Master Guide — Sudoku Companion Toolkit

Welcome! This guide walks you from zero to a complete demo: **photo → rectify → candidates → next moves → overlays → storyboard → GIF/MP4**

# Who this is for

Creators or PMs who want to see the end-to-end experience without deep coding.

# What you need

• A computer with Python 3.10+ installed (macOS, Windows, or Linux).

• A Sudoku photo (from your book).

• Basic terminal access (Terminal on macOS, PowerShell on Windows).

# 1) Download the toolkit

Unzip the archive into any folder. In your terminal, cd into the folder (the one containing README.md).

# 2) Create a virtual environment

macOS / Linux:

python -m venv .venv && source .venv/bin/activate

Windows (PowerShell):

.venv\Scripts\Activate.ps1

# 3) Install minimal dependencies

pip install -r requirements.txt

# 4) Run a quick end-to-end demo

This demo uses a mock classifier (no ML training needed):

a) Generate overlays and moves JSON:

python apps/cli/demo\_cli\_overlay.py --image /path/to/photo.jpg --out demo\_export --mode demo --max\_moves 5 > demo\_export/moves.json

b) Open demo\_export/ in your file explorer—you should see:

• board\_warped.png — rectified board

• overlay\_move\_01.jpg, overlay\_move\_02.jpg, ... — annotated moves

• moves.json — the printed JSON with suggested moves

c) Create a storyboard sheet (Letter or A4):

python apps/cli/storyboard\_sheet.py --dir demo\_export --out demo\_export/storyboard --paper letter --cols 2 --json demo\_export/moves.json

d) Make animations:

python apps/cli/animate\_gif.py --dir demo\_export --out demo\_export/moves.gif

python apps/cli/animate\_mp4.py --dir demo\_export --out demo\_export/moves.mp4

# 5) (Optional) Train the small cell model

If you want the model to read digits & notes instead of using the mock classifier:

a) Install training extras:

pip install -r requirements-train.txt

b) Generate synthetic data:

python vision/cell\_model/make\_synthetic\_cells\_plus.py --out data/synth\_plus --train 5000 --val 1000

c) Train:

python vision/cell\_model/train\_cell\_model.py --root data/synth\_plus --epochs 3 --bs 256 --train\_steps 200 --val\_steps 40

d) Convert to TFLite or CoreML (for mobile):

python vision/cell\_model/convert\_to\_tflite.py --keras checkpoints/sudoku\_cell\_model.keras --manifests data/synth\_plus/train\_manifest.jsonl data/synth\_plus/val\_manifest.jsonl --out checkpoints/sudoku\_cell\_model\_int8.tflite

# 6) (Optional) Run the Tool API

pip install -r requirements-api.txt

uvicorn apps.api.sudoku\_tool\_api:app --reload

# Troubleshooting

• If Python says 'command not found', try 'python3' instead of 'python'.

• If GIF/MP4 generation fails, ensure 'imageio' & 'imageio-ffmpeg' installed (already in requirements).

• For training: TensorFlow can be heavy; ensure you have enough disk/RAM. CPU-only installs are fine for small experiments.

# Folder overview

docs/ — Master Guide, LLM prompt, training notes

vision/ — OpenCV rectifier + model training

solver/ — human-style solver + tool layer

apps/cli/ — demos & visualization

apps/api/ — optional FastAPI server

mobile/ — Android/iOS inference stubs

That's it—enjoy the demo!

# Appendix — Sample & Makefile

Use the provided sample image and Makefile to run everything with a few commands:

1) Open Terminal/PowerShell in the toolkit folder.

2) Run: make install

3) Run: make demo

4) Optionally: make storyboard, make gif, make mp4, or make all

The sample is at samples/sample\_puzzle.jpg; outputs go to demo\_export/.