上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

SOFTWARE ENGINEERING COURSE PROJECT

REPORT

NAME:	Jiaxian Guo Ouyu Lan
STUDENT ID:	5140309093 5140309001
MAJOR:	COMPUTER SCIENCE
COMPLETE TIME:	2017.1.04

Content

1 Introduction

1.1 Purpose of the Project

It is acknowledged that for any business, the best method of publicity is advertising. So they often buy the rights of broadcast advertising in the commercial center of the screen. But in fact, commercial center display ads on the display is random, not targeted, so the Advertising yields is not high. To save the cost and be effective, precise marketing is necessary.

Our software is to solve this problem, show the most likely attracting advertisements to different people, which is based on human's age and gender. For example, if the software captures a baby face in the video, it may play advertisements of milk. Or if a middle-aged man appearing in the video, it may advertise a car. To this end, we make the method of machine learning, deep convolution neural network specifically, to get a satisfying accuracy to predict the age and gender.

1.2 Scope of the Project

During the process of project, we involve the field of machine learning and software development. The software can be applied in following scenarios:

- Taxi back seat advertising;
- Commercial center advertising screen;
- Any scene where you can customize your advertisements.

1.3 Overview of the Document

The following document introduce and analyze our software in detail. Section

2 is about requirements specification, including functional, non-functional and domain requirements. Section 3 illustrates the whole software design, including the model, development tools, architectural design and object identification. Section 4 shows the testing to our software, including the test plan and design, case, procedure specification. Section 5 is the conclusion while Section 6 is the references in the report and the project.

2 Requirements Specification

2.1 Functional Requirements

1) Predict the age and gender of faces in the video;
2) Play ads based on the estimated gender and age;
3) Upload image, and check image's age and gender;
4) A Caffe model which reaches a high accuracy in predicting age and gender;
5) C# interface for Caffe;
6) Open computer's camera;

2.2 Non-functional Requirements

1) The maximum processing time span is 1 second;
2) The software can run for 24 hours without stop;
3) The software processing has no memory leaks;
4) The rights of administrators.

2.3 Domain Requirements

Any individual or organization that wants a higher return on advertising.

2.4 Environment Requirements

1) CPU: i3 1.3Ghz or better;
2) Main memory: 1GB in minimum; 2GB at recommendation;
3) Platform: x64 windows;
4) User: click.

2.5 Requirements Analysis Models

2.5.1 Use Cases

This system is divided into five use cases: Upload Image, Image Prediction, Open Camera, Close Camera, Switch Camera.

Each use case is specified in the use case specification document. Please refer to each use case specification document to get detail information to know more about each use case.

2.5.2 Functional Analysis Model

We use Rational Rose to create the use case model. Here is the view of use case model.

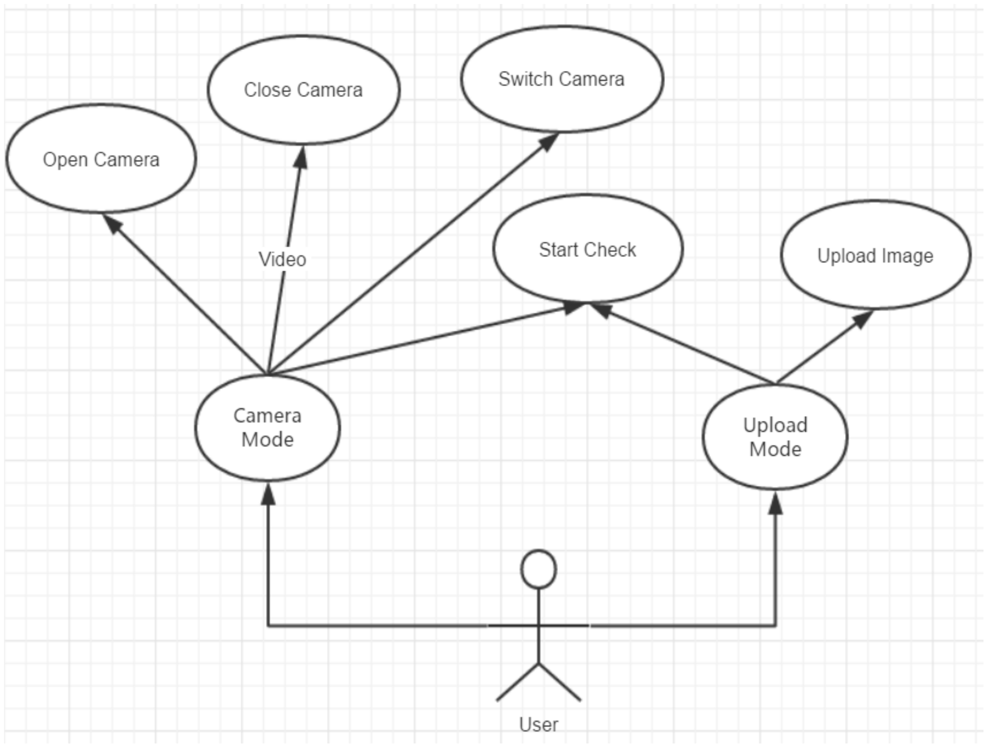

Fig 1. Use cases.

2.5.3 Architecture

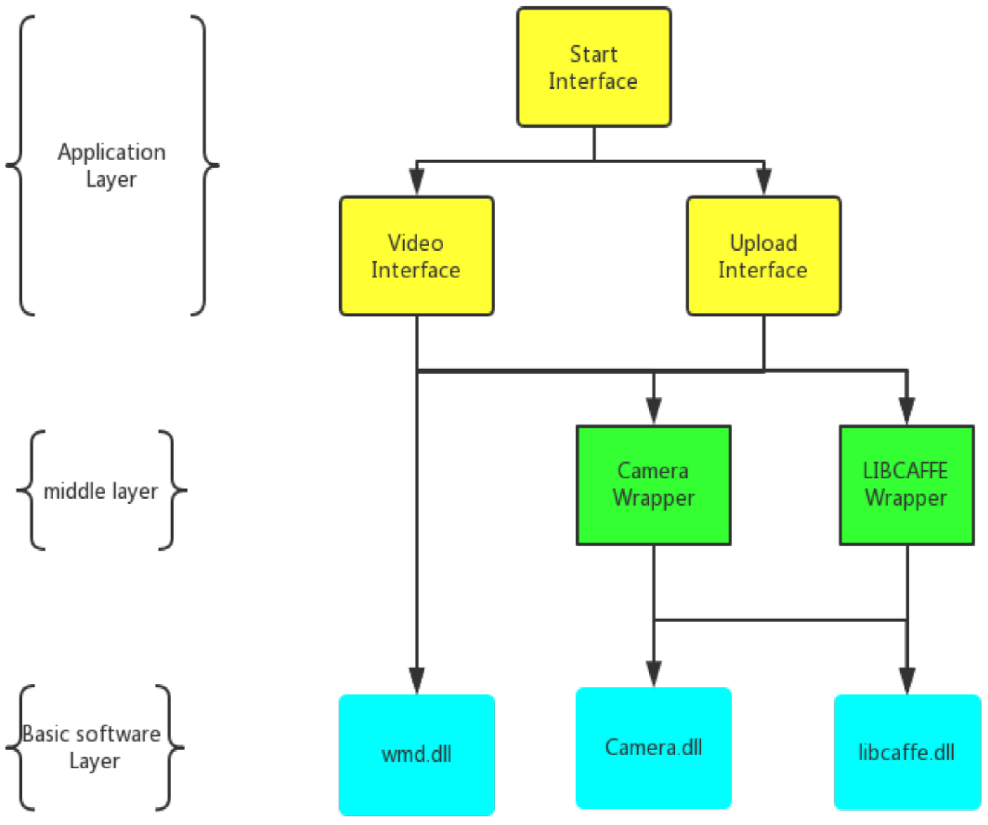

Fig 2. Software architecture.

2.5.4 Bill of Materials

Seq	Item	Qty	Functional Description	Vendor/ Manufacturer	Product Name	Version	Part No.	Purchase Reuse Upgrade	Standard (Y/N)	CR#*
1	System	High	Provide database service, response all the request from the clients	ACER, China	Laptop	1.0	10001	R	N	2
2	Operation System	High	Platform to develop system	Microsoft, America	Windows 10	10 Professional	10002	U	N	3
3	UML tools	Middle	Use to design system	IBM Rational, America	Rational Rose	2003	10003	R	Y	2
4	Develop Environment	Middle	Integrated develop environment to develop system.	Microsoft, America	Visual studio	2013 community	10004	R	Y	4
5	Office Tools	Middle	To edit documents, table, etc.	Microsoft, America	Office	2016 Edition	10005	R	Y	1

Table 1. Bill of materials.

2.6 Requirements Traceability

In phase of BTM developing, we will make three modules: Use-case Module, Analysis Module, and Design Module, so there exists the traceability among those modules.

In UML, we describe it as Fig 3.

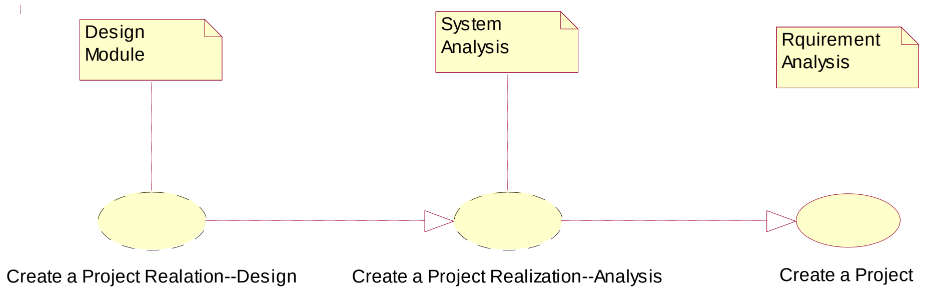
Design Module
System Analysis
Rquirement Analysis
Create a Project Realation--Design
Create a Project Realization--Analysis
Create a Project

Fig 3. UML description of VTM developing.

2.7 User Interface

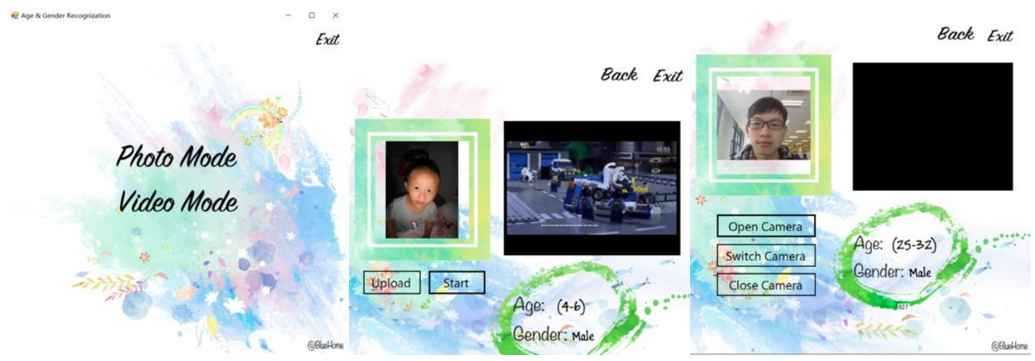

Fig 4. User interface. From left to right: start interface, photo interface, video interface.

3. Software Design

3.1 Software model

The architecture of the application is represented following the recommendations of the Rational Unified Process and the Rational Architecture Practice guidelines. And this document presents the architectural as a series of views:

1) Use Case View
2) Logical View
3) Process View
4) Implement View
5) Deploy View

3.2 Software development tools

1） Caffe open-source framework

2） Visual Studio 2013

3.3 Architectural design

3.3.1 Use-Case View

A description of the use-case view of the software architecture. The Use Case View is important input to the selection of the set of scenarios and/or use cases that are the focus of an iteration. The functionality of software is captured in the use-case diagram in Fig 5.

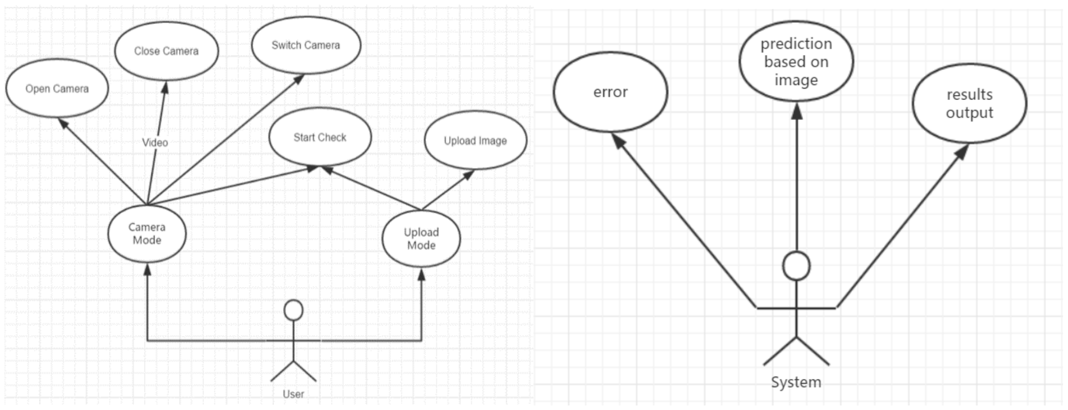

Fig 5. Use-case diagram.

All implemented use cases have an associated Use Case Specification document. References to these documents can be found in the same directory.

The architecturally-significant use cases are those, that “exercise” the most critical parts of the system architecture and demonstrate the core system functionality. In this system they are:

1. Open Camera (Upload Image): This use case allows user to input an image to Back-end detection system. When a develop group apply for using this system

to get image's age and gender, this use case starts.

Basic Scenarios:

a) User uploads an image.

b) User Open Camera

c) System receives the image and estimates the image's age and gender.

2. Start check and obtain results: This use case allows the software to check images and show the output on the interface.

Basic Scenarios:

a) User uploads an image.

b) Software estimates the image.

c) Users click the start button

d) Software output an results.

Actually, you can reference to the sequence diagrams and collaboration diagrams of these very important use cases in the analysis model.

3.3.2 Logical View

This section describes the architecturally significant parts of the design model, such as its decomposition into subsystems and packages. It describes the logical structure of the system. It starts from the overview of the architecture and then presents its key structure, behavioral elements and mechanisms.

There are three dominant structures in the application design model:

a) Logical decomposition of the system into three layers.

b) The structure of the use case realizations derived from model templates of architectural mechanisms.

c) The design mechanisms package contains a pre-designed solution to a common problem.

The high-level diagram of above is showed in Fig 6.

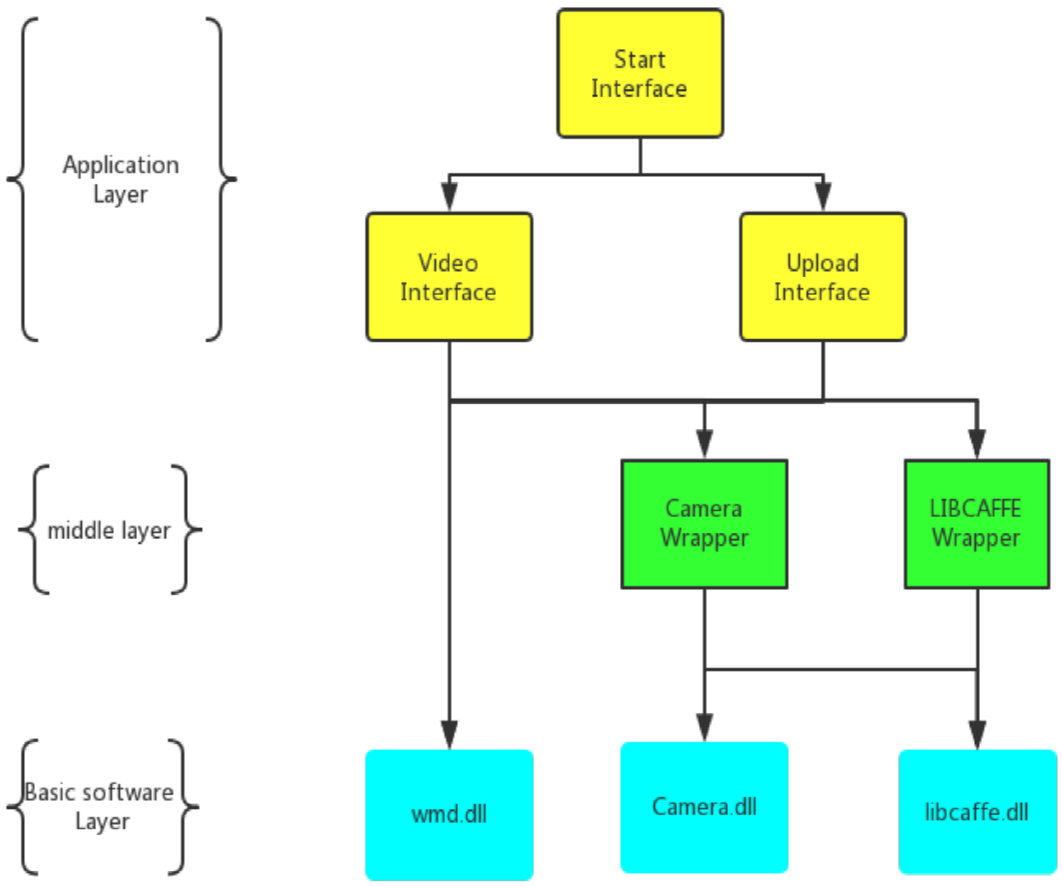

Fig 6. High-level diagram of structures.

The three layers are introduced below:

1) Basic software Layer: This layer have some basic function and algorithm of software including caffe and camera.

   a) libCaffe.dll : including all caffe’s core function and interface

   b) Camera.dll: including all camera’s function.

   c) Wmd.dll: including windows media player ‘s function

2） Middle layer: The function let C# can call C++

a) libcaffeWapper: Convert memory from C++ to C#., let C# can call libCaffe .dll’s funtion

b) CameraWrapper: Convert memory from C++ to C#., let C# can call camera.dll’s funtion

3） Application Layer: This Layer complete interface and function’s call logical

a) Start interface: include start interface and it can call Video interface and upload interface.

b) Video interface: it complete video interface, and can recognition age and gender real time and back to start interface.

c) Upload interface: it complete Upload interface, and can recognition age and gender image and back to start interface.

3.3.3 Process View

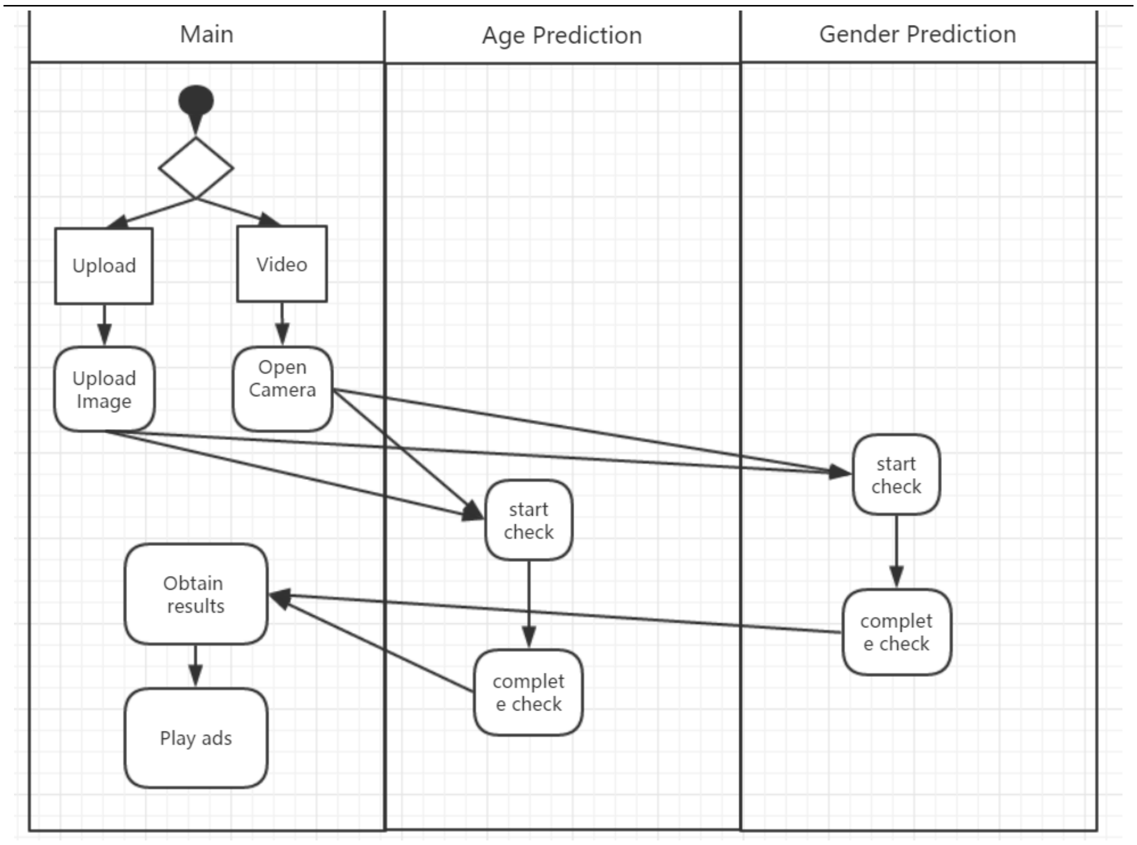

Fig 7. Process view.

3.3.4 Deployment View

The deployment view of a system shows the physical nodes on which the system executes and the assignment of the system processed to the nodes. The diagram below shows the most typical deployment configuration used by the development team.

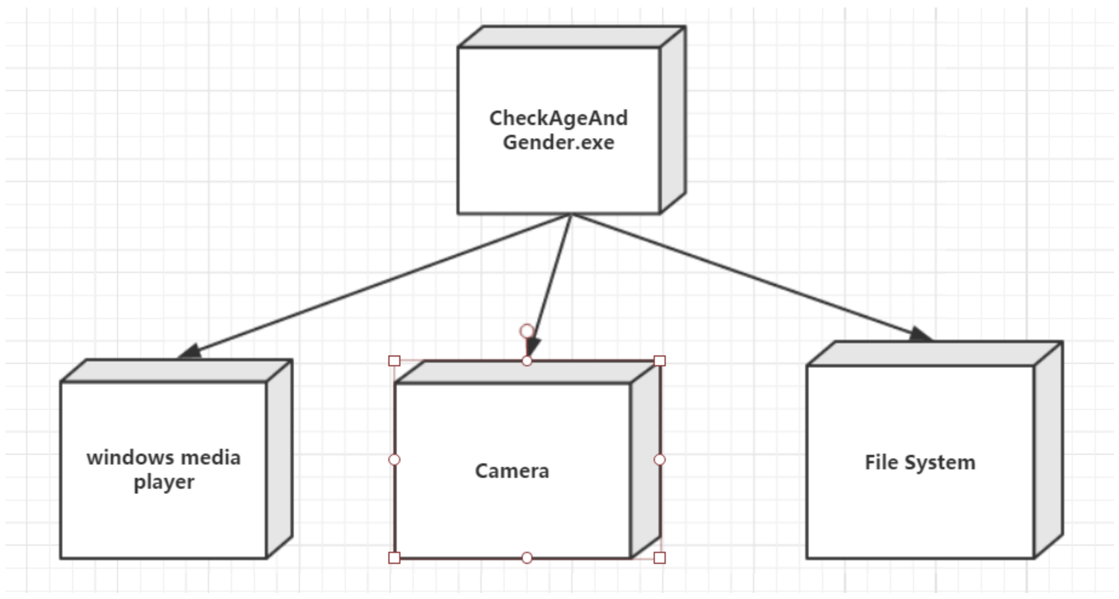

Fig 8. **Deployment View**

3.3.5 Implementation View

Organized by IDE, in this system, we use Visual Studio.

3.4 **Object identification**

4 Use-case Specification

4.1 Open Camera Use-Case Specification

4.1.1 Definition

This is the requirement description for the open camera use case. This use case is for user to open his own camera.

4.1.2 Preconditions and Post Conditions

4.1.2.1 Preconditions

The user open the camera and enter Video mode.

4.1.2.2 Post Conditions

When user click the close camera button ,the user case is over.

4.1.3 Basic Scenarios

1) User click the Video button

2) User click the Open Camera Button

3) Computer open Camera and show it.

4) Software check face’s age and prediction real time.

4.1.4 Exceptions or Branches

System have no camera or camera has been occupied by other process, the software may have no image to show.

4.1.5 Note

Null

4.2 Switch Camera Use-Case Specification

4.2.1 Definition

This is the requirement description for the switch camera use case. This use case is for user to switch his own camera to another camera.

4.2.2 Preconditions and Post Conditions

4.2.2.1 Preconditions

The user open the camera and enter Video mode.

4.2.2.2 Post Conditions

When user click the close camera button , the user case is over.

4.2.3 Basic Scenarios

1) User click the Video button

2) User click the Open Camera Button

3) User click the Switch Camera Button

4) Computer switch to another Camera and show it.

5) Software check face’s age and prediction real time.

4.2.4 Exceptions or Branches

System have no camera or only one camera, it will stay origin picture

4.2.5 Note

Null

4.3 Close Camera Use-Case Specification

4.3.1 Definition

This is the requirement description for the close camera use case. This use case is for user to close his own camera.

4.3.2 Preconditions and Post Conditions

4.3.2.1 Preconditions

The user open the camera and enter Video mode.

4.3.2.2 Post Conditions

Camera be closed.

4.3.3 Basic Scenarios

1) User click the Video button

2) User click the Open Camera Button

3) User click the Close Camera Button

4) Computer close the camera.

4.3.4 Exceptions or Branches

System have no camera or camera has been closed it will stay origin picture

4.3.5 Note

Null

4.4 Upload Image Use-Case Specification

4.4.1 Definition

This is the requirement description for the upload image use case. This use case is for user to upload his own image to check.

4.4.2 Preconditions and Post Conditions

4.4.2.1 Preconditions

The user enter Upload mode.

4.4.2.2 Post Conditions

The user exit this mode

4.4.3 Basic Scenarios

1) User click the Upload mode button

2) User click the Upload Image Button

3) Choose one image to upload

4) Image show on the screen.

4.4.4 Exceptions or Branches

User open the diagram but not choose any image, the software will show error information.

4.4.5 Note

Null

4.4 Upload Image Use-Case Specification

4.4.1 Definition

This is the requirement description for the upload image use case. This use case is for user to upload his own image to check.

4.4.2 Preconditions and Post Conditions

4.4.2.1 Preconditions

The user enter Upload mode.

4.4.2.2 Post Conditions

The user exit this mode

4.4.3 Basic Scenarios

1) User click the Upload mode button
2) User click the Upload Image Button
3) Choose one image to upload
4) Image show on the screen.

4.4.4 Exceptions or Branches

User open the diagram but not choose any image, the software will show error information.

4.4.5 Note

Null

4.5 Start Check Use-Case Specification

4.5.1 Definition

This is the requirement description for the start check use case. This use case is for user to check the age and gender of image's face .

4.5.2 Preconditions and Post Conditions

4.5.2.1 Preconditions

The user Open the software and choose any mode.

4.5.2.2 Post Conditions

The user exit mode

4.5.3 Basic Scenarios

1) User click the Upload mode button or Video mode
2) User click the Upload Image Button or Open camera
3) User click the Start button(in upload mode)
4) Image result show on the screen.

4.5.4 Exceptions or Branches

User have not run it in the rights of administrators, and system will reminder it.

4.5.5 Note

Null

5 Testing

5.1Test plan

5.1.1 Definition

1) L0: DLL test. We will test the basic function of SDK built by ourselves.
2) L1: Invoking test. We will test if C# can invoke the function of SDK without error.
3) L2: Integration test. We will test if the whole software can run without error.
4) HAPPY_PATH: The test of user running the software procedure successfully.
5) SAD_PATH: The test of user running the software in exception process.

Test Entry	If Passed
L0	✔
L1	✔
L2	✔
HAPPY_PATH	✔
SAD_PATH	✔
Compatibility Test	✔
Stability Test	✔
Performance Test	✔
Speed Test	✔
Memory Test	✔

Table 2. The test plan.

5.2Test-design specification

5.2.1 L0 Test (DLL Test)

Test Entry	Method	Prediction	If Passed
Functional Test	Call DLL by C++	Return the age and gender of the test image	✔
Memory Test	Recursively call DLL for 100 times and observe	The memory difference is almost	✔

	the difference between the real-time memory and the initial memory	same in each recursion	
Stability Test	Same as before	DLL works normally	✔
Unit Test	Test every function in the DLL	All functions work normally	✔
Exceptional Test	Send empty image to the DLL	Programs does not crash	✔

Table 3. L0 test specification

5.2.2 L1 Test (C# Calls)

Test Entry	Method	Prediction	If Passed
Functional Test	Call DLL by C#	Return the age and gender of the test image	✔
Memory Test	Recursively call DLL for 100 times and observe the difference between the real-time memory and the initial memory	The memory difference is almost same in each recursion	✔ (little memory leak)
Stability Test	Same as before	DLL works normally	✔
Exceptional Test	Send empty image to the DLL	Programs does not crash	✔

Table 4. L1 test specification

5.2.3 L2 Test (Integration Test)

Test Entry	Method	Prediction	If Passed
Functional Test	Call DLL by C++	Return the age and gender of the test image	✔
Memory Test	Recursively call DLL for 100 times and observe the difference between the real-time memory and the initial memory	The memory difference is almost same in each recursion	✔ (little memory leak)
Stability Test	Same as before	DLL works normally	✔
Exceptional Test	Send empty image to the DLL	Programs does not crash	✔
Performance Test	Test the accuracy in backlights, sidelights, normal lights and dusky environment	Programs does not crash	✔

Table 5. L2 test specification

5.3Test-case specification

5.3.1 L0 Test

Test Case	Input	Output	Prediction	If passed
Image Prediction Test	Image path	Image classification and confidence	Return the class and confidence of test image	✔
Model Upload Test	Model file	Return 1 if model uploaded successfully	Return 1	✔

Table 6. L0 functional test.

Test Case	Input	Output	Prediction	If passed
Image Prediction Test	Other file type path	Error	Return error message	✔
Image Prediction Test	No existing path	Error	Return error message	✔
Model Upload Test	No existing path	Error	Return error message	✔
Model Upload Test	Other file type path	Error	Return error message	✔

Table 7. L0 unit test.

Test Method	Prediction	If passed
Recursively call DLL for 100 times and observe the difference between the real-time memory and the initial memory	The memory difference is almost same in each recursion and DLL works.	✔

Table 8. L0 memory and stability test.

5.3.2 L1 Test

Test Case	Input	Output	Prediction	If passed
Image Prediction Test	Image path	Image classification and confidence	Return the class and confidence of test image	✔
Model Upload Test	Model file	Return 1 if model uploaded successfully	Return 1	✔

Table 9. L1 functional test.

Test Case	Input	Output	Prediction	If passed
Image Prediction Test	Other file type path	Error	Return error message	✔
Image Prediction Test	No existing path	Error	Return error message	✔

Model Upload Test	No existing path	Error	Return error message	✔
Model Upload Test	Other file type path	Error	Return error message	✔

Table 10. L1 unit test.

Test Method	Prediction	If passed
Recursively call DLL for 100 times and observe the difference between the real-time memory and the initial memory	The memory difference is almost same in each recursion and DLL works.	✔

Table 11. L1 memory and stability test.

5.3.3 HAPPY_PATH Test

Test Case	Prediction	If passed
Click Photo Mode	Enter photo mode interface	✔
Click Upload	Show the file selection box and the chosen image in the interface	✔
Click Start	Show the prediction results and play the advertisement	✔
Click Back	Enter main interface	✔
Click Exit	Exit the software without any remaining process	✔
Click Video Mode	Enter video mode interface	✔
Click Open Camera	The camera is opened and show in the main interface. The software predict the age and gender in real-time video and play the advertisement	✔
Click Switch Camera	Switch the camera, remain the same if only one camera	✔
Click Close Camera	The camera is closed	✔
Click Back	Enter main interface	✔
Click Exit	Exit the software without any remaining process	✔

Table 12. HAPPY_PATH test.

5.3.4 SAD_PATH Test

Test Case	Prediction	If passed
Click Upload without uploaded image	Pop message box to inform user while the software works well	✔
Click Start without any image	Pop message box to inform user while the software works well	✔
Click Open Camera while camera opened	The software works well	✔
Click Switch Camera	The software works well	✔

when only one camera		
Click Close Camera while camera closed	The software works well	✔

Table 13. SAD_PATH test.

5.3.5 Compatibility Test

WIN10	Win7	Win8.1
✔	✔	✔

Table 14. Compatibility test.

5.3.6 Speed Test

Processor	I5 4210M	MAC PRO	I3 4000M
Memory	8G	MAC PRO	4G
Speed(ms)	380	350	410

Table 15. Speed test.

5.3.7 Performance Test

Test Environment	Success Times(m/n)	Accuracy
Normal light	5/5	100%
Backlight	2/5	40%
Sidelight	3/5	60%
Dusky	2/5	40%

Table 16. Performance test.

5.4Test-procedure specification

L0 test code is written in C++, enclosure in the folder.

L1, L2 test code is written in C#, enclosure in the folder as well.

Others **are** all software test.

5.5Test Conclusion

When we done all test be mentioned above, we verify that our software have only little memory leak, it ‘s not influence software’s use, besides, we also find that our software have a satisfied speed(0.4s) in I3 CPU, this means our software can run a low configure computer. From the results of performance testing, our software have better performance in the normal light compared to the Backlight, side light, dim. It gives us direction to optimize it. In general, our software passed the whole test, it have little memory leak, and it can run in x64 platform windows including win7/win8.1/win10. It can complete the function that play ads based on age and prediction

6 Conclusion

It is the first time for us to design and implement a software from scratch. We meet a lot of difficulties in the process, technical or untechnical. Some technical problem can be solved easily by using the third library, but we choose to figure it out by ourselves. This project not only improve our engineering skill, but also teach us how to design the framework and workflow in software engineering, as well as work effective in a team.

7 References

[1] Y. H. Kwon and N. da Vitoria Lobo. Age classification from facial images. In Proc. Conf. Comput. Vision Pattern Recognition, pages 762–767. IEEE, 1994. 1, 2

[2] L. Rabiner and B.-H. Juang. An introduction to hidden markov models. ASSP Magazine, IEEE, 3(1):4–16, 1986. 2

[3] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski. Sexnet: A neural network identifies sex from human faces. In Neural Inform. Process. Syst., pages 572–579, 1990. 2

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Neural Inform. Process. Syst., pages 1097–1105,

2012. 3, 4

[5] Gil Levi and Tal Hassner, Age and Gender Classification using Convolutional Neural Networks, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, June 2015

[6] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv technical report, 2014