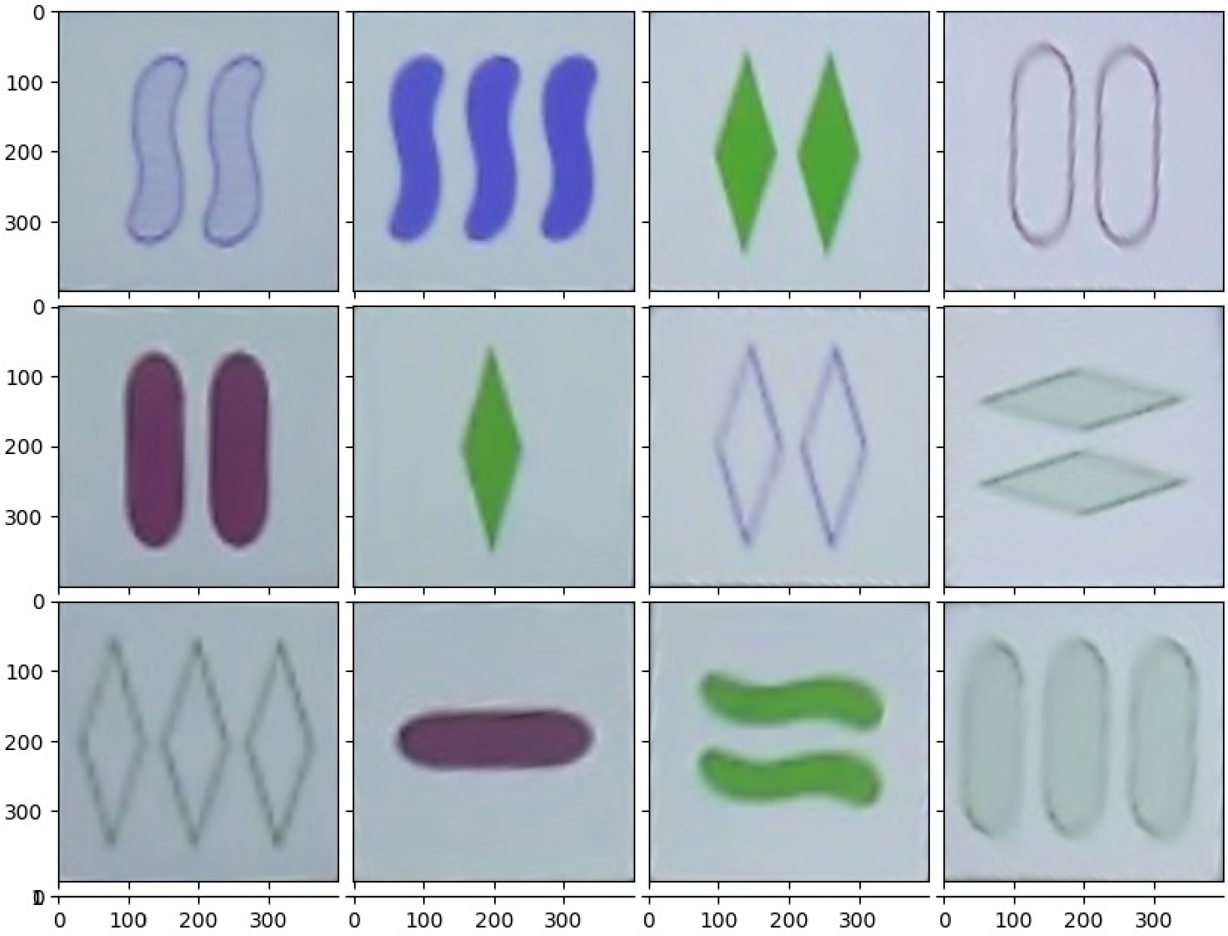
```
import numpy as np
import cv2
from matplotlib import pyplot as plt
from mpl_toolkits.axes_grid1 import ImageGrid
import math

img = cv2.imread('C:/Users/Fernando/Desktop/visao computacional -
jupyter/teste.jpg')
cv2.imshow('imagem original', img)


gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray,(1,1),1000)
#cv2.imshow('filtro blur', blur)



flag, thresh = cv2.threshold(blur, 120, 255, cv2.THRESH_BINARY)
#cv2.imshow('threshold', thresh)


contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
contours = sorted(contours, key=cv2.contourArea, reverse=True)



# selecionando perímetros maiores
perimeters = []
for i in range(len(contours)):
    perimeters.append(cv2.arcLength(contours[i],True))

listindex = []
for i in range(15):
    if perimeters[i] > perimeters[0]/2:
        listindex.append(i)

numcards=len(listindex)

imgcont = img.copy()
[cv2.drawContours(imgcont, [contours[i]], 0, (0,255,0), 5) for i in
listindex]

cv2.imshow('imagem contono', imgcont)


warp = []

for i in range(numcards):
    card = contours[i]
    peri = cv2.arcLength(card,True)
    approx = cv2.approxPolyDP(card,0.02*peri,True)
    rect = cv2.minAreaRect(contours[i])

    h = np.array([ [0,0],[399,0],[399,399],[0,399] ],np.float32)
    approx = np.array([item for sublist in approx for item in
sublist],np.float32)

    transform = cv2.getPerspectiveTransform(approx,h)

    warp.append(cv2.warpPerspective(img,transform,(400,400)))


fig = plt.figure(1, (10,10))
grid = ImageGrid(fig, 111,
                nrows_ncols = (4, 4),
                axes_pad=0.1,
                aspect=True,
                )

for i in range(numcards):
    grid[i].imshow(warp[i])


while (1):
    if cv2.waitKey(10) & 0xFF == 27:
        break
# # para sair deve-se pressionar ESC
cv2.destroyAllWindows()
```


0
100
200
300
0
100
200
300