

Research Proposal

Title: Intelligent Driving System Based On Android

Author: Yao Xiao

Supervisor: Xinguo Yu

Student Number: 2019180015

Introduction and Aim

Image processing is a method of converting an image into a digital form and performing some operations on it to extract or extract some useful information from it. This is a signal distribution where the input is an image, such as a video frame or photo, and the output can be an image or feature error associated with the image! The reference source was not found. Generally, an image processing system includes a method of processing an image using a two-dimensional signal while applying a signal processing method thereto. Image processing has two types of methods, one is analog image processing, and the other is digital image processing. Analog image processing can be used for hard copies such as printouts and photos. When using these vision techniques, image analysts use a variety of interpretation bases. Computer technology can effectively assist digital image processing technology in manipulating digital images. Preprocessing, enhancement and display, and information extraction are the three general stages that all types of data must undergo when using digital technology.

Image processing, one of the technologies that tells development today, is applied to all walks of life. Image processing is also a core research area in engineering and computer science. Today, entering the Internet era, as the most convenient mobile terminal for the mass access, the average user needs more and more intensely for the Android platform image processing software in order to enjoy more intelligent services. As a relatively difficult platform for Android platform, graphics image processing is highly professional. At present, there are few researches on image processing technology that can maximize the advantages of mobile platforms such as Android.

Based on the above research background and other related descriptions, this topic based on the image processing of the Android platform has been studied in depth, and a software for intelligent assisted driving has been designed. The project focuses on the fast-developing autonomous driving industry, combining the excellent

performance and high openness of the Android platform to maximize the software advantages, provide real-time vehicle condition information and automatically detect and analyze the alarms and regulate the driver's driving habits, it is used to meet the needs of imaginative autopilot users.

Literature Review

Because of its high openness and portability, Android is widely used in various scenarios, and autonomous driving continues to attract attention as technology continues to evolve and improve. In recent years, more and more researches have intended to achieve vehicle detection and recognition on the android platform, and to improve the accuracy of the recognition and the efficiency of the algorithm, and strive for the most complete automatic driving experience.

As early as 1999, Stan Z. Li et al. [] proposed a face recognition method called the nearest feature line (NFL), which can summarize any two feature points of the same type of person by deriving at the same time by passing through two points of FL, while FL can capture more facial image changes. The author makes a systematic analysis and review of the classification points based on the nearest distance. On this basis, the research prospects of NFL-based pattern recognition methods are prospected, and it also lays a certain foundation for the development of face recognition. In order to better realize the recognition of feature regions, the texture of the face needs to be further processed. Joan Medina et al. [] Developed a Bayesian formula AMM. This model can use Gaussian noise and Gaussian priori to marginalize the potential texture space, which is convenient for processing face images with less noise. At the same time, the statistical model is established by annotated reference points, and these reference points are one of the important factors in determining the feature area. Due to the complexity of the environment, the face may have non-linear deformation. In order to better handle this kind of situation, Yu Chen et al. [] combined a method based on joint cascade regression [] and deep learning to use RNN and DNN from rough to fine architecture to learn and improve the shape of the face, finally the low-resolution face image input can be directly mapped to the estimated face shape. Based on the research of the above article, Nannan Wang et al. [] proposed a method based on parametric shape model and a method based on non-parametric shape model (FFPD), which can comprehensively detect facial feature points with the help of rich manual labeling, and choose different methods according to different situations to complete feature point recognition. In terms of breadth, these studies have undoubtedly achieved considerable achievements. However, in terms of depth, the portability of these recognition technologies on different platforms is far from sufficient, not only depending on the hardware, but also severely restricted by the software system.

On the other hand, real-time vehicle condition detection based on the Android platform is an important research content for autonomous driving. Artis Mednis et al. proposed an Android-based irregular road detection system, which can obtain comprehensive real-time road conditions in favor of autonomous driving. Teddy Gunawan et al. [1] studied the ANPR algorithm based on ANN (Artificial Neural Networks) versus template and template-based OCR, including deep image process-

ing and character recognition. Yujie Jheng et al. [2] proposed a forward collision warning detection algorithm, established an ROI, and improved the error detection using a Bayesian classifier, which made the recognition result more robust than using only shadow detection. It can be seen that there have been a lot of great studies, but there are still many problems that need to be solved. For example, how do you integrate many algorithms into the android system instead of using them alone? How many recognition algorithms and techniques should design in order to achieve a complete autonomous driving experience? How to best integrate facial feature point recognition technology with Android platform? How to design a detection system that is safe for the driver [4]? How to detect if the driver is in compliance with the driving rules during autonomous driving?

Based on the above mentioned situation, in order to better combine these excellent research and make innovations and useful research contributions, the research work in this paper is based on face feature region recognition technology and real-time vehicle condition detection technology. The research content of this thesis is based on the Android platform, mainly from the face feature area recognition, real-time vehicle condition detection and analysis, adaptation and optimization. Compared with other papers, this article is not limited to algorithms and single Android system functions, but is closely related to the Android platform.

For example, face feature point recognition, combined with the recognition algorithm of excellent papers, integrates the Tensorflow framework under the Android platform, and then the background open thread periodically calls the camera to take pictures, and uses the Bitmap to use the relevant pictures, and then enters the recognition. First, the 256-bit image data is preprocessed from the 0-255-bit int value to the float value according to the provided parameters, which is a standardized floating-point number, and finally converted into a matrix. For the converted matrix, from top to bottom, from left to scan, use a custom matrix. The custom matrix is defined as follows:

$$array[b * 3 + 0] = (byte)array[b] \quad (1)$$

$$array[b * 3 + 1] = (byte)(array[b] \gg 8) \quad (2)$$

$$array[b * 3 + 2] = (byte)(array[b] \gg 16) \quad (3)$$

Generally speaking, the matrix is much smaller than the converted matrix as a scanning tool. After scanning, in each small matrix block, the number of times each color is rendered from 0 to 255 is counted, thereby expressing the characteristics of each block, and then the data is input into Tensorflow, the corresponding interface is called back, and the output data is sequentially placed into the corresponding array created. The feature blocks corresponding to the eyes, mouth, and ears are represented by an array of positions using an Android custom image to draw a rectangular area, by defining array a and parameter m and input n, the areas are given by

$$\sum_{i=0}^3 a[4 * m + i] * n \quad (4)$$

and the initial data is set to a default value. This is highly integrated with the recognition algorithm and Android, enabling it to quickly and intuitively give the

driver feedback compared to the previous one-dimensional output, thus achieving intelligent driving.

For real-time vehicle condition detection, it also closely combines with Android to provide smarter driving modes. The purpose of this function is to detect whether the lane is deviated during driving or if the front vehicle is too close, and the feedback from the screen is used to remind the driver whether it is necessary to intervene in the driving route and mode of the vehicle. Integrate the Tensorflow framework under the Android platform, and then the background open thread periodically calls the rear camera to take pictures. Trying to use too large preview size may exceed the bandwidth limit of the camera bus, although it will produce excellent display effect, but it will take up a lot of storage space. And it is difficult to capture the data. Therefore, it is necessary to give a list of input sizes supported by the camera and a list of sizes that support the expected output class, constantly adjusting, selecting a minimum value at least as large as the corresponding texture view size, and at most as large as the corresponding maximum size, and width and height. The analogy specifies that the values match, the slope k_1, k_2 are defined as follows:

$$k_1 = -3.0f \quad (5)$$

$$k_2 = (top - bottom)/(right - left) \quad (6)$$

If $k_2 < k_1$, it is judged as a lane departure. Use canvas to draw the red font "Lane Departure", check whether the center point of the front car is in the range of one-quarter to three-quarters of the horizontal screen; If yes, use the rectangular area of the canvas layout to cover the vehicle. The judgment center distance is given as follows, the judgment center distance is given by

$$distance = (top - bottom)/2 + bottom \quad (7)$$

Methodology

The proposed study will include literature research, developing research methods, data collation, data testing, and draw conclusions, design, construct and test system requirements. Method for intelligent driving system: Mainly rely on deep learning models and some algorithms, such as CNN, to achieve face feature point detection, determine important facial features, such as the presence and location of nose, eyes, mouth, ears, etc. The purpose of introducing this technology is It can detect whether the driver is in compliance with the driving specifications, and should use the edge detection algorithm to improve the recognition accuracy. In the vehicle detection, the algorithm is designed and matched with the corresponding deep learning framework to detect whether the lane is deviated during driving or whether the preceding vehicle is too close. The feedback of the screen is used to remind the driver whether to intervene in the driving route and mode of the vehicle. Can provide a certain reference value and know the meaning of automated driving.

Software testing is the process or method of finding errors in a software application or program so that the application runs according to the requirements of the end user, called a software test. Software testing is the process of verifying the system

and is designed to identify any errors, gaps or lack of demand and actual demand. Software testing can be broadly divided into two categories, one for functional testing and the other for non-functional testing. White-box testing and black-box testing are common software testing methods.

Data Analysis

Simulate experiments in different environments with different equipment. The experimental results show that because Tensorflow does not support GPUs on mobile platforms, and the CPU is slow for floating point operations, the project has higher requirements for the processor. For low-performance processors, the recognition rate is required in both normal and low-light environments. It is significantly lower than high-performance processors, but accurate feature area positioning can be obtained on high-performance platforms.

However, when using a mobile phone to take photos in real time, it is susceptible to other external factors such as lighting conditions, weather characteristics and coverings when taking photos. Most of them can be accurately positioned, but the recognition of low-light environments needs to be improved.

Table 1: Pixel environmental experiment results statistics

	Normal environment	Low light
Number of images	50	15
Number of successes	49	12
Positioning success rate	98%	75%

Table 2: Meizu Note6 environmental experiment results statistics

	Normal environment	Low light
Number of images	50	15
Number of successes	20	5
Positioning success rate	40%	33.3%

Conclusion

Under the Android platform, the visual operation can accurately detect the driver's state and regulate driving habits in real time, and can meet the identification of some extreme conditions. At the same time, through the feedback of the screen to remind the driver whether it is necessary to intervene in the driving route and mode of the car, to detect whether the lane is deviated during driving or whether the preceding vehicle is too close, so as to achieve the effect of intelligent driving or even automatic driving.

References

- [1] Gunawan, Teddy Surya et al. Design of Automatic Number Plate Recognition on Android Smartphone Platform, Indonesian Journal of Electrical Engineering and Computer Science Vol. 5, No. 1, January 2017, pp. 99-108.
- [2] Y. Jheng, Y. Yen and T. Sun, A symmetry-based forward vehicle detection and collision warning system on Android smartphone, 2015 IEEE International Conference on Consumer Electronics - Taiwan, Taipei, 2015, pp. 212-213.
- [3] M. Y. Aalsalem and W. Z. Khan, CampusSense — A smart vehicle parking monitoring and management system using ANPR cameras and android phones, 2017 19th International Conference on Advanced Communication Technology (ICACT), Bongpyeong, 2017, pp. 809-815.
- [4] T. H. Yee and P. Y. Lau, Mobile vehicle crash detection system, 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, 2018, pp. 1-4. doi: 10.1109/IWAIT.2018.8369671