Objective: This project pieces the torn photo parts or torn document parts together, and generates a result photo or document. The torn photo repair also fills the missing pixel along the edges, detects image parts from multiple original photo, and generates an album for the multipage case. The torn document repair can be used as an OCR for torn documents, and generate a text file containing all the torn parts scanned.

Users:

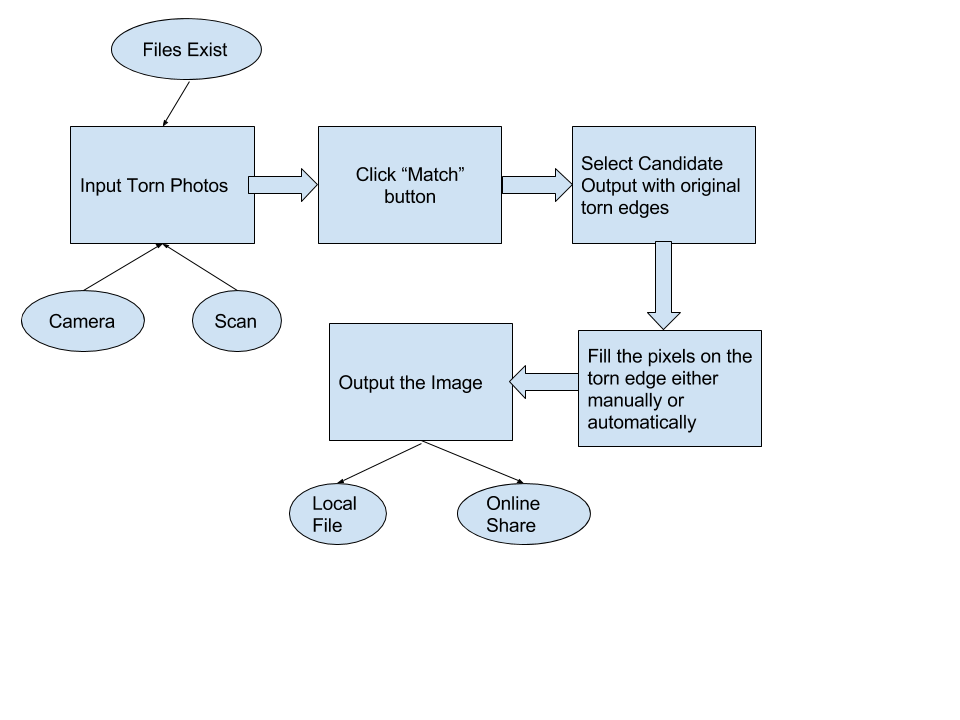
1. Normal people use this software for torn photo repair, like piecing together torn old family photo
2. Normal people use this software for torn document repair, like restoring a torn book, scanning text from shredded document.

Use cases:

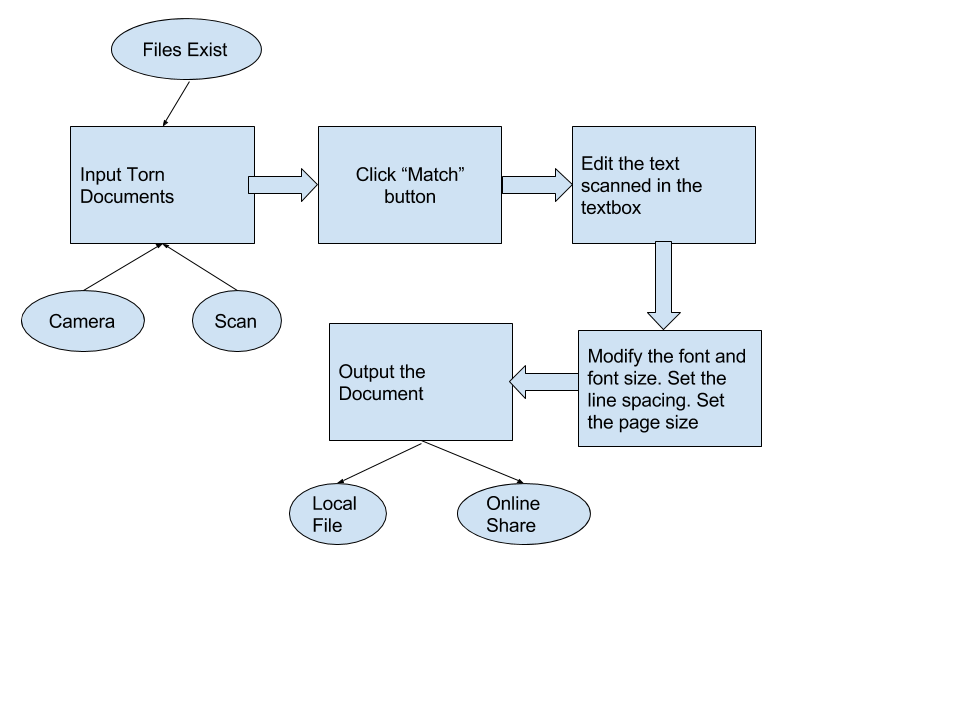
1. Torn photo repair: Users need to import their torn photo parts into the software. The image can from scanner, camera or image files in most of the popular file types (JPG, PNG, BMP...). The image must be in white or black background, and the color of the photo parts must have a significant contract to the background. In addition, the torn edge or the color near the torn edge must not be very uniform ( rectangle parts with only one color will not be accepted ). The size of the torn parts must be at least 100p\*100p with at least 72dpi, and the torn parts must not be scaled. Once user uploaded all the parts and click "Match" button, this use case will be triggered.
2. Torn document repair: Users need to import their torn document parts into the software. This document must be in image file format, not in a document file format. The torn parts can be file in computer, or can come from scanners. Input from camera is generally acceptable if the character is eligible. The torn parts must be at least 100p\*100p with at least 72dpi. The font size must be greater than 12. In order to use the OCR correctly, no image is allowed in document repair. Once user uploaded all the parts and click "Match" button, the torn document parts will be scanned, and place the result text in a textbox. The user can modify the text in the textbox and click "Export" to get the output text file.

Workflow:

1. Torn photo repair:



2. Torn Document Repair



Functional Components with Functional Requirements and Non-functional Requirements:

Core:

An algorithm that can use the contour map, the color near the contour or the feature of characters in an alphabet on the edge to determine the best solution to match the pieces together.

Functional Requirements:

* Correctly match the pieces together
* Detect multipage documents/photos
* Provide some alternative solutions for user to choose the best one
* Automatically fill the pixels on torn edges

Non-functional Requirements:

* Efficency
* Provide progress to user
* Show the matching edges to user
* Fill the pixel in a nice way and recover more of the original photo

General:

UI: Windows Form Application or Universal App depend on the platform

Functional Requirements:

* Can trigger all the required actions in the program
* Provide tools to import the sources and export the result
* Can be used to configure some setting of the program

Non-functional Requirements:

* User friendly with minimal tutorials or guides
* Nice looking
* Nice user experience in different platforms
* Multitouch compatible

Input Manager: Provide a tool to manage and verify inputs

Functional Requirements

* Can take all three sources of input
* Read most of the popular file types
* Interact with camera and scanner API
* Validate if the inputs fit the requirements

Non-functional Requirements:

* User friendly messages, suggest user how to provide valid input
* Different warnings for different kind of error input

Image Processing Library: Emgu CV. Use this to get the contour map of the input fragments and the color of the pixels nearby the contour. This is also used for image transformations and filling the pixel on the torn edges. This library also supports OCR so that text can be automatically generated

Functional Requirements:

* Provide all the APIs for image processing, like image transformation and modifcation
* Detect and extract the contour maps and the color near the contours
* Read characters from images of document

Non-functional Requirements:

* Efficency
* Nice Documentation on API
* Nice code style

Online Sharing: APIs provided by online service providers

Input: APIs for camera on phones for camera input. Codes to call scanner program for that computer and fetch the result from scanner.

Functional Requirements:

* Do all the online sharing to a particular service correctly

Non-functional Requirements:

* Nice looking
* Nice Documentation on API

Torn Photo Repair Specific:

Image Viewer: used in viewing the candidate results and choosing the most accurate one by user

Functional Requirements:

* View all the candidate photo results
* Choose the best one as user specified

Non-functional Requirements:

* Have nice positioning of the buttons

Image Editor: used in manual fixing of the result image.

Functional Requirements:

* Transformations: scale, rotate, shear etc.
* Modifications: change color of pixels, interpolation.

Non-functional Requirements:

* Nice positioning on buttons
* Better modification experience

Album generator: If the torn parts is from multiple photos, this will be used to generate a file that contains all the complete photos

Functional Requirements:

* Generate a file in popular file format (like PDF) that contains all of the complete photos

Non-functional Requirements:

* Efficency
* Have nice transition between pages
* Provide multiple album frames and nice album cover

Image Output Manager: Manage the image output and destination

Functional Requirements:

* Output file into different resolutions
* Output file into different file types
* Provides choices for single output or album output

Non-functional Requirements:

* User Friendly
* Provides error information if something wrong happen

Torn Document Repair Specific:

Text Editor: used in verifying the text result and modifying the format of the text.

Functional Requirements:

* Modify the text content
* Modify the format of the text, like font and font size
* Modify the paragraph format, like line spacing
* Set the page size and margins

Non-functional Requirements:

* Better user experience, like editing text in their familiar word processing software
* Support some less frequently used fonts

Text Output Manager: manage the output options for text output

Functional Requirements:

* Can output text into different file types
* Provide all the information, including text and format information, to outside libraries for specific text formats

Non-Functional Requirements:

* User friendly
* Provide different error messages for different situations

Marking Languages generator: Something like Freemarker. Used in generating different output file format like HTML or XML.

Functional Requirements:

* Generate the required file format with all of the contents and text formats.
* Translate the text formats specified in the program to text formats supported by different file types
* Supports major marking languages

Non-functional Requirements:

* Efficency
* Easy to code
* The document generated is well structured

Microsoft Office API: used for generating MS Word files.

Functional Requirements:

* Generate the MS Word file with all of the contents in the output text

Data Flow Graph:

