

Advanced Computer Vision

Second Project Report - Sudoku Solver

Suhas Cholleti(sc3614@rit.edu)

Venkata Karteek Paladugu(vp3982@rit.edu)

Introduction :

In this project our goal is to solve a sudoku puzzle in a live video and then show the solved sudoku puzzle in the video itself using computer vision. Sudoku puzzle is a 9 X 9 grid containing 3 x 3 subgrids of 1 to 9 digits. In order to have a valid sudoku no digit can be used twice in any row, column or 3 x 3 subgrid. All the preprocessing steps starting from extracting the frame from the video to extracting the sudoku puzzle was done using OpenCV and solving the puzzle was done through backtracking algorithm and finally showing the solution in the frame was also done using OpenCV.

Motivation:

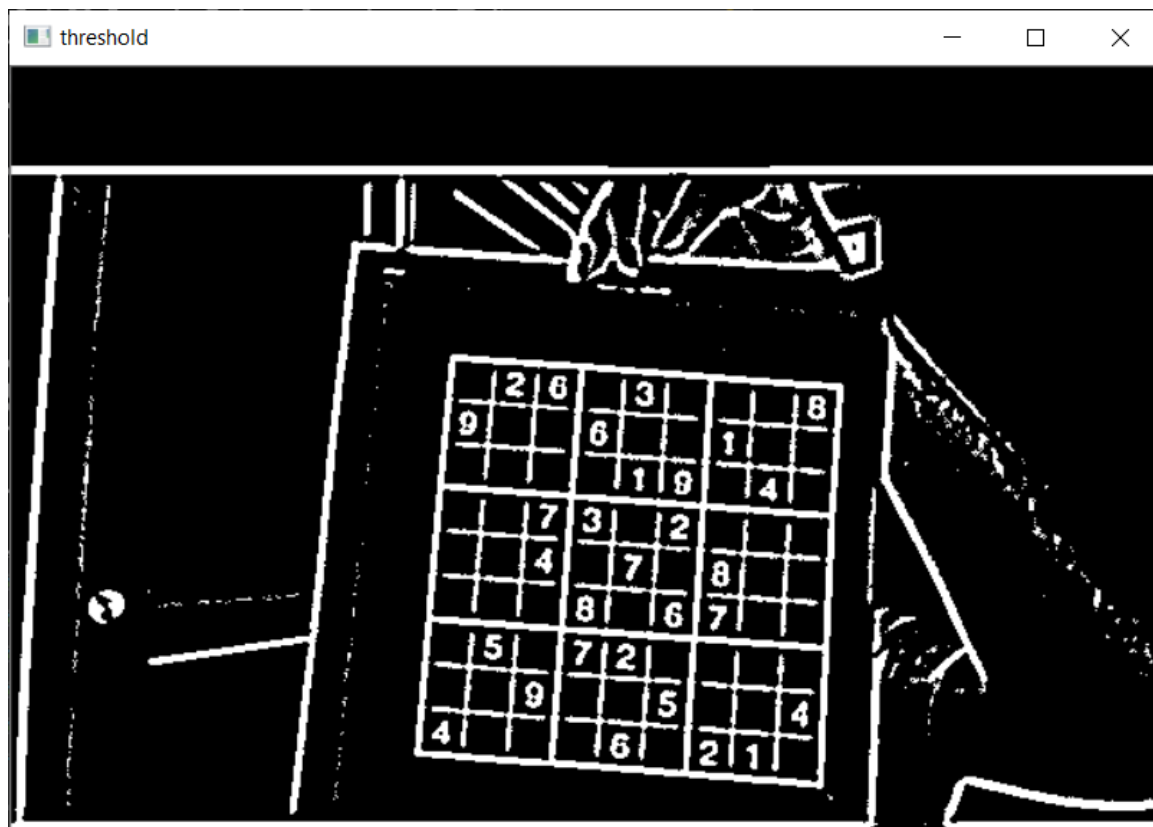
The motivation behind this project is it looked very interesting and we grew up solving sudoku puzzles and it amazed us when a machine is able to solve the puzzle in live feed and show it to the user rather than just printing it on the console. Also we wanted to apply the concepts learnt in class mainly stereo matching and camera calibration for this project.

Implementation and Results:

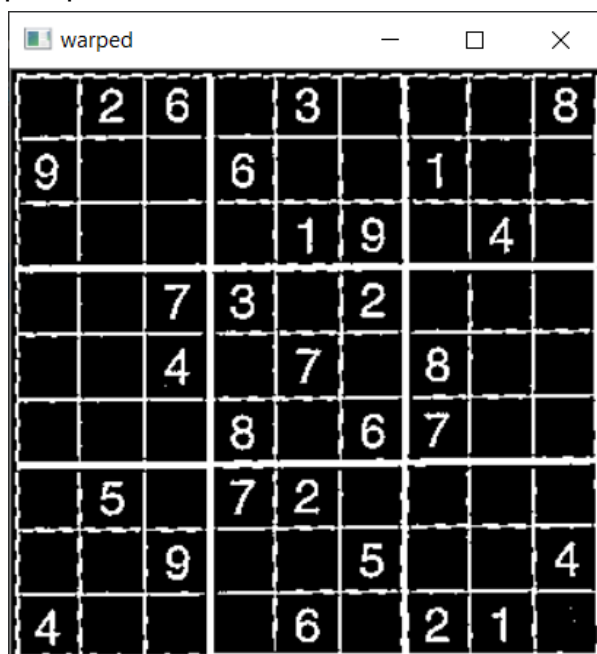
We initially tried to use stereo imaging techniques taught in class, as a sudoku is a grid like pattern. But these techniques were not able to capture the corners cleanly as the puzzle isn't a checkerboard pattern that is expected for the camera calibration. So instead we used contours. We find the largest contour and warp the image. We then use simple math to divide the image into 9 x 9 smaller images and predict the numbers present in each image. We solve the sudoku using the predictions and then add text onto the frame to display the solved sudoku on a live feed.

We used cv2, keras, and numpy modules for the implementation of this project. We read in frames using the videoCapture from cv2. For each frame we apply a gaussian

blur and perform an adaptive thresholding. The resultant frame is given below.



From the above frame we find the largest contour, as the largest contour the A4 sheet will always be the puzzle. We get the 4 corners from the contour and perform a perspective transform on it. Below is the resulting frame.

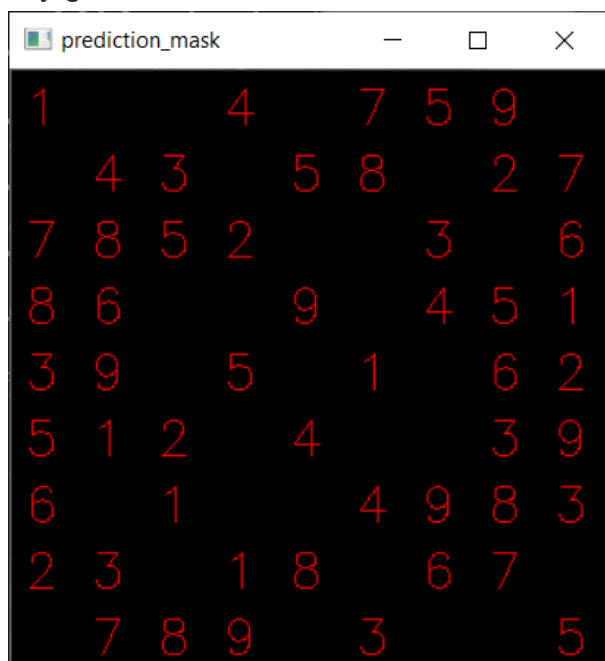


We divide this frame into 9 equal parts horizontally and vertically. This would give us 81 images with each image being the image of a cell in the grid. Below are a few examples. The first is a empty cell followed by cells with a 2 and a 6.

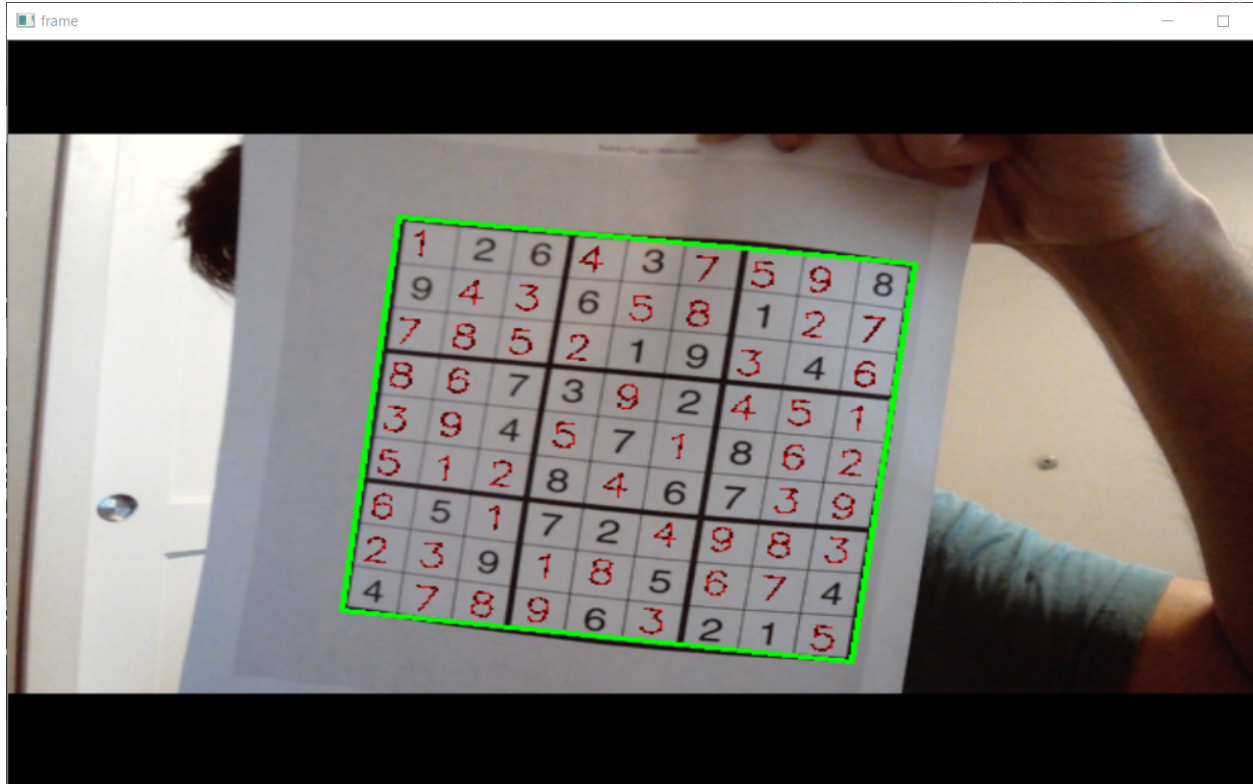


We send each of these images to a CNN model. The CNN model was trained using keras with printed characters from the different ubuntu fonts. Using the predictions made from the CNN model, we make a 9 x 9 matrix and then solve the sudoku.

We then make a prediction mask from the solved sudoku matrix. The prediction mask is of the same size of the warped image, and only the numbers of the solved cells without any grid.



This prediction mask is then warped into the plane of the original frame and then it is added to the frame. Below is the resulting frame from the output



Once we find the solution to the sudoku, we set a flag to 1. We solve the puzzle only when the flag is 0 and set it to one once we solve it. Once contour leaves the screen, we change the flag back to 0 as that could be that the puzzle has changed. This allows us to skip solving for sudoku once we solve for the first time and helps with the performance.

If any of the predictions from the CNN model for the numbers are wrong, it is almost certain that the sudoku solver cannot solve the given matrix. If such a situation arises, the code skips that frame and starts the whole process with the next frame.

Conclusion:

We have learnt a lot of things and enjoyed doing this project. Although there are issues with the model like cases where when the lighting is low or if any of the number is not scanned properly then the sudoku won't be solved and it again scans the frame and tries to solve the same problem again. Apart from it we learned and understood thresholding, finding contours, warping and transforming images, extracting digits from the puzzle using a neural network and then solving the puzzle using backtracking.