
초고속망 최종 보고서

[프로젝트명: Image Analysis And Translation]

2019. 06. 11

성명	학번
상 매	201910578
손 흠	201910579

CONTENTS

목차

1

프로젝트 계획서

2

프로젝트 구조 다이어그램

3

Ocr/text translate service create

4

Building code and page design

5

Result analysis

프로젝트 계획서

프로젝트명: Image Analysis And Translation

프로젝트 목적 [개요와 해결할 문제를 기술한다]

Microsoft Cognitive Services include five fields: image, voice, language, search and knowledge. Through the independent or combined use of these cognitive services, many real world problems can be solved.

The content of our project is to use Microsoft Cognitive Service OCR and text translate to translate the Image into the language we want to translate. Our project is to translate English or Japanese into Chinese.

프로젝트 범위 [개발할 기능을 중심으로 기술한다]

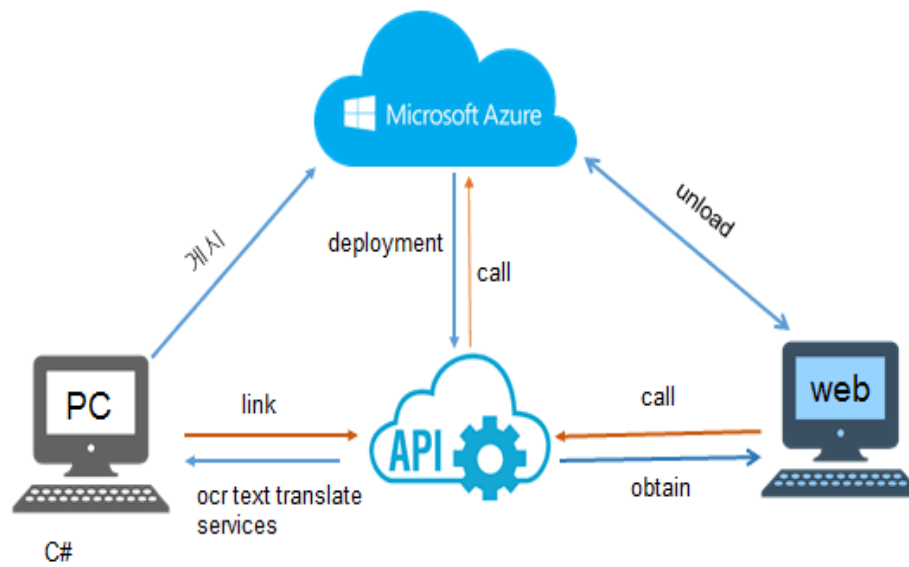
기능:

1. OCR (Optical Character Recognition) service can recognize all the text on the image.
2. Text Translate(Text Translation) can translate Japanese or English into Chinese.

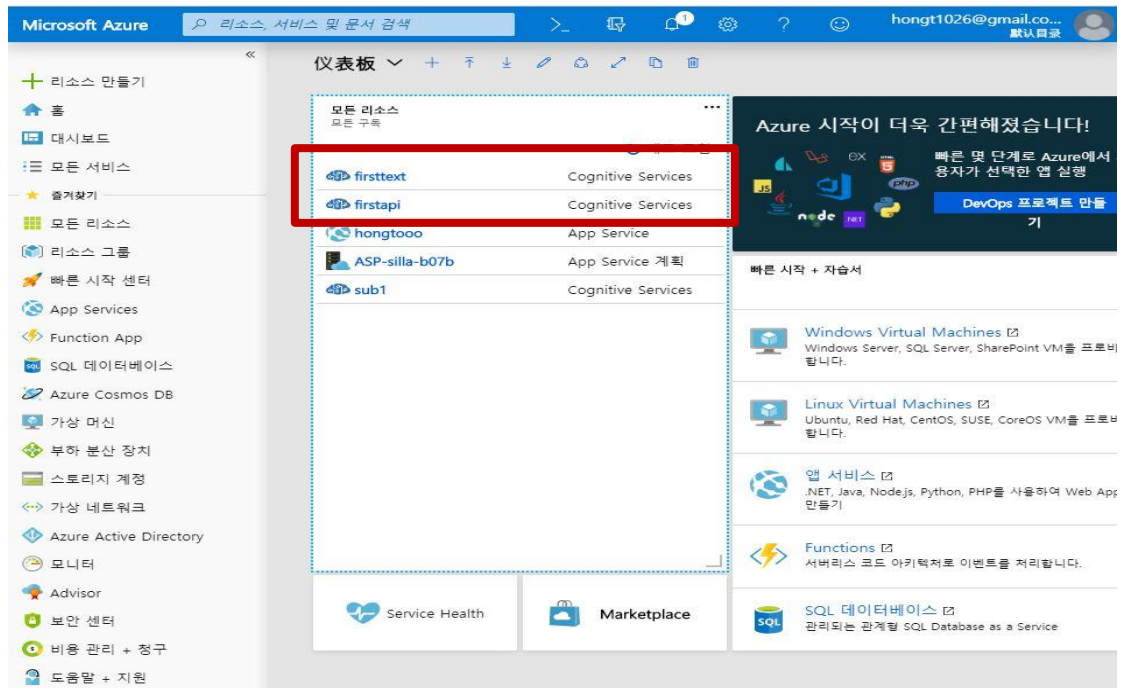
팀원 이름 / 학번	역할	
상 매/201910578	디자인, 코딩, 테스트	
손 홈/201910579	설계, 코딩	

개정 현황				
V1.0	2019. 06. 10	최종 작성	상매/손홈	90%
개정번호	일자	변경 사유	작성자	승인

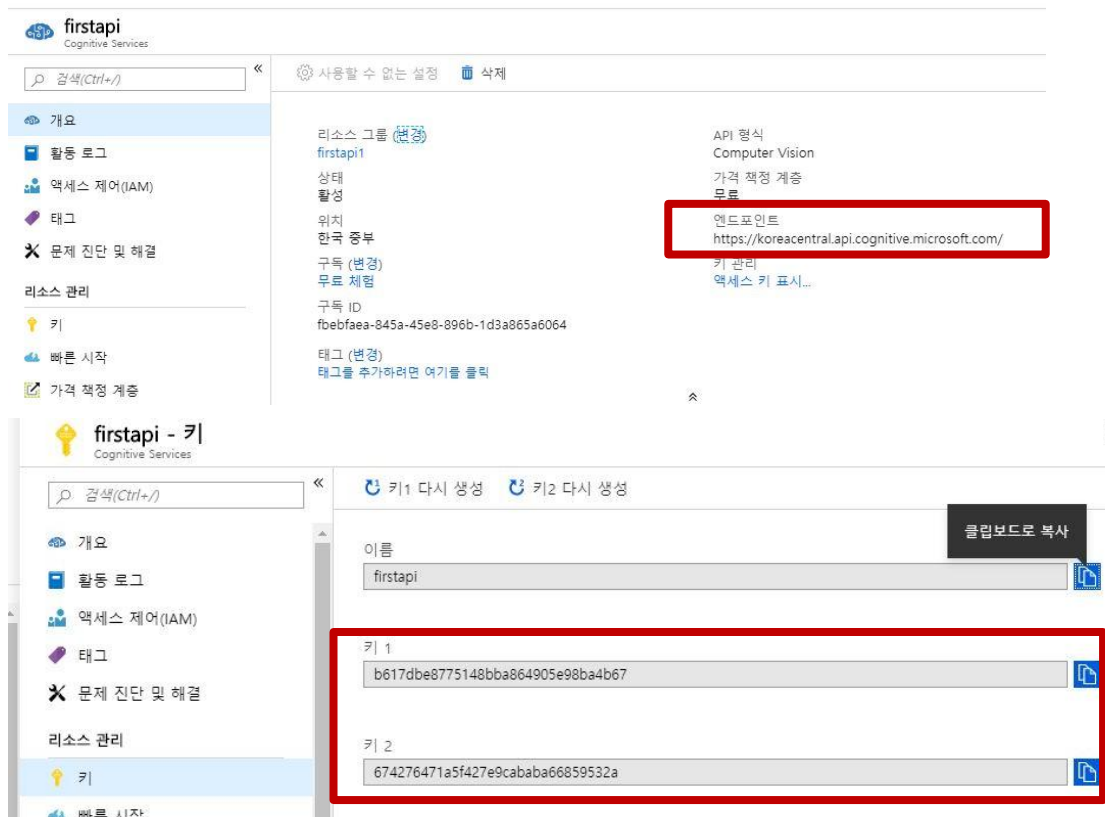
프로젝트 구조 다이어그램



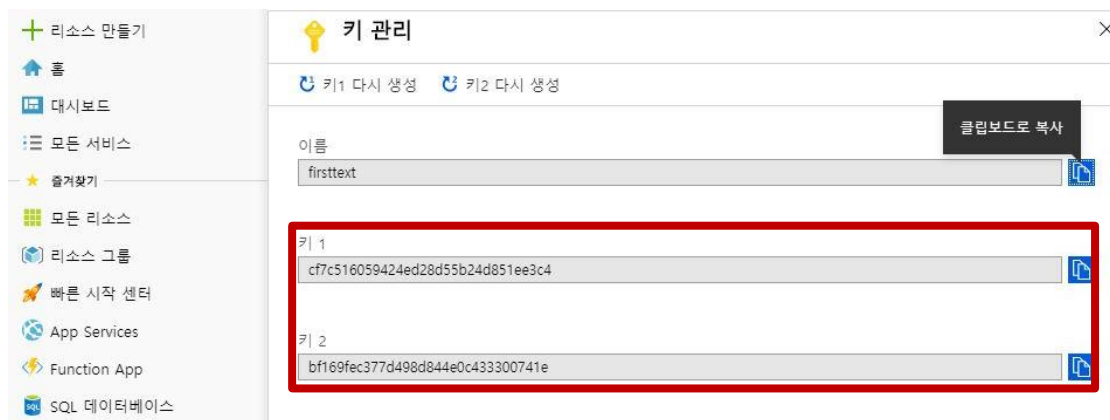
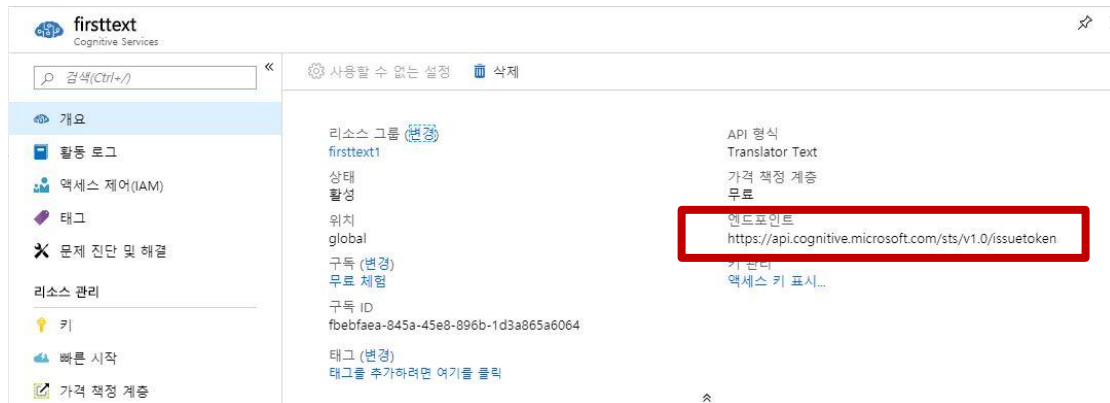
Ocr and text translate create on mircrosoft azure



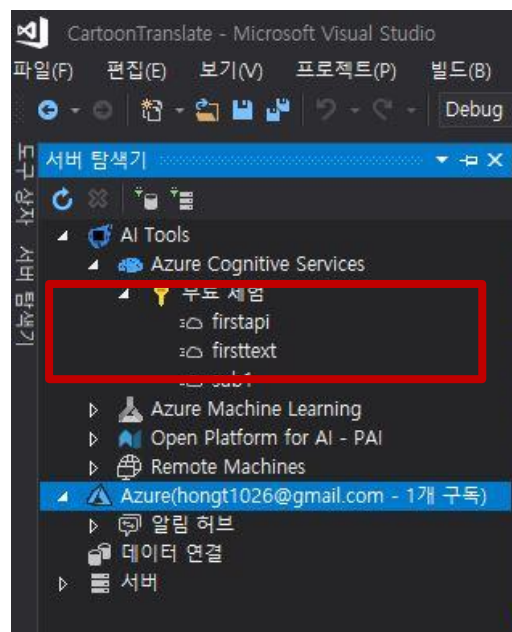
1.Applocation for ocr service Key and endpoint.



2. Application for text translate text translation service Key and endpoint

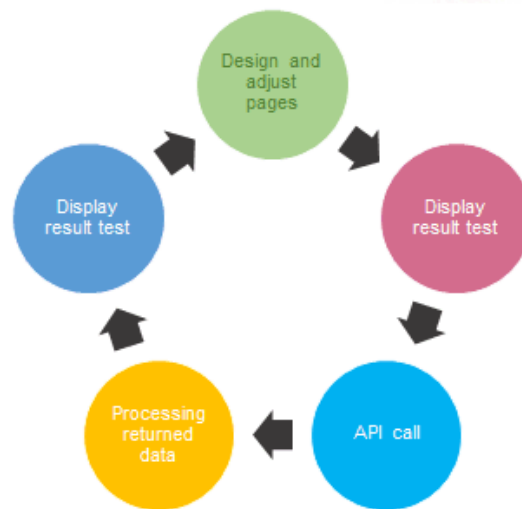


3. Installation Microsoft Visual Studio Tools for AI on Visual Studio 2017 Community

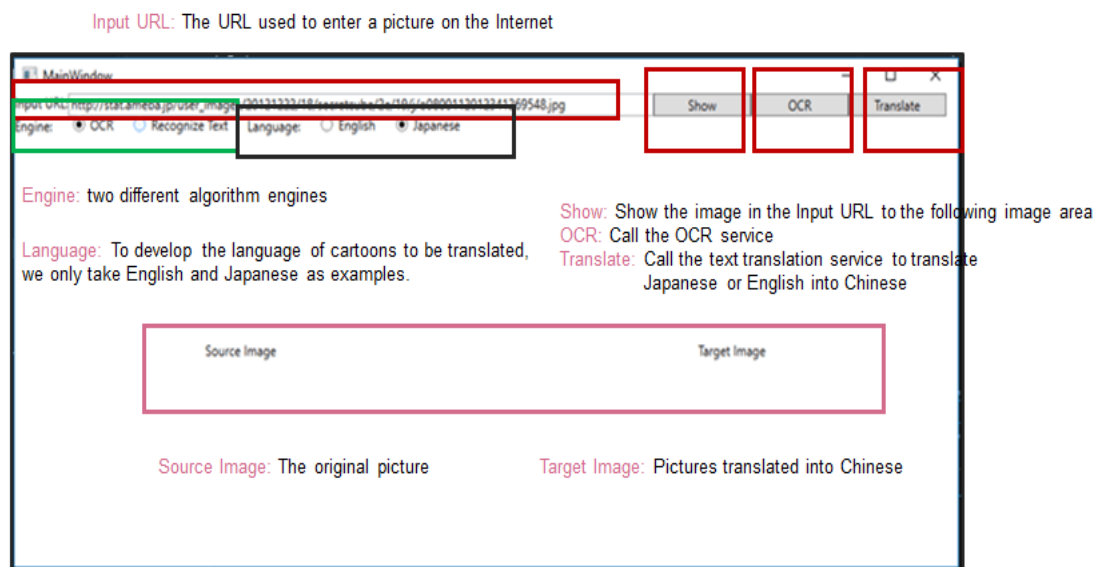


Building code and page design

1. Building code



2. page design



3. API cognitive service code

```
using System.Threading.Tasks;
using System.Web;

namespace CartoonTranslate
{
    class CognitiveServiceAgent
    {
        const string OcrEndPointV1 = "https://westcentralus.api.cognitive.microsoft.com/vision/v2.0/ocr?detectOrientation=true&language=";
        const string OcrEndPointV2 = "https://westcentralus.api.cognitive.microsoft.com/vision/v2.0/recognizeText?mode=Printed";
        const string VisionKey1 = "4c20ac56e1e7459a05e1497270022b";
        const string VisionKey2 = "97992f0987e4be6b5be132309b8e5";
        const string UriContentTemplate = "{0}*url*{1}*{0}*{1}";

        const string TranslateEndPoint = "https://api.cognitive.microsofttranslator.com/translate?api-version=3.0&from={0}&to={1}";
        const string TKey1 = "01023d634c409b1f02510b40026";
        const string TKey2 = "9f76381748549cb503dae4a0d80a80";
    }
}
```

Firstocr

firsttext

4. show button click code

```
private void btn_Show_Click(object sender, RoutedEventArgs e)
{
    if (!Uri.IsWellFormedUriString(this.tb_Url.Text, UriKind.Absolute))
    {
        // show warning message
        return;
    }

    // show image at imgSource
    BitmapImage bi = new BitmapImage();
    bi.BeginInit();
    bi.UriSource = new Uri(this.tb_Url.Text);
    bi.EndInit();
    this.imgSource.Source = bi;
    this.imgTarget.Source = bi;
}
```

Show button



5. OCR button click code

```
private async void btn_OCR_Click(object sender, RoutedEventArgs e)
{
    this.Engine = GetEngine();
    this.Language = GetLanguage();

    if (Engine == "OCR")
    {
        ocrResult = await CognitiveServiceAgent.DoOCR(imageUri);
        foreach (OcrResult.Region region in ocrResult.Regions)
        {
            foreach (OcrResult.Line line in region.Lines)
            {
                if (line.ConvertToText())
                {
                    Rectangle rect = new Rectangle
                    {
                        Margin = new Thickness(1),
                        Width = line.BB[2],
                        Height = line.BB[3],
                        Stroke = Brushes.Red,
                        //Fill = Brushes.White
                    };
                    this.canvas_1.Children.Add(rect);
                }
            }
        }
    }
}

public static async Task<OcrResult.RootObject> DoOCR(string imageUri, string language)
{
    try
    {
        using (HttpClient hc = new HttpClient())
        {
            ByteArrayContent content = CreateHeader(hc, imageUri);
            var uri = OcrEndPointV1 + language;
            HttpResponseMessage resp = await hc.PostAsync(uri, content);
            string result = string.Empty;
            if (resp.StatusCode == System.Net.HttpStatusCode.OK)
            {
                string json = await resp.Content.ReadAsStringAsync();
                Debug.WriteLine(json);
                OcrResult.RootObject ro = Newtonsoft.Json.JsonConvert.DeserializeObject<OcrResult.RootObject>(json);
                return ro;
            }
        }
    }
    catch (Exception ex)
    {
        Debug.WriteLine(ex.Message);
        return null;
    }
}
```

OCR button



6. Translate button click code

```
private async void btn_Translate_Click(object sender, RoutedEventArgs e)
{
    List<string> listTarget = await this.Translate();
    this.ShowTargetText(listTarget);
}

private async Task<List<string>> Translate()
{
    List<string> listSource = new List<string>();
    List<string> listTarget = new List<string>();
    if (this.Version == "OCR")
    {
        foreach (OcrResult.Region region in ocrResult.Regions)
        {
            foreach (OcrResult.Line line in region.Lines)
            {
                listSource.Add(line.TEXT);
                if (listSource.Count >= 25)
                {
                    List<string> listOutput = await
                    CognitiveServiceAgent.DoTranslate(listSource, Language, "zh-Hans");
                    listTarget.AddRange(listOutput);
                    listSource.Clear();
                }
            }
        }
    }
}
```

Translate button



Result analysis

결과 분석




Image1: 이미지에서 일본어를 다 식별할 수 있고 번역은 보통이다.

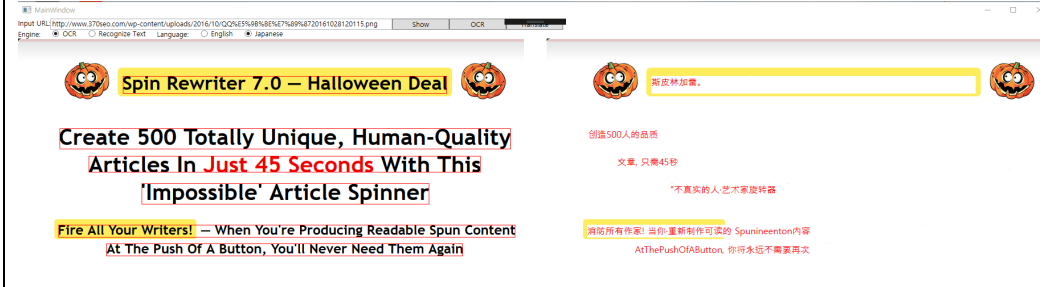


Image2: 이미지에서 영어를 다 식별할 수 있지만 번역 정확한 비율이 높지 않다.

이미지에서 영어와 일본어를 식별률이 높지만 번역은 정확도 좋지 않다.

각각의 식별된 텍스트를 각각 통역 엔진에 보내면 전후가 연결되지 않고 어구의 불순한 문제를 초래할 수 있다. 우리가 고려한 해결책은 먼저 사각형의 위치정보에 근거하여 이 텍스트를 동일한 말로 통역 엔진에 보내는 것이다. 이것은 클러스터링 문제이라서 아직 열심히 공부하고 있다.

이 프로젝트 업로드 Microsoft azure's Web service 하고 싶은데 그 방법을 안 찾아서 나중에 업로드 할 것이다.