%++++++++++++++++++++++++++++++++++++++++

% Don't modify this section unless you know what you're doing!

\documentclass[english , letterpaper,12pt]{article}

\usepackage{tabularx} % extra features for tabular environment

\usepackage{amsmath} % improve math presentation

\usepackage{graphicx} % takes care of graphic including machinery

\usepackage[margin=1in,letterpaper]{geometry} % decreases margins

\usepackage{cite} % takes care of citations

\usepackage[final]{hyperref} % adds hyper links inside the generated pdf file

\usepackage{setspace}

\usepackage{float}

\usepackage[linesnumbered,ruled,vlined]{algorithm2e}

\usepackage{babel,blindtext}

\doublespacing

\hypersetup{

colorlinks=true, % false: boxed links; true: colored links

linkcolor=blue, % color of internal links

citecolor=blue, % color of links to bibliography

filecolor=magenta, % color of file links

urlcolor=blue

}

%++++++++++++++++++++++++++++++++++++++++

\begin{document}

\title{Sudoku Puzzle Detection}

\author{Priyanka Dwivedi}

\date{\today}

\maketitle

\begin{abstract}

In this experiment we studied image processing. There were some test images given which had sudoku on the newspaper. The given task was to detect the sudoku and using OCR engine detect the numbers on the sudoku. The images may have different rotation of sudoku.The report will give how I did sudoku detection with number detection on it.

\end{abstract}

\section{Introduction}

The task to detect sudoku from newspaper was done using thresholding and morphological operation. Detect sudoku orientation using hough transform. Now we have sudoku detected we need to find numbers on the sudoku which is done using OCR engine.

\section{Theory}

Before proceeding with detection. I first analyze the image and think of various methods that can be done to detect sudoku. After thorough analysis, I did image segmentation by thresholding and using morphology operation like opening,closing,dilation,erosion. Assumption made during this process is that the sudoku is the largest blob in the image. The image obtained at this stage was binary image with sudoku as white blob. Using hough transform to get the sudoku orientation. Hough detect lines and also gives the angle. This angle was used to rotate the sudoku. After the sudoku detection was detected successfully, the sudoku was crop so as it becomes easy as we now have only sudoku. The cropped image is processed again to extract only the numbers which are passed to the OCR engine to detect the numbers on it.

\section{Procedures}

Given is a schematic of the experimental setup used in the assignment.There are many images, however I am using single image which will cover rotation and other background details.The below image will be used in the entire step by step description.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{original.png}

\caption{

\label{fig:original}

Original Image

}

\end{figure}

\begin{enumerate}

\item Convert into double and gray scale image.

Convert into double so that it becomes easy for thresholding as we have to deal values between 0 to 1 and gray scale for thresholding and binarise image.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{double\_gray.png}

\caption{

\label{fig:double}

Double and GrayScale Image

}

\end{figure}

\item Compare the index points of the outside co-ordinates and sudoku image to threshold image. I varied various numbers and 0.2 worked for the test images given.This image may not be showing great difference but for images which had black cardboard the thresholding made the black background white.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{theshold.png}

\caption{

\label{fig:threshold}

Threshold Image

}

\end{figure}

\item Binarize Image using threshold value. Here you can use graythresh which is inbuilt function of Matlab. I preferred experimenting with various values and 0.5 value worked for all test images.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{binary.png}

\caption{

\label{fig:binary}

Binarize Image

}

\end{figure}

\item We now have binary image and since we want detect sudoku using blob method I inverted image.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{invert.png}

\caption{

\label{fig:inverse}

Inverse of Binary Image in Fig.~\ref{fig:binary}.

}

\end{figure}

\item Fill image with holes so as to get blobs in the image.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{blobs.png}

\caption{

\label{fig:blob}

Blob using hole in Binary Image in Fig.~\ref{fig:inverse}.

}

\end{figure}

\item From Fig.~\ref{fig:blob} we can see there is more than one blob and many text in the image. To remove all this I used bwareafilt.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{segmented\_sudoku.png}

\caption{

\label{fig:segment}

Blob using hole in Binary Image in Fig.~\ref{fig:blob}.

}

\end{figure}

\item I had all images in which sudoku was only blob left after the above step.

At this stage it is possible to have more than one blob. In that case, we can assume that sudoku is biggest blob or check blob have 81 blocks or not. Some kind of degree of confidence to get sudoku detected. In my case i didn't do anything as all sudoku were remaining after the filling. Since the sudoku is rotated we need to rotate the sudoku to correct angle. To do that, I used Hough Transform. Before doing hough, I detected the edges using canny edge detection technique.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{edge\_detection.png}

\caption{

\label{fig:edge}

Canny Edge detection for Fig.~\ref{fig:segment}

}

\end{figure}

\item The above edge detected image was passed to hough transform which gives the angle of rotation. The theta of peaks value in hough was used to rotate the image.

This rotation of image along with sudoku detection is shown in image in magenta color.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{hough\_sudoku.png}

\caption{

\label{fig:hough}

Rotated Sudoku with sudoku detection

}

\end{figure}

\item Since now we have detected sudoku, I crop the detected region to work on number detection.Cropping will restrict our search to only numbers and 81 boxes.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{cropped\_sudoku.png}

\caption{

\label{fig:crop}

Cropped Sudoku after Detection

}

\end{figure}

\item We now have cropped sudoku, however passing this to the ocr as it is not going to work. It will for example detect 1 for the edges of the sudoku box. To get rid of such unwanted detection I have done some preprocessing before passing the image to ocr. Let's take look at in detail.\\

\textbf{Number Detection in sudoku}

\begin{enumerate}

\item We know that sudoku have different sizes in newspaper so we need to resize sudoku so that all of them are of same size. We first binarize the image since we need to perform morphological operation to get rid of noise and connected components to remove the grid.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{binarize.png}

\caption{

\label{fig:binarize}

Binarized Image

}

\end{figure}

\item We inverse the binarized Image.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{inv.png}

\caption{

\label{fig:inv}

Inverse of Binary Image

}

\end{figure}

\item Using bwconnected Components we find the biggest line and get rid of the line in between the sudoku.At this stage we only have numbers in the image, Since for my processing there were some dots in between I had to some morphological operation to get rid of those small objects.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{numbers.png}

\caption{

\label{fig:numbers}

Extract only numbers from the Image

}

\end{figure}

\item Now we have degree of confidence that we have only numbers in the image and can be used to detect the numbers using ocr engine. However, we are not sure about the orientation of the image. This stage, I have proper oriented image, but not all images have correction orientation. I just passed this image to ocr engine, it couldn't detect many numbers so as degree of confidence, I divided my image into 81 square before passing to ocr engine.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{81.png}

\caption{

\label{fig:81}

81 squares of the sudoku

}

\end{figure}

\item From the Fig.~\ref{fig:81} , we now know that each square can either be blank or 1 digit number between 1-9 by rules of sudoku.Passing each 81 squares in sudoku,

I concatenate each digit in a string with comma separated to get output as follows.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{output.png}

\caption{

\label{fig:output}

Final output of Sudoku

}

\end{figure}

\end{enumerate}

\end{enumerate}

\section{Analysis of Different approaches}

The Analysis of various approaches tried during the project. There were many different techniques I tried finally after reaching the approach where it was working for all Image.

\begin{enumerate}

\item \textbf{Sudoku Detection : } I tried using different technique for detecting sudoku. Initially I went with appproach of detecting only horizontal lines and vertical lines and combining the image to get the sudoku and got other parts of image. The problem with this approach was that it could not detect lines if sudoku was rotated.

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{img.png}

\caption{

\label{fig:img}

Original Image

}

\end{figure}

\begin{figure}[H]

\minipage{0.48\textwidth}

\includegraphics[width=\linewidth]{vert\_hor.png}

\caption{Vertical and horizontal line}\label{fig:ver\_hor}

\endminipage\hfill

\minipage{0.50\textwidth}%

\includegraphics[width=\linewidth]{combined.png}

\caption{Combined image of Vertical and Horizontal Lines}\label{fig:combined}

\endminipage

\end{figure}

\textbf{How did I solve :} I used technqiue we did in dice detection by filling holes and removing small text and blob by bwareafilt.

\item \textbf{Hough Transform :} I faced issues with hough transform, but while using it I realized I didn't segment my image properly or i could have done some degree of confidence as to find sudoku blob with 81 square box. \\

\textbf{How did I solve : } I did a lot of preprocessing of image before doing Hough Transform. I even used imdilate because edge detection had some gaps before passing it to hough Transform.

\item \textbf{Rotation of Sudoku} After successful sudoku detection , I resized so that all sudoku are of same size. The resultant image just had the sudoku, directly passing it to OCR engine will give some gibberish output. At this stage, though I had cropped sudoku but all didn't have upright rotation, I had to find a way for detecting the orientation of the sudoku. \\

\textbf{How did I solve : } I tried various technique to find proper orientation. I tried using template matching discussed in class using normal cross co-relation. But the filter created by subtraction of each pixel with mean of template didn't work out for me. I tried using feature matching like creating eight of the sudoku and detecting the 8 and its orientation. However, it didn't work as it was suppose to it used to detect something different in the image. Even the template matching didn't work so I broke the entire image into 81 boxes and found the numbers

\begin{figure}[H]

\centering \includegraphics[width=\textwidth]{temp.png}

\caption{

\label{fig:temp}

Template matching using normal cross co-relation

}

\end{figure}

\item \textbf{Number Detection :} Though the idea of breaking image into 81 boxes and detecting number was useful still the orientation was issue.

\textbf{How did I solve : } orientation for two images was 90 degree and for such image the ocr gave output of more than one digit. Using this I could detect that the image is rotated and them rotate image by 90 and repeat the same of dividing into 81 squares and this technique worked for me only because for rotated image, ocr gave me ouput of more than one digit.

\end{enumerate}

\section{Results of est Images}

\section{Conclusions}

Here you briefly summarize your findings.

%++++++++++++++++++++++++++++++++++++++++

% References section will be created automatically

% with inclusion of "thebibliography" environment

% as it shown below. See text starting with line

% \begin{thebibliography}{99}

% Note: with this approach it is YOUR responsibility to put them in order

% of appearance.

\begin{thebibliography}{99}

\bibitem{melissinos}

A.~C. Melissinos and J. Napolitano, \textit{Experiments in Modern Physics},

(Academic Press, New York, 2003).

\bibitem{Cyr}

N.\ Cyr, M.\ T$\hat{e}$tu, and M.\ Breton,

% "All-optical microwave frequency standard: a proposal,"

IEEE Trans.\ Instrum.\ Meas.\ \textbf{42}, 640 (1993).

\bibitem{Wiki} \emph{Expected value}, available at

\texttt{http://en.wikipedia.org/wiki/Expected\\_value}.

\end{thebibliography}

\end{document}