2. System building
3. Change managment
4. Release management

Some systems have multiple versions that might involve simultaneous development by multiple teams and in such a situation version management becomes even more important.

We have used git for as our version management tool
The way git works is there is a public or private repository that contains the master branch and then any of the developers in this case our team members can check out the components into their local repository work on those copies and check in those changes.

Git is distribute version control system and works in the following way.
A 'master' repository is created on a server that maintains the code produced by the development team.
Instead of checking out the files that they need, a developer creates a clone of the project repository that is downloaded and installed on their computer.
Developers work on the files required and maintain the new versions on their private repository on their own computer.
When changes are done, they 'commit' these changes and update their private server repository.
They may then 'push' these changes to the project repository

The benefits of using a distributed VCS were
1. It provides a backup mechanism for the repository. If the repository is corrupted, work can continue and the project repository can be restored from local copies.
2. It allows for off-line working so that developers can commit changes if they do not have a network connection.
Project support is the default way of working.
3. Developers can compile and test the entire system on their local machines and test the changes that they have made

Branching and Merging
1. Our project has one master branch and all changes were done by pushing into that master branch. If required we can create separate branches for each of the team members to work separately on their own changes.

Storage management
1. As disk storage is now relatively cheap, Git uses an alternative, faster approach. Git does not use deltas but applies a standard compression algorithm to stored files and their associated meta-information.
2. It does not store duplicate copies of files. Retrieving a file simply involves decompressing it, with no need to apply a chain of operations.
3. Git also uses the notion of packfiles where several smaller files are combined into an indexed single file.

◄ lecture 30

Jump to...