# Automatic Number Plate Recognition on FPGA

Xiaojun Zhai*, Faycal Bensaali† and Klaus McDonald-Maier*
*School of Computer Science & Electronic Engineering,
*University of Essex, Colchester, UK
*{xzhai, kdm}@essex.ac.uk

†Department of Electrical Engineering College of Engineering,
†Qatar University, Doha, Qatar
†f.bensaali@qu.edu.qa

*Abstract*—**Automatic Number Plate Recognition (ANPR) systems have become one of the most important components in the current Intelligent Transportation Systems (ITS). In this paper, a FPGA implementation of a complete ANPR system which consists of Number Plate Localisation (NPL), Character Segmentation (CS), and Optical Character Recognition (OCR) is presented. The Mentor Graphics RC240 FPGA development board was used for the implementation, where only 80% of the available on-chip slices of a Virtex-4 LX60 FPGA have been used. The whole system runs with a maximum frequency of 57.6 MHz and is capable of processing one image in 11*ms* with a successful recognition rate of 93%.**

## I. Introduction

ANPR systems have been successfully operated in the UK for several decades. First generation ANPR systems were invented in 1976 at the Home Office Scientific Development Branch in England (now known as the Home Office Centre for Applied Science and Technology, CAST) and they have successfully detected simple crimes: Tracking and finding stolen vehicles and prosecuting uninsured or un-taxed road users. One successful example is the UK's "Ring of Steel" around the city of London. The area covered includes London congestion charge zone, where motorists are required to pay a congestion charge. There are 1,500 ANPR cameras that monitor anywhere in the zone and around 98% of vehicles moving within the zone are caught on cameras. The video streams are transmitted to the National ANPR Data Centre (NADC) where ANPR software processes the registration plate of the vehicle [1]. Currently, the vehicle information is gathered from fixed cameras strategic sites (e.g. motorways and petrol stations), mobile units (e.g. police van) and CCTV in urban area  and there are 35 million number plate reads per day and this number is increasing every year [2]. ANPR becomes an important technology for intelligent infrastructure systems like electronic payment systems, access control, tracing of stolen cars, or identification of dangerous drivers [3, 4].

Typically, an ANPR system consists of three stages: Number Plate Localisation (NPL), Character Segmentation (CS), and Optical Character Recognition (OCR). The NPL stage is where the Number Plate (NP) being detected. The CS stage is an important pre-processing step before applying OCR, where each character from the detected NP is segmented before recognition. In the last stage, characters are segmented from the NP where the image format will be converted into characters by a character classifier [4].

Most methods in current ANPR systems utilise general purpose CPUs to perform complex and computationally intensive image processing algorithms. The CPU must read each instruction from memory, decode it and then execute it. Additionally, any operation needs to be implemented from basic arithmetic and logical operations in CPU, which slow down the execution speed for each individual operation. Therefore, in order to achieve real-time performance, specialist hardware platforms can be one of valuable solution for accelerating computationally intensive image processing algorithms. Some of researchers have chosen Digital Signal Processors (DSP) and/or Field Programmable Gate Arrays (FPGAs) as their platform for implementing ANPR systems [5] [6] [7]. However, most of the hardware based systems focus only on one or two stages of ANPR system due to limited hardware resources or complexity of the chosen algorithms. In this paper, a complete solution for ANPR system on FPGA is presented, where the algorithms and FPGA implementations of the NPL, CS and OCR stage are included.

The rest of this paper is organised as follows. The proposed ANPR system is introduced in Section II. The experimental setup and the implementation results are discussed in Section III. Section IV concludes the paper.

## II. Proposed ANPR System

### *A. NPL Module*

The proposed NPL module consists of two major stages: 1) Morphological operations for extracting plate features, and 2) Selection of candidate regions. The proposed NP feature extraction algorithm is mainly based on two open and one close morphological operations, the first open morphological operation is used to extract the features of the NP, the second open operation is used to remove noise, and the close operation is then used to fuse the pixels in the NP region together.

Figure 1 shows a block diagram of the proposed NPL system.

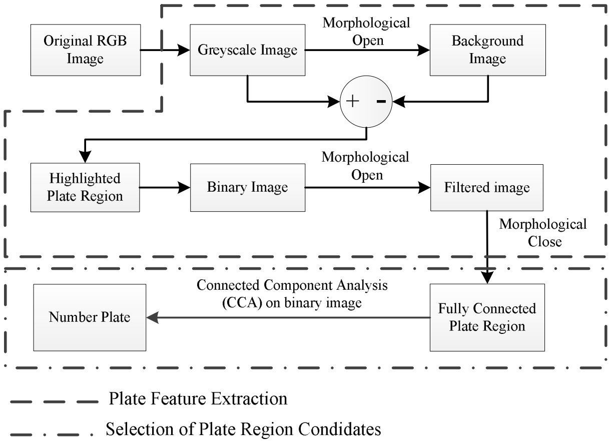

Figure 1. Block diagram of NPL system

The proposed NP feature extraction algorithm can be summarised in the following pseudocode.

**Proposed algorithm: NP feature extraction**

```
Input image: colour car image
Output image: Highlighted NP region image
for all pixels in the input image do
grayscale pixels = RGB2Gray(original colour pixels);
Shifting the pixels into 3×30 SE;
background pixels = open(SE);
highlighted NP pixels =  grayscale pixels -
background pixels
end
```

Compare to the existing algorithms, the proposed algorithm uses a morphological open operation and image subtraction to replace the edge detection operator. As morphological open operation has less computational intensity than the edge detection operator, the processing speed of the proposed system has been significantly improved. In addition to the above, with a specially designed Structuring Element (SE), the proposed algorithm could extract more accurate NP features rather than focus only on common edge features, which should improve the ability of proposed algorithm for tolerating the noise.

The output image from the previous stage consists of a set of groups of connected pixels. A Connected Component Analysis (CCA) labelling algorithm is used to mark these pixels. A set of potential candidates can be selected from the image using the known geometrical conditions, which mainly consists of the width, height and ratio of the plate region. More details about the proposed NPL module are presented in [8].

### B. CS Module

The proposed CS algorithm is mainly based on pixel projection and morphological operations. Compare to existing works based on pixel projection method [9-11], two optional morphological operations have been introduced in the proposed improved algorithm to minimise the impact of noise and the entire horizontal pixel projection step has been replaced by an NP height optimisation step. These modifications improve the robustness of the vertical projection and also accelerate processing speed.

The proposed method has three stages: 1) Pre-projection stage, 2) Vertical projection, and 3) Horizontal projection. Figure 2 shows the block diagram of the proposed CS module.

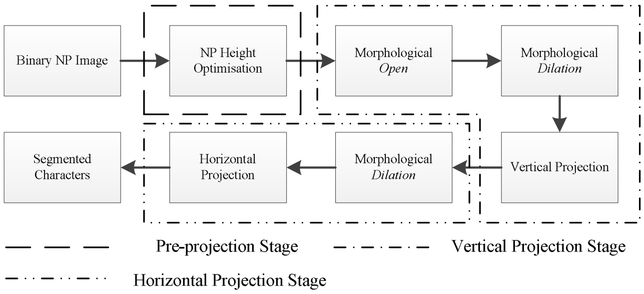

Figure 2. Block diagram of the proposed character segmentation module

The input binary NP images are the outputs of the NPL stage. All images from the NPL stage are binarised and inclined images must be roughly rotated before they are fed to the character segmentation stage. At the beginning, the unnecessary parts of NP are firstly removed by an NP height optimisation step in the pre-projection stage, and then, the morphological *open* and *dilation* are used to eliminate the noise impact on the NP followed by the vertical projection process. After obtaining the vertical positions of the characters, the horizontal projection stage is introduced to localise the horizontal positions of the characters.

The proposed CS algorithm can be summarised in the following pseudocode.

**Proposed algorithm: CS algorithm**

```
Input images: localised NP image ($a×b$), where $a$ is
height of NP, b is width of NP
Output images: Segmented character image
if (a>26 and b/a<7) then
    reducing height of NP $a$ to $0.7a$
    morphological open input image using 3×1 SE
    morphological dilation after the open
else
     morphological dilation for the input image
end
 obtain highlighted NP image
for all pixels in the highlighted NP image do
    generating vertical projection histogram
     finding the vertical critical points between two
     characters
     generating horizontal projection histogram for each
     vertical cropped character image
    finding the horizontal critical points for each character
end
```

Compare to the existing algorithms, two optional morphological operations have been introduced in the proposed CS algorithm to eliminate noise impact and the entire horizontal pixel projection step has been replaced by an NP height optimisation step. These modifications improve the robustness of the CS algorithm and also accelerate processing speed. More details about the proposed CS module can be found in [12].

### C. *OCR Module*

The proposed OCR algorithm uses a multi-layer feed-forward Neural Network (NN) to translate scanned character images into machine encoded text. Typically, an N-layer NN consists of a set of input vectors, N-1 hidden layers, one output layer and a set of output vectors. Each layer consists of a set of neurons and corresponding transfer function (e.g. sigmoid, linear) [13]. Figure 3 shows a two-layer feed-forward network with one hidden layer and one output layer.

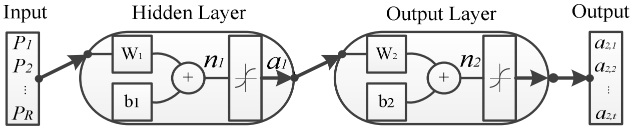

Figure 3. The architecture of two-layer feed-forward network

The most commonly used images in OCR are binary images. They are also used in other stages in an ANPR system. Binary images require less computational intensity compare to other types of images which significantly decreases the computation for real-time applications such as ANPR. In the proposed work, a 2-D binary image matrix $w \times l$ is transformed into 1-D vector **p** with $R$ elements $[p_1, p_2, ..., p_R]^T$ to be used to form the inputs of NN, where pixels of binary image are read row by row to form the **p**. Due to only having 33 possible characters, the output of NN has been decided as $a_2 = [a_{2,1}, a_{2,2}, ..., a_{2,33}]^T$. In order to find the most suitable NN architecture for OCR task, the different number of neurons $S$ and size of input vector **p** are used to create different NNs.

Because the training speed of Scaled Conjugate Gradient (SCG) is much faster than traditional Back Propagation algorithm (BP), and also it gives better results compared to other training methods, the SCG is chosen as the training method of the NN. Before the start of training, the NN was initialised using Nguyen-Widrow initialisation algorithm [14], where the weights and biases in each layer are initialised and distributed approximately evenly over the input space. More details about the proposed algorithm and its hardware implementation can be found in [15].

Based on experimental results, the size of the input binary character image was decided to be $34 \times 22$. The entire character database has been divided into two non-overlapping groups: the first group has 45% of characters, which is used only for neural network training, and the second group has 55% of characters, which is used only for neural network testing. The number of neurons used in the NN was 50 for obtaining the best results in terms of performance and accuracy.

## III. FPGA Implementation and Results

The whole ANPR system has been simulated using the PAL Virtual Platform (PALSim) [16]. After simulation, the system has been successfully implemented and verified using the Mentor Graphics RC240 FPGA development board [16]. The ANPR modules run in parallel and pipelining has also been used in their implementation to achieve high throughput rate.

Handel-C has been used for hardware description of the proposed architectures, which is a high-level language that is at the heart of a hardware compilation system known as the Mentor Graphic Development Kit (DK) [17], which is designed to compile programmes written in a C-like high level language into synchronous hardware. The main advantages of Handel-C over the other hardware description languages are the rapid prototyping and the software liked simulator.

Due to the low complexity of each ANPR module, the proposed ANPR system requires only 80% of the on-chip FPGA slices. From these slices, 61% LUTs are used to implement logic operations and RAMs in the design. 23% flip-flops are mainly used as registers to buffer the data for enabling the high throughput pipeline manner in the design. 31% BRAMs are mainly used to buffer the image pixels. The DSP48s slices are used to perform the arithmetic calculations in each module. The remaining 20% of the FPGA area can be used for implementing the communication and display units. Table I summarises the consumed on-chip resources.

Table I: Usage of FPGA on-chip Resources

| On-chip resources | Used | Available | Utilisation |
|---|---|---|---|
| Occupied Slices | 14,775 | 18,432 | 80% |
| LUTs | 22,556 | 36,864 | 61% |
| Flip-Flops | 8,547 | 36,864 | 23% |
| Block Rams | 30 | 96 | 31% |
| DSP48s | 12 | 64 | 18% |

The maximum running frequency is 57.6 MHz and the number of clock cycles needed for one image to be processed is between 506686-683278, which depends on the number of the characters within the NP image. The execution time for processing one frame is $11ms$. This means that the proposed architecture satisfies the minimum requirement for real-time processing. The used database contains 1000 images with a resolution of 640×480 UK NPs [8].

The power consumption of the designed circuit has also been analysed using Xilinx XPower Analyser [18], the total power consumption of the proposed architectures is 910 mW, which is comprised of 459 mW dynamic power and 451 mw quiescent power. The total power consumption is very low compared to computer-based ANPR systems.

The successful segmentation/recognition rate for NPL, CS and OCR rates are 97.8%, 97.7% and 97.3%, respectively, and the overall system accuracy is around 93.0%.

A comparison of the experimental computational speed and successful rate with existing PC, DSP and FPGA based implementations of ANPR system is shown in Table II.

Table II: Performance Comparison

| ANPR System | Character Set | Hardware Platform | Successful Rate (%) | Speed (*ms*) |
|---|---|---|---|---|
| [5] | Australia | TI C64 DSP | 85 | 52.11 |
| [19] | Turkey | FPGA Virtex-4 | 73 | 500 |
| [20] | Japan | PC Intel Core 1.8 GHz | 93.54 | 284 |
| [21] | Chinese | PC 3 GHz | 93.9 | 293 |
| Proposed System | UK | FPGA Virtex-4 | 93.0 | 11 |

Results achieved in terms of computational speed and successful rate in comparison with existing PC [20] [21], DSP[5] and FPGA[19] based implementations have shown that the proposed FPGA based ANPR system outperforms the fastest software and hardware based ANPR systems by a factor of 26 and 4.7 respectively, it also outperforms the existing hardware solutions in terms of accuracy. Although the recognition rate of the proposed system is close to that of some PC-based systems, it presents an advantage over software-based solutions in terms of cost, size and energy consumption.

## IV. Conclusion

In this paper, a complete solution for ANPR system on FPGA is presented. This includes the algorithms and implementations for the NPL, CS and OCR stages. The Mentor Graphics RC240 FPGA development board was used for the implementation and testing of the proposed work. The entire system consumes only 80% of the available on-chip slices of a Virtex-4 FPGA, runs with a maximum frequency of 57.6 MHz and is capable of processing one image in 11*ms* with a successful recognition rate of 93%.

The achieved results have shown that the entire ANPR system can be implemented on a single FPGA chip, which can be placed within an ANPR camera housing to create a stand-alone unit which will drastically improve energy efficiency and remove the installation and cabling costs of bulky PCs situated in expensive, cooled, waterproof roadside cabinets.

## Acknowledgment

Xiaojun Zhai's and Klaus McDonald-Maier's work have partially been supported by the UK Engineering and Physical Sciences Research Council under grant EP/K004638/1 and the EU Interreg IV A 2 Mers Seas Zeeën Cross-border Cooperation Programme – SYSIASS project: Autonomous and Intelligent Healthcare System (project's website http://www.sysiass.eu/).

## References

[1] R. Gurney, M. Rhead, S. Ramalingam, and N. Cohen, "Working towards an International ANPR Standard: an initial investigation into the UK Standard," in *46th IEEE International Carnahan Conference on Security Technology*, USA, 2012, pp. 331-337.

[2] S. Connor. (2005, Oct). *Surveillance UK: why this revolution is only the start*. Available: http://www.independent.co.uk/news/science/surveillance-uk-why-this-revolution-is-only-the-start-520396.html (Accessed on Oct, 2012)

[3] A. O. Yerdut, Y. B. Eldeniz, and H. G. Ilk, "Automatic license plate recognition based on a projection method," in *IEEE 19th Conference on Signal Processing and Communications Applications (SIU)*, 2011, pp. 182-185.

[4] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos and E. Kayafas "License plate recognition from still images and video sequences: A survey," *IEEE Transaction Intelligent Transportation System* vol. 9, pp. 377-391, 2008.

[5] C. Arth, F. Limberger, and H. Bischof, "Real-Time License Plate Recognition on an Embedded DSP-Platform," presented at the IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[6] C. Arth, C. Leistner and H.Bischof, "TRIcam: an embedded platform for remote traffic surveillance," in *Proceedings of IEEE Computer Vision and Pattern Recognition Conference*, 2006, pp. 125-134.

[7] T. Kanamori, H. Amano, M. Arai, D. Konno, T. Nanba, and Y. Ajioka, "Implementation and Evaluation of a High Speed License Plate Recognition System on an FPGA," in *7th International Conference on Computer and Information Technology*, 2007, pp. 567-572.

[8] X. Zhai, F. Bensaali, and S. Ramalingam, "Improved Number Plate Localisation Algorithm and its Efficient FPGA Implementation," *IET Circuits, Devices & Systems,* Vol. 7, (2), pp. 93 – 103, June, 2013.

[9] S. Chang, Chen, L., Chung, Y. and Chen, S., "Automatic license plate recognition," *IEEE Transaction on Intelligent Transpotation Systerms,* vol. 5, pp. 42-53, 2004.

[10] W. Tsang-Hong, N. Feng-Chou, L. Keh-Tsong, and C. Yon-Ping, "Robust license plate recognition based on dynamic projection warping," in *IEEE International Conference on Networking, Sensing and Control*, 2004, pp. 784-788.

[11] H. Xiangjian, Z. Lihong, W. Qiang, J. Wenjing, B. Samali, and M. Palaniswami, "Segmentation of characters on car license plates," in *IEEE 10th Workshop on Multimedia Signal Processing*, 2008, pp. 399-402.

[12] X. Zhai and F. Bensaali, "Improved Number Plate Character Segmentation Algorithm and its Efficient FPGA Implementation," *Journal of Real-Time Image Processing*, pp. 1-11, 2012.

[13] H. Demuth, M. Beale and M. Hagan, *Neural Network Toolbox 6 User's Guide*: The MathWorks, Inc. , 2008.

[14] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," in *International Joint Conference on Neural Networks*, 1990, pp. 21-26.

[15] X. Zhai, F. Bensaali, and R. Sotudeh, "Real-Time Optical Character Recognition on FPGA for ANPR," *IET Circuits, Devices and Systems,* August, pp. 1-11, 2013.

[16] Mentor Graphics Corporation.. *PAL User Manual*. Available: http://www.mentor.com/ (Accessed on June, 2011)

[17] Mentor Graphics Corporation. *Handel-C User Manual*. Available: http://www.mentor.com/ (Accessed on June, 2011)

[18] Xilinx, Inc.. *Xpower Tutorial: FPGA Design*. Available: http://www.xilinx.com/ (Accessed on June, 2011)

[19] Y. Wen, Y. Lu, J. Yan, Z. Zhou, von Deneen K.M. and P. Shi, "An Algorithm for License Plate Recognition Applied to Intelligent Transportation System," *IEEE Transactions on Intelligent Transportation Systems,* vol. 12, pp. 830-845, 2011.

[20] Y.-P. Huang, C.-H. Chen, Y.-T. Chang, and F. E. Sandnes, "An intelligent strategy for checking the annual inspection status of motorcycles based on license plate recognition," *Expert Systems with Applications,* vol. 36, pp. 9260-9267, 2009.

[21] H. Caner, H. S. Gecim, and A. Z. Alkar, "Efficient Embedded Neural-Network-Based License Plate Recognition System," *IEEE Transactions on Vehicular Technology,* vol. 57, pp. 2675-2683, 2008.