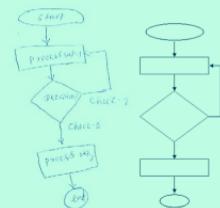


Digitalization of Offline Handdrawn Flow Diagrams



Abhay Kumar Singh | UG201310003 | CSE
Deepshi Garg | UG201313008 | SS-CSE

Mentor : Dr. Gaurav Harit

Project Overview

The software takes in an image(jpg) as input which contains flowcharts and various other polygons with text and outputs a rendered digitalized image with perfect shapes and place the recognised the handwritten text into it.

Importance of Work

For scientific documents, which nowadays usually created in digital form, it is often necessary to include diagrams or models.

The faster and more convenient method to create those diagrams however, is still drawing them by hand.

Scanned or photographed, those drawings are rarely in such neat quality that they smoothly integrate into the digital document.

For this project, a program is created to recognise the individual shapes in a diagram and redraw it with a drawing function.

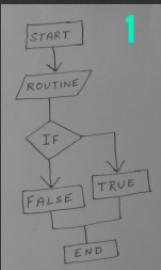
Technologies Used

- **Language Used :** Python for feature implementation, CSS Bootstrap for building User Interface
- **Web Application :** Python Flask (micro web framework written in Python based on the Werkzeug toolkit and Jinja2 template engine)
- **Environment Used :** OpenCV Python for image processing tools and scikit learn tools for neural network

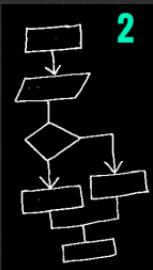
Salient Features

- Noise in image is removed using Adaptive threshold
- Images drawn on ruled paper can also be detected.
- Incomplete polygon and polygon with crossing vertices can also be detected.
- Able to differentiate between rectangle, rhombus & parallelogram
- Able to detect line even if it's broken
- Able to detect arrow heads
- New robust algorithm for shape detection using cycle detection algorithm in graphs
- Adjustment of connecting lines according to the rendered shapes
- Hand written text recognition with an accuracy of 97%

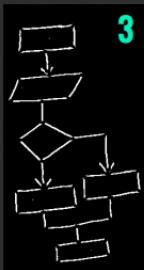
Modules



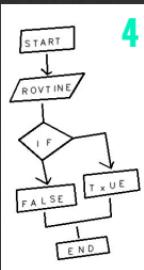
1



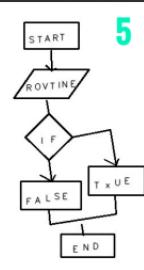
2



3



4



5

Description

- 1 Input Image, it contains hand drawn flowchart with text, arrows and conditional branches. Input image can be drawn on ruled paper with shadows and noise. It can contain polygons with incomplete edges or crossing edges.
- 2 High quality binarization using Adaptive thresholding. Adaptive thresholding leads to breaks in some long edges, therefore morphological closing is done (first dilating and then eroding the image). Later, extraction of text from the image using connected component analysis and considering components with small bounding box as text. Finally those connected components are cropped out from the main image and given to the text detection algorithm for proper output.
- 3 Sharp corners are detected using Harris Corner Detection algorithm. Then those corners are blacked out from the image, so that the image is left with only straight lines. All Lines are detected and equations are obtained.
Lines with almost same slope and close to each other are joined because these lines are basically the lines which are disconnected because of corner detector. If three lines are concurrent, then those are joined first, and using this approach we are able to restore the arrows and make a conditional branch using rhombus.
- 4 Detected character are placed on their respective position on a white image and that image is used to place the intersected lines. Intersection of all line pairs are computed and converted into closed shape. Using the line equations and coordinates of end points, distance between intersection point and end point is obtained, and if that value is less than a permissible distance, those two lines are joined together. After applying these operation on all the line pairs, we are left with closed polygons and connecting lines.
- 5 Considering each line segment as a node in graph and adjacent line segments as adjacent nodes, a graph is created. Any cycles in this graph is considered as a closed polygon. Therefore, a robust recursive cycle detection algorithm is called to detect all the possible cycles in the graph and remaining lines that do not make a graph is considered as arrows. Finally, perfect shapes are rendered on the image and line segments are adjusted according to the final rendered figure.