UPPSALA
UNIVERSITET

Report for Computer Assisted Image Analysis II

Project:Registration plate recognition and analysis

Group 05

Uvais Karni Mohideen Meera Sha and Praveen Swamy

March 8,2020

Contents

1 Subject Presentation

The project task is to localize the vehicle license plate, removal of noise, enhancing the region of interest, segmentation, and identification of the characters present on the number plate.

We have got a dataset of images that consists of both the front and rear sides of vehicle. These images have different lighting conditions and little noise. The region of interest, the number plate needs to be localized and it's a bit challenging task as the vehicles have some stickers with texts and properties of the stickers are like license plates.

Localizing the number plates from the images will make the process easier in real-time applications for e.g. the surveillance department can face difficulties to capture exactly the license plates of fast-moving vehicles. Noises in the images pose difficulties in segmenting the individual characters and then identifying it. Therefore, the noises in the images should be removed and enhancement methods can be utilized.

1.1 Existing solutions

Following are some of the existing solutions:

- License plate detection using OCR:

  In this method, the pre-processing stage followed grayscale conversion, edge detection using canny edge operator, morphological dilation operation to thicken the boundaries, median filter to remove the impulsive noise. And the built-in function of MATLAB, Optical Character Recognition (OCR) was used to identify the characters.

- License plate localization using histogram:

  In this method, grayscale conversion, morphological operation - erosion and dilation were performed in pre-processing stage to remove noise. In the localization step - low pass filter was applied to the images to suppress higher intensities to focus on the homogeneous region. Then the histogram of the images was computed across vertical and horizontal directions. Then the license plate region was identified by combining these two histograms.

1.1.1 Restrictions and similarities

In method 1 MATLAB's built-in function OCR was used which could be a limitation to run the application on different platforms. The use of dilation will lead to connecting the edges of the characters with the boundaries of the license plate. This would lead to the misclassification of characters. The region props method was used from this solution to extract the individual characters using bounding box property.

In method 2 the use of histogram to localize the license plate region will not be applicable to images with different lighting conditions and with multiple

homogeneous regions. There was no similar method from this solution which we used.

Yes, our project is application oriented. We used two types of image datasets localized number plate images[2] and another dataset[1] consists of rear and front view of vehicles.

2 Method

Following are the stages in our implementation:

1. Localization
2. Pre-processing
3. Character Segmentation
4. Character Identification

2.1 Localization

In this step we first convert the RGB image into grayscale and then binarize the image. Gaussian filter is applied to remove the noise and then sobel operator is used to identify the edges. Next morphological dilation operation is applied on the edge image to remove the gaps within the plate boundaries so that when we apply imfill the entire region of number plate will be masked.

Then we used region props to mark the region with the highest area i.e. the number plate. Later we use the bounding box coordinates to extract the number plate region from the original image.

Figure 1: RGB image

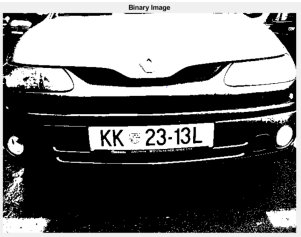

Figure 2: Binary image

Figure 3: Sobel edge image

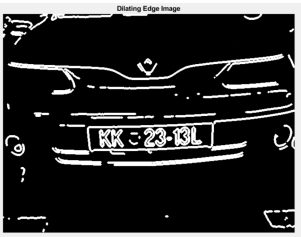

Figure 4: Dilated edge image

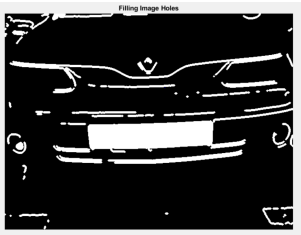

Figure 5: Image holes filled

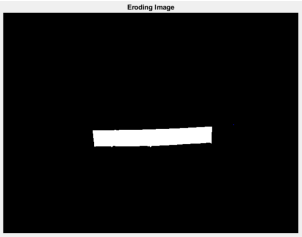

Figure 6: Eroded image

Figure 7: Localized ROI

2.2 Pre-processing

We perform two stages of pre-processing techniques. First, we perform grayscale conversion and then apply median filter to remove noise. We sharpen the image to improve the resolution using imsharpen function. Then we binarize the image using threshold value determined by graythresh function. We applied morphological close operation to get uniform structured characters. Then we apply bwlabel to label the different regions in the image. After which we apply imclearborder function to clear the boundaries of the license plate.

Next we use regionprops to get the regional properties of different regions. Then we filter by the following conditions: Area, Equivdiamater and solidity. Then we check if there are 7 characters, if not we run second pre-processing technique which also includes histogram equalization. The reason behind this is to enhance the contrast in all the regions.

Figure 8: Gray scale image

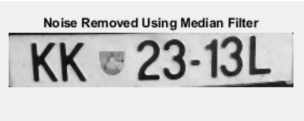

Figure 9: After applying median filter

Figure 10: Sharpened image

Figure 11: Binarized image

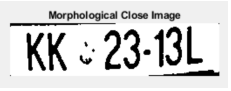

Figure 12: Morphological close operation

Figure 13: Labeled image

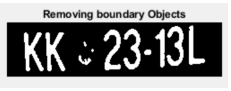

Figure 14: After removing boundaries

2.3 Character segmentation

We applied region props on the result of fig 14 and then filtered the regions with conditions for Area and Equivdiameter properties. The segmented characters are shown in the following figure.

Figure 15: Segmented Characters

2.4 Character identification

After identifying individual characters, we extracted the HOG features for each character and trained the SVM model with this feature vector. The model was trained for 207 training images with defined labels for each character. Then we predicted the characters for 75 test images, and we achieved an accuracy of 95.43%.

In localization step we referred a video tutorial[2] for the different methods to be used. We referred a MATLAB tutorial[1] - HOG feature extraction for digit classification for feature extraction and training SVM model.

3 Implementation

- First, we acquired the images from the dataset(reference) and processed 1st pre-processing technique wherein the grayscale image was subjected to noise removal, then the image was sharpened. After which we binarized the image and applied closing morphological operation with disk of size 1 as structuring element. Then we labeled the regions of the image and removed the borders of the plate area.

```
filt_res = medfilt2(gray_image);
sharpen_res = imsharpen(filt_res);
thresh = graythresh(sharpen_res);
binary_res = imbinarize(sharpen_res,thresh);
morph_close_res = imclose(binary_res,se);
label_res = bwlabel(morph_close_res,8);
bord_res = imclearborder(label_res,8);
```

- Then we apply region props on the above processed image to identify the characters by filtering the blobs on Area, Equivdiameter and solidity parameters.

```
object_stats = regionprops(bord_res,'all');
index1 = ([object_stats.Area] >= 70 & [object_stats.Area] <=
    350);
index2 = ([object_stats.EquivDiameter] >= 9 & [object_stats.
    EquivDiameter] <= 25);
index3 = ([object_stats.Solidity] <= 0.82);
indexes = find(index1 & index2 & index3);
character_image = ismember(image_labeled,indexes);
character_stats = regionprops(character_image,'all');
```

- Then we check for the condition i.e. if we segmented 7 characters, as our number plate dataset has 7 characters. If we fail to get 7 characters from the image, then we apply an additional pre-processing step i.e. histogram equalization to enhance the contrast throughout the image. Then we apply regionprops again to get the exact individual characters. The below pseudo code was coded using reference from MATLAB tutorial[1].

```
histeq_res = histeq(sharpen_res);
```

- Next we perform feature extraction on the segmented characters using extractHOGfeatures and stored it in a list. After feature extraction, we apply supervised learning i.e. train the SVM model using training dataset and their defined labels. Next we test the model using the test dataset with 75 images. We achieved an accuracy of 95.43% for individual character classification.

```
for i = 1 to length(list_segmented_characters)
     features(i,:) = extractHOGFeatures(cell2mat(
    list_segmented_characters{1,i}),'CellSize',[3 3]);
end
```

```
4
5 %%Train the model using SVM classifier
6 model = fitcecoc(features,       labels_arr,'Coding','onevsone'
      ,'Learners','svm','ObservationsIn','rows','Verbose',2);
7
8 %%Test the model
9 predictedLabel = predict(model,test_features);
10 Accuracy = (sum(predictedLabel == pred_labels)/length(
      pred_labels));
```

- Then we tested the model for the dataset with rear and front view images to predict the characters. Here we localized the images and extracted the number plate region. Then we preprocessed the image and segmented the characters from the plate region and predicted the characters using the pre-trained model from the previous step. We referred a video tutorial for localization methods[2].

4 Results

We used dataset[3] with localized number plates and split the dataset into training–201 images and testing set–75 images, then we trained the SVM model and tested the model using the test set. We achieved an accuracy of 95.43% for individual character classification. We evaluated the accuracy by checking how many of the predicted labels were correctly classified. We also evaluated the accuracy for individual number plates, and we achieved an accuracy of 82.67%.

We were able to localize the number plates for around 200 images from the dataset[4] which consisted 457 images.

5 Discussion

Based on our implementation techniques used, we had to tweak different values for regionprops, thresholding, noise filtering and sharpening to get decent results. We were able to localize only 50% of images from the entire dataset. We could have overcome the issue of localizing by using CNN based semantic segmentation which involves the use of networks like FCN or U-net. Our method did not work for harsh lighting conditions like too bright and dark. We could not process the number plates with different colored backgrounds. The efficiency of the code was good enough that it needed only 0.648 seconds to predict characters for each label.

We did not consider images with blur , deformed and images which were captured in odd angles. We learnt how histogram of gradients(HOG) is font invariant.

References

[1] License plate detection using Optical Character Recognition (OCR), *https://pdfs.semanticscholar.org/5ef1/9f9fad5d79119db9de8942a00b5197e5383a.pdf*

[2] License plate localization using histogram, *https://www.youtube.com/watch?v=ACKu0kK3dU8t=170s*

[3] Dataset 1, *https://medusa.fit.vutbr.cz/traffic/research-topics/general-traffic-analysis/holistic-recognition-of-low-quality-license-plates-by-cnn-using-track-annotated-data-iwt4s-avss-2017/*

[4] Dataset 2, *http://www.zemris.fer.hr/projects/LicensePlates/english/results.shtml*

Appendices

Please refer the following files for the implementation code:

1. Segmenting_using_localized_images.m, It includes code for segmenting the characters from localized image dataset.
2. Numberplate_localisation.m, It includes code for localizing number plates from front and rear view image dataset.
3. Integrated_code.m, It includes the code for predicting the characters of localized images using trained model from code 1.