Sudoku Solver: -

Project Report

Abstract: -

Take pictures of unsolved Sudoku puzzles, store it on your laptop or mobile, run the project to get their solutions!

Implementation: -

First, we need to install all the dependencies.

Dependencies: -

A python environment and following libraries: -

OpenCV, numpy, matplotlib, PIL(Pillow), easyocr

Keras.tensorflow, tensorflow.

And tkinter

Optional: -

If you want to upload pictures from your phone to a flask server for solving, the following needs to be installed: -

A java environment and Android Studio

Then pull the code from: - https://github.com/ahmedfgad/AndroidFlask author-Ahmed Gad

Before building the code as an app to your phone,

Turn on developer mode on your phone, then turn on USB debugging.

Then in the application part of Android Manifest file, add

```
android:requestLegacyExternalStorage="true"
```

Find out sdk of your Phone, make sure its KitKat

Then in build.gradle(:app), change the compileSDKVersion, targetSDKVersion and minSDKVersion to what suit your phone.

Go to permissions of the app, (once built and installed) and turn on permissions!!

Also, the app works better when laptop is connected to your phone's hotspot.

Hopefully this saves your time in running the application!

The detailed tutorial is written by Ahmed Gad himself: - https://heartbeat.fritz.ai/uploading-images-from-android-to-a-python-based-flask-server-691e4092a95e

Step-By-Step Procedure to use the program: -

The project file will contain Main.py, Flask_Server.py, Process_Image.py and My_Sudoku_Solver.py

It will also contain myModel.h5 written by Murtaza Hassan. A model trained by him to read numbers.

And a few images of unsolved Sudoku's.

To run the project, run Main.py

You will get a choice to upload from laptop OR from your android mobile device.

If you choose laptop, only the path name is required.

If you choose the phone option, open EasyUpload, Ahmed Gad's application, and when the python program discloses its IP Address and port its listening too, put it on your phone. The flask server is now ready for receiving files.

Select a photo from gallery by the "select photo option" and then connect to server and upload.

Once a photo is fetched by the program, it will be notified.

Next, you will be asked if the photo needs to be rotated, as at times in transmission as a byte stream, photo gets rotated.

Then the photo is scanned and read and it creates a Sudoku puzzle.

It creates a tkinter window and the user can correct the puzzle Sudoku grid (**if** there was any error in reading the numbers)

Once you click on the solve button, The Sudoku will be solved and an image will be returned.

And in the terminal too a solved Sudoku grid will be shown.

Future Aspect: -

On a personal note, this in itself is a future aspect of a project I had written earlier, which was my own algorithm to solve a Sudoku. I always wanted to process an image and use it for I/O operations.

This does not mean the project stops here. A complete Android application can be built which can be connected to the laptop as a virtual host. Since the computer is using a Neural Network to read numbers, it is advisable to not run such a program on your phone. Ahmed Gad himself said he wrote the EasyUpload applications for AI/ML applications and using a computer and its processing power for running computational heavy applications on images.

Right now, the application is running on the laptop. It would still continue to be running on the laptop, but the I/O can be completely shifted to mobile and laptop communication. And A VR headset or smart lens can also implement the same. Additionally, real time vision of the Headset or lens can be used to freeze images, extract the Sudokus and overlay its solution in real time.

On a personal note, I want to continue the project and extend it to other puzzles too, like a Rubix Cube, in which its input coordinates and colors takes forever to input manually. Image processing can be done to speed up the process and I am currently working on it. I expect to get it done by October end.

Acknowledgements: -

I would now like to thank Zense for giving me an opportunity to explore Open-Source development. I have learnt a lot about image preprocessing as I really dug deep into OpenCV. From the Murtaza Hassan tutorial, I also learnt about AI/ML and how he trained a model to read numbers using Deep Learning and Neural Networks. I also learnt about Web Dev as I had to work on a flask server and its thread. I also learnt about Andorid app dev, as I had to build the app and fix the bugs myself by opening up the xml and java files. I also learnt about Multithreading and the dangers of using a non-thread safe application like matplotlib and also synchronous and asynchronous programming and how to avoid program warnings while running a flask server. Overall, I really enjoyed working with the open environment and its applications.