Objective:

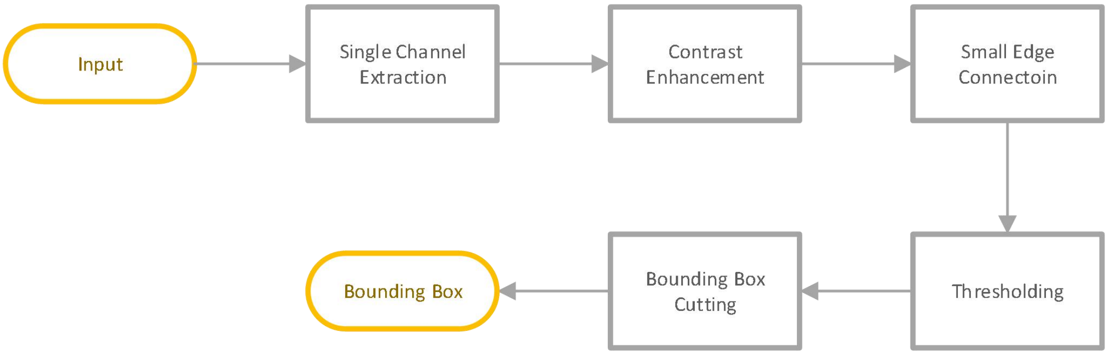
1. License Plate Detection
2. Character Segmentation and Recognition

Method:

1. Prerequisites

1) Python3.6
2) numpy==1.13.1
3) opencv

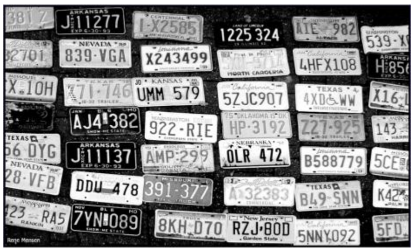
2. License Plate Detection



1) Single Channel Extraction

We extract the value channel in HSV color model to form a single channel image for the following process.

Original Image



Value Channel

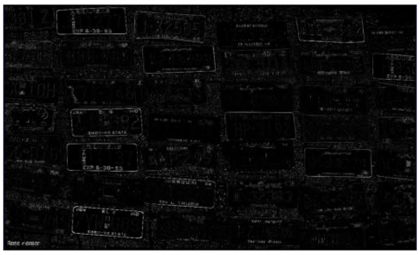
2) Contrast Enhancement

Top-hat transform is an operation that extracts small elements and details from given images. White top-hat returns an image containing some elements that are smaller than structuring element and brighter than their surroundings. Black top-hat returns an image containing some objects that are smaller than the structuring element and darker than their surroundings.

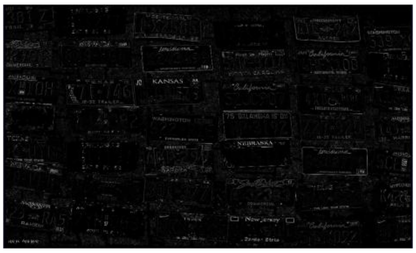
To enhance contrast, enhanced image can be formulated as:

$$I_{\text{enhanced}} = I_{\text{original}} + I_{\text{white top-hat}} - I_{\text{black top-hat}}$$

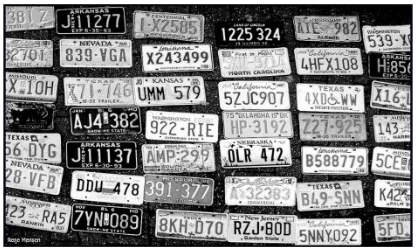
where I_{enhanced}, I_{original} are the enhanced image and input/original image, and $I_{\text{white top-hat}}$, $I_{\text{black top-hat}}$ are the images transformed by white top-hat and black top-hat operation.



$I_{\text{white top-hat}}$



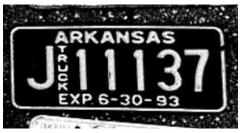
$I_{\text{black top-hat}}$



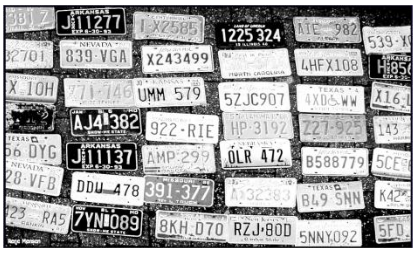
I_{enhanced}

3) Small Edge Connection

After contrast enhancement, we find that edges of some license plates boundaries are not connected. Dilation operation is used for expanding the edges.



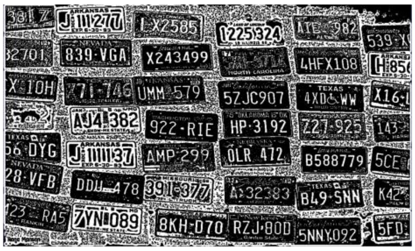
Boundary not connected



Dilation Operation

4) Thresholding

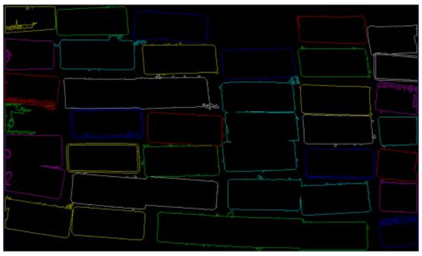
Adaptive thresholding is adapted to convert the single channel image to a binary image.



Thresholding

5) Bounding Box Cutting

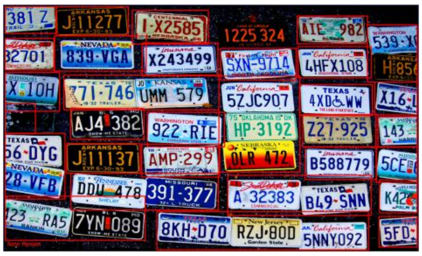
First, we detect contours by cv2.findContours. And the hierarchy of every contour is also obtained. Detecting a license plate can be formulated as finding an inner edge while joint of plates and background can be the outer edge. In order to find the maximum inner boundaries, which is the exact boundaries of license plates, we organize all of the contours into a two-level hierarchy and filter the contours whose area are smaller than a set threshold(2000 is proved the best threshold).



Inner Contours

For each inner contour, we count the largest and smallest value for x-dim and y-dim. End points in the enclosing rectangle are calculated by these four numbers.

Specifically, some license plates are so close that we can’t divide, and enclosed by a bigger rectangle. To solve it, width and height of well divided license plates are counted to crop the bigger enclosing rectangle horizontal or vertical on average.



Bounding Box

However, a wrong detection is conducted in the blurred plate. But we can filter it by the statistical width.

A wrong detection

In the end, we detect 40 license plates in the original image while the total number of valid license plates is 41. The one that we missed is the following one. It is observed that it has almost the same background with the background in the whole image. It is acceptable that we miss it.

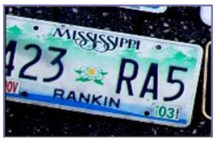
In conclusion, a great result in the license plate detection is conducted.

The Missed One

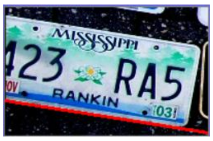
3. Character Segmentation and Recognition

1) Skew Correction

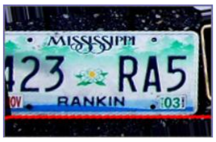
Skew Correction is important because tilted license plates and characters are hard to process. Hough transform is concerned with the identification of lines in the image. We apply it to identify the top edge of license plate. Besides, the angle of this line can help to correct skew.



Original bounding box



Line detected by Hough transform



Skew Correction

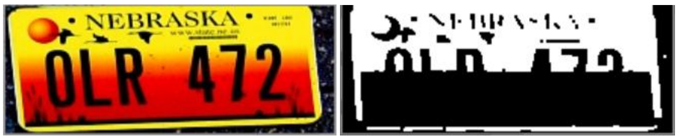
2) Preprocess the Cropped License Plates

The license plate obtained before the previous chapter is only a roughly divided license plate. The useful part of the license plate needs to be extracted to make the font clear.

First convert the BGR image to a gray image, and then use the OTSU method to convert the gray image to a binary image to easily match the template of the binary image.

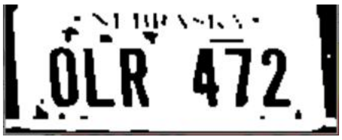
Model character

However, some backgrounds are gradient and dark images are difficult to separate well:

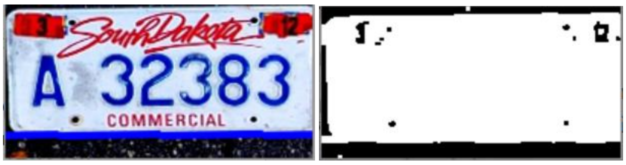


Difficult to separate due to background gradient in image_19

We tried to enhance the contrast, which is helpful for this image, but it is disastrous for other images:



Helpful for image_19



Harmful for others

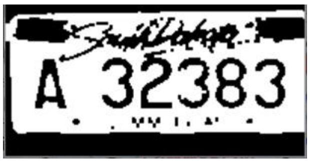
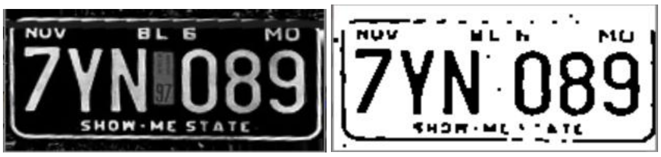


Image without contrast

We don’t considere for enhanced contrast.

Then, the opening operation and the closing operation are performed. The purpose of the opening operation is to remove the unevenness of the character edge because some character edges are uneven, and the closing operation is to fill the unclear part of the font.

Because some license plates in the image are black and different from other license plates, color inversion is required. First of all, let's determine what the percentage of pixel values in the license plate is. In the previous step, the segmented pixels reached two extremes, 0 and 255. So when a black license plate is used, its white pixels are few, less than 30%. Use this condition to find the black license plate and reverse the color, 255-pixel.

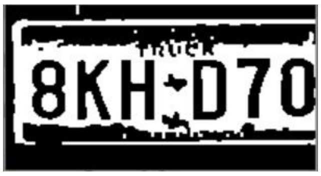


Black car plate

3) License plate search

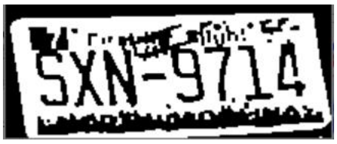
License plate search is divided into finding upper and lower boundaries.

For upper boundaries, we find the maximum gray value of each line, and use the maximum gray value to find and determine the upper and lower boundaries. The difference is not used because the differentiation will cause the boundary to stay, and the license plate boundary is sometimes far from the text. When the previous session finds a row with a certain parameter larger than the next one, it is determined as the upper boundary: row > x * max, max is the maximum row pixel value currently found, and x is set to 0.93, because sometimes the upper boundary is empty and it is pure white, so you must find a Bottom and closest to the border of the character.



There is a long white border on the top of the picture, and the pixels are very large.

However, there are some images that are difficult to separate by this method:

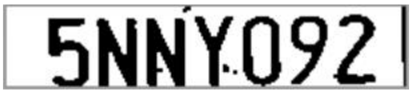


Characters connected with background

Some characters are too low:

Characters are low

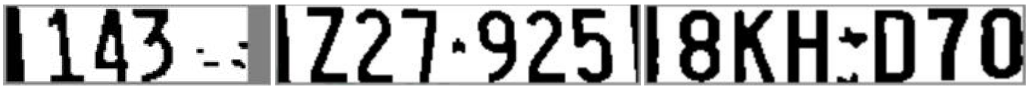
So the range of the upper boundary is found to be 0.55 of the entire image, and the separation effect is not bad:：



Cut well

The lower boundary is a similar method. When the boundary changes suddenly, the boundary value will be found. This method is similar to the derivative to observe the degree of change in gray value. Sometimes the background color and noise will interfere with the judgment, and at the same time the font itself will interfere. Some letters have very wide lower borders, for example: "Z", "2", "D", "L", etc. Some letters have very narrow lower borders. , So set a parameter to balance these.

The processed license plate was obtained by the above method:

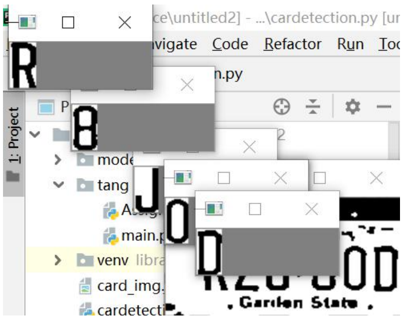


Some car plate images

4) Character Segmentation

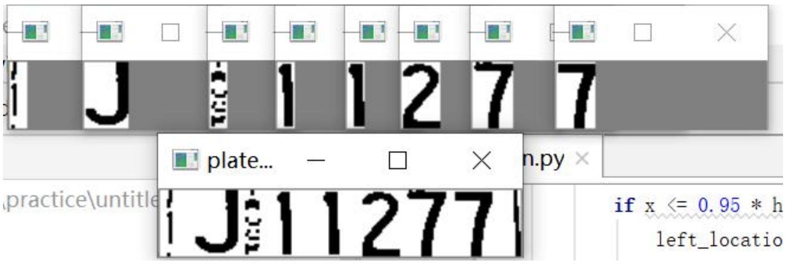
The method of character segmentation is similar to the upper and lower borders. You need to calculate the gray values of the columns and determine whether they are left and right borders. The difference is that he needs to find some characters continuously.

By judging the gray value of the column, if the gray level on the left reaches x * his_max, then it is determined to be the left boundary, because there is noise and the background is not perfect, so his_max is row * 255, and x is 0.95, 0.95 You can find narrow character boundaries such as "T", while also limiting noise and background to a limited extent.


ce\untitled2] - ...\cardetection.py [ur
vigate Code Refactor Run Too
n.py
1: Project
mode
tang
main.
venv libra
card_img.
cardetect
Garden State

Result of cutting

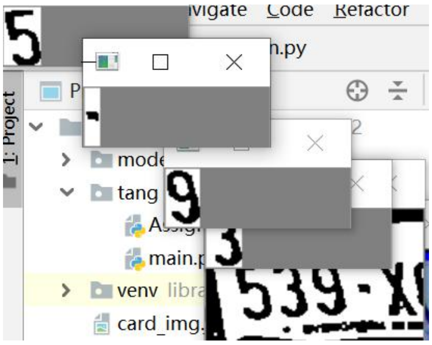
In the image, the background pattern in the middle of the font is often cut into it. Initially, it was expected to limit the size of the background pattern. However, when the font size is limited, some characters such as "I", "1", etc. It is also very small and will be excluded. In order to avoid excluding license plates, we have retained the background pattern.



```
if x <= 0.95 * h
    left_locatio
```

Parts with background

In the original image, some pictures are oblique. In order to obtain a horizontal image, we obtain a horizontal image through Hough transform. Although the license plate reaches the horizontal position, sometimes the fonts of some image fonts obtained are often slanted, which will affect the font segmentation.



Not cut “X”

5) Recognitions

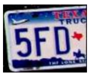
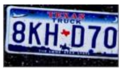
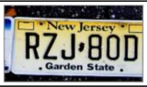
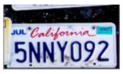
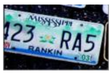
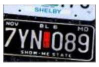
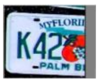
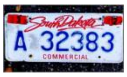
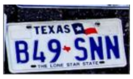
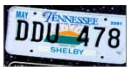
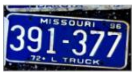
The template picture is compared with the segmented characters, and the matching formula using the normalized correlation matching method is as follows:

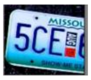
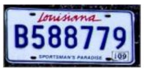
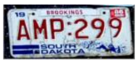
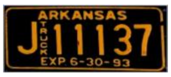
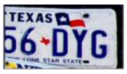
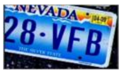
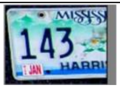
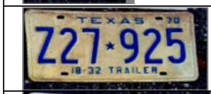
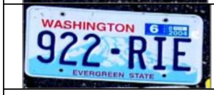
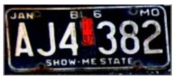
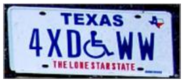
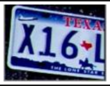
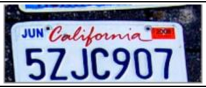
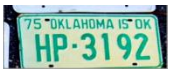
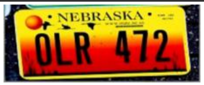
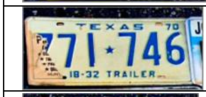
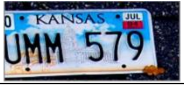
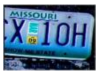
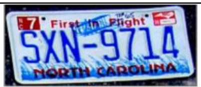
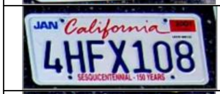
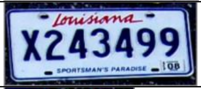
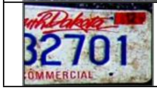
$$R(x,y)=\frac{\sum_{x',y'}(T'(x',y')\cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'}T'(x',y')^2\cdot\sum_{x',y'}I'(x+x',y+y')^2}}$$

Select the output with the best matching result to determine:

Result:

The recognition effect is as follows: there are some similar characters, such as "0" "O", "3" "8", "0" "D", "I" "1", "2" "Z", etc., they are not easy distinguish. Therefore, the effect is not good during recognition. The other edges of the characters are difficult to recognize. The background is dark and the gradient is not distinguished, so the effect is not good. The final recognition success rate is about 60.67%.

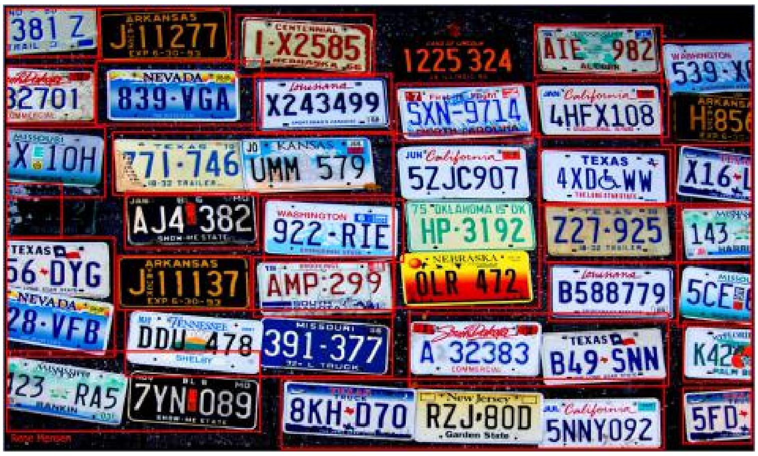
Cropped license plate	Recognized characters(*, / represent noise and characters not cropped)	Correct Rate(%)
	5 * F D *	100
	8 K H * D 7 /	83
	R Z J * 8 0 D	100
	5 N N 7 G 9 Z	57.14
	/ 2 3 * * R A 5	83.3
	7 V N 0 8 8	66.67
	K 4 /	66.67
	A 8 8 8 8 8	33.33
	* 8 A 9 * 5 A W	16.67
	8 D U / 7 8	66.67
	9 9 * / / 7	33.33

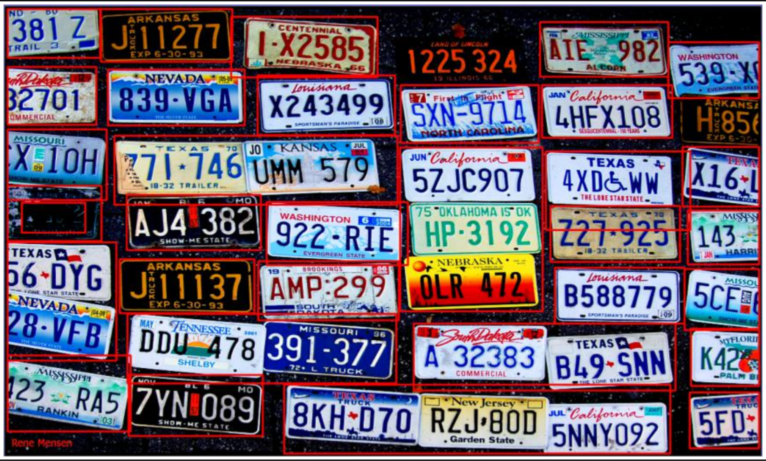
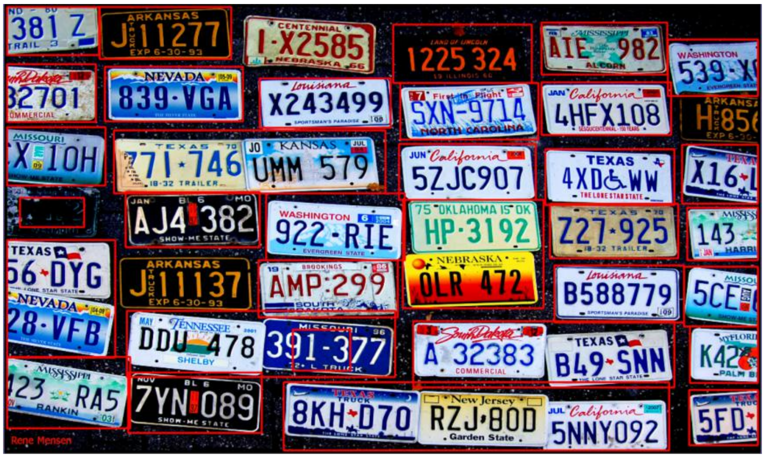
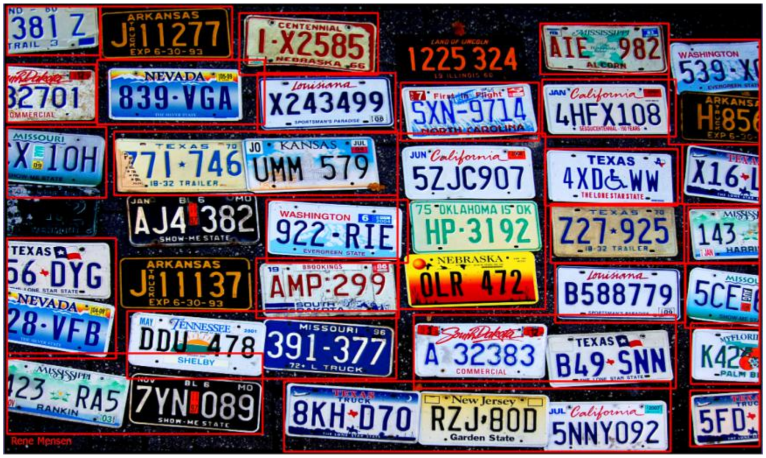
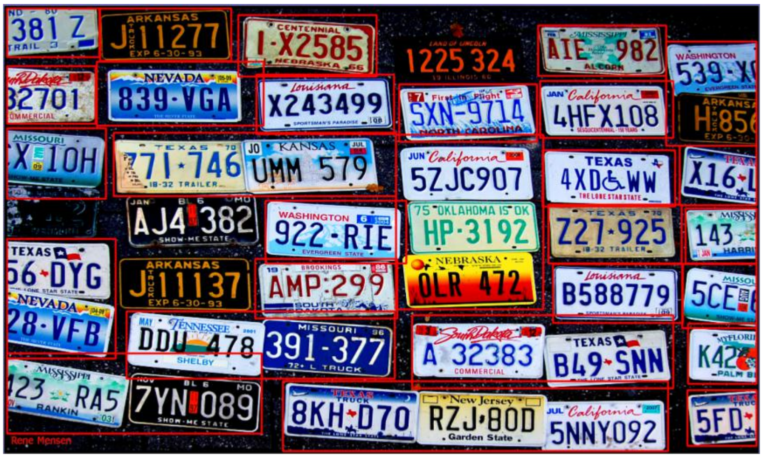
	5CE*	100
	8588779	85.71
	AN9*289	50
	*J*44I87	33.33
	/6*DYC	60
	/////	0
	143*	100
	Z20*8Z5	50
	8ZZ*RIE	50
	*AZ488Z	50
	4X8*WW	80
	XI6*	100
	5ZWC909	71.43
	*8P*849Z	33.33
	//////	0
	A7I*A46	80
	/MM579	83.33
	X*30H	75
	MX///I	16.67
	AW9X109	42.86
	XZ48499	71.43
	2784	50

	/ / / / / /	0
	5 3 8 * /	75
	* H * 8 5	100
	A I Z 9 8 2	83.33
	I * X 2 5 8 5 *	100
	* Z * 4 4 Z 7 7	28.57
	G 8 4 Z	50
Average		60.67

Discussion:

1. License Plate Detection

	Bounding boxes on the image	Detected plates number
Our Method		40

w/o contrast enhancement		39
w/o small edge connection		39
Threshold(4000) in bounding box cutting		38
Threshold(5000) in bounding box cutting		38