**Abstract -** Computer Vision is one of the most real-world applicable technologies used today, because of its many modules, versatility, and new developments in Deep Learning and Convolutional Neural Networks which have aided us to extract various features from images. Implementing this Artificial Intelligence paragon to our daily lives will make our learning and interpretation easier. Sudoku was invented in Japan in 1979 but was unknown to the western world until 2004. Sudoku is played by more than a million people every day. This paper explains the technology used to solve Sudoku Puzzle using OpenCV and Deep Learning. The input can be given in the form of an Image or using the live camera. The main motivation for this paper is that it will help beginners and advanced players to solve sudoku and check their solutions.

1. **Introduction**

Puzzles have always fascinated humans because of the thrill and challenge involved in solving them. Sudoku can be accessed by people from the web, daily newspapers, magazines. The building blocks of main architecture are:

* OpenCV: Real-time Computer Vision Library with a variety of tools and functions. The few used in this model are Contour Detection, Resize and changing images from 3 Dimensional RGB to 1 Dimensional.
* Neural Network: A Convolutional Neural Network which works by convoluting a kernel over image pixels is used to train on images.
* Back tracing Algorithm:

1. **Dataset**

The Dataset used to train neural networks is the MNIST dataset which is images of numbers 0-9 in different patterns and their corresponding labels, the dataset was further divided 60000 training samples and 10000 testing samples. Each image is a one-dimensional 28\*28 image.

A comprehensive detail about the trained neural network and OpenCV operations used to detect images is given in Section 3. The results are presented in Section 4.

1. **Related Work**
2. **Method**
3. **Data**

The Sudoku puzzle to be solved is taken as input in the form of an image. The image can be of any dimensions. Below is the image of example used to explain every process in the paper.

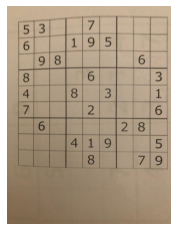


Figure -Example used

1. **OpenCV**

OpenCV is a python library for real time computer Vision. We have used OpenCV to detect Sudoku boundaries and later resize it by using following operations:

1. First the image is converted to grayscale because contour detection will become easier.
2. Next a gaussian blur is used on image to smooth it. To remove any noise and irrelevant detail to identify contours from it.
3. Thresholding is done to separate foreground from background. The adaptive threshold in OpenCV allows to calculate threshold values for smaller regions. If the value to pixel is above threshold it is set to 255 and when it is below threshold it is set to 0.
4. A bitwise not operation of the image allows us interchange the black and white of foreground and background.
5. The next operation is to use dilate from cv2, so that contours become more wider for detection.
6. The next task is to identify the contours of the image:
7. The find contours function returns all of the contours it finds in the image.
8. Next, we sort the contours by the descending area size enclosed
9. The contours that enclose the maximum area will be the corners of Sudoku puzzle.
10. After identifying the corners of puzzle, we formulate the pixel position of sudoku on the original image.
11. Then, we extract the sudoku puzzle from the original image by cropping and resizing it to 450\*450 image.
12. The next step is to crop blocks of numbers from sudoku using cropping and resizing them to 28\*28. Because our model was trained using 28\*28 images.
13. **Convolutional Neural Network**

A convolutional neural network is a combination of multi-layer perceptron that help extract features from an image. These features are calculated using convolution of image with kernel.

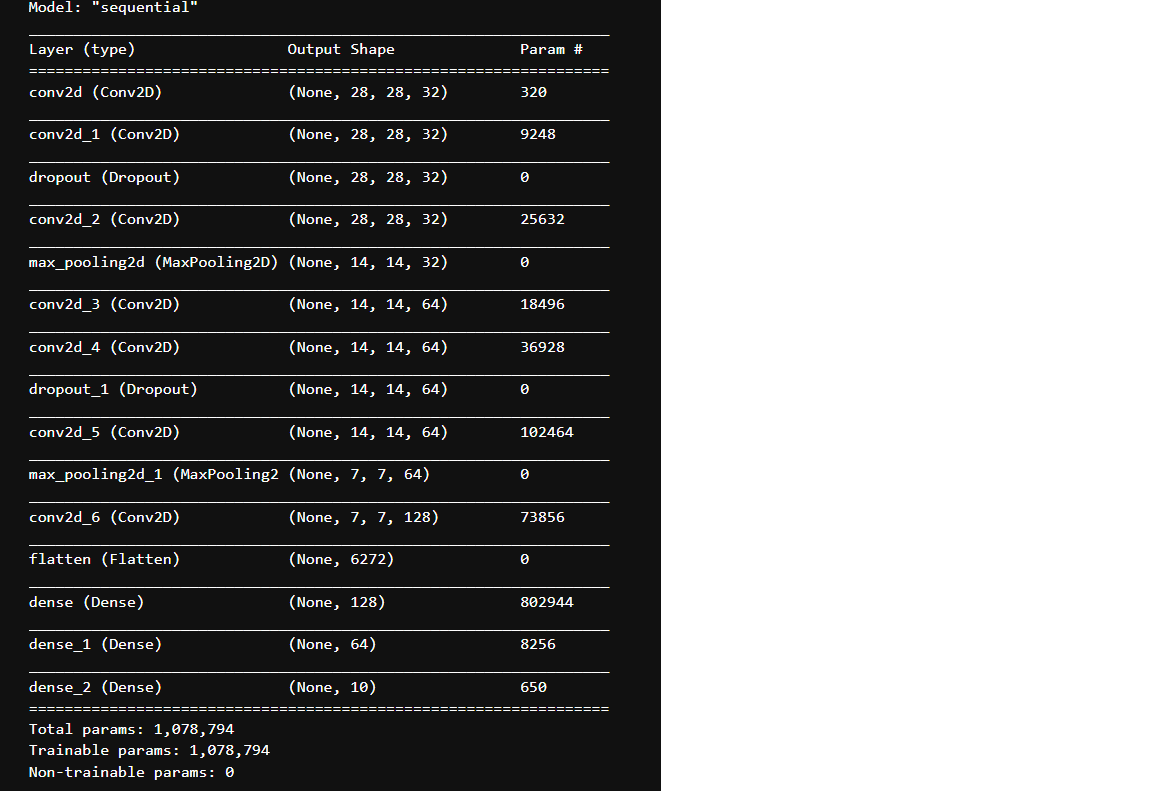


Figure - Model Details

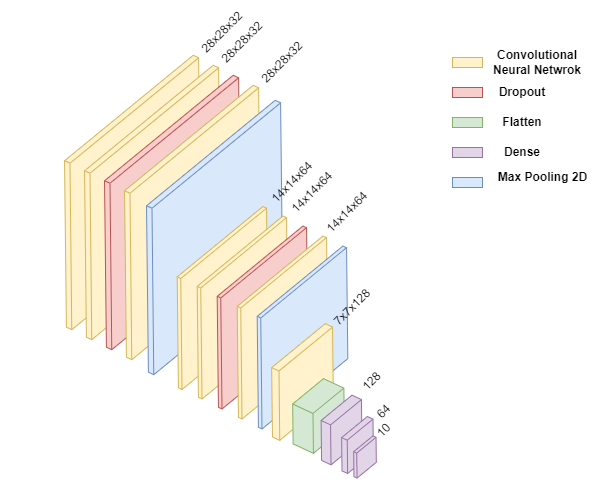


Figure - Sequential Model Layout

The constructed neural has 15 layers comprising of 7 Convolutional Layers,3 Dense Layer, 2 Dropout layers,2 Max Pooling Layers and 1 Flatten Layer. The model was trained over 10 epochs, with a validation split of 20%. The accuracy and validation accuracy achieved was 99.17% and 99% respectively. After evaluation of model on test dataset the accuracy was 99.22%. Therefore, the model was not overfitted on dataset.

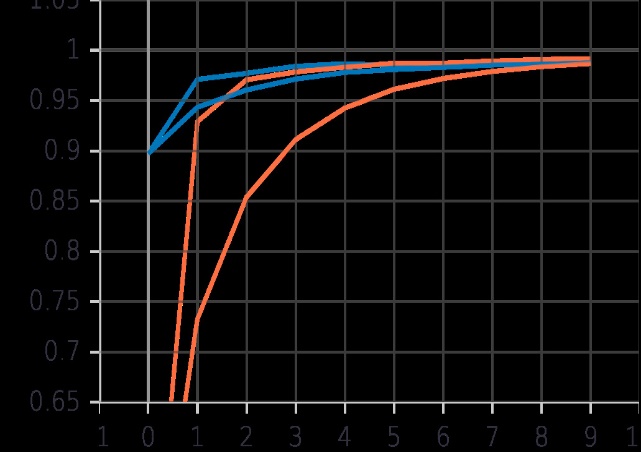


Figure - Model ACCURACY OVER EPOCHS

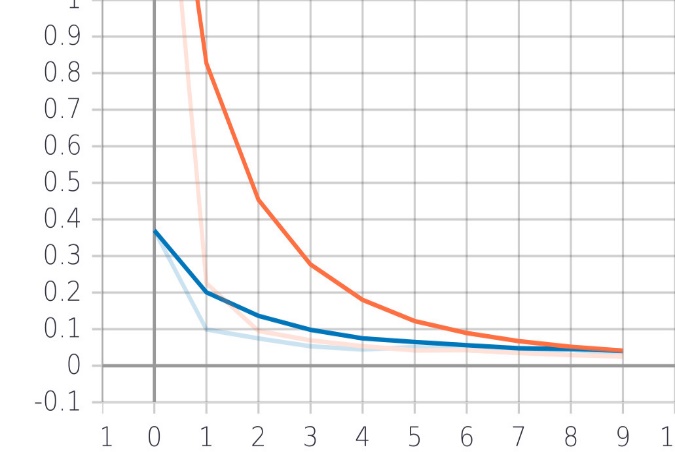


Figure - LOSS FUNCTION OVER EPOCHS

1. **Workflow**

* The workflow starts with taking an image of puzzle and finding the boundary of the desired sudoku puzzle using various image processing steps.
* Next is to crop the image of Sudoku from the provided image and resizing it to 450\*450.
* Next step is to crop the 50\*50 squares of Sudoku from the image.
* Next step is to resize the image to 28\*28 and predict the number from trained Neural Network.
* After a matrix of 81 numbers is identified in the form of a 9\*9 matrix, it is passed on to the backtracking algorithm for Sudoku to solve it.

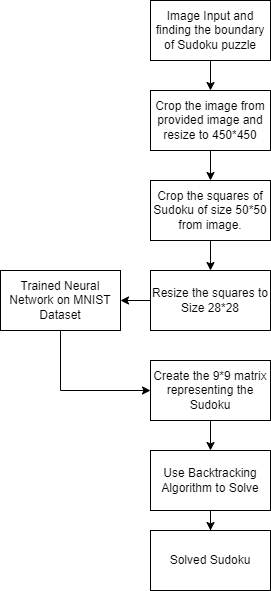


Figure - WORKFLOW FLOWCHART

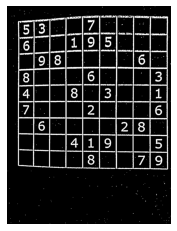
****

Figure - STEP 1

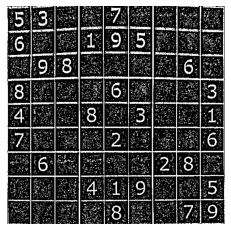
****

Figure - STEP 2

1. **RESULTS**

After training, the model has achieved a high accuracy of 99.20% on training dataset. The graph in figure 4 shows the model accuracy over epochs. The graph in figure 5 shows the loss function of model over different epochs.

1. **CONCLUSION AND FUTURE WORK**

**A. CONCLUSION**

In this paper an effective process to solve sudoku puzzle has been discussed using technologies such as OpenCV, Convolutional Neural Network and Backtracking Algorithm. The process starts with finding the boundary of the puzzle and extraction of digits using the square structure of sudoku and then using Backtracking algorithm to solve the extracted array.

1. **Merits and Limitations**

* Merits of this model is that it is easy to use and can be implemented using any web applications.
* The computer vision technologies used are latest.
* The accuracy of the model maybe effected due to quality, angle and shape of the puzzle images.
* The model accuracy is high and therefore the predictions are very accurate.
* A real time model would be more efficient as it would take multiple images and process these images and produce an averaged output.

1. **FUTURE WORK**

This model can be developed as an mobile application and more image processing techniques can be added to remove errors and incorrect detection of digits.

1. **REFRENCES**