Car License Plate Recognition with different image filtering techniques

Young-Shiuan Hsu
ENEE631 Image Processing Project
ENEE University of Maryland College Park, Prince George County, USA
yhsu1235@umd.edu

***Abstract*—The car license plate recognition system is proposed to get a better glimpse of how filter affects an image and the different methods we could use to improve the quality of the image and different techniques on how to implement noisy image. We insert the different processed images into the car license plate recognition neural network and analyze the performance of each image inserted.**

Keywords—image processing, edge detection, car license detection, DNN, noise reduction for image, convolution, FFT

I. Introduction

In the real world, it is impossible for people to obtain a "perfect" image that could be directly used for one's purpose due to channel noise that causes distortion. Therefore, image processing has become a big field when we are interested in different parts of the image and through these years people have developed various filtering techniques and image processing algorithms in different fields to help us analyze images.

This project will focus on how different filters affect an image and how the effected image affects the neural network to determine the car license. We will first add white noise and peppered noise to a certain image and pass it through the system. Then by applying different noise reduction techniques we can obtain better performance for the car plate license recognition system. Car plate license recognition for an image, we have to first implement edge detection to find the edges of the whole picture. Then with contour detection, we can extract the car license image from the given image. Last, we have to do letter and digit recognition of the extracted car plate image.

II. Filtering Techniques

There are various filtering techniques used in this system. We will be implementing low-pass filter, high pass filter(Laplacian filter), median filter, bilateral filter, Sobel filter, and Robert filter.

A. *Convolution*

The formula of convolving two images X and Y

$$\mathrm{x[m,n]} * \mathrm{y[m,n]} = \sum_{k=0}^{N-1}\sum_{l=0}^{M-1} x[k,l]\, y[m-k, n-l]$$

B. *Low pass filter*

The kernel is a general gaussian low pass filter:

$$\mathrm{kernel} = \frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

and by convolving the kernel with the original image we can see it blurs the image. The image on the left is the original image and the image of the right is the image after convolving with the gaussian kernel.

Figure 1

C. *High pass filter*

The kernel for high pass filter:

$$\mathrm{kernel} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

If we convolve the original image with the kernel above it sharpens the image. The image on the left is the original image and the image of the right is the image after convolving with the Laplacian kernel.

Figure 2

D. *Edge detection*

There are different edge detection methods. The well-know methods are Canny edge detector, Sobel filter and Robert filter.

The original image before convolving with the kernel:

Figure 3

The Sobel filter is to convolve the original picture two times with two different kernels that picks up the horizontal and parallel edges in the image.

$$\mathrm{kernel_x} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\mathrm{kernel_y} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

By convolving the two kernels, this is the image we get:

Figure 4

The Robert filter has the same operation with Sobel filter but it has a different kernel:

$$\text{kernel}_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\text{kernel}_y = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

By convolving the two kernels, this is the image we get:

Figure 5

However, Canny edge detector has a more complicated operation than Sobel kernel and Robert kernel and it is widely used in computer vision. Canny edge detector algorithm is to first apply a gaussian filter to smooth the image in order to remove white noise.

$$\text{kernel} = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Then by finding the gradient of the image, we set a threshold to get rid of the spurious response to prevent pixels with weak gradient value which might cause noise. Last apply the edge tracking by hysteresis. This is the image we get after implementing the algorithm.

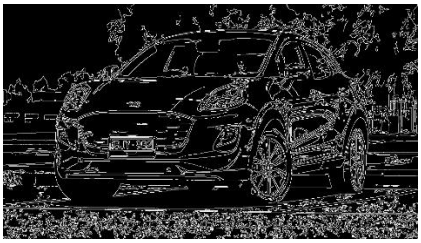


Figure 6

E. Noise reduction filter

The commonly used filter to fight white noise is a low pass filter mentioned above. The bilateral filter is another filtering technique by smoothing the image in order to reduce noise while preserving the edges.

Another commonly used filter to deal with pepper and salt noise image is the median filter. We convolve the median filter with the noisy image and the difference will be shown in the next section.

$$\text{kernel} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

III. Result of Car License Recognition

After constructing different kinds of filtering techniques, we now generate our noisy image. The image on the left is when we add white noise with mean equal to 0 and variance equal to 10.

After adding white noise, we pass it through a low pass filter in order to do noise reduction. Figure 7 is the image added with gaussian noise and Figure 8 is the image after convolving with the low pass filter. Obviously, after convolving with the low pass filter, we could see less white dots in the front of the car which is considered as a better image quality.

Figure 7

Figure 8

We also generate a peppered and salt image and apply median filter to fight the noisy image. From the sky in the image(top left region), we can clearly see the dots. On the other hand, after convolving with the median filter, the sky looks clearer which is consider as a better image quality.

Figure 9

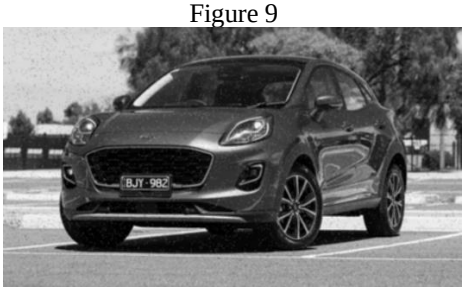


Figure 10

To accurately extract the license plate in the image, we have to apply edge detection to help the system to detect the contours in the image. By applying Sobel filter and Robert

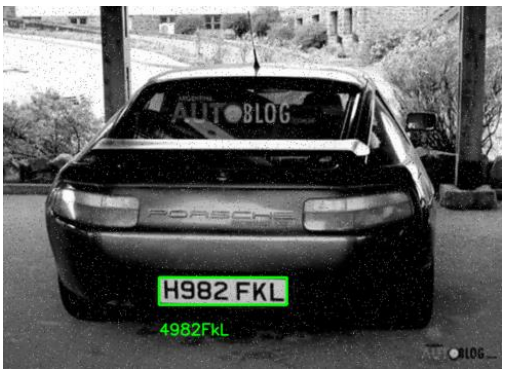
filter, they pick up noise without adjusting it which makes them very sensitive to noise. Therefore, after experimenting different edge detection techniques, canny edge detector algorithms gives the best result so the experiment will all apply canny edge detector.

The result of implementing edge detection for gaussian noise image and the image with noise reduction, we can clearly see the image contains a lot of noise than the one implemented noise reduction. We could barely determine the position of the car license plate in the noisy image. After contour detection, it would only be able to detect part of the license place and extract it from the original image.

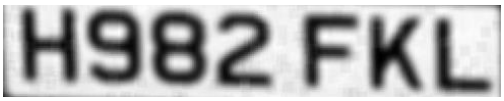
The car plate is recognized as “bJ” with gaussian noise image.

Now we insert the noise reduction image into the system. Since the image is blurred with the low pass filter applied earlier, we sharpen the image through the Laplacian filter. After applying edge detection the system is able to pick up the edges or the car license and extract it from the image. The recognized plate value is “EBJY 982)”, which is quite close to the real value of the car license plate “BJY 982”.

By doing the same thing to pepper and salted noise image, we could expect a similar result. The result of edge detection on the peppered noise contains great amount of noise which makes contour detection very hard. If we successfully detected the license place, we can see the extracted image contains small dots everywhere that greatly affect the result of the recognized value. The detected value of the image is “4982FkL”.



On the other hand, if we implement the median filter, the result is surprisingly good. Obviously, after edge detection, the image is able to successfully pick up the license plate with a very high resolution and the recognition of the plate value is “H982 KFL”, which is exactly the value of the license plate.



IV. Conclusion

By experimenting with all these different filters we are able to see how different filters affect the original picture and try to add different combinations to enhance the image quality or to present different effects of an image. By inserting different images into the car plate license recognition model, we are able to see how we can use different filters to fight different types of noise and the recognition result shows the result if we have different filters involved.

From the experiment and output of the image, we can see low pass filter blurs the image and the high pass filter sharpens the image. There are three different edge detection methods while Sobel filter and Robert filter are relatively sensitive to noise comparing to Canny edge detector algorithm, which is why we decided to use Canny edge detector algorithm to help us detect contours in the image.

While we insert a gaussian noise image into the network, it is extremely hard to determine the contours in the noisy image so that we are only able to pick up part of the car license which does not matches our expectation. With poor quality of images, we have to implement low pass filter to fight with white noise. After convolving with the gaussian low pass filter, we were able to detect the contours of the license plate so that we could extract the desired license plate from the original image and recognize it. Although the results might now be very precise but it is much better than the result from the gaussian noisy image. Me repeat the same operation with the peppered and salted image. The system is able to detect the contours but it could now recognize the digits on the license plate very well. However, if we pass it through a median filter the system would be able to recognize all the values which is why median filter are considered to be able to efficiently fight peppered and salted noise.

References

[1] https://www.geeksforgeeks.org/python-bilateral-filtering/

[2] https://en.wikipedia.org/wiki/Canny_edge_detector

[3] https://www.researchgate.net/publication/346583845_Image_Filtering_Techniques_Algorithm_and_Applications#:~:text=Image%20filtering%20is%20a%20technique,graphic%20design%20and%20editing%20software.

[4] https://en.wikipedia.org/wiki/Kernel_(image_processing)

[5] https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9

[6] https://wandb.ai/authors/kaggle_license/reports/Kaggle-License-Plate-Detection--VmlldzozMDMwNTU

[7] https://github.com/JaidedAI/EasyOCR

[8] https://www.irya.unam.mx/computo/sites/manuales/IDL/Content/GuideMe/ImageProcessing/LaplacianFilters.html#:~:text=A%20Laplacian%20filter%20is%20an,an%20edge%20or%20continuous%20progression.