VIETNAM GENERAL CONFEDERATION OF LABOR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY

NGUYEN GIA MY – 521H0272

TRAFFIC SIGNS RECOGNITION

FINAL PROJECT

COMPUTER SCIENCE

HO CHI MINH CITY, 2023

VIETNAM GENERAL CONFEDERATION OF LABOR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY

NGUYEN GIA MY –521H0272

TRAFFIC SIGNS RECOGNITION

FINAL PROJECT

COMPUTER SCIENCE

Advised by
Dr. Pham Van Huy

HO CHI MINH CITY, 2023

ACKNOWLEDGEMENT

I would like to express our sincere gratitude to Mr. Pham Van Huy, our instructor, and mentor, for his valuable guidance and support throughout the final report of our project on Traffic Signs Recognition. He has been very helpful and patient in providing me with constructive feedback and suggestions to improve my work. He has also encouraged me to explore new technologies and techniques to enhance my project functionality and performance. I am honored and privileged to have him as our teacher and supervisor.

Ho Chi Minh city, 7th January 2024.

Author

(Signature and full name)

My

Nguyen Gia My

DECLARATION OF AUTHORSHIP

I hereby declare that this is my project and is guided by Mr. Pham Van Huy ; The content research and results contained herein are central and have not been published in any form before. The data in the tables for analysis, comments and evaluation are collected by the main author from different sources, which are clearly stated in the reference section.

In addition, the project also uses some comments, assessments as well as data of other authors, other organizations with citations and annotated sources.

If something wrong happens, I'll take full responsibility for the content of my project. Ton Duc Thang University is not related to the infringing rights, the copyrights that I give during the implementation process (if any).

Ho Chi Minh city, 3rd September 2023

Author

(Signature and full name)

My

Nguyen Gia My

TRAFFIC SIGNS RECOGNIZATION

ABSTRACT

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CHAPTER 1. SOLVING METHODS

This project is using for recognize prohibition signs. Most of prohibition signs have the specific attributes:

- Circular shape.
- White background.
- Outer border in red.
- A red (or orange) diagonal cross or object draw in black inside the circle.

1.1 Detect and draw contours around the prohibition signs

To detect and draw contours around the prohibition signs, we need to:

- Change the color channel from BRG (OpenCV) to HSV.
- Create two masks (mask_r1, mask_r2) to isolate the colors red (or orange) and blue (or violet) from the HSV range. These are red and blue HSV range:
    - (0, 100, 100), (10, 255, 255).
    - (160, 100, 100), (180, 255, 255).
- Combine these two masks (mask_r1 and mask_r2) into mask_r to have a final mask that contains red and blue color.
- Apply the mask_r to the original image to retain only the regions of red or blue colors.
- Blurs the image using a Gaussian filter to reduce noise and make the edges smoother.
- Uses the Canny algorithm to detect edges in the blurred image.
- Finds contours in the blurred image, using the RETR_EXTERNAL method (OpenCV) to only return external contours and CHAIN_APPROX_SIMPLE (OpenCV) to compress the contours.

- Iterates through all the found contours:
    - Calculates the area of the contour.
    - Only the contours with an area larger than 700 pixels are drawn.
    - Calculates the bounding rectangle around the contour.
    - Generates a random color for each contour detected.
    - Draws a rectangle bounding with a random color around the prohibition sign that is detected.

1.2 Training model to recognize prohibition signs

To train the model that can classify prohibition signs, I used Tensorflow to build a Deep Neural Network (DNN) [1]. The dataset used in this project is a combination of two datasets from Kaggle:

- GTSRB - German Traffic Sign Recognition Benchmark.
- Traffic Sign Dataset Classification.

This project's dataset includes 22 classes representing 22 common prohibition signs that are widely used in most countries (labels.csv):

- TDTD x km/h, x = {5, 15, 20, 30, 40, 50, 60, 70, 80, 100, 120}. (Max speed x km/h).
- Cam Dung Va Do Xe (No stopping).
- Cam Di Nguoc Chieu (No Entry).
- Duong Cam.
- Cam Di Thang Va Re Trai (Do not go straight and left).
- Cam Di Thang Va Re Phai (Do not go straight and right).
- Cam Di Thang (Do not go straight).
- Cam Re Trai (Do not turn left).
- Cam Re Phai (Do not turn right).
- Cam Re Trai Va Phai (Do not turn left and right).
- Cam Quay Dau Xe (No U-turn).

- Cam Xe Oto (No car).

Model’s accuracy and loss value:

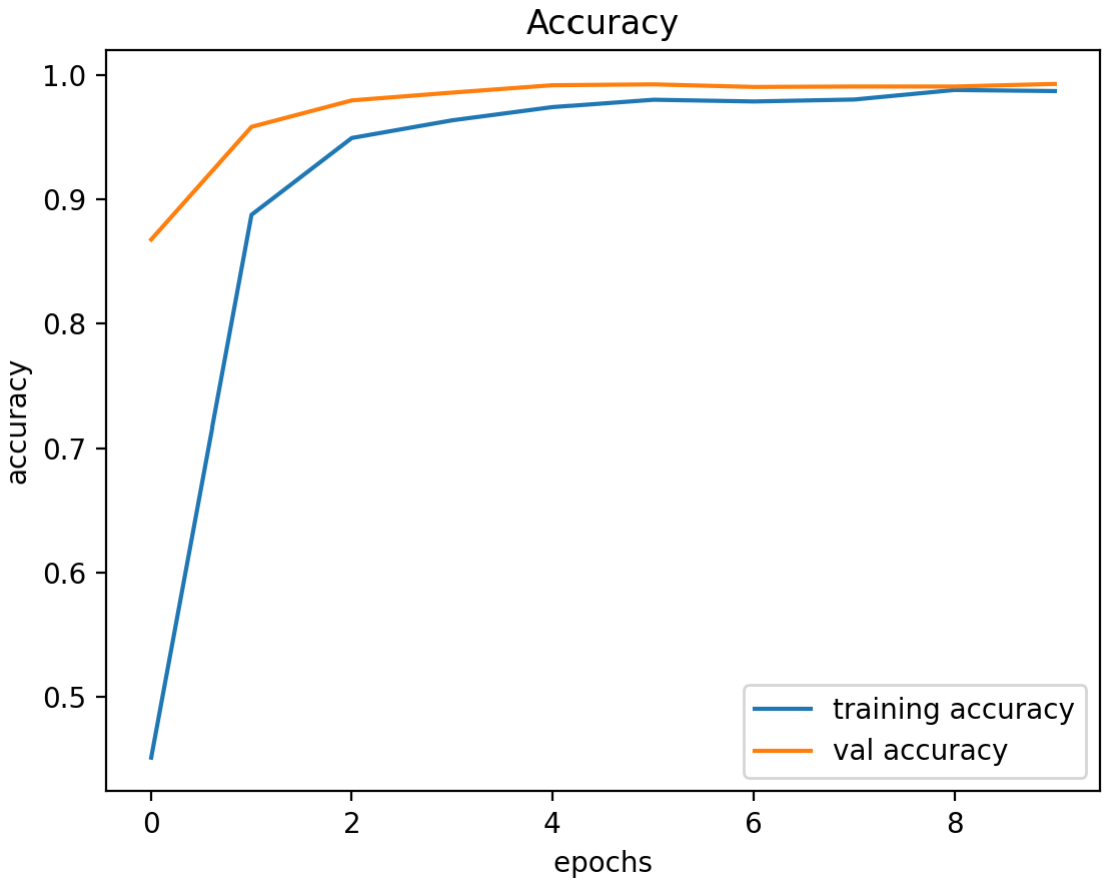
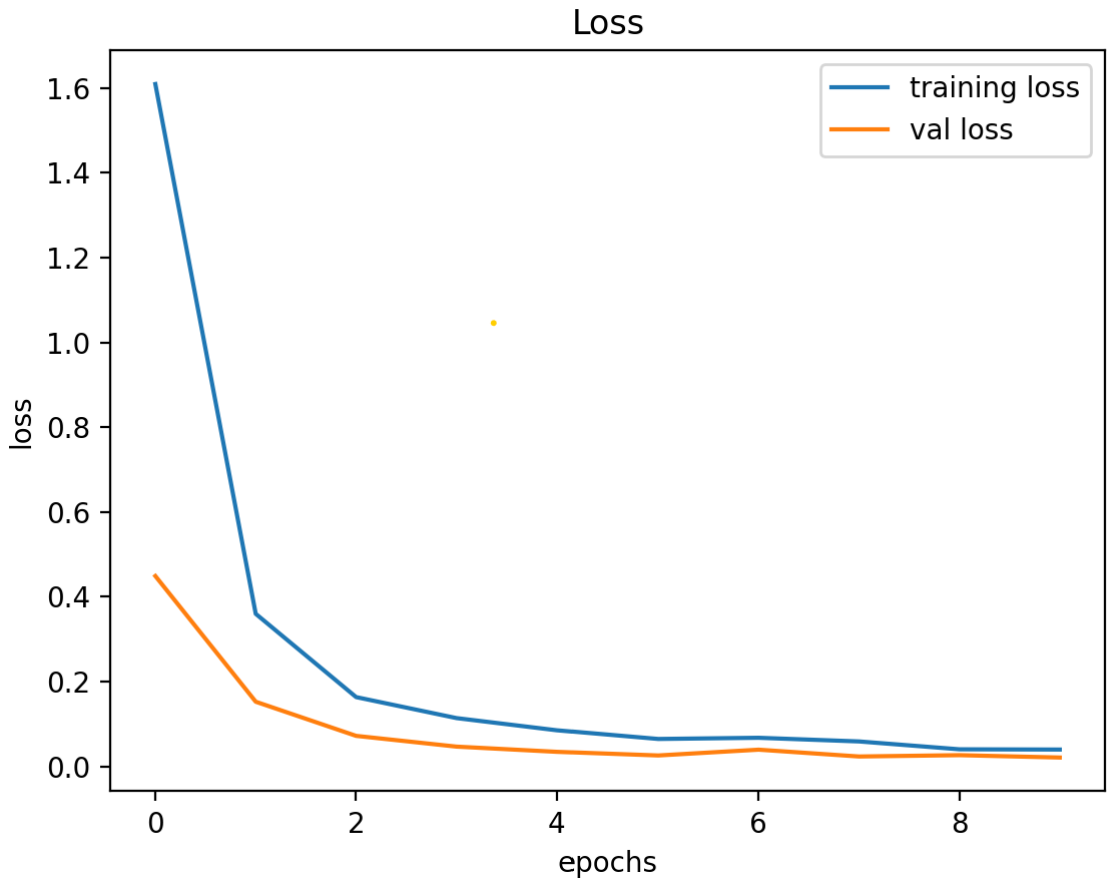


Figure 1.1 Model’s accuracy.



Picture 1.2 Model's loss value.

1.3 Predict the prohibition sign

After drawing the rectangle box for the prohibition signs in an image, we need to classify it. The object (prohibition sign) was cropped and saved into a variable named crop, which we are going to use the model to predict.

The cropped image is converted to a numpy array resized to 30x30. Then, use the model to predict the class of that prohibition sign.

CHAPTER 2. BASIS EXPERIMENTAL STEPS AND RESULT

2.1 Project Structure (Source code)

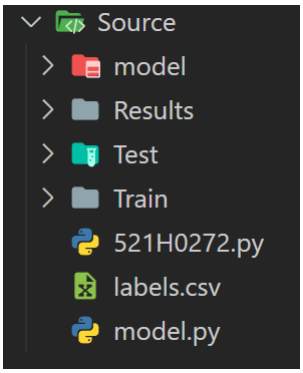


Figure 2.1 Source Code Structure.

The Source folder concludes all the code for processing images, training images, testing images, and a deep learning model.

- model (folder): include the model.h5 file, which is the model that is saved after creating in the model.py.
- model.py (file): create a deep learning model.
- Results (folder): contains all the result images (image after predicting, crop images from the input image is saved at crop_images folder, result.txt is used for saving crop image paths and the class name).
- Test (folder): testing images.
- Train (folder): training images.
- 521H0272.py (file): processing the input image.
- Labels.csv (file): 22 class names (prohibition signs' name).

2.2 Run this project

To run this project, please open 521H0272.py file, then enter the image path at line 133. After that, open the terminal and execute the 521H0272.py file.

Let’s predict the Test/Test_00_06.jpg image.

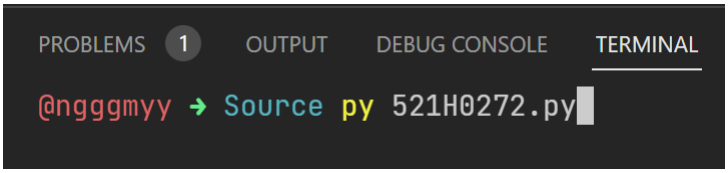
Figure 2.2 Original image.

```
131  def main():
132      print("--- Run Main Function ---")
133      img_name = 'Test/Test_00_06.jpg'
134      img = read_img(img_name)
135      detect_img(img, img_name, show=False, export=True)
136
137  main()
```

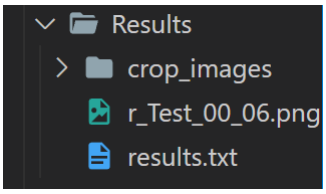
Picture 2.3 Change the image path at line 133 in 521H0272.py file.

detect_img(img, img_name, show, export) is used to show or export the predicted image. If ‘show’ is equal to False, this application is not showing the predicted image. Otherwise, it will show the predicted image using matplotlib library. If ‘export’ is equal to True, the application will export the predicted image to the Results folder, and all the prohibition signs that are detected are saved in

crop_images folder. The prohibition signs' names and it image paths are saved at results.txt file.



Picture 2.4 Execute 521H0272.py file



Picture 2.5 Results folder after running 521H0272.py.

Figure 2.6 Image after predicting.

The input image after predicting is saved at Results folder with the syntax: r_original-image-name.png. In this example, the result image name is r_Test_00_06.png.

The crop images (prohibition signs detected) are saved at the crop_images folder, following the syntax: original-image-name_specific-number.png (specific-number is a random number from 1 to 1000).

Figure 2.7 First cropped image. (Test_00_06_329.png)

Figure 2.8 Second cropped image. (Test_00_06_328.png)

The result.txt file contains the cropped image paths (prohibition signs) and the class names.

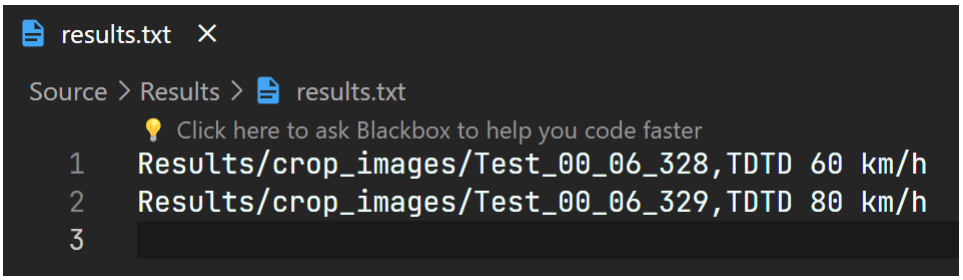


Figure 2.9 result.txt file

2.3 Additional links

To get the training images or get more result images, please visit this Google Drive link:

https://drive.google.com/drive/folders/1HtJIia1Q86al7mUBu1NG0c-pbu1Yg78l?usp=sharing

REFERENCES

[1]. Chirag Samal. (2019). Traffic-Sign-Recognition. Retrieved January 1 2024, from GitHub website:

chiragsamal/Traffic-Sign-Recognition: Traffic Signs Detection and Recognition with Keras (github.com)