Sri Lankan Vehicle Plate Recognition System

CO543 - Image Processing

Mini-Project

Group A

Balasuriya I. S.
E/17/018
Dept. Computer Engineering
Faculty of Engineering
University of Peradeniya

Francis F.B.A.H.
E/17/090
Dept. Computer Engineering
Faculty of Engineering
University of Peradeniya

Karunanayake A.I.
E/17/154
Dept. Computer Engineering
Faculty of Engineering
University of Peradeniya

Abstract

The vehicle number plate is the unique identifier for every vehicle in Sri Lanka. Number plate identification is very important for various purposes such as traffic statistics, identification of traffic law violations, identification of criminals' vehicles, identification of fake number plates, etc. Automating such a system is a really challenging task due to different reasons. In the image acquisition step, a lot of undesired noise, lighting issues, and blurring effects are introduced to the image due to these images being acquired from CCTV cameras under non-ideal conditions. The motion of the vehicle with respect to the camera also introduces some level of motion blurring to the image. Hence, in order to identify the license plate, a large amount of preprocessing needs to be done to mitigate these effects so that a clear image is obtained. Our approach utilizes a series of operations such as upscaling, noise removal, and deblurring to restore the image to a legible representation and performs optical character recognition (OCR) on the restored image to identify the license plate. The upscaling step attempts to overcome the low resolution of the captured image by using a deep learning model. The noise removal step utilizes a simple Median filter to get rid of high-frequency noise in the captured image, especially under low light conditions. The deblurring step models the Point Spread Function (PSF) of the motion blur in the image by estimating the blur length and angle, and using the PSF to deblur the image. Coupled with some additional basic image processing techniques such as contrast stretching, this approach produces a reasonably good image to be used for OCR.

Introduction

The focus of the project is to create an automated number plate identification system. The inputs considered here are the images captured from CCTV cameras. CCTV cameras are mainly focused because of their availability and coverage of the current environment. Almost every urban area has these cameras for surveillance and has immensely affected identifying criminal activity, accidents, and other unexpected scenarios. Although they have been helpful for identifying people and cars nearby the most prominent issue has been the low-resolution images captured. As a result, using them for identifying vehicle number plates has been proven to be rather difficult. This is also due to the conditions in which the images were captured being mostly undesirable as they introduce an added complexity to the already low-resolution images with noise and defects. The main objective of our project is to recover a readable license plate number from the images even when they are proven to be in an undesirable state.

The task at hand is undertaken with a step-by-step approach wherein each step we use various techniques to have a final result outputting the license plate number. The set of images used is mostly non-human-readable and, as such, certain image processing techniques are used to enhance features and enhance their characteristics. Once they are of satisfactory quality to be observed by an OCR we pass them through to recognize the characters. The whole process consists of the following steps,

1. As the images are of substantially low resolution the first step entails enhancing the resolution of the images. This is done by super-resolution imaging using OpenCV and existing deep learning models.
2. The resulting image contains defects of various kinds and the focus here is to identify them correctly to achieve a better result in image enhancement for feature recognition. This involves mainly delving into the histogram plots, Fourier domain, and log transforms to visualize defects of noise such as motion blur, blur, low contrast, etc.
3. The identified defects are dealt with using techniques of deblurring, sharpening, contrast stretching, etc.
4. Once a clearer image of the number plate is received optical character recognition is done to identify the license plate.

Background/Related Work

The identification of vehicle number plates is an important operation that can save a lot of time if automated. Hence, there has been extensive research conducted in this regard. However, a lot of the existing algorithms and techniques rely on having a high-quality image of the license plate. These techniques are generally ineffective at accurately recognizing the number plate using images of low resolution such as the ones obtained from CCTV cameras.

During the development of our solution, we have referred to the works of Ajanthan T. et al who have developed an algorithm for extracting license plate data from low-quality videos [5]. The algorithm utilizes deep learning-based SVM classification in order to identify characters in the license plate. Since such advanced techniques are beyond the scope of this project, we have opted for classical techniques that attempt to make the characters themselves legible enough to be accurately identified by an OCR engine such as Tesseract.

Our solution relies heavily on the pre-processing techniques used to improve the clarity of the image. The most challenging artifact to remove was the motion blur present in captured images. Several solutions were explored and we have settled on the techniques outlined in the algorithm developed by Kishore R. Bhagat and Puran Gour for estimating motion blur parameters [4]. This algorithm was used in conjunction with the super-resolution upscaling technique proposed by Lim et al [1] to form the basis of our pre-processing solution.

Approach

Super-resolution

The low-resolution images have been proven to be difficult to process as the features are seemingly too unrecognizable. In an effort to get a better resolution and features of the images, super-resolution is used. After attempting basic methods of noise removal, it was observed that a significant difference was not perceived doing so. As a result, the focus was moved towards existing deep learning techniques on superresolution with OpenCV. Hence three models are tested for the task and the Super-resolution enhanced deep super-resolution network (EDSR) method proposed by Lim et al is used for our approach.

For the purpose of superresolution a ResNet style architecture is used while removing the Batch Normalization layers. The authors found out that the normalization layers reduce the range of flexibility from features' networks in improving performance. As a counter for the instability that would occur in larger models a residual scaling factor of 0.1 is used in each residual block. Initially, the model employed a scaling factor of 2. Then the pre-trained weights are used for scaling 3 and 4. The used model in our approach is the 4x super-super resolution model as a result of the EDSR method.

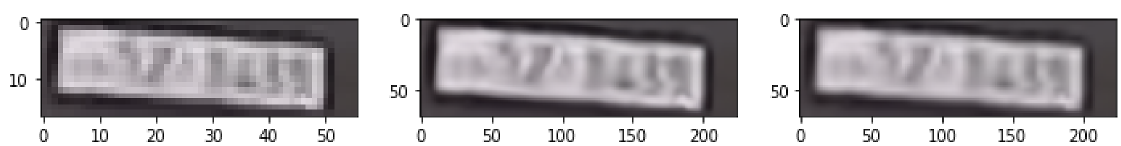


Figure 1: Left - original image , Middle - EDSR_x4 upscaled image, Right - Image upscaled using OpenCV's resize function

The operation removes the pixelated nature of the images and leaves the features as it is which makes it desirable in feature extraction for the next steps of our approach.

Identifying noise

As a preliminary step to addressing the defects of the images, it is necessary to identify which type of noise and other characteristics are affecting the images. To have a quantitative idea about them we have used methods of Histogram analysis and Fourier transform analysis. After identifying the noises and the blur type they are mathematically modeled, The original images considered are the ones assigned to the group (Group A).

Figure 2: Original Images

Fourier Domain Observations

As an initial step for Fourier domain analysis the direct Fourier representation of the images is taken. The representation of the magnitude calculated from the complex result shows a centered DC value.

Figure 3: Fourier Domain Representation of Images 1,2,3

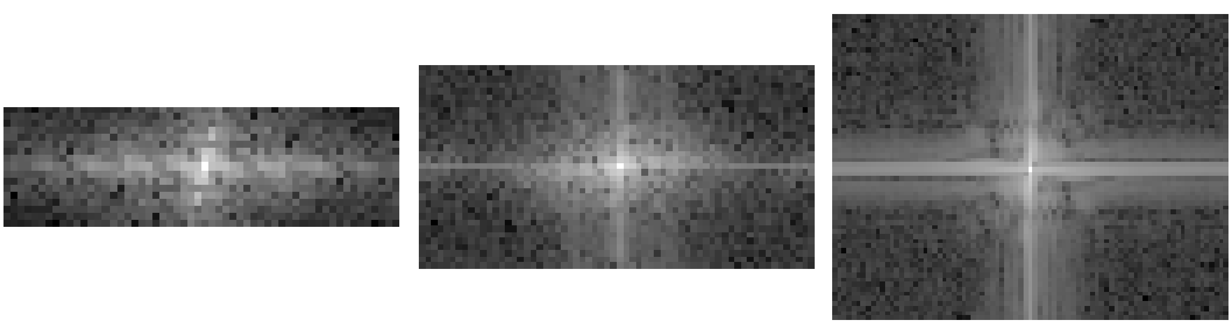


Figure 4: Log-enhanced Fourier Domain Representation of Images 1,2,3

It can be observed that there are some high-frequency components in the images therefore it is required to apply a low pass filter to remove these components. Some images contain motion blur. Therefore, additional processing is required to remove the motion blur as well.

Histogram analysis

The following figure shows the histogram of the image,

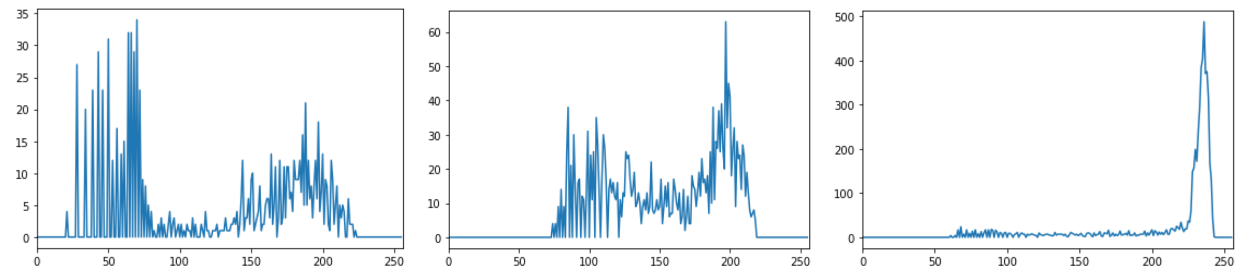


Figure 5: Histograms of the considered images A1, A2, A3

Observing the histogram of the images, it is difficult to identify any noise, but we can observe images with low contrast. This can be rectified by the application of contrast stretching.

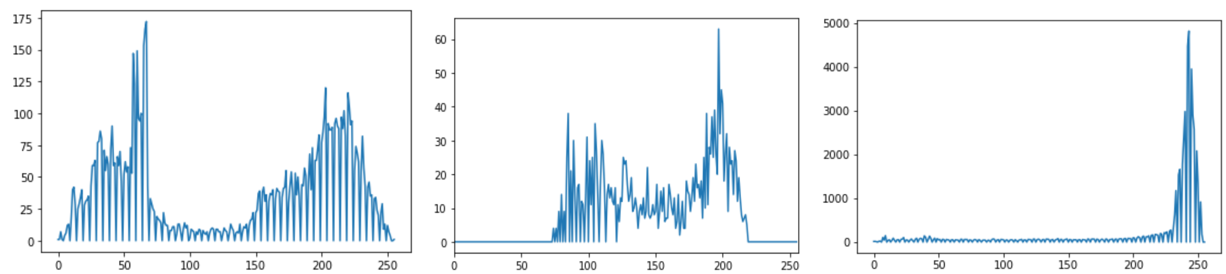


Figure 6: Contrast Stretched Histograms of the considered images A1, A2, A3

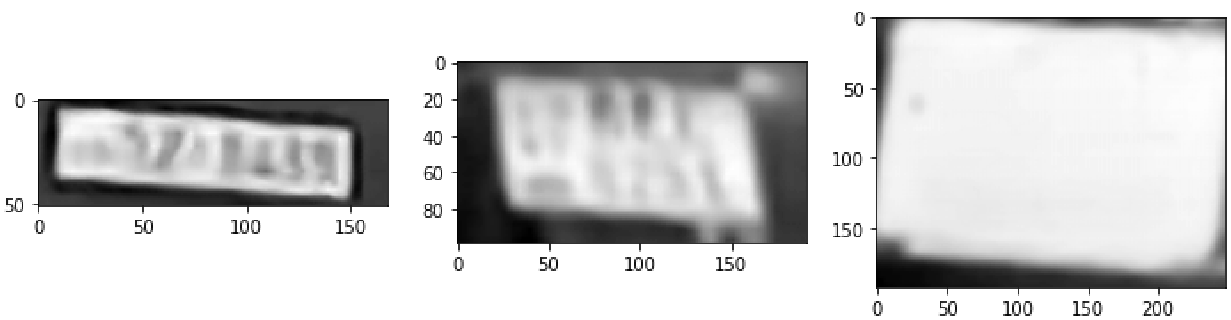


Figure 7: Contrast Stretched images of the considered images A1, A2, A3

Image Acquisition Model

After the analysis of the images, it was determined that the following image acquisition model gives a reasonable estimate of the captured image's characteristics.

$s(x, y) = t(x, y) * u(x, y) + n(x, y)$

where,

t(x,y) is original image

s(x,y) is observed image

u(x,y) is the point spread function

n(x,y) is the additive noise

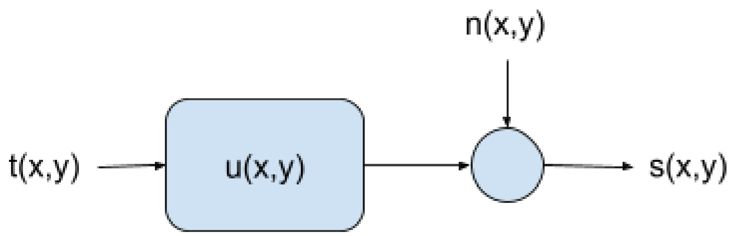


Figure 8: Image Acquisition Model

Hence, the approach to reconstruction is the removal of the noise through low-pass filtering followed by deblurring to obtain a clear image.

Noise Reduction

The median filter is used to reduce the noise caused by low light conditions in the images and preserves the image features as well.

$g(x, y) = s(x, y) - n(x, y) = t(x, y) * u(x, y)$

Where g(x,y) is the noiseless image.

After reducing noises in the image it is passed to the deblurring step,

Deblurring

In our case the main and common blur effect in the images is the motion blur. The following figure shows the propose stages to remove the motion blur,

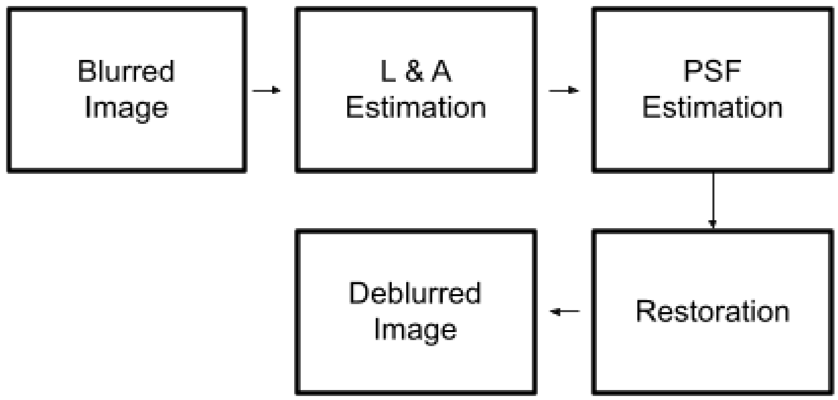


Figure 9: Deblurring Algorithm

Blur angle(A) and the Blur length(L) are the main two variables of the Point Spread Function PSF, PSF which introduces the blur to the image. If we can find the Blur Length and the Blur Angle, we can easily estimate the PSF of the image. That PSF can be used to deblur the image.

Estimation of Blur Angle

The motion blur kernel low passes the image in the direction of the blur. The anisotropy of the frequency spectrum is always perpendicular to the motion direction. Therefore, Hough transform is used to detect the orientation of the line in the spectrum.

Steps followed for finding the blur angle,

- Determine the Fourier spectrum FT(k,l) of the image s(x,y)
- Calculate the log spectrum of the S(k,l) of the F(k,l)
- Apply the Hough transform on the S(k,l)
- Derive the accumulator array a(L,A)
- Take the maximum repeating angle in the a(L,A), which is the blur angle

Estimation of Blur Length

Steps followed for finding blur length

- Determine the Fourier spectrum FT(k,l) of the image s(x,y)
- Calculate the log spectrum of the S(k,l) of the F(k,l)
- Convert the S(k,l) into binary by thresholding
- Binary structure is rotated in direction opposite to the blur angle.
- Convert the 2D data into 1D by taking the average along the columns
- Take the inverse fourier transform of 1D array
- The real part of the first negative value is found as blur length (L)

Character Extraction

The algorithm attempts to localize the license plate itself from the image as an initial step. This is done using Canny edge detection after which the algorithm attempts to find the largest rectangular contour in the image, which would most likely be the license plate. This can be done by generating a list of all the contours and sorting them by their size. Then, the largest contour with four points in the rough shape of a rectangle can be identified as the desired contour by iterating over the sorted list. This contour will provide the bounds of the license plate to be cropped out.

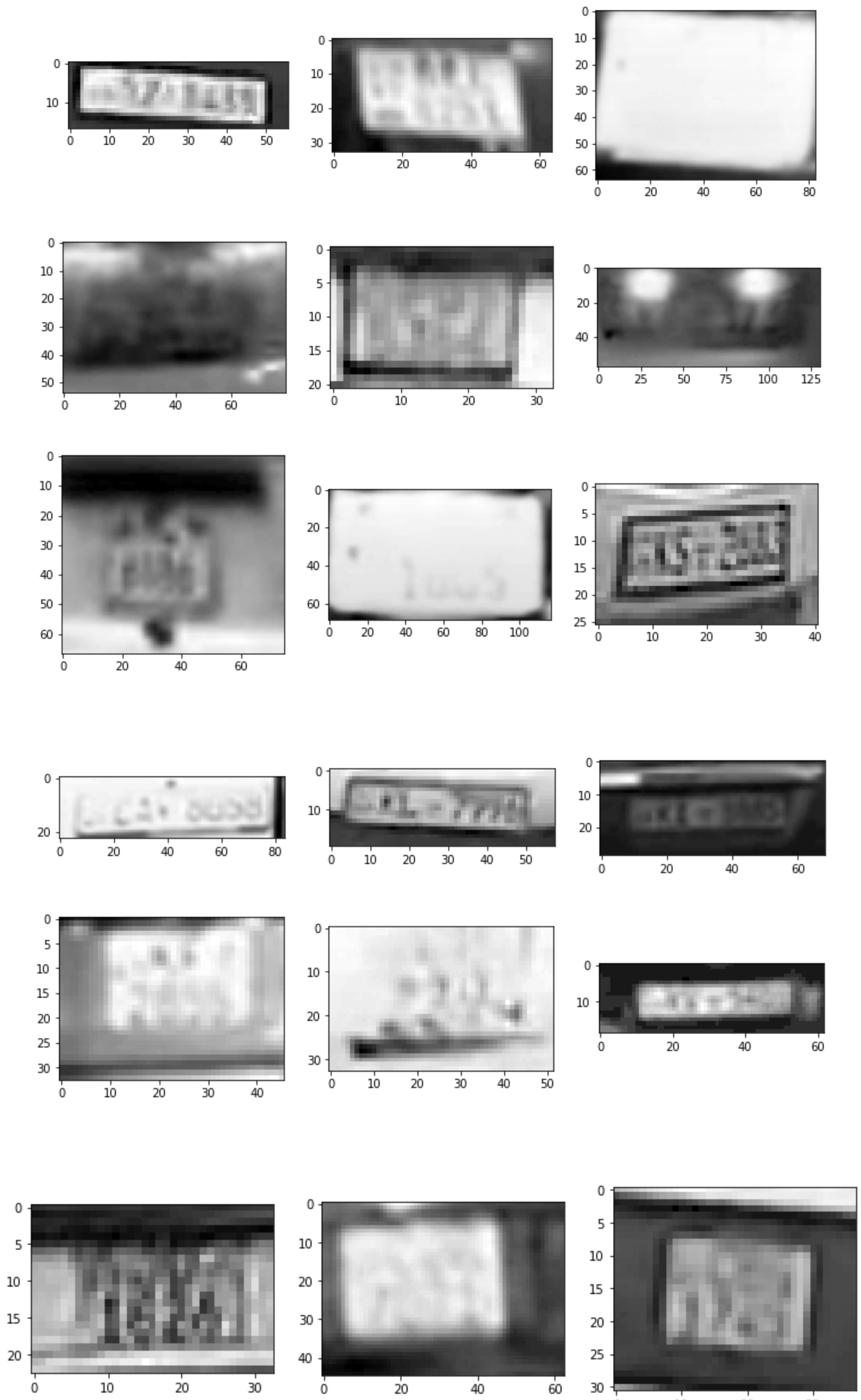
Having identified the bounds of the license plate, the algorithm crops out that section from the input image and applies a perspective transform on the license plate. This is to remove any warping of the characters caused by the alignment of the license plate in the image.

Finally, a threshold operation is applied to make the characters stand out and well-defined for easier recognition. This is done through OpenCV’s inbuilt adaptive thresholding function. Then, it is fed to the Tesseract OCR engine which processes the image and produces the recognized characters as a string output.

Some clean-up is required on the received string of text since the OCR is far from perfect. Tesseract will often identify the provincial markings and other smudges in the image as special characters. To correct this, the alphanumeric characters are filtered out from the produced text and provided as the final output.

Experimentation

The experimentation for the implemented model was done mainly using the provided [Test Dataset] for the project. This dataset consists primarily of low-resolution license plate images cropped to include just the license plate.



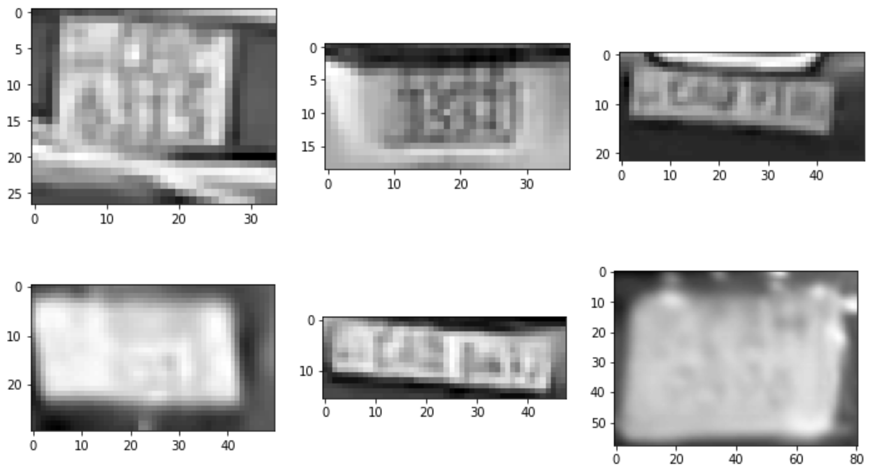


Figure 10: Low-Resolution Test Dataset

Along with this, a set of higher resolution images were also used to evaluate the character extraction and the OCR implementation.

Figure 11: High-Resolution Test Dataset

Super-Resolution

As mentioned before the approach uses Deep neural networks for the improvement of the low-resolution nature of the images. Following are a few other methods experimented with for our task.

Efficient Sub-Pixel Convolutional Neural Network (ESPCN)

This method looks at performing super-resolution by extracting feature maps from the low-resolution images themselves and uses complex upscaling filters to get the required result, to make sure the complex operations happen in the lower dimensions, and to make it faster by deploying the upscaling filters only at the end.

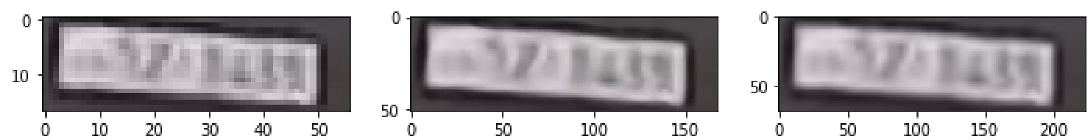


Figure 12: Left - original image. Middle - ESPCN_x3 upscaled image. Right - Image upscaled using OpenCV's resize function

Laplacian Pyramid Super-Resolution Network (LapSRN)

The approach was born as an in-between method of upscaling the start and the end. As the name suggests the method consists of an architecture similar to Laplacian pyramids which does the upscaling of the low-resolution image until the end. The basic model contains two branches: feature extraction and image reconstruction. Parameter sharing occurs on different scales which means one pyramid is used for scaling 2x, two for 4x, and three 8x, and so on. For our implementation, we have used the 8x super-resolution result of the method.

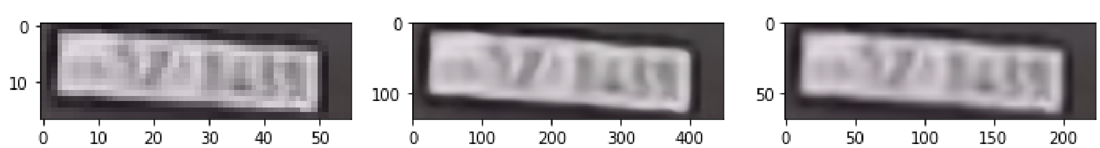


Figure 13: Left - original image . Middle - LapSRNx8 upscaled image. Right Image upscaled using OpenCV's resize function

Images in the given dataset were used to check the parameters of the implementation and to check whether a satisfactory result can be obtained for any of them.

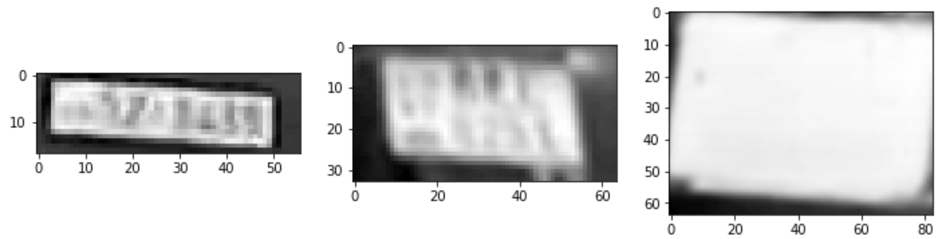


Figure 14: Original images (A)

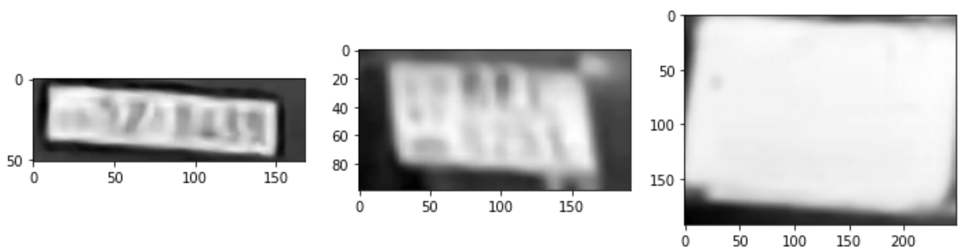


Figure 15: ESPCN super-resolution operation (A)

As the results show each model de-pixelates the low-resolution images and gives out a smoother representation of the features. Although the task at hand requires a more detailed result the seemingly highest result we could achieve was certain images such as A1.PNG

As the direct results of the operation seemed to be with low contrast, sharpening filters were used before to get a better depiction of the features.

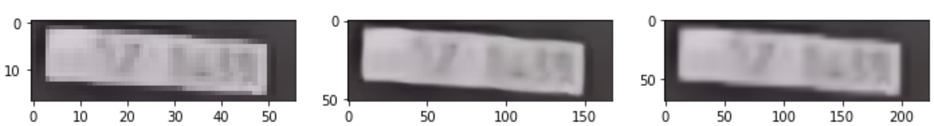


Figure 16: Bilateral filter before ESPCN super-resolution operation (A1.PNG)

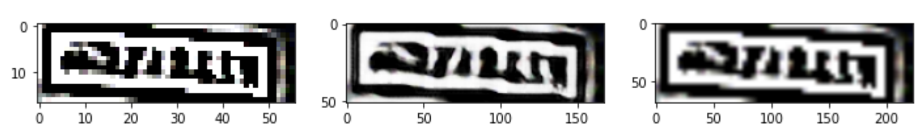


Figure 17: High pass filter (kernel 5) before ESPCN super-resolution operation (A1.PNG)

However even though some images received an enhancement with the sharpening filters most of the others became more degraded as the filters could not do an accurate identification of the image characteristics. Hence we settled upon using only the super-resolution implementation without any additional filters.

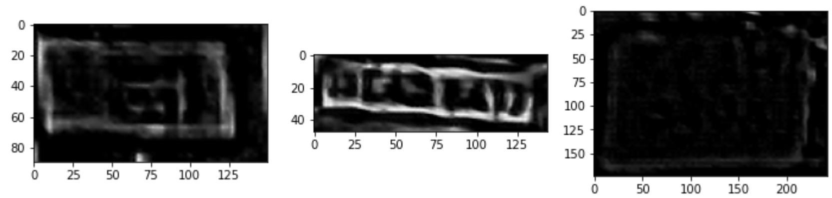


Figure 18: Further degraded set of images (H)

Noise Detection and Noise Reduction

As the images are with noticeable degradations a significant amount of time was spent identifying them. In the beginning, most of the fundamental image processing techniques were used for the given test images to achieve a human-readable output.

Image processing techniques

As the results of these experiments yielded little to no difference when compared with the original images an effort was made to identify common characteristics of the images that can be useful in the construction of a suitable solution for noise reduction. One thing we noticed is the low contrast nature of the images. Hence an effort was made to observe the single-channel histograms.

Histogram analysis

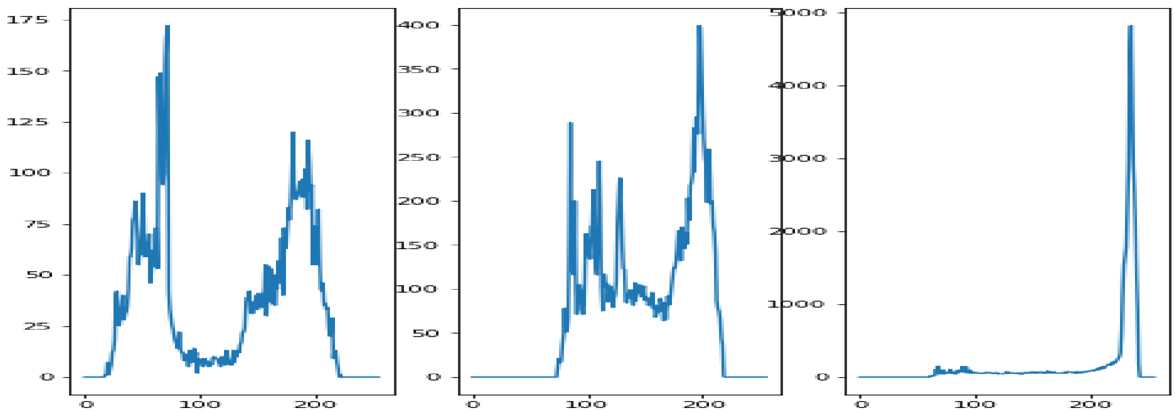


Figure 19: Histograms of image set Group A

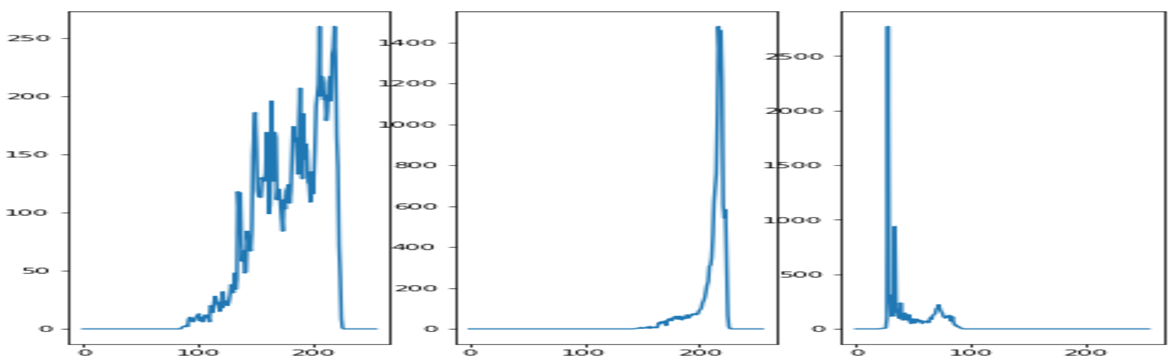


Figure 20: Histograms of image set Group E

As seen by the plots the histograms revealed that most of the images contain low contrast. To deal with this detected quality we used contrast stretching to gain more characteristics of the images. Although it was deemed to be only a little effective it was considered much better to use the resultant images rather than the original.

Noise Reduction

High-frequency noise components were identified during the analysis of the image and a low-pass Median filter was used to reduce the noise. This got rid of most of the noise in the captured images due to poor lighting conditions while preserving the details of the license plates. We also experimented with the Bilateral filter but, despite its edge-preserving nature, it blurred out details in the image that were better handled by the Median filter.

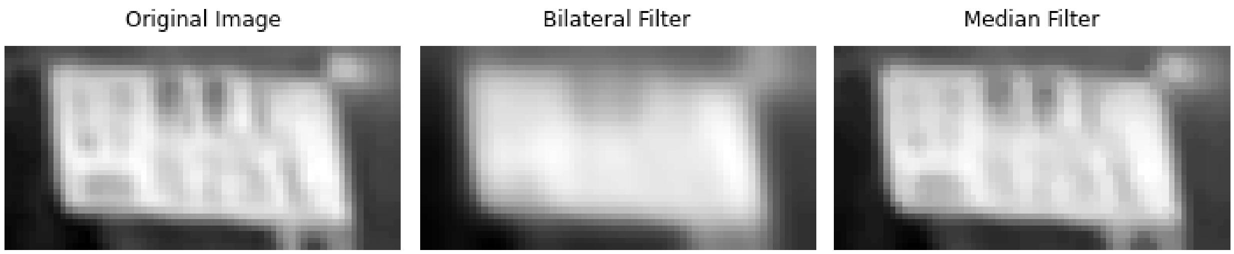


Figure 21: Results of applying low-pass filters to the “A2.PNG” image

Fourier Transform

Even though we tried approaching the problem in the spatial domain at the early stages, later on, we moved with an analysis in the Fourier domain to visualize the frequencies of the images. A basic approach was taken to remove the high-frequency noise of the images by masking them using a thresholded value. The following shows an implementation of one of the images (A1). It can be seen that the mask itself becomes a black square noting the inability to detect the noises of the given image. This result was the same for all the images found on the dataset.

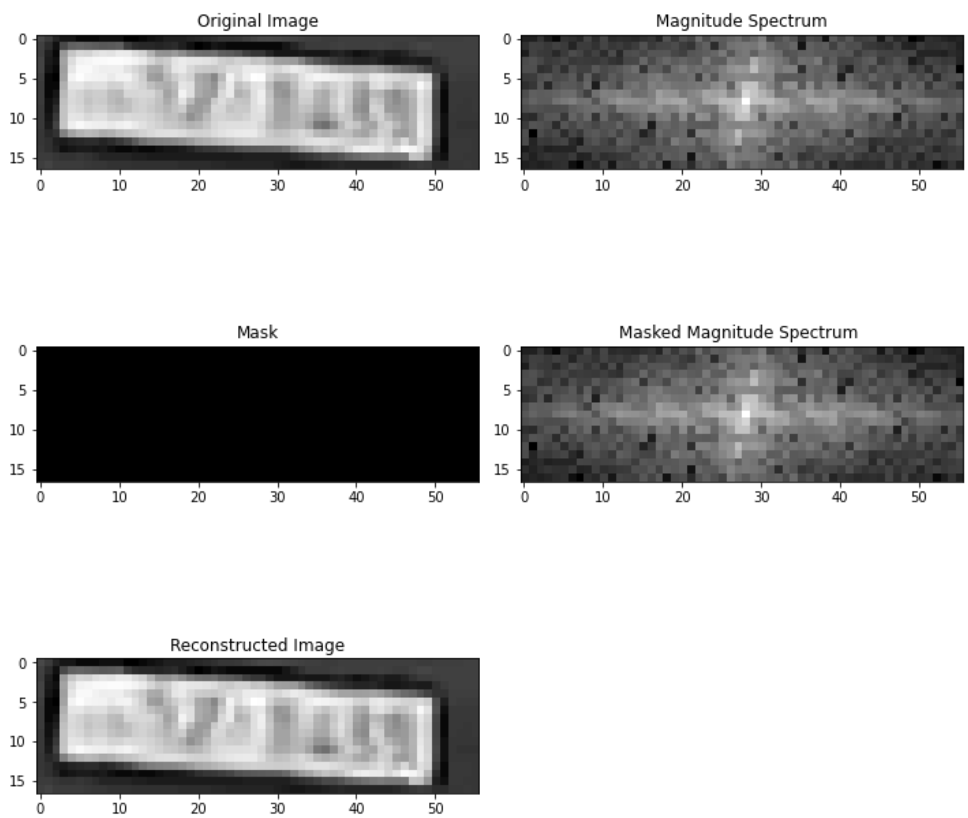


Figure 22: Attempt at Fourier domain noise removal

Optical Character Recognition

After some research on OCR tools, the widely-used, publicly-available Tesseract OCR engine developed by Google was picked for the character recognition.

Tesseract recommends cropping out the image to only the text portion to be scanned. To conform to this, we added a preliminary step to the character extraction algorithm that localizes the license plate. Then, the algorithm was tested with the high-resolution dataset.

It was seen that the engine still struggled to produce a correct output when the license plate was skewed. The warped perspective prevented the OCR engine from picking up the characters correctly. Tesseract performs best when the characters appear to be forward-facing with no warping. Since CCTV cameras are typically placed overhead on roads, it is likely that the images will require a perspective transform in practice as well. Therefore, an additional step to correct the perspective was added. This resulted in an improvement to the accuracy of recognizing images with warped perspective.

Tesseract has in-built binary thresholding to improve the accuracy of the engine. However, it was noted during our experiments that the engine performs better when a thresholded image is provided. Hence, the cropped image was also thresholded before feeding into the OCR engine.

During our testing, this methodology produced reasonably accurate results with some erroneous results with images of extreme perspective. It was also observed that Tesseract consistently misidentified certain characters in the font used for Sri Lankan license plates. Since Tesseract can be trained with custom datasets, we propose that the solution can be improved further by training Tesseract with a specialized dataset containing only Sri Lankan license plates.

Figure 23: Results of OCR on high-resolution images

Table 1: Accuracy of the OCR algorithm on the test dataset

Percentage of Correct Readings with No Perspective Correction or Thresholding	**Percentage of Correct Readings with Perspective Correction**	**Percentage of Correct Readings with Perspective Correction and Thresholding**
36.7%	40.5%	64.7%

Conclusion

During the course of the project, we have discovered that a large amount of pre-processing is required to obtain a reliable image that can be used for optical character recognition. An especially challenging artifact to remove was the motion blur in the captured images. It was also apparent that the images needed to be of higher quality to achieve any form of success with OCR.

As future developments over our work, it may be prudent to explore more advanced approaches to solving the problem such as deep learning. As outlined in this document, more accurate results may be obtained by training the OCR engine with a specialized dataset of Sri Lankan number plates. Alternatively, it may also be possible to avoid OCR entirely and use techniques such as SVM classification to identify the characters in the images. This may produce more accurate results, especially with low-resolution images where the OCR method notably fails to produce any correct results.

References

[1] Lim, B., Son, S., Kim, H., Nah, S., and Lee, K., 2022. *Enhanced Deep Residual Networks for Single Image Super-Resolution*. .

[2] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A., Bishop, R., Rueckert, D. and Wang, Z., 2022. *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*.

[3] Lai, W., Huang, J., Ahuja, N. and Yang, M., 2022. *Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution*.

[4] Kishore R Bhagat and Puran Gour. Article: Novel Approach to Estimate Motion Blur Kernel Parameters and Comparative Study of Restoration Techniques. *International Journal of Computer Applications* 72(17):21-26, June 2013.

[5] Ajanthan, Thalaiyasingam & Kamalaruban, Parameswaran & Rodrigo, Ranga. (2013). Automatic number plate recognition in low-quality videos. 2013 IEEE 8th International Conference on Industrial and Information Systems, ICIIS 2013 - Conference Proceedings. 566-571. 10.1109/ICIInfS.2013.6732046.

Appendix

Experiments:
https://colab.research.google.com/drive/1x4B-n-2Ml9wrHx7Tru-AQyEDu7gXPtO2?usp=sharing

Final Solution:
https://colab.research.google.com/drive/1-pCZbtF5JKPPn2CHD0W6MJRfY2JnbGmg?usp=sharing