OCR Web Application

Abstract:

Our OCR Web Application enales users to take a screenshot of something written on a piece of paper using web cam and convert it into machine-readable text. Probably the most well-known use case for OCR is converting printed paper documents into machine-readable text documents. Once a scanned paper document went through OCR processing, the text of the document can be edited with word processors like Microsoft Word or Google Docs. Before OCR technology was available, the only option to digitize printed paper documents was to manually re-type the text. Not only was it massively time consuming, it also came with inaccuracy and typing errors. It is a simple web application to convert English printed text into machine readable text.

OCR is often used as a “hidden” technology, powering many well-known systems and services in our daily life. Less known, but as important, use cases for OCR technology include data entry automation, indexing documents for search engines, automatic number plate recognition, as well as assisting blind and visually impaired persons.

OCR technology has proven immensely useful in digitizing historic newspapers and texts that have now been converted into fully searchable formats and had made accessing those earlier texts easier and faster.

OCR:

Literally, OCR stands for Optical Character Recognition. It is a widespread technology to recognize text inside images, such as scanned documents and photos. OCR technology is used to convert virtually any kind of images containing written text (typed, handwritten or printed) into machine-readable text data.

OCR Technology became popular in the early 1990s while attempting to digitise historic newspapers. Since then the technology has underwent several improvements. Nowadays solutions deliver near to perfect OCR accuracy. Advanced methods like Zonal OCR are used to automate complex document based workflows.

OCR Web Application

Overview of project:

Step #1 - MediaDevices.getUserMedia()

MediaDevices.getUserMedia is a browser API that allows web apps to access user's camera and microphone.

Step #2 - glfx.js, JCrop

glfx.js was used for image effects (sharpening, contrast, etc.). Cropping functionality (with touch support) is provided by jQuery plugin Jcrop.

Step #3 - Tesseract.js

Tesseract.js was used for OCR (Optical Character Recognition). It is a javascript version of the Tesseract Open Source OCR Engine.

Team Members:

1. Aman Srivastava : 18100004(CSE)
2. Avinash Sharma: 18100012(CSE)
3. Daksh Singh: 18100015(CSE)

References:

- Tesseract.Js: https://tesseract.projectnaptha.com/
- MDN: https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia
- Capturing Video in HTML5 - https://www.html5rocks.com/en/tutorials/getusermedia/intro/
- glfx.js - http://evanw.github.io/glfx.js
- evroneCrop - https://github.com/evrone/evroneCrop