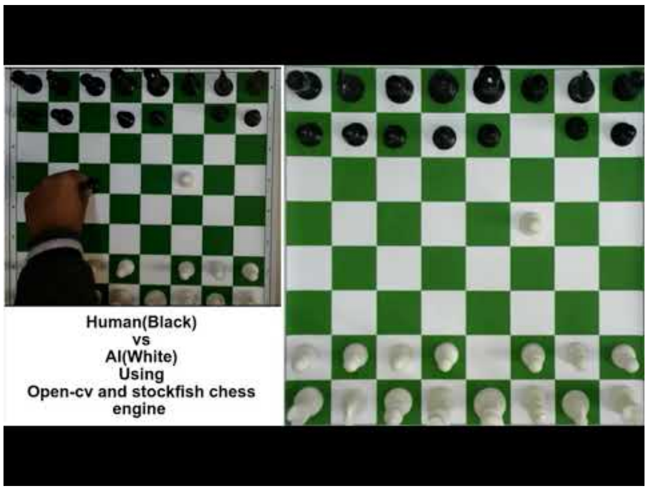
opencv-chess

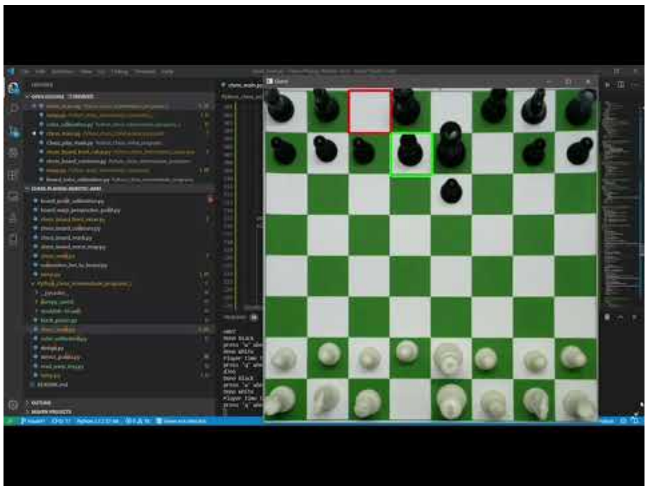
Human vs AI (Stockfish engine)

Camera captures the image of chessboard then the imageis analyzed using imageprocessing to identify the moves made by opponent and stockfish engine calculates the best possible move.

CV Open-CV Py Python3

Youtube Video

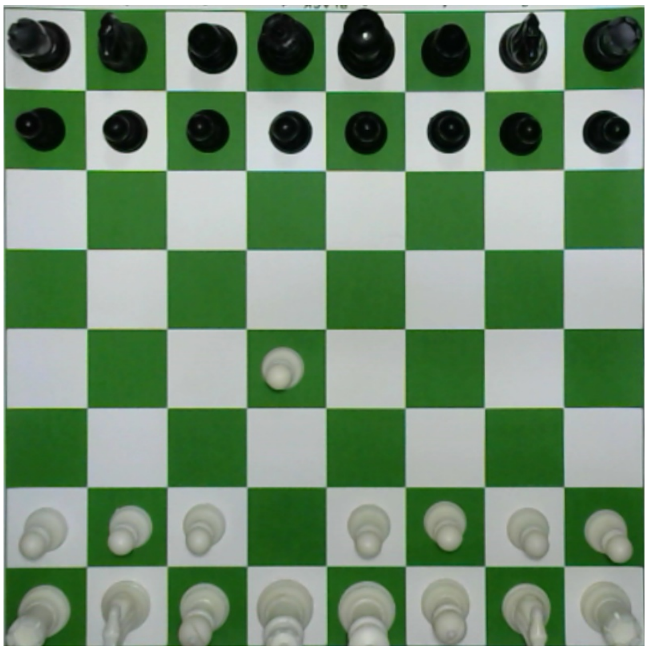
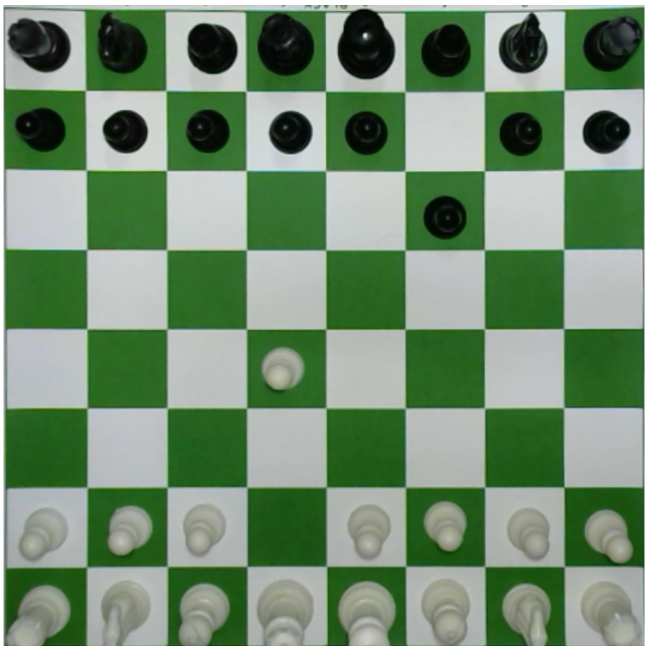




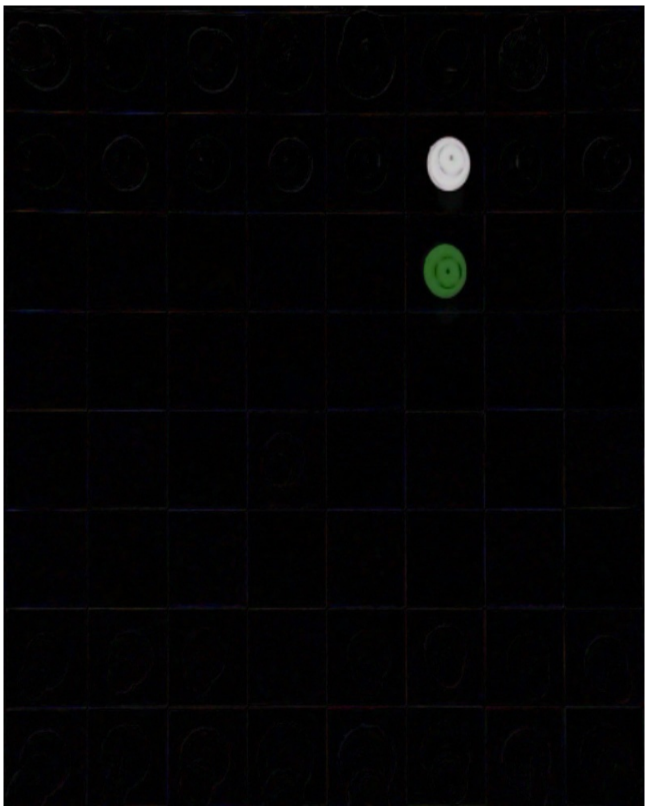
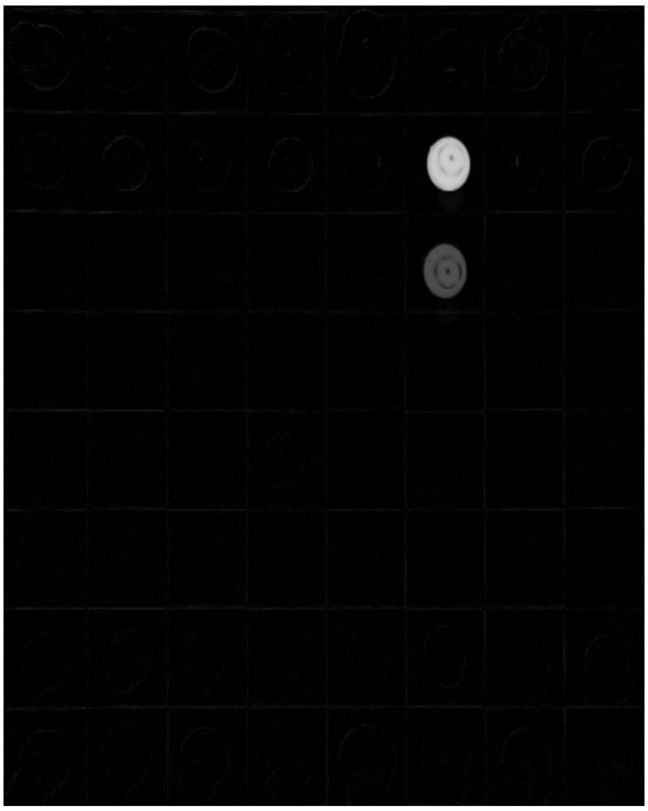
Method of Working

Step - 1

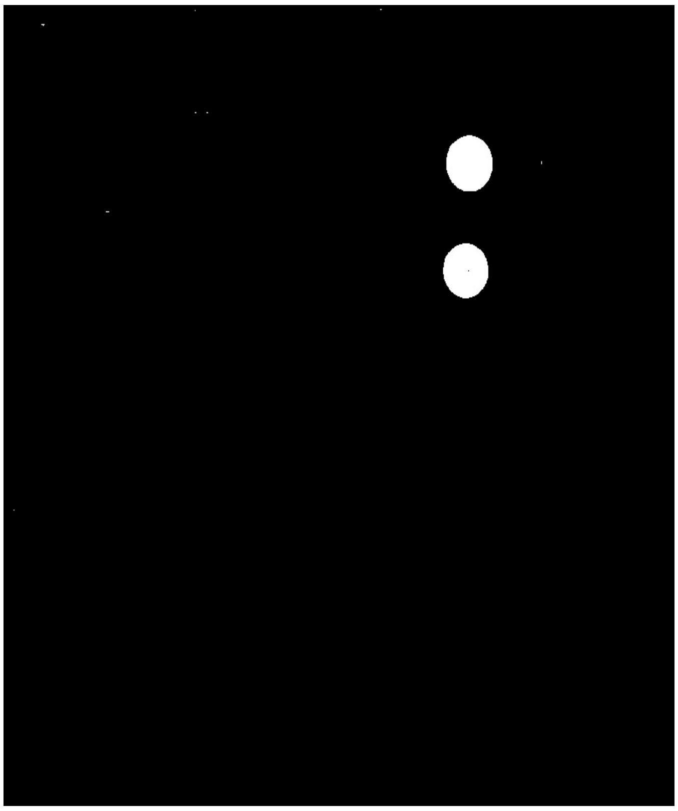
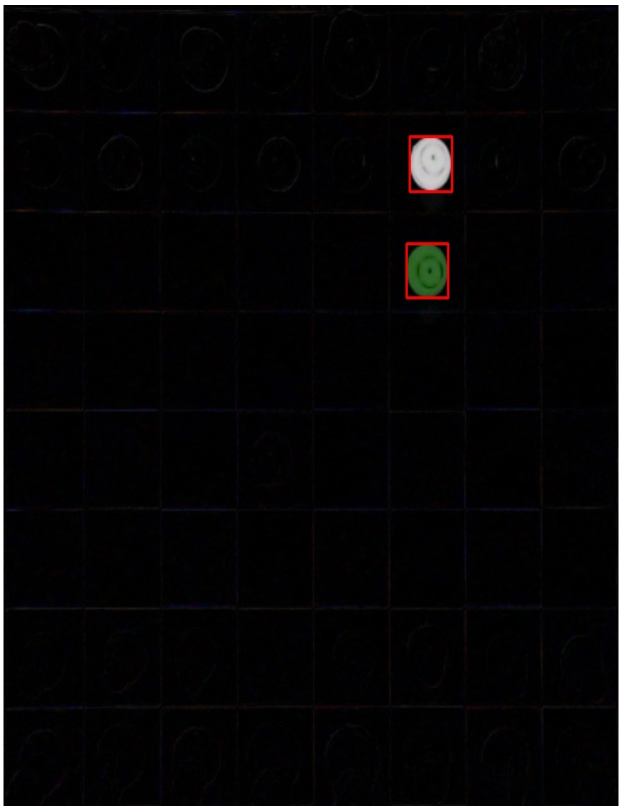
Image1 : Image of Chess Board befor player move piece	**Image2 : Image of Chess Board after player move piece**

Image1 : Image of Chess Board befor player move piece	Image2 : Image of Chess Board after player move piece
	

step - 2

Difference of image by using function absdiff in CV2	Change Difference_image to Gray scale image
diff = cv2.absdiff(image1,image2)	diff_gray = cv2.cvtColor(diff,cv2.COLOR_BGR2GRAY)
	

step - 3

Apply thresholding on Grayscale image	Find Contours on threshold image
matrix,thresold = cv2.threshold(diff_gray,value,255,cv2.THRESH_BINARY)	cnts,_ = cv2.findContours(thresold, cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
	

Main Variables

Variables	Explain
points = []	# contains chess board corners points
boxes = np.zeros((8,8,4),dtype=int)	# contains top-left and bottom-right point of chessboard boxes
fen_line = 'rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR'	# fen line of chess board
board = chess.Board(fen=fen_line)	# object of chess board
dir_path = os.path.dirname(os.path.realpath(file))+"/numpy_saved"	# path of current directory
device = cv2.VideoCapture(1)	# set devidce for read image (1: for tacking input from usb-webcam)
img_resize = (800,800)	# set o/p image size
engine = chess.engine.SimpleEngine.popen_uci("stockfish-10-win\Windows\stockfish_10_x64.exe")	# stockfish engine
chess_board = []	# it will store chess board matrix

Variables	Explain
bool_position = np.zeros((8,8),dtype=int)	# store bool matrix of Board
number_to_position_map = []	# map move values for [0,0]-> (8,a) , [0,1]-> (8,b).... so on

Main Functions

Function Name	Explain
get_points(img,n)	select n points on image by double click and returns list of selected points
get_warp_img(img,dir_path,img_resize)	return warp prespective of image taken by camera and resize it to img_resize value
map_function()	makes a dictonary to map values { "a8":[0,0],"b8":[0,1],.... so on }
fen2board(fen_line)	retuen a 8X8 matrix of chess player piece name and bool position
board2fen(chess_board)	return fen line of chess board
map_function_for_number_2_position()	makes a list for map values [0,0]="8a", [0,1]="8b", [0,2]="8c",... so on
rectContains(rectangle,mid_point)	logic function for checking given mid_point is inside the rectangle or not
show_game(game_img,board,player_move)	This function shows all game in proper format with plane turn, opponent's last move, current chess **board, red and green boxes on moved piece, etc.**
set_legal_positions(game_image,board,boxes)	if Illegal move found in chess it shows last correct state of chess board

Author

```
Vatsal Parsaniya
```