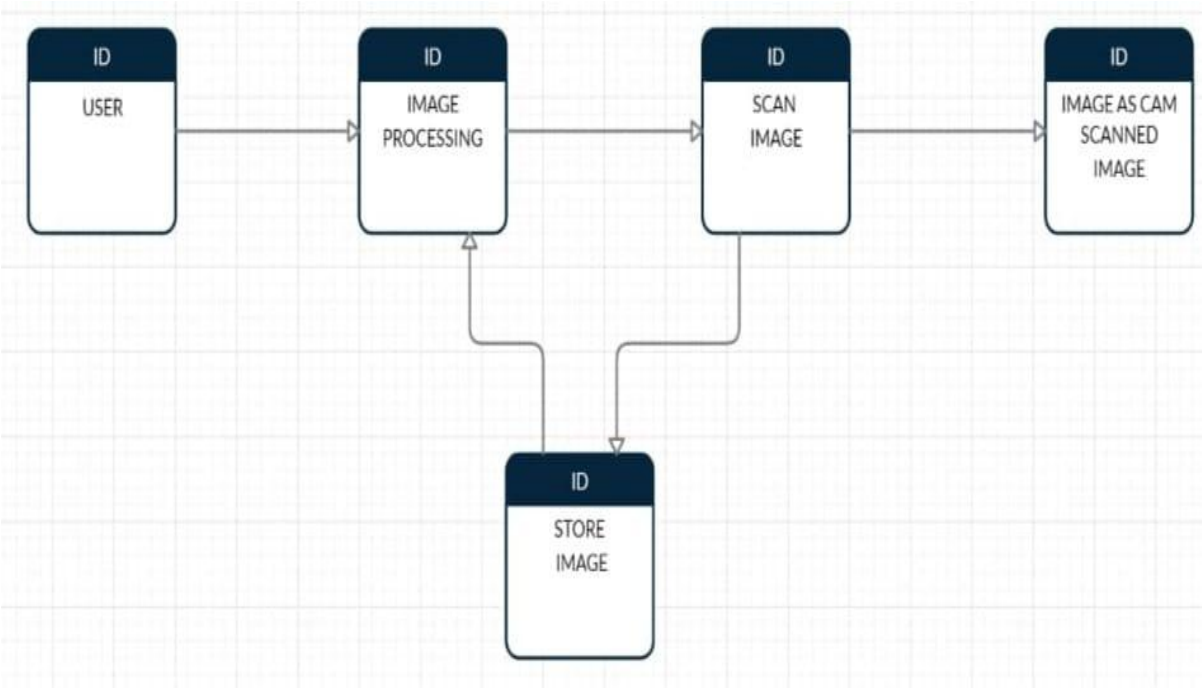
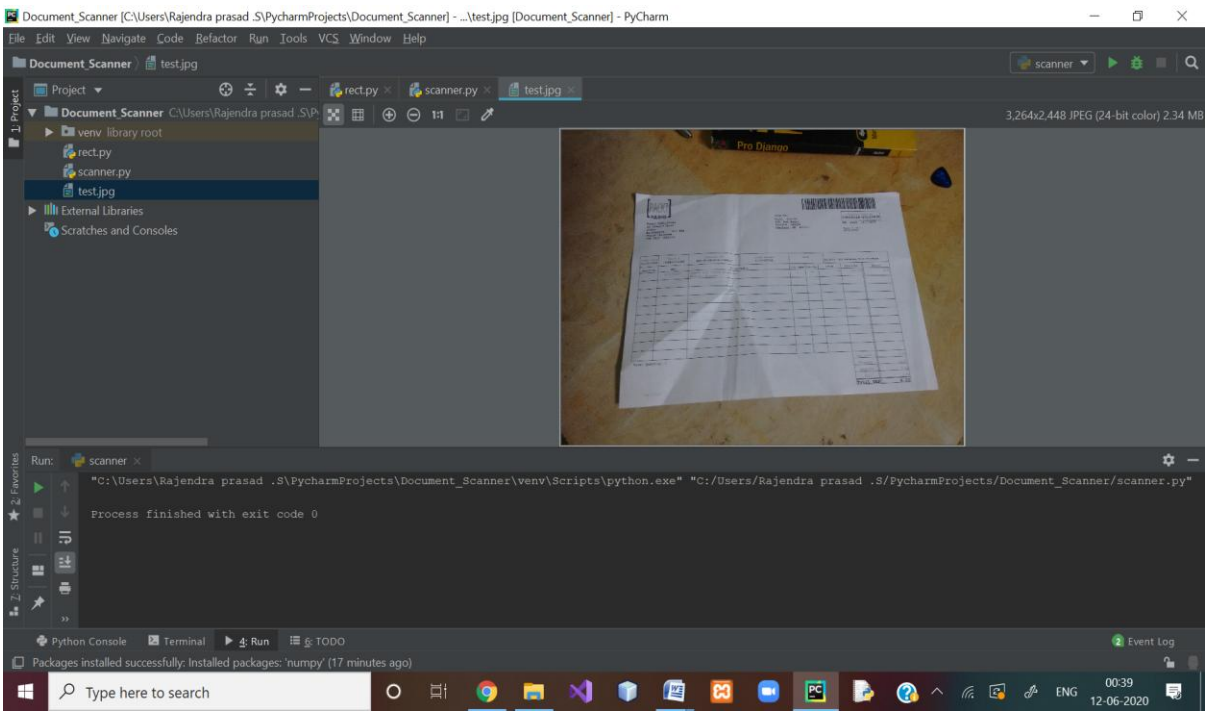
DOCUMENT SCANNER

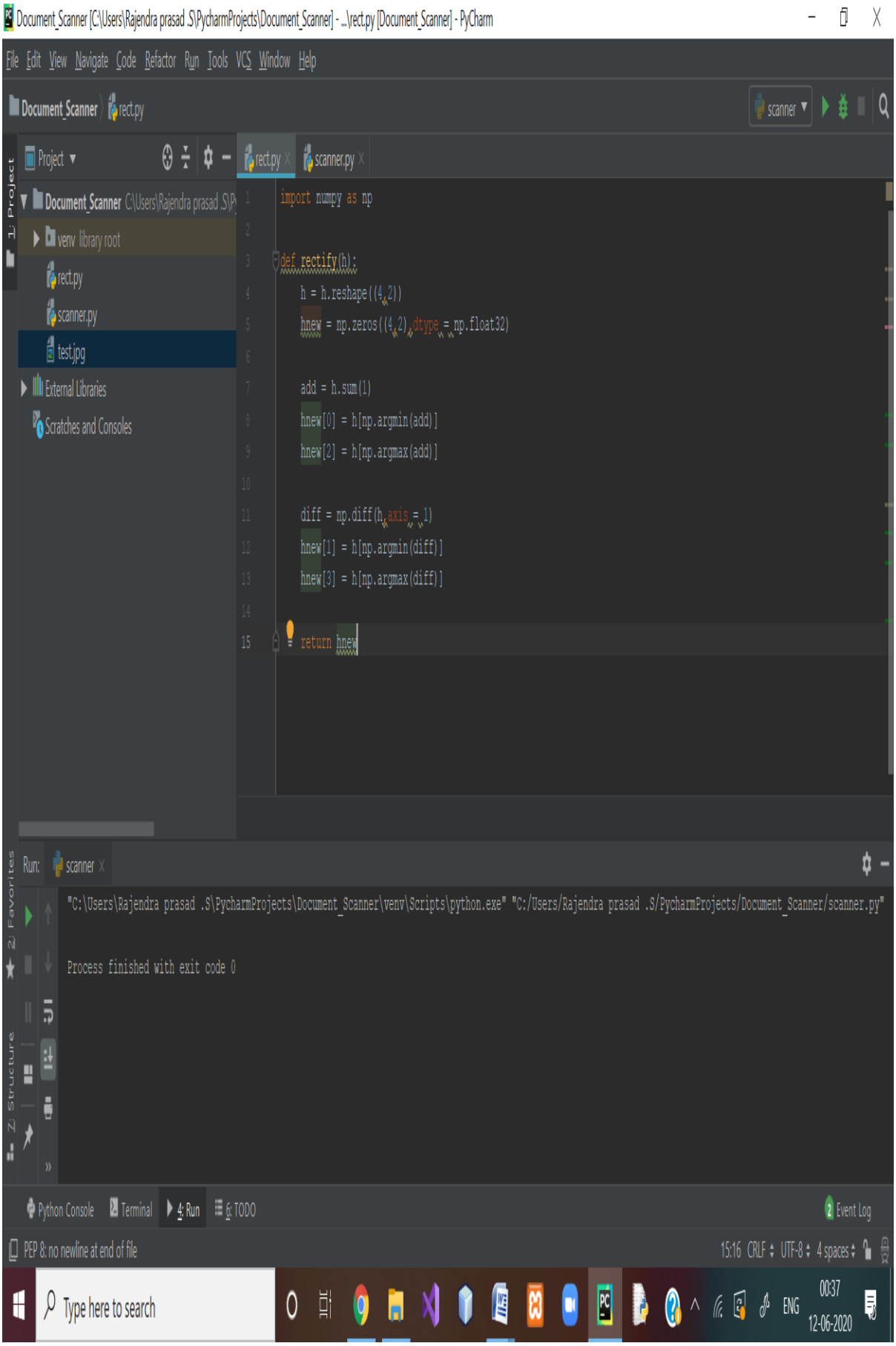
Data flow diagram:



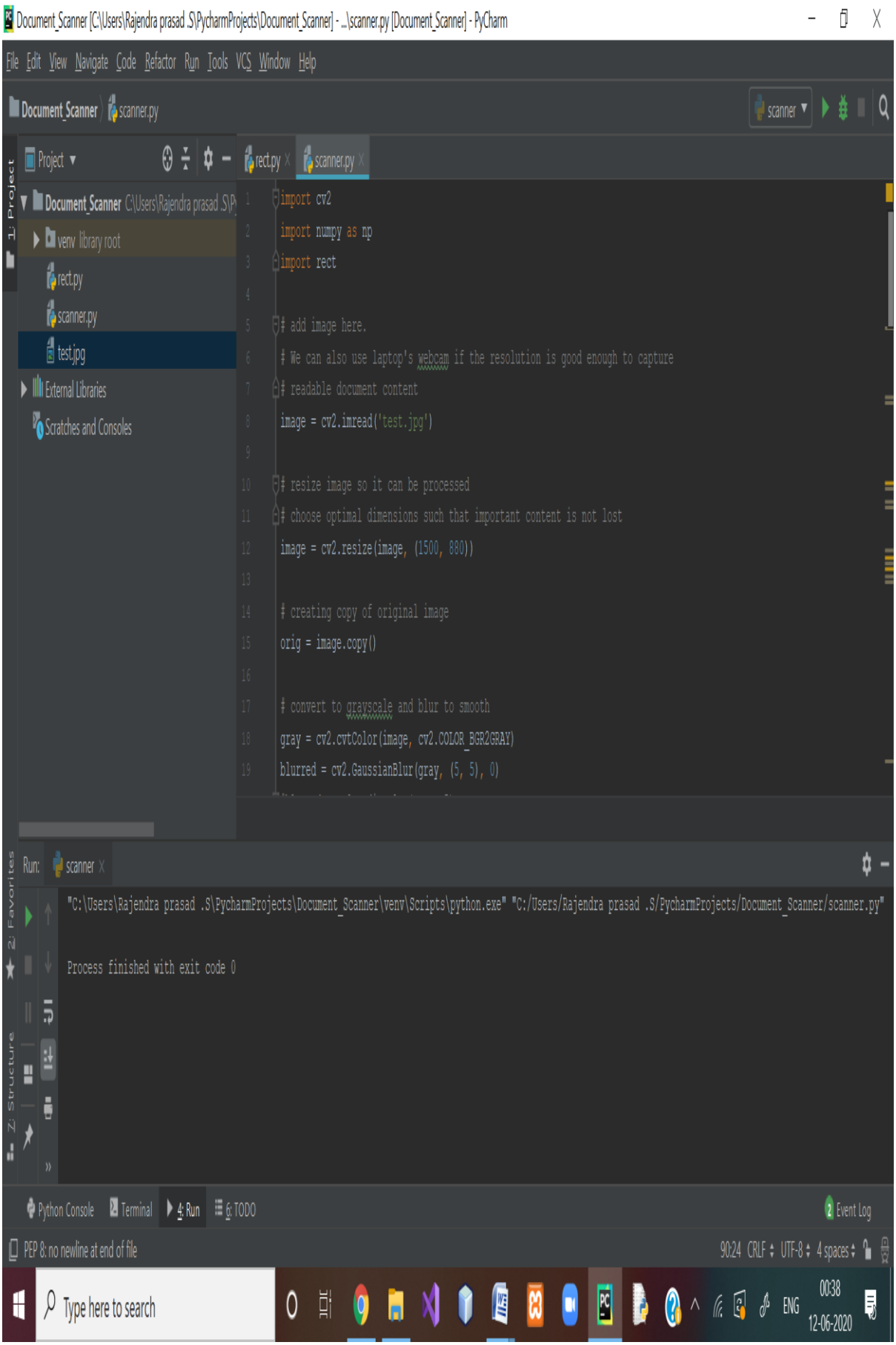
Project demonstration:



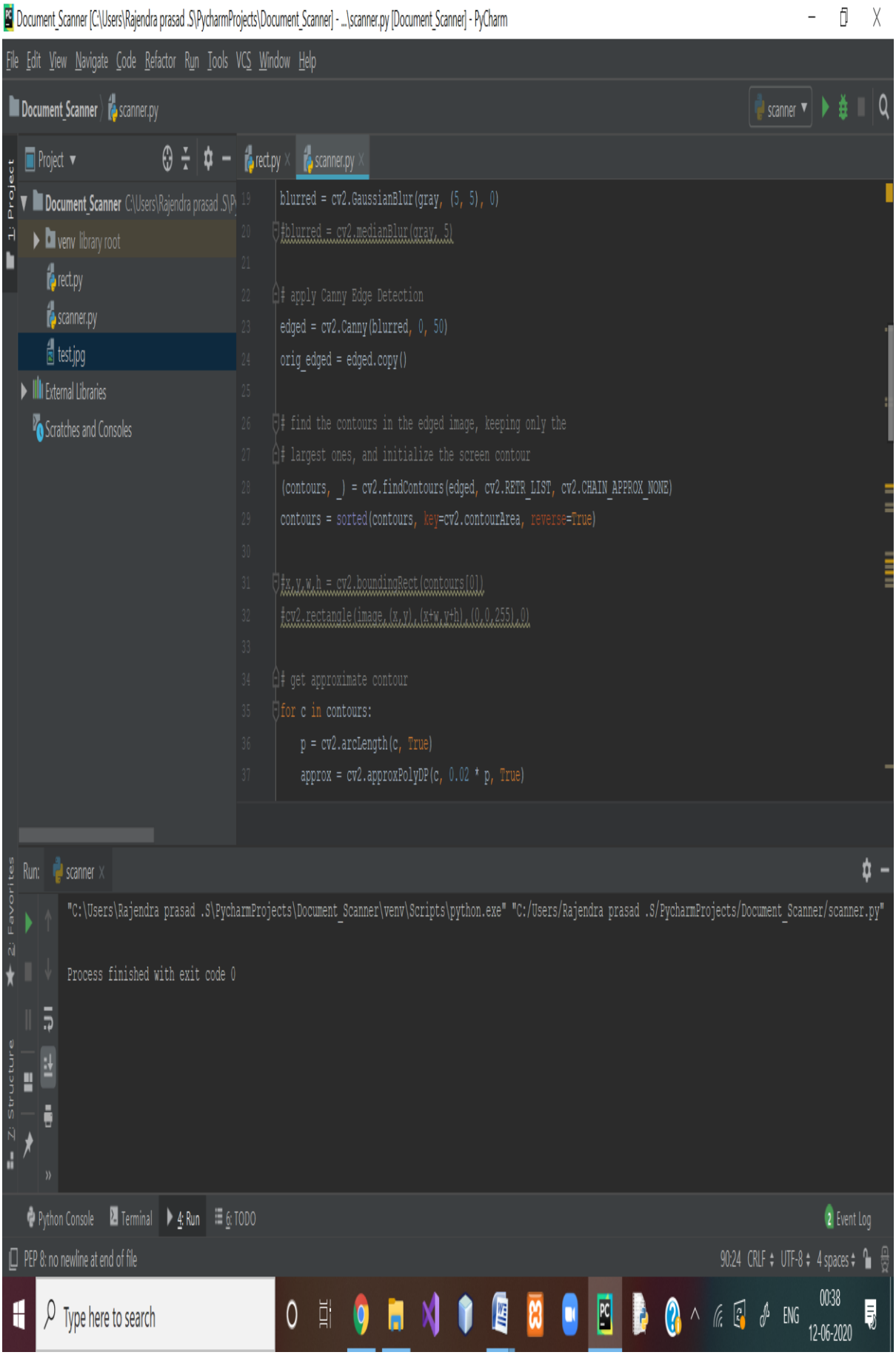
The original image

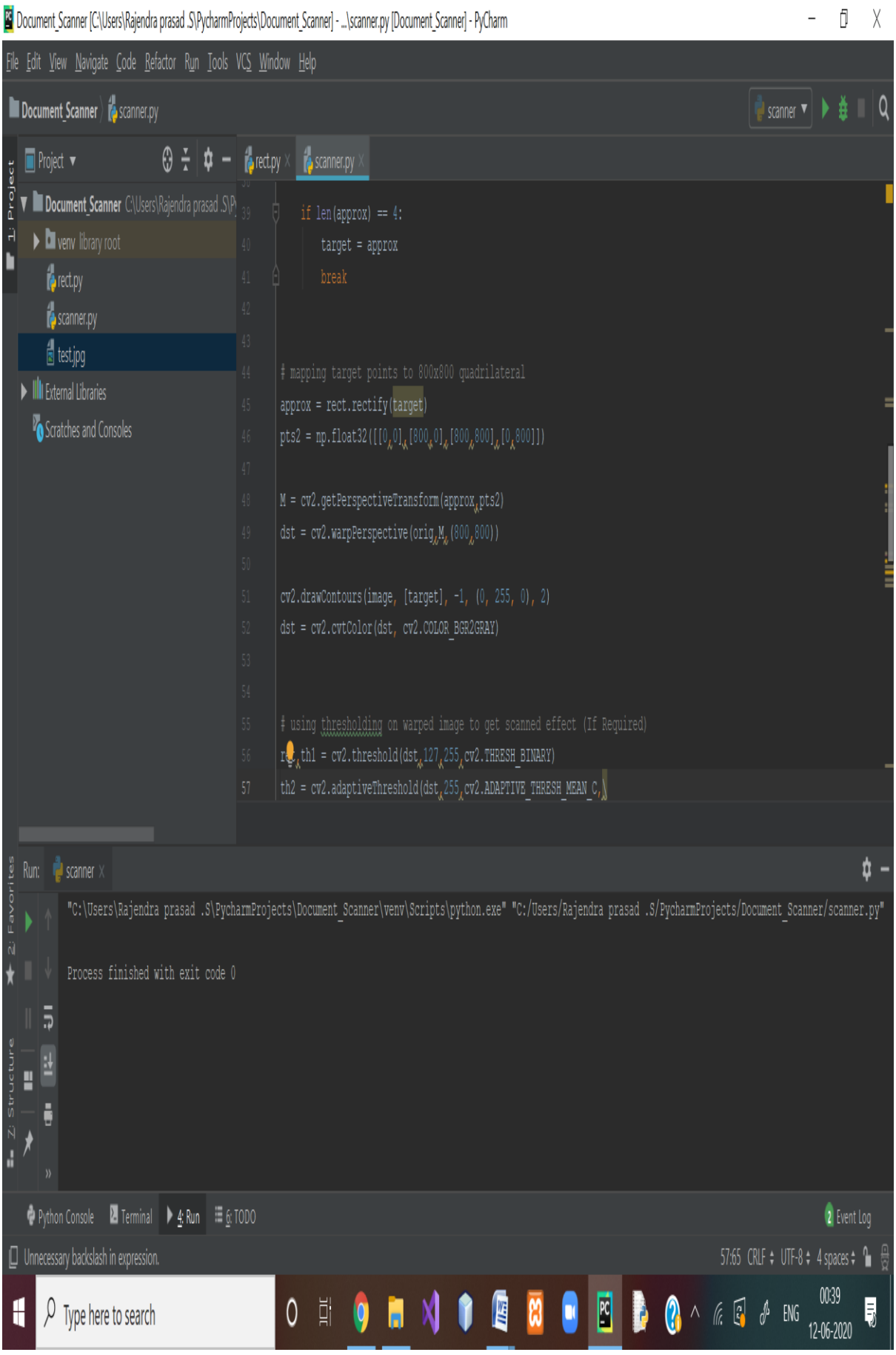

Document_Scanner [C:\Users\Rajendra prasad .S\PycharmProjects\Document_Scanner] - ...\rect.py [Document_Scanner] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Document_Scanner rect.py
scanner
Project
rect.py
scanner.py
Document_Scanner C:\Users\Rajendra prasad .S\P
venv library root
rect.py
scanner.py
test.jpg
External Libraries
Scratches and Consoles
import numpy as np
def rectify(h):
h = h.reshape((4,2))
hnew = np.zeros((4,2),dtype = np.float32)
add = h.sum(1)
hnew[0] = h[np.argmin(add)]
hnew[2] = h[np.argmax(add)]
diff = np.diff(h,axis = 1)
hnew[1] = h[np.argmin(diff)]
hnew[3] = h[np.argmax(diff)]
return hnew
Run: scanner
"C:\Users\Rajendra prasad .S\PycharmProjects\Document_Scanner\venv\Scripts\python.exe" "C:/Users/Rajendra prasad .S/PycharmProjects/Document_Scanner/scanner.py"
Process finished with exit code 0
Python Console
Terminal
4: Run
6: TODO
Event Log
PEP 8: no newline at end of file
15:16 CRLF UTF-8 4 spaces
Type here to search
ENG
0037
12-06-2020

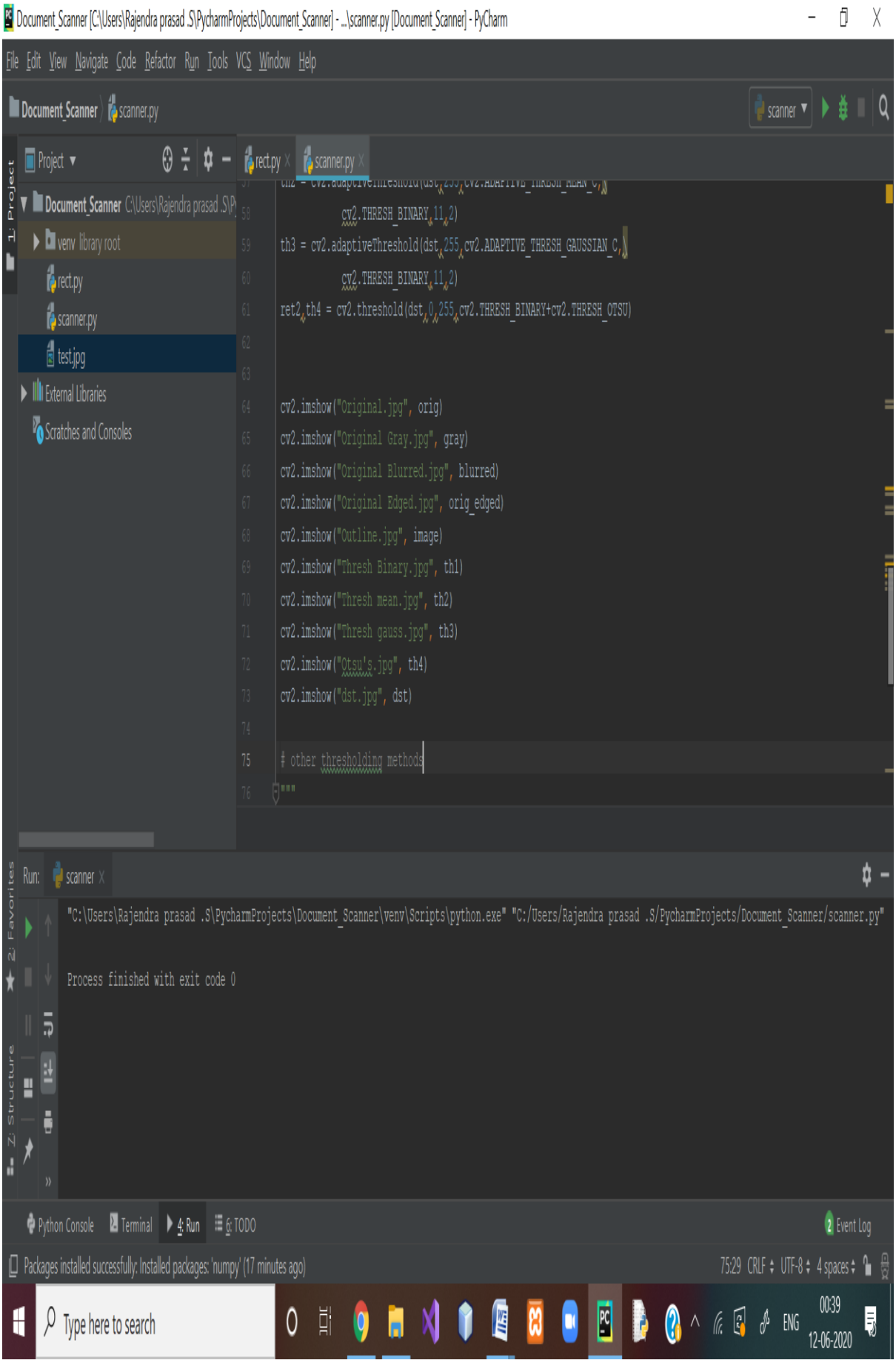
rect.py file

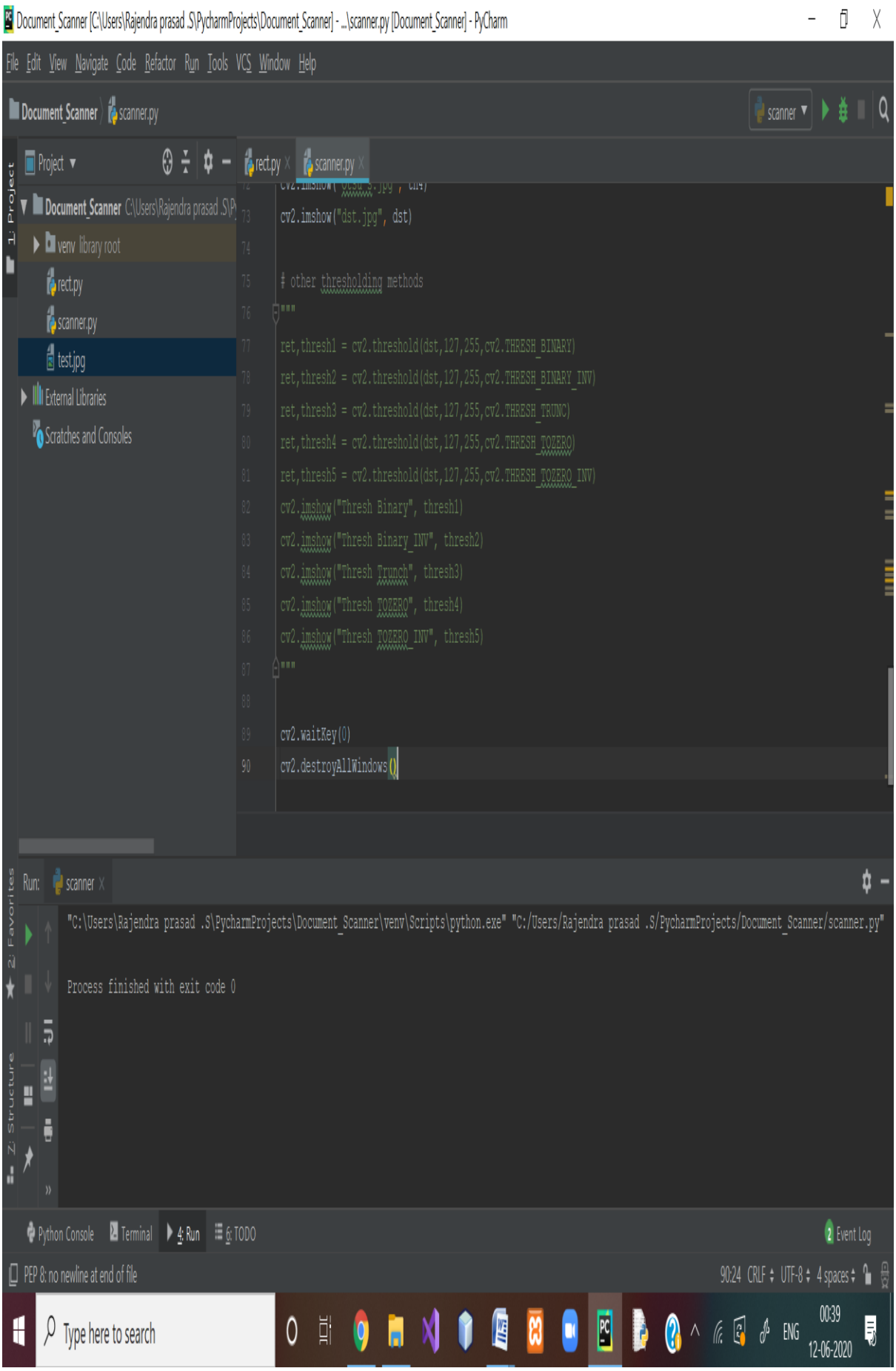


scanner.py file

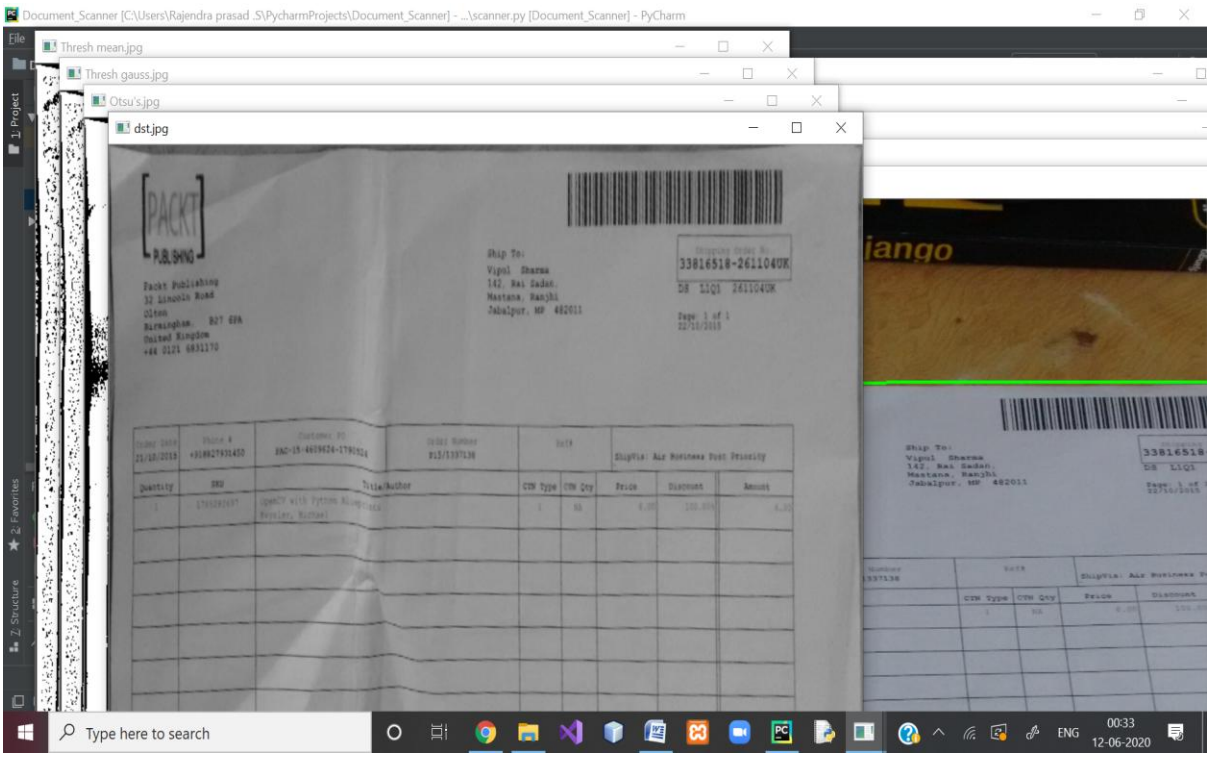

Document_Scanner [C:\Users\Rajendra prasad .S\PycharmProjects\Document_Scanner] - ...\scanner.py [Document_Scanner] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Document_Scanner scanner.py
scanner
Project
rect.py
scanner.py
Document_Scanner C:\Users\Rajendra prasad .S\P
venv library root
rect.py
scanner.py
test.jpg
External Libraries
Scratches and Consoles
blurred = cv2.GaussianBlur(gray, (5, 5), 0)
#blurred = cv2.medianBlur(gray, 5)
# apply Canny Edge Detection
edged = cv2.Canny(blurred, 0, 50)
orig_edged = edged.copy()
# find the contours in the edged image, keeping only the
# largest ones, and initialize the screen contour
(contours, _) = cv2.findContours(edged, cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)
contours = sorted(contours, key=cv2.contourArea, reverse=True)
#x,y,w,h = cv2.boundingRect(contours[0])
#cv2.rectangle(image,(x,y),(x+w,y+h),(0,0,255),0)
# get approximate contour
for c in contours:
p = cv2.arcLength(c, True)
approx = cv2.approxPolyDP(c, 0.02 * p, True)
Run: scanner
"C:\Users\Rajendra prasad .S\PycharmProjects\Document_Scanner\venv\Scripts\python.exe" "C:/Users/Rajendra prasad .S/PycharmProjects/Document_Scanner/scanner.py"
Process finished with exit code 0
Python Console
Terminal
4: Run
6: TODO
Event Log
PEP 8: no newline at end of file
90:24 CRLF UTF-8 4 spaces
Type here to search
ENG
00:38
12-06-2020


Document_Scanner [C:\Users\Rajendra prasad .S\PycharmProjects\Document_Scanner] - ...\scanner.py [Document_Scanner] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Document_Scanner \ scanner.py
scanner
Project
rect.py
scanner.py
Document_Scanner C:\Users\Rajendra prasad .S\P
venv library root
rect.py
scanner.py
test.jpg
External Libraries
Scratches and Consoles
if len(approx) == 4:
target = approx
break
# mapping target points to 800x800 quadrilateral
approx = rect.rectify(target)
pts2 = np.float32([[0,0],[800,0],[800,800],[0,800]])
M = cv2.getPerspectiveTransform(approx,pts2)
dst = cv2.warpPerspective(orig,M,(800,800))
cv2.drawContours(image, [target], -1, (0, 255, 0), 2)
dst = cv2.cvtColor(dst, cv2.COLOR_BGR2GRAY)
# using thresholding on warped image to get scanned effect (If Required)
ret,th1 = cv2.threshold(dst,127,255,cv2.THRESH_BINARY)
th2 = cv2.adaptiveThreshold(dst,255,cv2.ADAPTIVE_THRESH_MEAN_C,
Run: scanner
"C:\Users\Rajendra prasad .S\PycharmProjects\Document_Scanner\venv\Scripts\python.exe" "C:/Users/Rajendra prasad .S/PycharmProjects/Document_Scanner/scanner.py"
Process finished with exit code 0
Python Console
Terminal
4: Run
6: TODO
Event Log
Unnecessary backslash in expression.
57:65 CRLF UTF-8 4 spaces
Type here to search
ENG
0039
12-06-2020

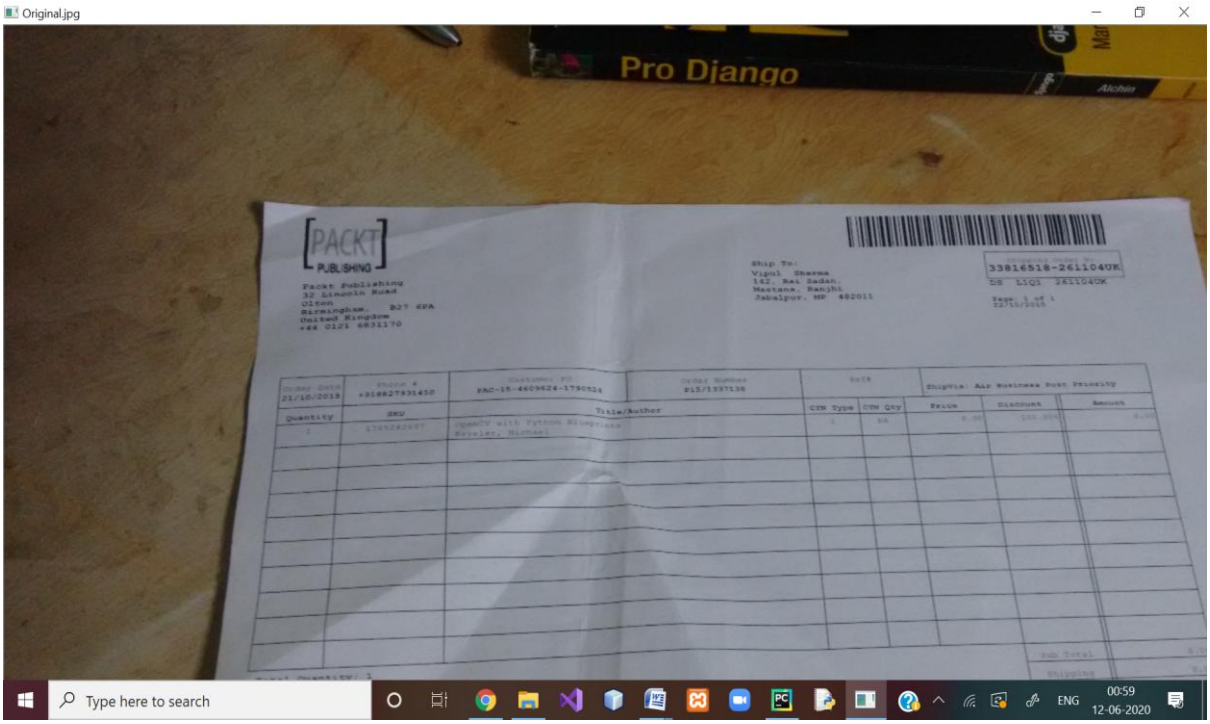

Document_Scanner [C:\Users\Rajendra prasad .S\PycharmProjects\Document_Scanner] - ...\scanner.py [Document_Scanner] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Document_Scanner \ scanner.py
scanner
Project
rect.py
scanner.py
Document_Scanner C:\Users\Rajendra prasad .S\P
venv library root
rect.py
scanner.py
test.jpg
External Libraries
Scratches and Consoles
cv2.THRESH_BINARY,11,2)
th3 = cv2.adaptiveThreshold(dst,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY,11,2)
ret2,th4 = cv2.threshold(dst,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
cv2.imshow("Original.jpg", orig)
cv2.imshow("Original Gray.jpg", gray)
cv2.imshow("Original Blurred.jpg", blurred)
cv2.imshow("Original Edged.jpg", orig_edged)
cv2.imshow("Outline.jpg", image)
cv2.imshow("Thresh Binary.jpg", th1)
cv2.imshow("Thresh mean.jpg", th2)
cv2.imshow("Thresh gauss.jpg", th3)
cv2.imshow("Otsu's.jpg", th4)
cv2.imshow("dst.jpg", dst)
# other thresholding methods
"""
Run: scanner
"C:\Users\Rajendra prasad .S\PycharmProjects\Document_Scanner\venv\Scripts\python.exe" "C:/Users/Rajendra prasad .S/PycharmProjects/Document_Scanner/scanner.py"
Process finished with exit code 0
Python Console
Terminal
4: Run
6: TODO
Event Log
Packages installed successfully: Installed packages: 'numpy' (17 minutes ago)
75:29 CRLF UTF-8 4 spaces
Type here to search
ENG
00:39
12-06-2020

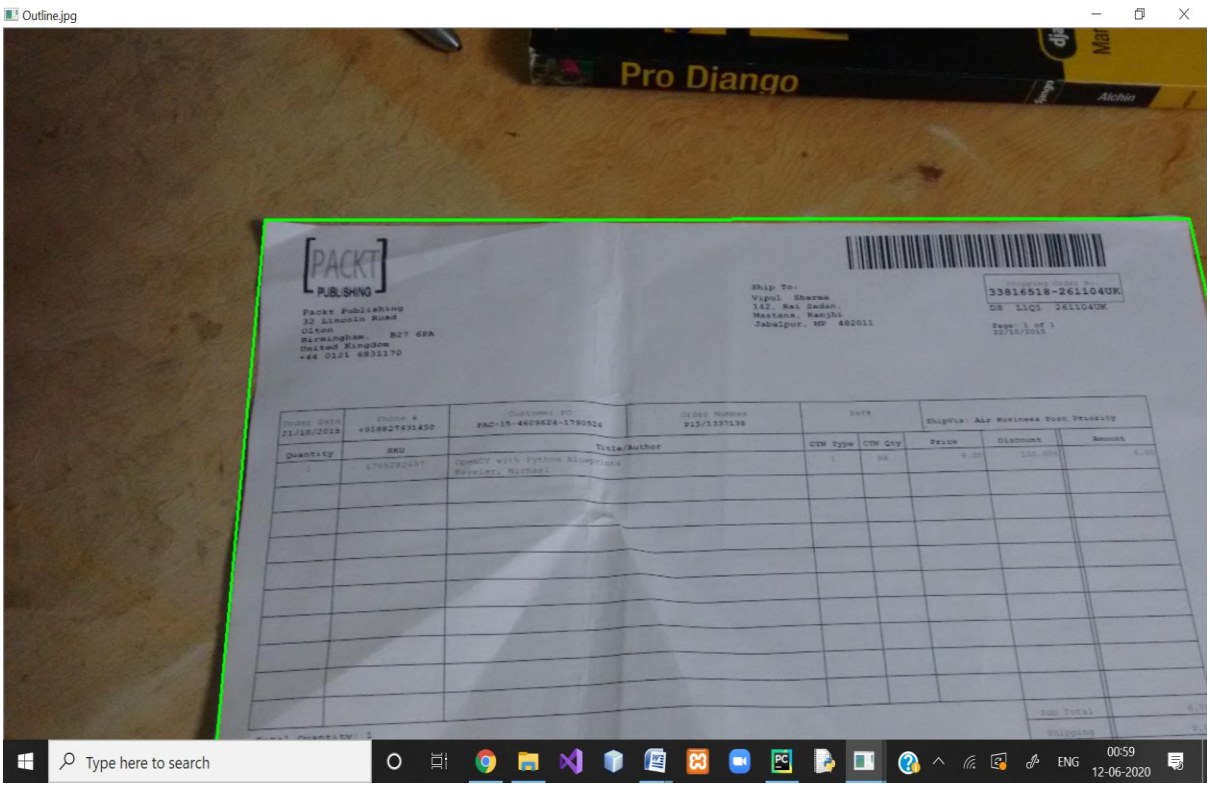

Document_Scanner [C:\Users\Rajendra prasad .S\PycharmProjects\Document_Scanner] - ...\scanner.py [Document_Scanner] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Document_Scanner scanner.py
scanner
Project
rect.py
scanner.py
Document_Scanner C:\Users\Rajendra prasad .S\P
venv library root
rect.py
scanner.py
test.jpg
External Libraries
Scratches and Consoles
cv2.imshow("dst.jpg", dst)
# other thresholding methods
"""
ret,thresh1 = cv2.threshold(dst,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(dst,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(dst,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(dst,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(dst,127,255,cv2.THRESH_TOZERO_INV)
cv2.imshow("Thresh Binary", thresh1)
cv2.imshow("Thresh Binary_INV", thresh2)
cv2.imshow("Thresh Trunch", thresh3)
cv2.imshow("Thresh TOZERO", thresh4)
cv2.imshow("Thresh TOZERO_INV", thresh5)
"""
cv2.waitKey(0)
cv2.destroyAllWindows()
Run: scanner
"C:\Users\Rajendra prasad .S\PycharmProjects\Document_Scanner\venv\Scripts\python.exe" "C:/Users/Rajendra prasad .S/PycharmProjects/Document_Scanner/scanner.py"
Process finished with exit code 0
Python Console Terminal 4: Run 6: TODO
Event Log
PEP 8: no newline at end of file
90:24 CRLF UTF-8 4 spaces
Type here to search
ENG
00:39
12-06-2020

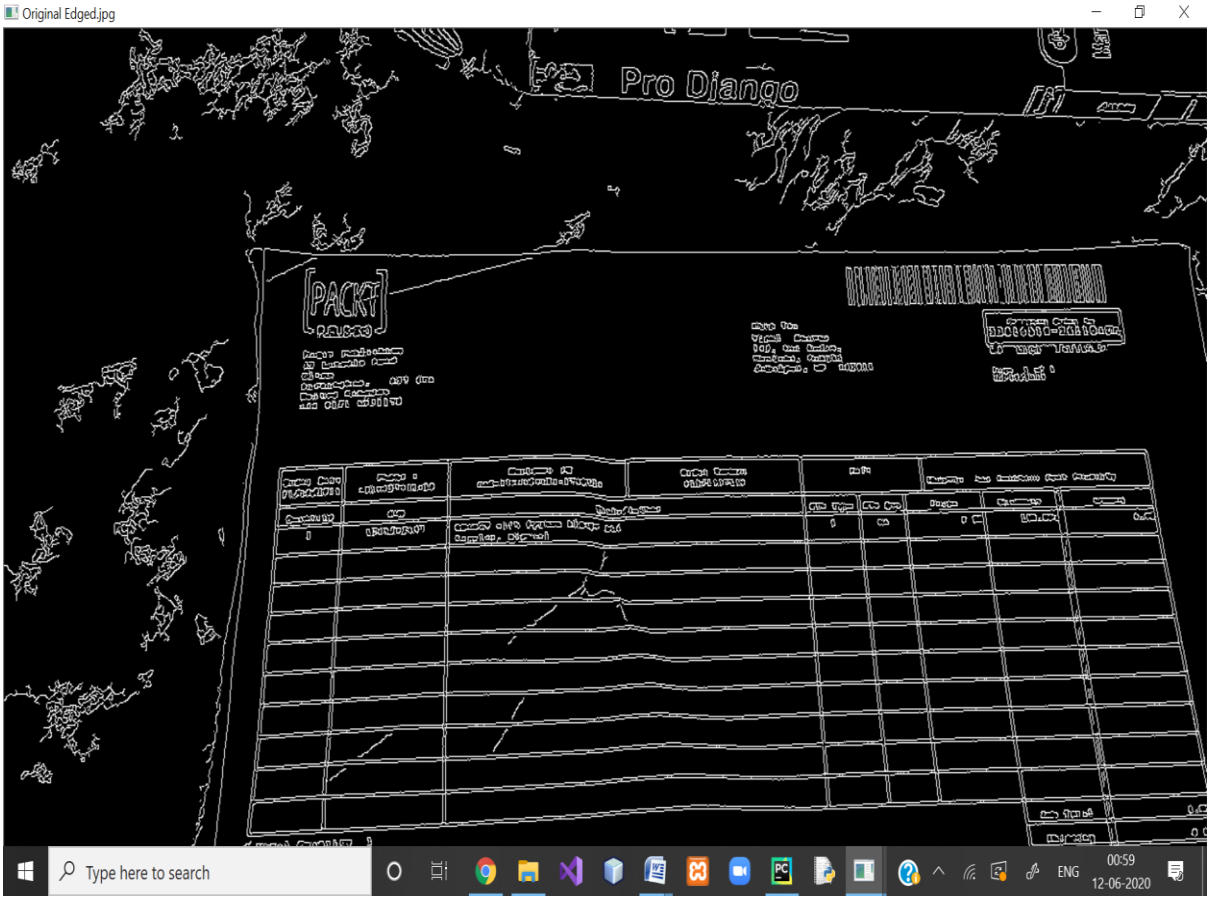
Testing:

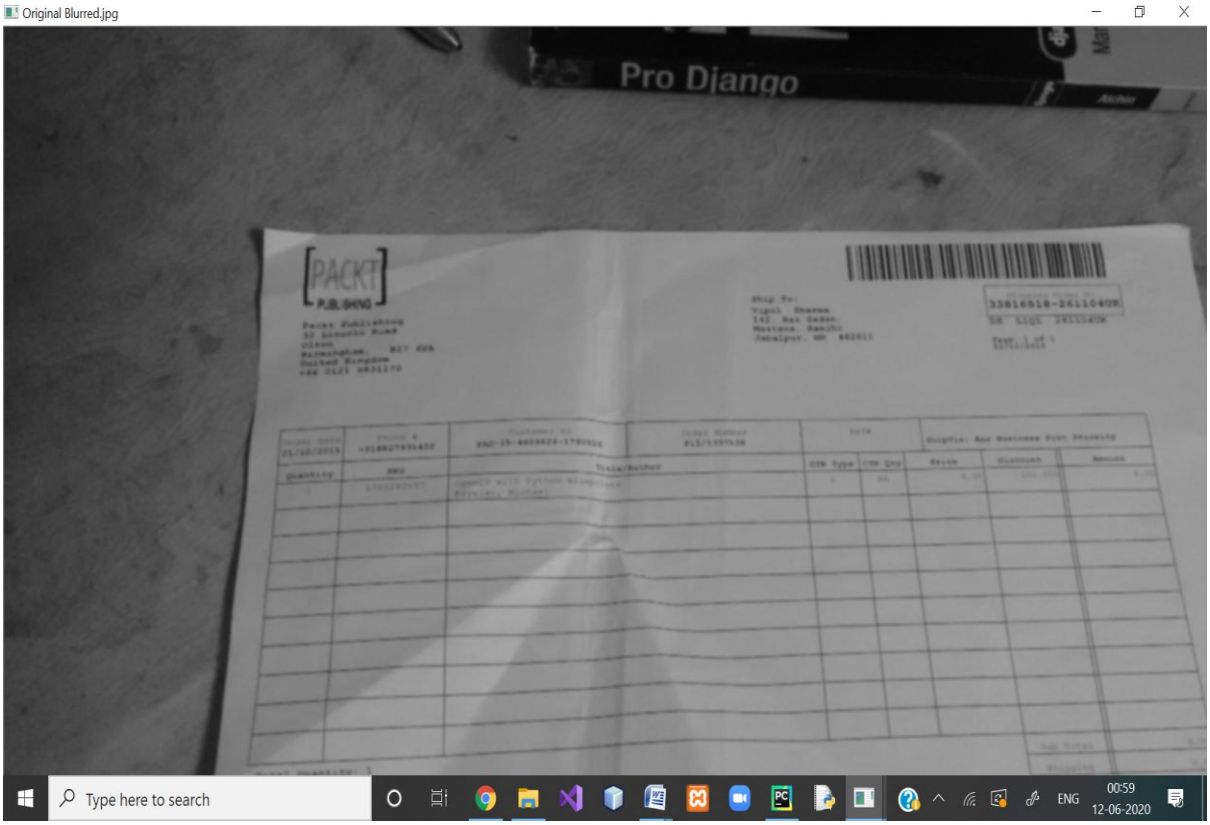


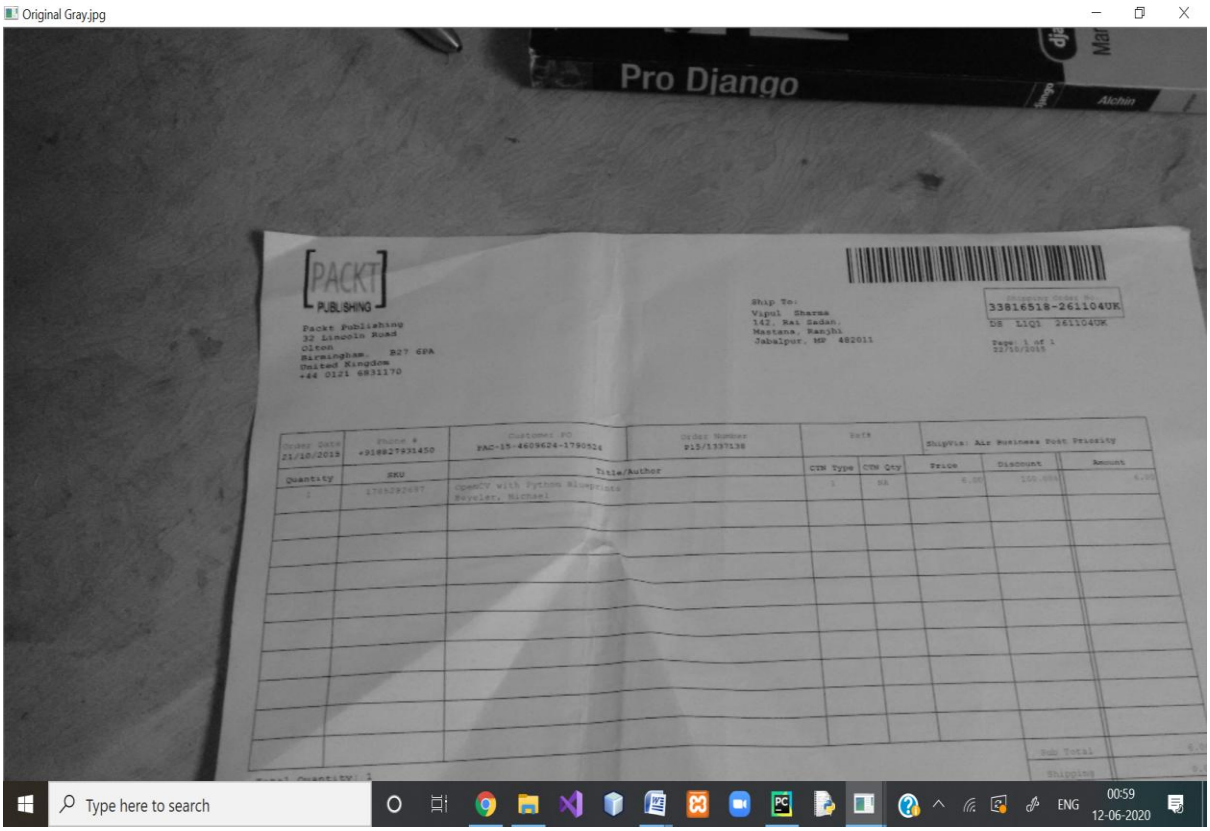
Initial minimized output

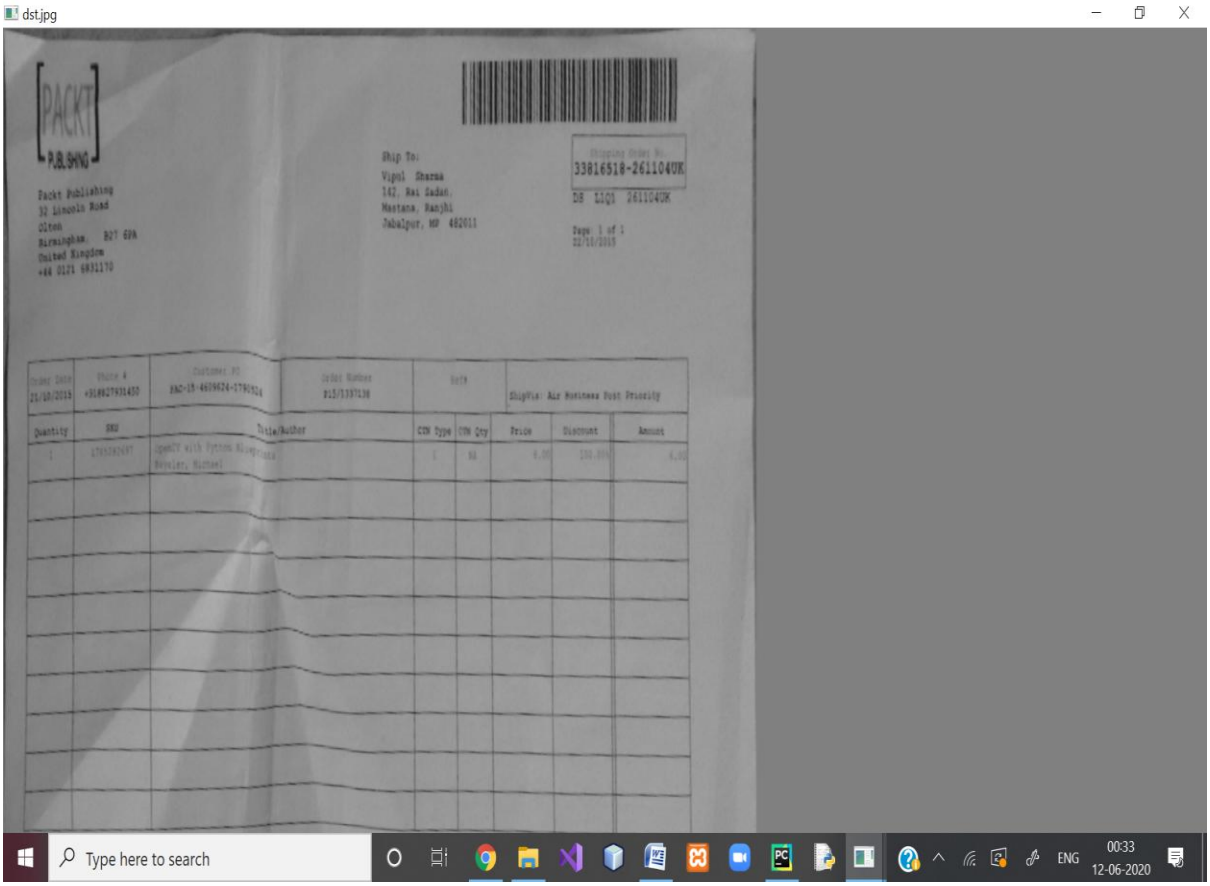


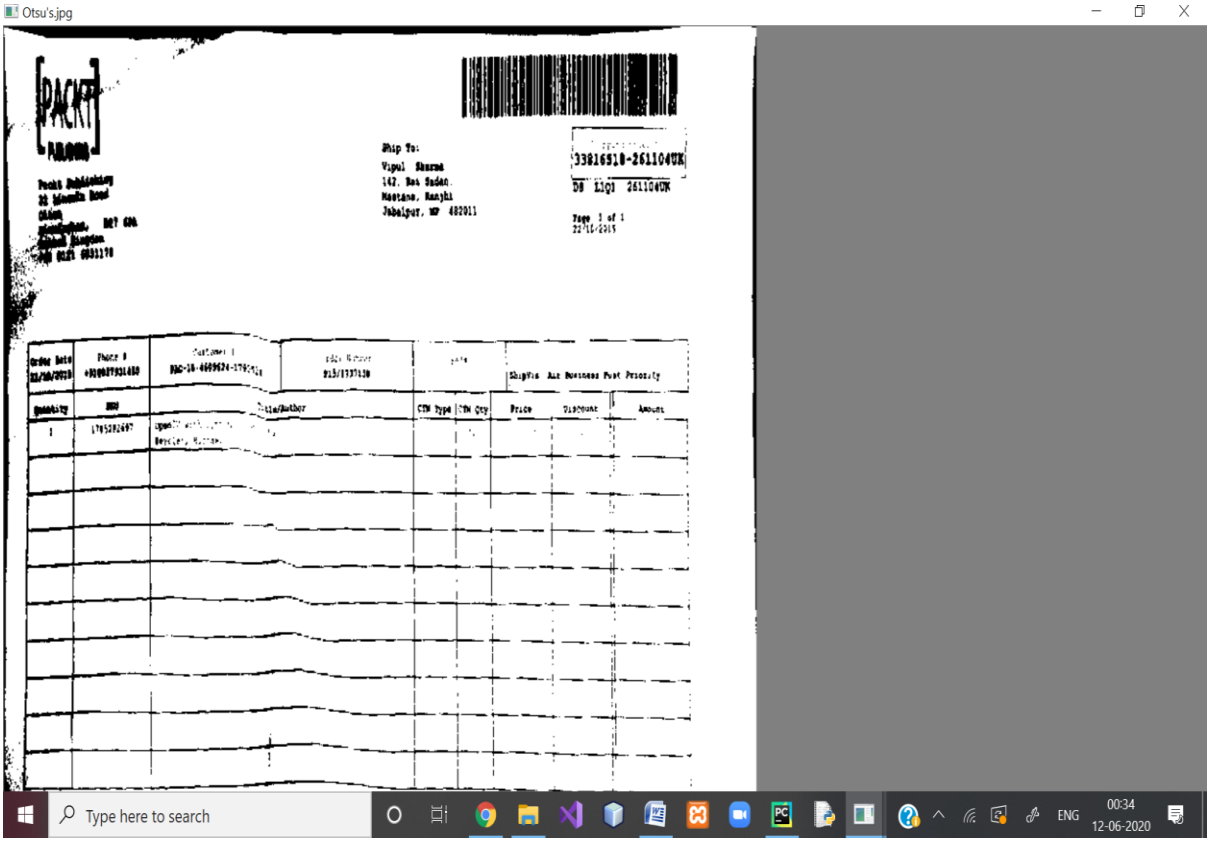

Outline.jpg
Pro Django
PACKT
PUBLISHING
Type here to search
ENG
00:59
12-06-2020

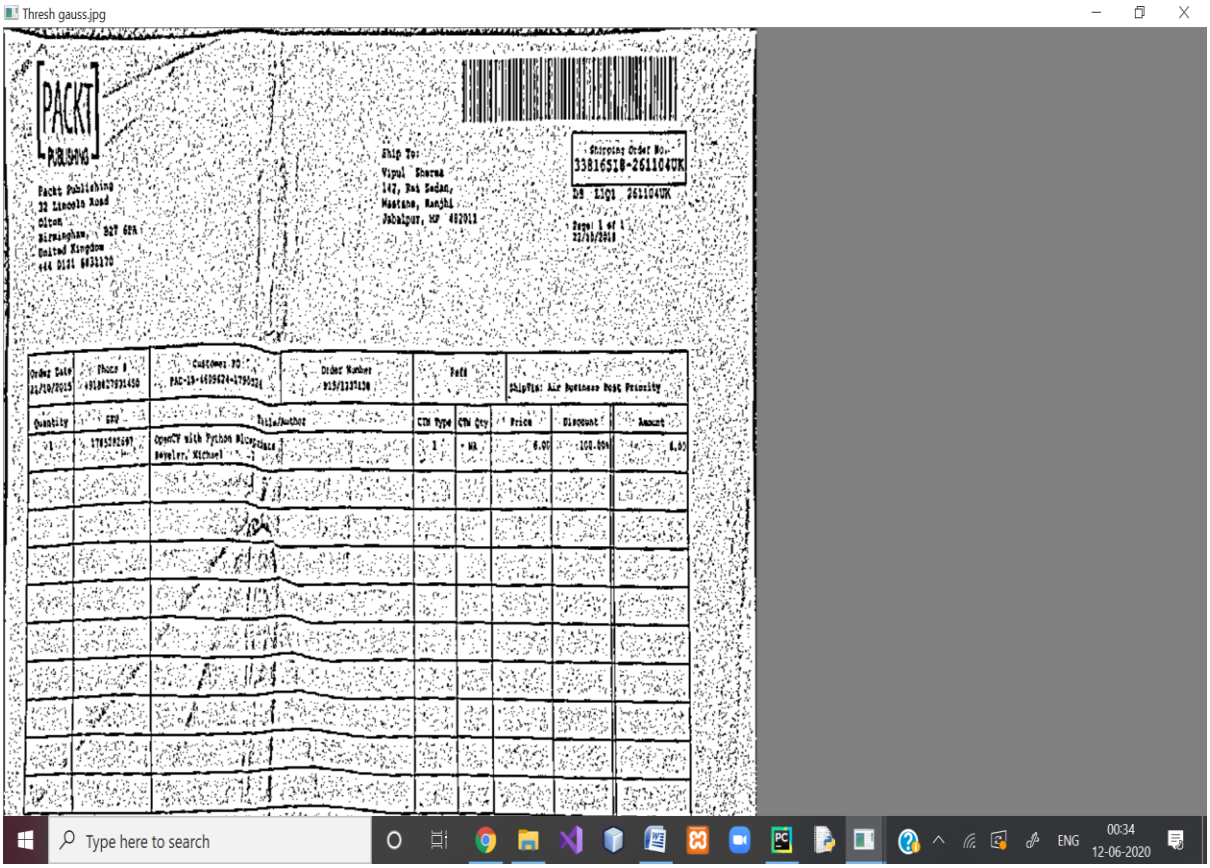

Original Edged.jpg
Pro Django
PACKT
PUBLISHING
Type here to search
ENG
00:59
12-06-2020

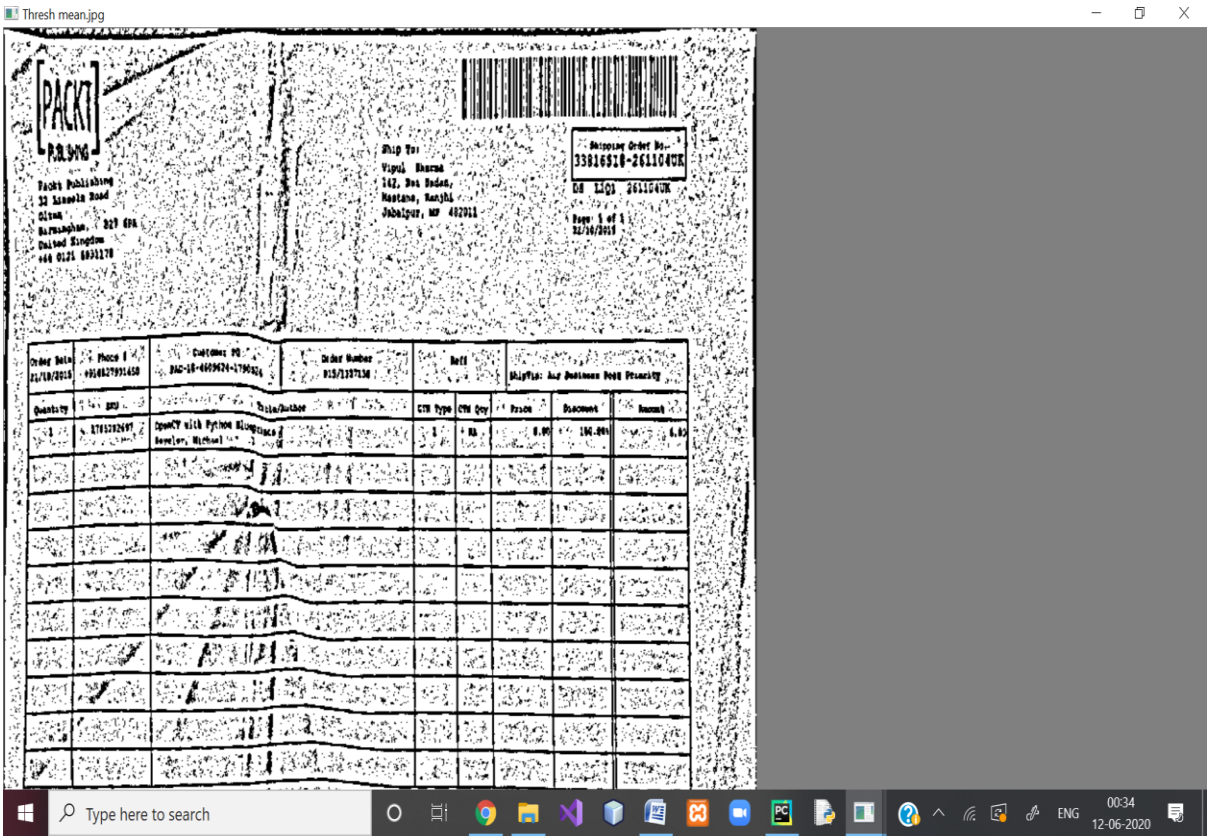

Original Blurred.jpg
Pro Django
PACKT
PUBLISHING
Type here to search
ENG
00:59
12-06-2020


Original Gray.jpg
Pro Django
PACKT
PUBLISHING
Type here to search
ENG
00:59
12-06-2020


dst.jpg
PACKT
PUBLISHING
Ship To:
Vipul Sharma
142, Ras Sadan,
Mastana, Ranjhi
Jabalpur, MP 482011
33816518-261104UK
Type here to search
ENG
00:33
12-06-2020


Otsu's.jpg
PACKT
PUBLISHING
Ship To:
Vipul Sharma
142, Ras Sadan,
Mastana, Ranjhi
Jabalpur, MP 482011
33816518-261104UK
Type here to search
ENG
00:34
12-06-2020


Thresh gauss.jpg
Type here to search
ENG
00:34
12-06-2020


Thresh mean.jpg
Type here to search
ENG
00:34
12-06-2020

Conclusion & Future Enhancements:

We managed to get our document scanned and transformed it to augment its look and feel. This code can be put across a web interface by using flask. Flask is one of the most used frameworks for Python based web application. We can further make our code to run on demand independently by using docker framework. This code is just a base to build further application. For example, this code can be generalized to create a mobile/web out of it.

One important point we would like to bring is that for various functions parameters, we have used those values which worked best in our case. We arrived at these values after trying various combinations. Moreover, if we plan to put this in real action, i.e. an app or document scanner, then we will try to capture document from top instead of from an angle. This way we will get best results.