Image Text to Speech Conversion

Problem Overview

The project is to make the image to speech. An image is processed and segmentedto identify the text in the image. Then the characters are combined to form words and save it as a text file. This text file is converted to speech.We use two tools for the completion of image to text to speech conversion. They are OCR (Optical Character Recognition) and TTS (Text to Speech) engines. Using OCR, we can optically recognize the characters in an image. TTS is used to convert the text file to speech. The audio output can be heard by using a python library Pygame for playing the audio at runtime.

Project Objective

Objective of our project is to reduce the complexity in reading and understanding the information's in text format. This is done by identifying the text content in the captured image and then the text is converted to speech.

Relevance of project

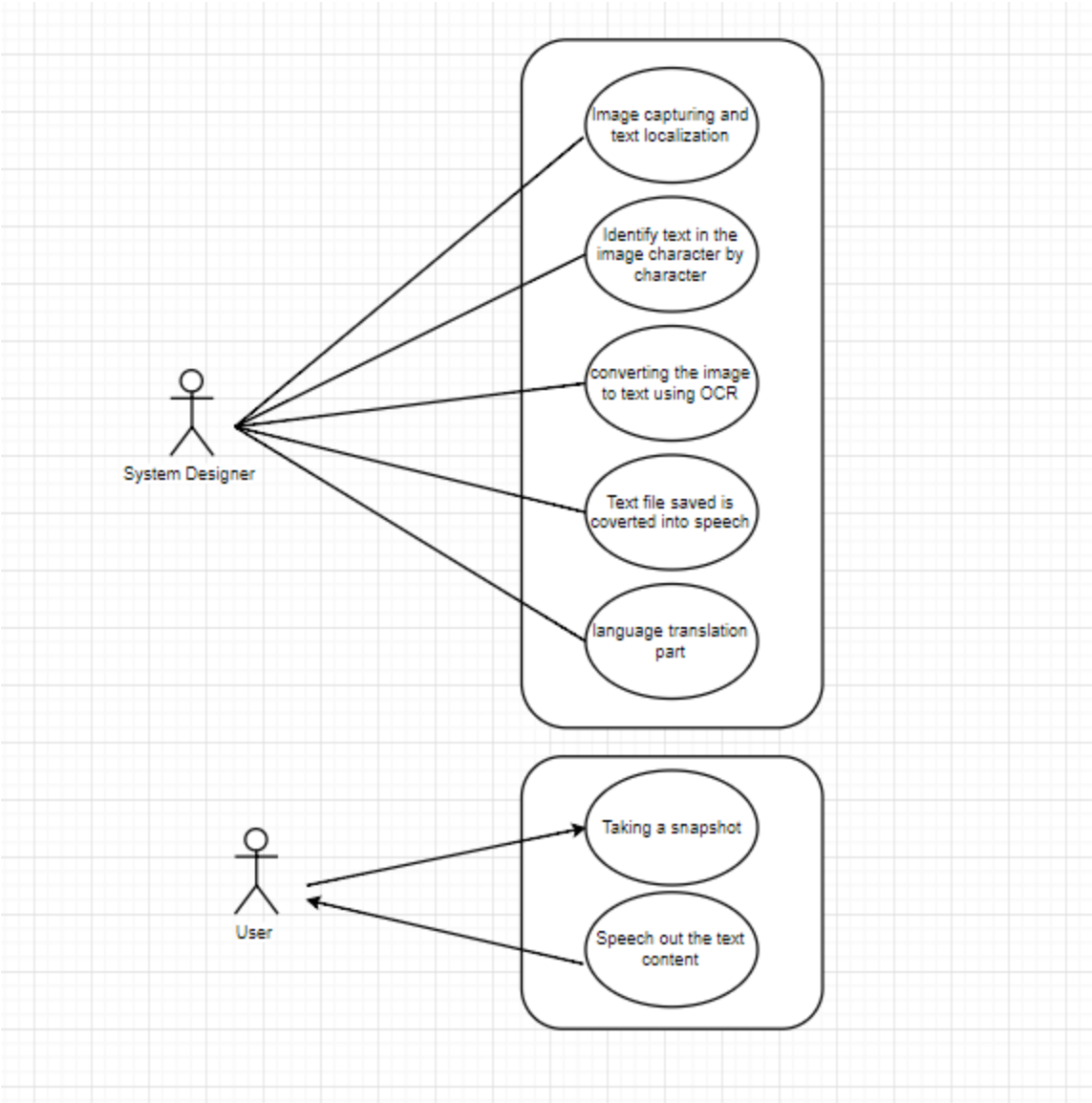
The image to text conversion has many levels of application nowadays. We can implement this project as solution for many applications. One of the important relevancy of this topic is in the development of devices like an automatic reader to blind people and people those who have visual impairments.

Methodology

The problem is to make a device which convert the text in to an image to speech. As a future advance also, we need to make the speech in any desired language.

First, we try with the R-PI 3 and the PY-CAM. We implemented the required libraries and with the help of pytesseract and GTTS (google text to speech) we make a system which takes the photo and identify the text parts. This text is saved into text file and then convert to a speech file. But the problem with this is the images with white background and text with black letters will be identified using pytesseract. So, we change the idea to implement it using OCR technique.

Use case diagram



In the section of Image to Text we use OCR concept and tried with various codes and different types of dataset available to train the model and make an output. But it's not working properly and didn't get much accuracy with it. In the part of Text to Speech conversion we use the help of GTTS (Google Text to Speech) API. Thus, we make a trial text file document and that text file get read by the computer at runtime with help of PYGAME library and MXPlayer.

In the Language Translation part, we made two kind of solutions, one by training a model to convert the English words to French using deep learning concept and next is done by interfacing the API of Google Translator.

Dataset Collectons

The data set for OCR implementation is collected from the Kaggle site. A collection of all alphabets both in capital letter and small letter with the numbers from 0-9 is taken as the dataset for character recognition. After using these dataset, we get two folders of images, one with 32768 images called trained data and another folder with 256 images called test data, after running the code we got two .CSV file with these two set of data.Another set of datasets used is the dataset with English words and corresponding French words, which is used to train the translator.

Classifiers and Models

Image to text:

- **Convolution 2d**: This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. If `use_bias` is True, a bias vector is created and added to the outputs. Finally, if `activation` is not `None`, it is applied to the outputs as well.
- **Max Pooling:**This is a form of non-linear down-sampling. There are several non-linear functions to implement **pooling**among which **max pooling** is the most common. It partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the **maximum**.
- **Padding:**If we don't use padding the information at the borders will be lost after each Conv layer, which will reduce the size of the volumes as well as the performance.
- **Activation Function** (Tanh, Relu, Sigmoid, Leaky Relu): We have used the combination of Leaky Relu and Tanh because our model was very highly biased.
- **Flatten:**We need to convert the output of the convolutional part of the CNN into a 1D feature vector, to be used by the ANN part of it. This operation is called flattening. It gets the output of the convolutional layers, flattens all its structure to create a single long feature vector to be used by the dense layer for the final classification.

Text to Speech

- **GTTS:** This is google-text-to-speech which is used to convert the text to speech.
- **PYgame:**Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.