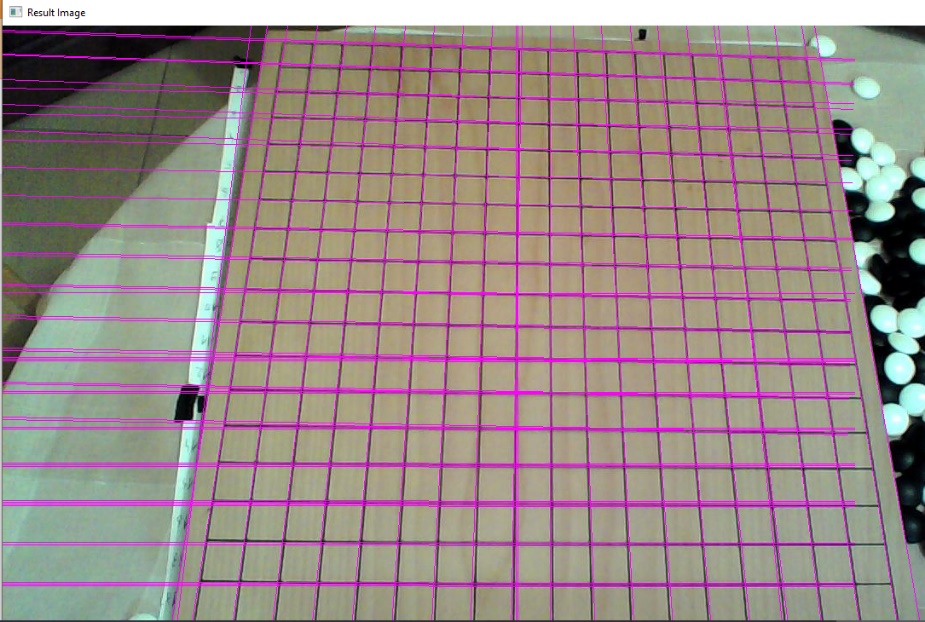
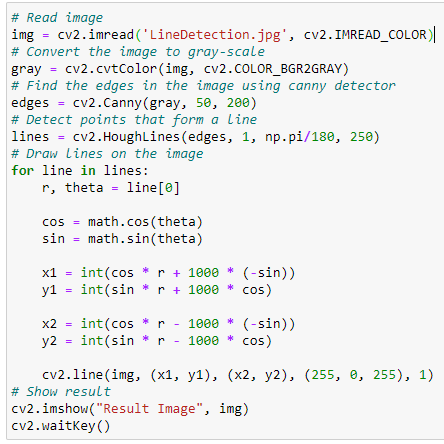
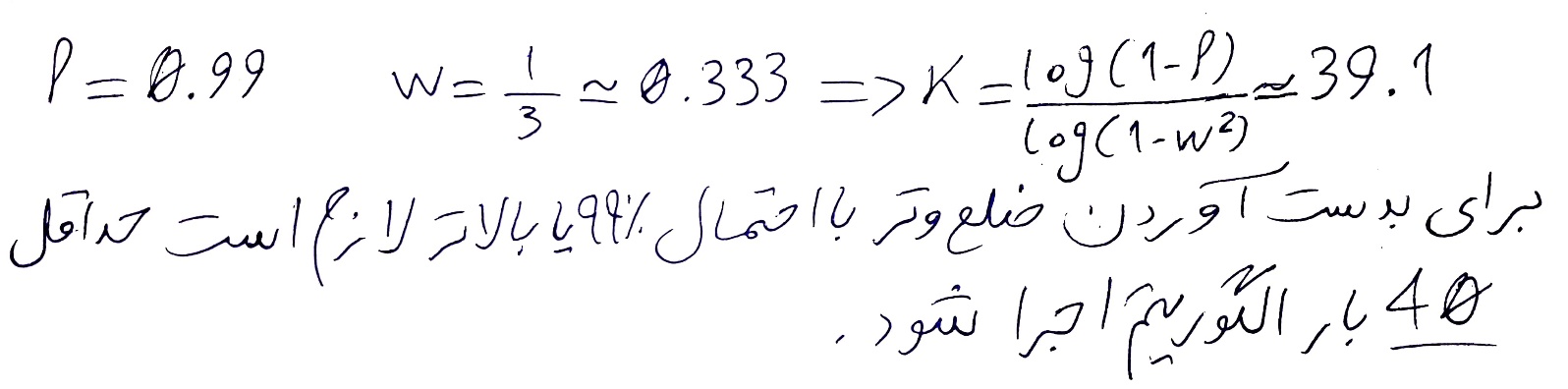
1.

۲. تابع **HoughLines**



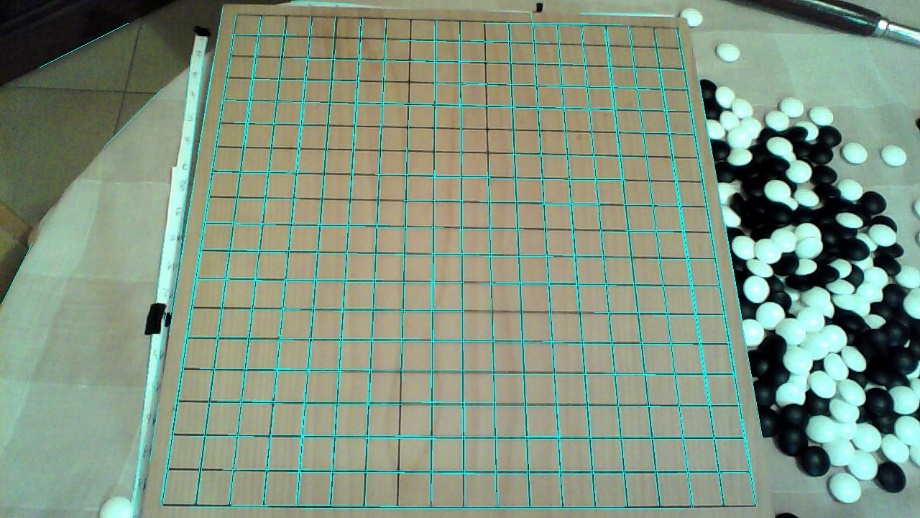
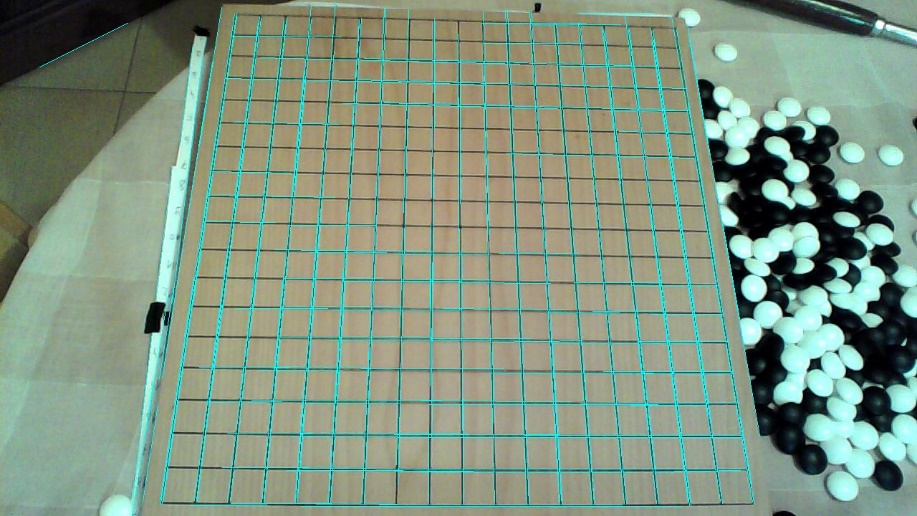
**خروجی تابع HoughLines خطوط یافت‌شده است بطوری که هر خط بصورت آرایه‌ی تک‌عضوی [[r,theta]] خروجی داده می‌شود. پس از بدست‌آوردن دو نقطه دلخواه از خط، آن را رسم می‌کنیم. (اگر بجای ۱۰۰۰ ضریب کوچکتری می‌گذاشتیم خطوط طول کمتری می‌داشتند و عدد بزرگتری می‌گذاشتیم خطوط طولانی‌تر بودند چون دو نقطه‌ی دلخواه را با فاصله بیشتری انتخاب کرده بودیم)

تابع **HoughLinesP**

مقادیر مختلف زیادی برای پارامترها بررسی کردم. بهترین نتیجه را از این دو مقداردهی گرفتم:

lines = cv2.HoughLinesP(edges, 0.25, np.pi/360, 20, minLineLength=100, maxLineGap=15)

lines = cv2.HoughLinesP(edges, 0.25, np.pi/360, 20, minLineLength=80, maxLineGap=12)

**بطور کلی با رزولوشن پیکسلی 0.25 و رزولوشن درجه نتایج بهتری می‌گرفتم.در هر دو مقداردهی حد آستانه رای‌گیری خط معتبر را ۲۰ رای گرفتم. کمینه طول موردقبول خطوط و حداکثر فاصله موردقبول نقطه از خط را در مقداردهی اول و دوم به ‌ترتیب (۱۰۰، ۱۵) و (۸۰، ۱۲) گرفتم.

Result Image1.jpg Result Image2.jpg

*منابع:*

*سوال۱.الف:* [Hough Transform using OpenCV | LearnOpenCV](https://learnopencv.com/hough-transform-with-opencv-c-python/)

[Line detection in python with OpenCV | Houghline method - GeeksforGeeks](https://www.geeksforgeeks.org/line-detection-python-opencv-houghline-method/)