

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

Software Project II

AD Project

프로젝트 명	imagerator
팀 명	태성이와 현승이의 입대 전 발악
문서 제목	최종보고서
git-hub-repo	https://github.com/hyeonseung-lee/swProject2_ADproject

팀원	20191653	이현승
	20191668	정태성

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagelator
	팀 명	태성이와 현승이의 입대 전 발악

목차

1. 프로젝트 소개
 - 1 - 1. 주제
 - 1 - 2. 요구사항 수집, 분석
 - 1 - 3. 역할분담
2. 수행 내용
 - 2 - 1. API 선정 과정
 - 2 - 2. 소프트웨어 구조 설계
 - 2 - 3. 소프트웨어 상세 설계
 - 2 - 4. 구현 (coding)
3. 테스트
 - 3 - 1. 단위 테스트
 - 3 - 2. 시스템 통합
 - 3 - 3. 통합 테스트
4. 결과
 - 4 - 1. 시연
 - 4 - 2. 후기

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

1. 프로젝트 소개

1 - 1. 주제

이번 학기에 들어서 선형대수(linear algebra)라는 과목을 수강한다. 이 강의에서는 영어로 집필된 원서를 이용하는데, 모르는 단어나 의미파악이 어려운 문장을 마주하게 되면 곤혹을 치르기 마련이다. 우리는 이런 상황에 닥칠 때 마다 인터넷 사전에 타이핑을 하는 번거로운 일을 겪는다. 이런 상황에는 강의 교안을 통째로 번역하고 싶다는 욕망을 느끼게 된다. 우리는 이런 욕망을 해소하고자 이미지를 바로 번역해주는 프로그램을 짜자는 아이디어를 떠올렸다. 강의교안을 보다가 모르는 부분을 캡처하고, 이 이미지를 프로그램에 업로드만 해주면 바로 목적언어로 번역되어 나오는 프로그램을 구상했다.

따라서 이 프로그램의 이름은 image 에 translator를 더한 image-lator라 칭한다.

1 - 2. 요구사항 수집, 분석

이미지를 불러오고, 이미지에서 텍스트를 추출한다. 이 텍스트를 목적 언어로 번역하는 기능이 필요하다.

– 요구사항 명세서 (Software Requirement Specification, SRS)

1. 기능적 요구사항

1-1. User Desktop에서 target image를 load하는 기능이 우선되어야한다.

1-2. image에서 text를 추출해내는 기능이 필요하다. 이 기능을 광학 문자 인식(Optical Character Recognition, OCR)으로 구현한다.

1-3. 인식된 Text가 어떤 language 기반으로 작성된 언어인지 인지하는 기능이 필요하다. 언어감지(language detection)

1-4. 어떤 언어로 번역할 것인지에 대해서 setting 하는 기능이 필요하다. 이를 target language라 칭한다.

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

1-5. 언어감지에서 인식된 language를 기준으로 text를 target language로 번역하는 기능을 수행할 번역기가 필요하다.

2. 사용자 인터페이스 요구사항

2-1. 프로그램의 사용자 인터페이스를 구성하는 요소는 다음과 같다.

2-1-1. file open button

2-1-2. setting target language button

2-1-3. change button

2-1-4. translate button

2-1-5. clear button

2-1-6. text to translate screen

2-1-7. result screen

2-2. 각각의 button과 screen들은 이름이 명시하고 있는 바의 기능을 수행한다.

3. 비기능적 요구사항

3-1. 이 소프트웨어의 구현에는 python과 PyQt5를 이용한다.

3-2. 광학 문자 인식(Optical Character Recognition, OCR) 기능의 구현에는 python-tesseract를 사용한다.

3-2. 언어감지(language detection) 기능의 구현에는 Naver Developers 에서 제공하는 Papago 언어감지 API를 사용한다.

3-3. 번역을 수행할 번역기(translator) 기능의 구현에는 Naver Developers 에서 제공하는 Papago 번역 API를 사용한다.

이렇게 선정한 이유에 대해서는 2-1에서 다루도록 한다.

1 - 3. 역할분담

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 임대 전 발악

프로그램이 요구하는 바에 따라 역할을 나눠 진행하도록 한다.

이현승

- 언어 감지 기능 구현 (language Detect)
- 번역 기능 구현 (translate)
- 보고서 작성 (report)
- 제안발표, 결과발표 진행 (presentation)

정태성

- 광학 문자 인식 기능 구현 (Optical Character Recognition, OCR)
- PyQt GUI 구현 (Graphic User Interface, GUI)
- 제안발표, 결과발표 발표자료 제작 (PowerPoint)

분담된 역할은 배타적으로 진행하지 않고, 상호간의 교류를 원활히 하며 도움을 주고받으며 진행했다.

2. 수행 내용

2 - 1. API 선정과정

초기에 설정한 API는 총 3개였다.

2-1-1. 광학문자인식(OCR)

: google vision API를 이용하려고 했으나 이용 절차와 코드가 복잡하여 비교적 단순하고 쉽게 이용할 수 있는 python-tesseract를 사용했다.

2-1-2. 문자인식(language recognition)

: 다양한 API가 존재했지만 국내 회사가 제공하는 API가 편리할 것이라 판단하여 Naver Developers에서 제공하는 API를 사용하기로 결정했다.

2-1-3. 번역기(translator)

: 2-1-2와 동일.

2 - 2. 소프트웨어 구조 설계

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

- 소프트웨어 구조 설계 (Architecture Design)

모듈(module)	클래스(class)	역할(role)
main.py	MyWindow	사용자 인터페이스의 대부분 위젯을 포함하는 UI component
		프로그램 실행의 핵심
image.py	Image	대상 이미지를 보여줌
OCR.py	OCR	이미지에서 텍스트를 추출
DetectTranslator.py	DetectTranslator	목적언어 인식 후 번역

- 클래스 인터페이스 설계 (Class Interface Design)

클래스(class)	메서드(method)	입력 인자(input)	출력 인자(output)	기능
MyWin dow	setupUI	-	-	사용자 인터페이스의 대부분 위젯을 포함하는 UI component
				목적 언어(target language)를 설정
				다른 class들을 실행, 프로그램 실행의 핵심
	ButtonClicked	-	-	클릭된 버튼에 따른 기능 구현, 버튼 클릭시 실행됨
	keyPressEvent	e	-	Qt.Key_Escape 키가 눌리면 UI를 닫음
	textChanged	-	-	현재 입력된 텍스트의 글자 수를 구함

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

Image	-	data	-	image의 경로를 받아와 경로 초기화, initUI 실행
	initUI	-	-	초기화시 받아온 경로에서 이미지를 불러옴. 이 이미지를 보여주는 UI component 구현 및 실행
	keyPressEvent	e	-	Qt.Key_Escape 키가 눌리면 UI를 닫음
OCR	-	dir	-	추출 대상 이미지의 주소를 받아와 이미지 초기화
				python-tesseract를 사용, image를 입력받고 텍스트를 추출
	getString	-	filtered	추출한 텍스트를 한문장 단위로 list에 저장하고 이 list를 return
Detect Translator	translateText	text	trans_text	문자열과 목적언어를 입력받고 이 문자열이 어떤 언어로 구성되어 있는지를 인식, 이 문자열을 목적언어로 번역해서 return

- 소프트웨어 구조 설계서 (Architecture Design Specification, ADS)

모듈(module)	클래스(class)	역할(role)
main.py	MyWindow	사용자 인터페이스의 대부분 위젯을 포함하는 UI component
		프로그램 실행의 핵심
image.py	Image	대상 이미지를 보여줌
OCR.py	OCR	이미지에서 텍스트를 추출
DetectTranslator.py	DetectTranslator	목적언어 인식 후 번역

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

2 - 3. 소프트웨어 상세 설계

- 구현 상세 설계 (Detailed Design)

1. main.py에 번역할 문자와 번역된 문자를 담을 QTextEdit창과 기능을 수행하는 버튼들을 추가함
2. file open버튼 누르면 파일을 선택하는 창이 뜸
3. 사진의 경로를 얻어 이를 image.py와 OCR.py클래스에 경로값 전달
4. image클래스에서 받은 경로를 바탕으로 해당 이미지를 QPixmap함수를 이용해 화면에 실행
5. OCR클래스에서 받은 경로를 바탕으로 해당 이미지를 pytesseract를 이용해 추출하고 불필요한 빈칸을 정제하여 리스트로 리턴
6. 'translate'버튼을 눌러 번역을 수행함 DetectTranslator클래스에 번역할 언어 기값과 번역할 언어 값을 전송, 같은 언어로 번역을 시도하면 그대로 출력되도록 설정
7. DetectTranslator클래스에서 넘겨받은 텍스트의 언어를 감지하여 넘겨받은 기값의 언어로 번역하여 리턴
8. 리턴받은 값을 Result화면에 출력함

- 소프트웨어 상세 설계서 (Detailed Design Specification, DDS)

2 - 4. 구현 (coding)

main.py

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

```

1  # -*- coding: utf-8 -*-
2  import sys
3      from PyQt5.QtWidgets import *
4      from image import Image
5      from OCR import OCR
6      from DetectTranslator import DetectTranslator
7  from PyQt5.QtCore import Qt
8
9  class MyWindow(QWidget):
10     def __init__(self):
11         super().__init__()
12         self.setupUI()
13
14     def setupUI(self):
15         self.setGeometry(800, 200, 700, 700)
16         self.setWindowTitle("Imagerator")
17         self.setStyleSheet("background-color: white;")
18
19         lang = ['한국어', '영어', '중국어 간체', '중국어 번체', '스페인어', '프랑스어', '베트남어', '태국어', '인도네시아어']
20
21         self.original = QLabel("Text to Translate")
22         self.result = QLabel("Result")
23
24         self.changeButton = QPushButton("Change(F1)")
25         self.changeButton.setStyleSheet("border: 1px solid #D0D3D1")#버튼 테두리 색 변경
26         self.openButton = QPushButton("File Open(F2)") #파일 선택창을 띄우는 버튼
27         self.openButton.setStyleSheet("border: 1px solid #D0D3D1")
28         self.language = QComboBox()
29         self.language.setStyleSheet("border: 1px solid #D0D3D1")
30         self.clearButton = QPushButton("Clear(F3)")
31         self.clearButton.setStyleSheet("border: 1px solid #D0D3D1")
32         self.transButton = QPushButton("Translate(F4)")

```

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

```

35 self.changeButton.clicked.connect(self.ButtonClicked)
36 self.changeButton.setShortcut("F1")# f1을 누르면 change 작동 하도록 단축키 설정
37
38 self.openButton.clicked.connect(self.ButtonClicked)
39 self.openButton.setShortcut("F2") # f4를 누르면 파일 열기 창 실행
40
41 self.clearButton.clicked.connect(self.ButtonClicked)
42 self.clearButton.setShortcut("F3")# f3를 누르면 모두 지우기
43
44 self.transButton.clicked.connect(self.ButtonClicked)
45 self.transButton.setShortcut("F4") # f2를 누르면 번역
46
47 self.origin = QTextEdit("", self)
48 self.trans = QTextEdit("", self)
49
50 #테스트창 색 변경
51 self.origin.setStyleSheet("border: 2px solid #89fa7d;")
52 self.trans.setStyleSheet("border: 2px solid #89fa7d;")
53 self.transButton.setStyleSheet("""background:solid #59FF7A;
54 |                                border: 2px solid #59FF7A;
55 |                                color : white;""")
56
57 #입력된 글자가 변경되면 작동
58 self.origin.textChanged.connect(self.textChanged)
59 self.trans.textChanged.connect(self.textChanged)
60
61 for lan in lang: # 언어의 키값을 combo 에 담음
62 |     self.language.addItem(lan)
63
64 hlayout = QHBoxLayout()
65 buttonlayout = QVBoxLayout()
66 textlayout = QVBoxLayout()
67
68 buttonlayout.addWidget(self.language)

```

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

```

68     buttonlayout.addWidget(self.changeButton)
69     buttonlayout.addWidget(self.openButton)
70     buttonlayout.addWidget(self.clearButton)
71     buttonlayout.addWidget(self.transButton)
72
73     textlayout.addWidget(self.original)
74     textlayout.addWidget(self.origin)
75     textlayout.addWidget(self.result)
76     textlayout.addWidget(self.trans)
77     hlayout.addLayout(textlayout)
78     hlayout.addLayout(buttonlayout)
79
80     self.setLayout(hlayout)
81     self.show()
82
83     # 사진의 경로를 얻어오는 버튼
84     def ButtonClicked(self):
85         button = self.sender()
86         if button.text() == "File Open(F2)":
87             # 경로 선택창을 띄움
88             fname = QFileDialog.getOpenFileName(self)
89             # fname[0] == 사진의 경로
90             try:
91                 self.dir = fname[0]
92                 # Image 클래스에 사진 경로를 입력, 이미지를 화면에 출력함
93                 self.pix = Image(self.dir)
94                 # OCR 클래스에 사진 경로를 입력, 이미지의 텍스트값을 리스트로 리턴
95                 ocr = OCR(self.dir)
96                 self.sentences = ocr.getString()
97                 for i in self.sentences:
98                     self.origin.append(i)
99             except:
100                 pass

```

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagelator
	팀 명	태성이와 현승이의 입대 전 발악

```

101 elif button.text() == "Translate(F4)":
102     if self.origin.toPlainText() != '':
103         # 같은 언어 번역 시도시 발생하는 오류 제거
104         try:
105             translator = DetectTranslator(self.language.currentText())
106             self.trans.setText(translator.translateText(self.origin.toPlainText()))
107         except:
108             self.trans.setText("failed!" + self.origin.toPlainText())
109     else:
110         self.trans.setText("입력받은 문자가 없습니다!")
111
112 elif button.text() == "Clear(F3)":
113     self.origin.clear()
114     self.trans.clear()
115
116 elif button.text() == "Change(F1)":
117     save = self.origin.toPlainText()
118     self.origin.setText(self.trans.toPlainText())
119     self.trans.setText(save)
120
121 # esc 입력받으면 창을 종료
122 def keyPressEvent(self, e):
123     if e.key() == Qt.Key_Escape:
124         self.close()
125         try:
126             self.pix.keyPressEvent(e)
127         except:
128             pass
129
130 # 현재 입력된 글자수를 세줌
131 def textChanged(self):
132     self.original.setText("Text to Translate " + str(len(self.origin.toPlainText())) + "/" + "5000")
133     self.result.setText("Result " + str(len(self.trans.toPlainText())) + "/" + "5000")

```

DetectTranslator.py

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

```

1 import os
2 import sys
3 import json, pprint
4 import urllib.request
5 class DetectTranslator():
6
7     def __init__(self, targetLang):
8
9         self.client_id = "qzJp_knRrBfUxE3Hly" # client ID received from naver developer center.
10        self.client_secret = "3DRB_k92uK" # client secret receiver from naver developer center.
11
12        self.trans_url = "https://openapi.naver.com/v1/papago/n2mt"
13        self.detect_url = "https://openapi.naver.com/v1/papago/detectLangs"
14
15        self.target_langCode = {"한국어": "ko",
16                                "영어": "en",
17                                "중국어 간체": "zh-CN",
18                                "중국어 번체": "zh-TW",
19                                "스페인어": "es",
20                                "프랑스어": "fr",
21                                "베트남어": "vi",
22                                "태국어": "th",
23                                "인도네시아어": "id"}
24        # possible combinations : ko->en, ko->zh-CN, ko->zh-TW, ko->es, ko->fr, ko->vi, ko->th, ko->id, en->ja, en->fr
25
26        self.target_Lang = self.target_langCode[targetLang] # set target code
27
28    def translateText(self, text):
29        ##### detect language #####
30        detect_encQuery = urllib.parse.quote(text)
31        detect_data = "query=" + detect_encQuery

```

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

```

33     # request to WEB
34     detect_request = urllib.request.Request(self.detect_url)
35     detect_request.add_header("X-Naver-Client-Id", self.client_id)
36     detect_request.add_header("X-Naver-Client-Secret", self.client_secret)
37
38     # get result
39     detect_response = urllib.request.urlopen(detect_request, data=detect_data.encode("utf-8"))
40
41     # successful responds
42     detect_rescode = detect_response.getcode()
43     if (detect_rescode == 200): # success
44         response_body = detect_response.read()
45         detect_data = response_body.decode('utf-8')
46         detect_data = json.loads(detect_data)
47         #print("This language code : ", detect_data["langCode"])
48
49     else: # fail
50         print("Error Code:" + detect_rescode)
51
52
53     ##### translaor #####
54     trans_encText = urllib.parse.quote(text)
55     srcLang = detect_data["langCode"] # setting source from language detector
56     tarLang = self.target_Lang # setting target language from user selected
57     trans_data = "source={}&target={}&text=".format(srcLang, tarLang) + trans_encText
58
59
60     # request to WEB
61     trans_request = urllib.request.Request(self.trans_url)
62     trans_request.add_header("X-Naver-Client-Id", self.client_id)
63     trans_request.add_header("X-Naver-Client-Secret", self.client_secret)
64
65     # get result
66     trans_response = urllib.request.urlopen(trans_request, data=trans_data.encode("utf-8"))
67
68     # successful responds
69     trans_rescode = trans_response.getcode()
70
71     if(trans_rescode==200): # success
72         trans_response_body = trans_response.read()
73         trans_data = trans_response_body.decode('utf-8')
74         trans_data = json.loads(trans_data) # to dictionary
75         #pprint.pprint(data) # data print at looks good
76
77         trans_text = trans_data['message']['result']['translatedText']
78
79     else: # fail
80
81         print("Error Code:" + trans_rescode)
82
83
84     return trans_text

```

image.py

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

```

1  #-*- coding: utf-8 -*-
2  import sys
3  from PyQt5.QtWidgets import *
4  from PyQt5.QtGui import *
5  from PyQt5.QtCore import Qt
6
7  class Image(QMainWindow):
8      def __init__(self, data): #data = 파일의 경로
9          super().__init__()
10         self.dir = data
11         self.initUI()
12         self.setWindowTitle('Image')
13
14
15     def initUI(self):
16         self.lbl = QLabel(self)
17         pixmap = QPixmap(self.dir) #파일 경로에 해당하는 이미지를 불러옴
18         height, width = pixmap.height(), pixmap.width() #이미지의 크기를 구함
19         self.setGeometry(800 - width, 200, width, height) #사진을 프로그램 옆에 띄움
20         self.lbl.resize(width, height)
21         self.lbl.setPixmap(QPixmap(pixmap))
22
23         self.show()
24
25     def keyPressEvent(self, e):
26         if e.key() == Qt.Key_Escape:
27             self.close()

```

OCR.py

```

1  #-*- coding: utf-8 -*-
2
3  from pytesseract import *
4  from PIL import Image
5
6  class OCR:
7      def __init__(self, dir):
8          img = Image.open(dir) #경로를 받아와 이미지를 엽니다
9          #pytesseract를 이용해 이미지의 단어를 추출 lang은 추출할 언어(한국+영문)
10         self.text = pytesseract.image_to_string(img, lang='kor+eng')
11
12     def getString(self):
13         #전체 텍스트를 문장별로 분리함
14         sentence = self.text.split('\n')
15
16         filtered = []
17
18         #부자연스러운 띄어쓰기를 제거하고 단어들만 리스트에 담음
19         for i in sentence:
20             if len(i) != 0:
21                 filtered.append(i)
22
23         return filtered

```

3. 테스트

국민대학교 소프트웨어융합대학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

3 - 1. 단위 테스트 -> 단위테스트보고서(UTR)

Test Case	Unit Under Test	Description	Results
UT 1-1	OCR.getString	문자 추출 기능 테스트	pass
UT 1-2	DetectTranslator.TranslateText	영문 번역 기능 테스트	pass
UT 1-3	DetectTranslator.TranslateText	한글 번역 기능 테스트	pass

3 - 2. 시스템 통합 -> 소프트웨어 패키지

3 - 3. 통합 테스트 -> 통합 테스트 보고서(ITR)나와야함.

4. 결과

4 - 1. 시연

실행화면

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악

Text to Translate 0/5000

한국어

Change(F1)

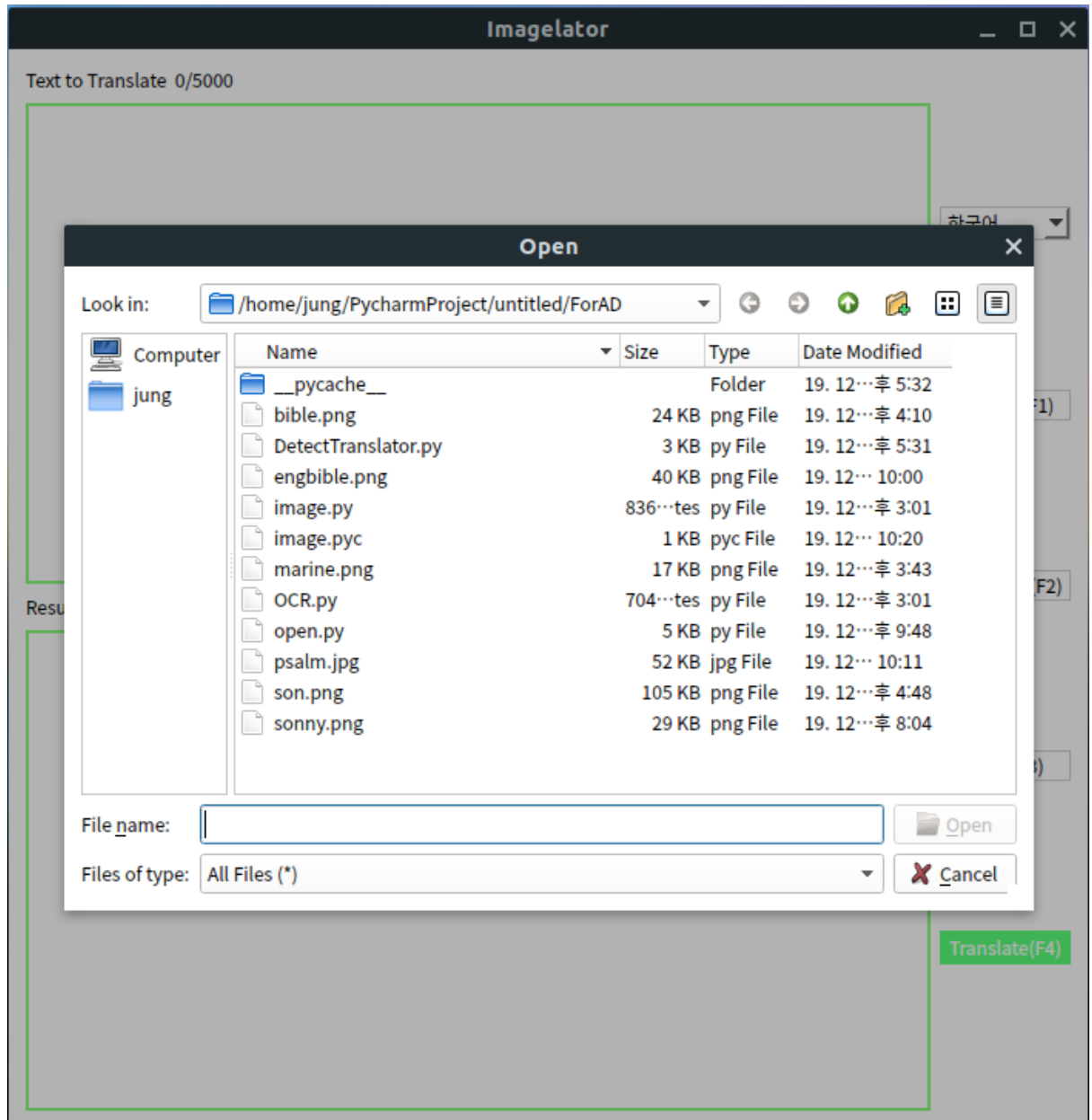
File Open(F2)

Clear(F3)

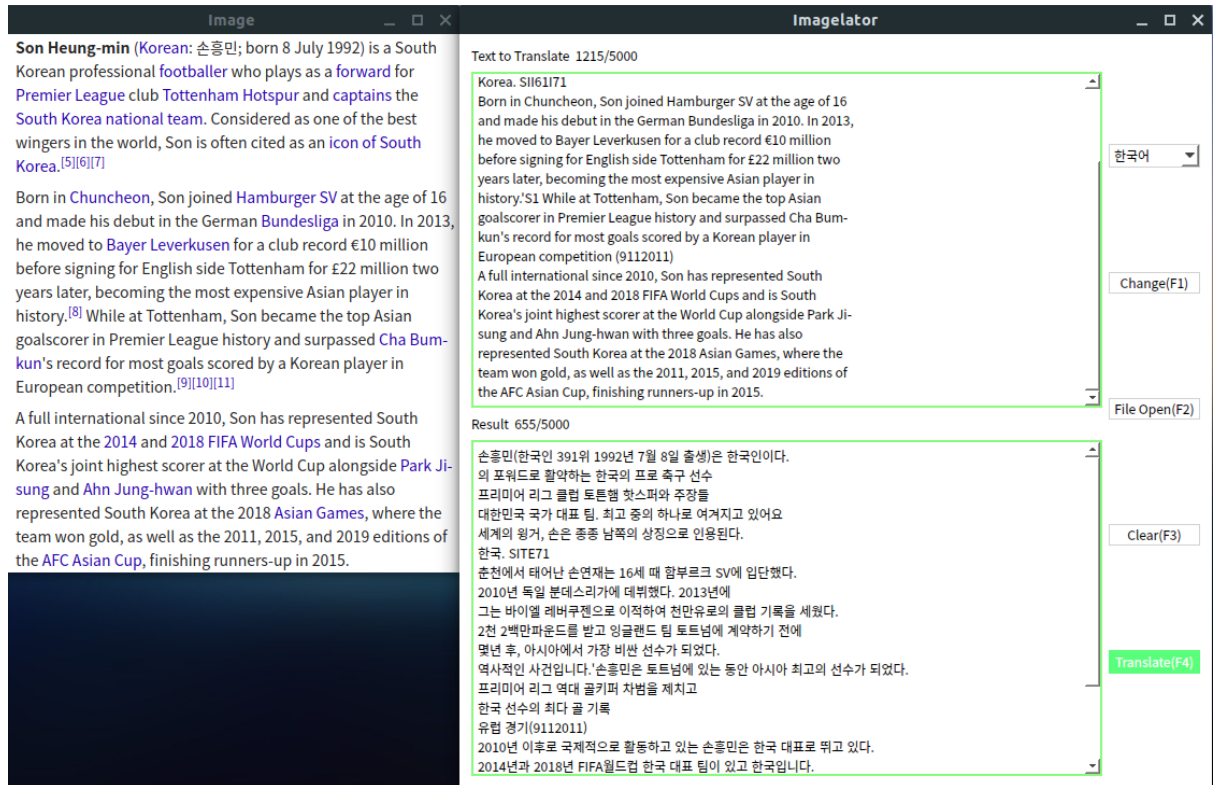
Translate(F4)

Result 0/5000

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imaelator
	팀 명	태성이와 현승이의 입대 전 발악



국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagerator
	팀 명	태성이와 현승이의 입대 전 발악



4 - 2. 후기

프로그램을 완성하고 시현해보니 pytesseract의 정확도가 기대만큼 높지 못한점이 아쉬웠다. 일부 알 수 없는 문자로 인식되어 표시되는 문제 때문에 papago를 이용한 번역까지 정확하게 이뤄지지 못하는 문제가 종종 발생하여 기대한만큼의 완성도를 구현하지 못했다. 인상깊었던 점은 목표했던 기능을 구현하는 시간보다 보기 좋게 UI를 개선하고 글자수를 세주거나 단축키를 이용하여 작동하게 만드는등의 사소하다고 생각한 부분들을 개선하고 추가하는 시간이 더 많이 소요되었는데 본래 목표했던 기능이 완전히 구현된 상태임에도 개선된 UI와 추가된 단축키들이 프로그램을 이용할때 느끼는 불편함을 해소해주며 프로그램이 편리하고 도움을 준다는 인상을 주는데 있어 큰 역할을 해주었고 사용하는 사람에게 편리함을 주는 요소는 프로그램의 주된 기능이외에도 다양한 부분이 열려있음을 알 수 있게 해준 부분이었다. 이번 AD프로젝트를 진행하며 전체적인 후기는 외부 라이브러리를 이용해 다양한 기능 구현이 가능함을 알 수 있었고 다양한 정보수집과 활용을 이용해 1학년임에도 다양한 기능을 가진 프로그램을 구현했다는 점에서 유의미한 경험을 했다 생각한다.

국민대학교 소프트웨어융합대 학	최종보고서	
	프로젝트 명	imagelator
	팀 명	태성이와 현승이의 입대 전 발악