**Computer Vision-Based** **Vehicle Integrated**

**Information Collection System**



Shangyuan Liu

25344136

[25344136@students.lincoln.ac.uk](mailto:25344136@students.lincoln.ac.uk)

School of Computer Science

College of Science

University of Lincoln

Submitted in partial fulfillment of the requirements for the

Degree of BSc(Hons) Computer Science

*Supervisor:* Dr. Shouyong Jiang

April 2021

**Acknowledgements**

Firstly, I want to thank somebody, and somebody else. Here is another thing.

**Abstract**

摘要在其整体上呈现了项目的简要摘要，并用来帮助读者快速确定项目的目的，上下文和结果。 摘要的目的是让读者在不需要阅读整个文档的情况下就能了解报告的内容，并帮助标记报告的内容。它还通常包含项目的关键结果是什么的简要说明。

Vehicle detection and vehicle license plate recognition (VLPR) is an indispensable component of an intelligent traffic management system, with the quick development of the information technique. Therefore, this project's purpose is to investigate and analyse the various fundamental theories about VLPR technology, which will be used in developing a computer vision-based vehicle integrated information collection system. The issue of accuracy and validity in complex light environments, low resolution, and random distribution has received considerable critical attention.

**Table of Contents**

[Introduction 1](#_Toc58407584)

[1.1 Some notes 1](#_Toc58407585)

[1.2 Testing some mathematics 1](#_Toc58407586)

[1.3 Undergraduate Project Report 1](#_Toc58407587)

[1.4 Referencing 1](#_Toc58407588)

[1.4.1 Ludography 2](#_Toc58407589)

[Literature Review 3](#_Toc58407590)

[2.1 Background 3](#_Toc58407591)

[2.2 Related Literature 3](#_Toc58407592)

[Methodology 4](#_Toc58407593)

[3.1 Project Management 5](#_Toc58407594)

[3.2 Software Development 5](#_Toc58407595)

[3.3 Toolsets and Machine Environments 5](#_Toc58407596)

[3.4 Research Methods 6](#_Toc58407597)

[Design, Development and Evaluation 7](#_Toc58407598)

[4.1 Software Development Projects 7](#_Toc58407599)

[4.2 Research Projects 8](#_Toc58407600)

[Conclusions 9](#_Toc58407601)

[Reflective Analysis 10](#_Toc58407602)

[References 11](#_Toc58407603)

**List of Figures**

3.1 A picture of the Brayford from Google Images. . . . . . . . . . . . . . 5

**List of Tables**

1. Here is a table. . . . . . . . . . . . 5

**Chapter 1**

# Introduction

## 1.1 Some notes

It is worth noting that this document is a project report template for the University of Lincoln, School of Computer Science. It should give you some direction and instruction for formatting and presenting your project report. If you have any suggestions or issues, please contact mdoughty@lincoln.ac.uk. It has been derived from the Latex PDF however, so there might be some issues – but ones I suspect you can overcome!

## 1.2 Testing some mathematics

Here are two equations using the equation editor (1, 2):

(1)

(2)

And here is some text with some nice inline maths, (*x,y*) wow *γ* so cool *ρ*.

## 1.3 Undergraduate Project Report

Currently, this template is set up for use with undergraduate project reports. However, the template can be modified fairly easily to conform to, for example, a MComp project report.

**Chapter 2**

# Literature Review

## 2.1 Background

In recent years, intelligent vehicle information monitoring systems have played a positive role in the management and safety of society. In addition, with the rapid development of computer vision technology, some cutting-edge applications in the automotive field have gained widespread popularity, such as VLPR, vehicle tracking, and autonomous driving. Take the VLPR system as an example, the previously published research on this field is circumscribed and problematic, with most of the literature focusing on the use of image processing and Optical Character Recognition (OCR) and other traditional methods. License plate positioning is a prerequisite for the implementation of license plate character recognition, however, Maglad’s (2012) analysis does not take account of the issue of license plate types adequately, nor does he examines how to accurately locate and segment the license plate area under the conditions of poor lighting background, blurred photos, and skewed angles. It is necessary to emphasize that the accuracy of OCR technology depends on the quality and quantity of the original character template in the database. As well as that, the identification speed of OCR should be improved as a real-time system.

Fig. 1: The license plate images under bad circumstances from Google



As is shown in the sample images Fig. 1, where it can be seen the negative influences of the above threats clearly. Therefore, I intended to locate the license plate, identify the license number, and other valid information more accurately via a new approach, which is combining convolutional neural network (CNN), U-Net (image segmentation), and OpenCV (cv2) technical etc. - to implement high-precision recognition of license plate and vehicle information such as plate number, brand, model, colour. In comparison to the traditional method of image processing that the proposed improvement approach (U-Net, CNN) for the VLPR system is accurate and robust. (Chowdhury et al., 2020). So in the following project work, it is necessary to compare the traditional identification methods with the new improved methods and to draw conclusions.

## 2.2 Related Literature

This section presents a detailed overview of the literature that relates to this project. The literature reviews will be conducted in two thematic directions, which are vehicle colour recognition, license plate location and recognition respectively. Due to the project focused features, this section will place emphasis on the second thematic direction.

### 2.2.1 Vehicle Colour Recognition

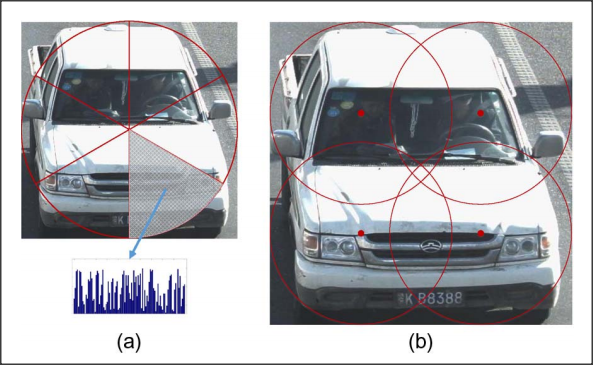
The traditional approach is to convert the image in RGB colour space to HSV space and determine which colour family the car in a given image most likely belongs to base on the Hue, Saturation, and Value (H, S, V) ranges of the nine common colours. [In](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) [allusion](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) [to](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) this question, Chen et al. (2014) has adopted a different approach to solve this issue based on the framework of Bag-of-Word (BoW) method, which has been put forward by [Zellig Harris](https://en.wikipedia.org/wiki/Zellig_Harris) (1954). Chen proposed methods is to recognize the dominant colour of vehicles via the region of interest (ROI) and separate the objects into subregions, and generate a histogram for each subregion, and use a linear SVM model to learn it. As can be seen from the Fig. 2, the feature context (FC) method divides the vehicle image into a number of sector sub-regions.

Fig. 2. FC Method to process the image (Pan Chen)

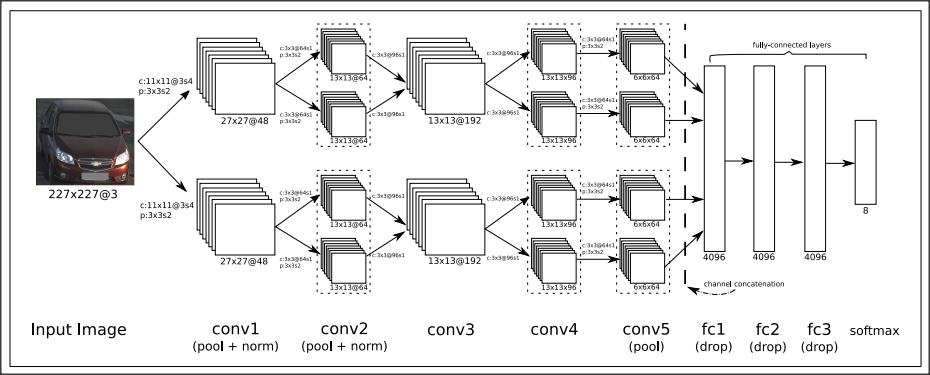
The research of Chen has come up with a good solution, however, several practical questions arise when dealing with multiple vehicles in one image. Particularly, the image gets excessively divided as the amount of subregions expands, so the discriminative capacity of the descriptor decreases. As a consequence, Rachmadi and Purnama (2018) had improved the vehicle colour recognition project based on Chen's research. They suggest a CNN-based method (the architecture image can be seen in Fig. 3) to recognize vehicle colour, the final recognition accuracy was 94.47%, and the CNN model's predictions were 2% more accurate than the results of Chen's study.

Fig. 3: The architecture of CNN (Reza Fuad Rachmadi)

### 2.2.2 License Plate Location and Recognition

It is fair to say that the VLPR technology has developed along with the maturity of artificial intelligence technology, and is a separate branch from the OCR system. In terms of OCR, it was invented in the 1950s and has been criticized and ignored because of its low recognition accuracy and slow speed. And Islam, N (2016) indicated that “the earliest OCR systems were not computers but mechanical devices that were able to recognize characters, but very slow speed and low accuracy”.

This system became commercially available in the late 1990s due to the demands of traffic management. And a systematic review of prospective observational studies found that the result offered by Jun-Wei Hsieh (2002) in [Morphology-based License Plate Detection from Complex Scenes] demonstrate that the robust approach to this issue has been proposed. And there was a great effort has been devoted to the study of multiple morphological operations, which used to find high-contrast areas as important features for license plate detection. Also,

V. Kocal (2003) indicated that using image fusion, neural networks, and iterative thresholding operation to identify license plate of vehicle. However, most of the previous studies do not take into account the future social demand and the rapid development of the automotive industry. Due to limited by the capabilities of conventional algorithms, and the low level of camera imaging, etc., so VLPR was not ready for widespread use compared to the current stage of technology. Therefore, it is concluded that these approaches may not be practical in all situations by means of literature review.

During the past twenty years, [many](D:/Dict/8.9.6.0/resultui/html/index.html" \l "/javascript:;) [investigations](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) have appeared in recent years documenting VLPR with the rise and technical maturity of machine learning and deep learning techniques. As a result, the previous research has demonstrated an algorithm that license plate recognition using a binary time-delay neural network (TDNN), which is a classification used for training the neural network. The TDNN architecture diagram is shown in Figure 4. According to the research from Ghasrodashti and Yazdi (2020) has shown that “hard conditions such as the distance and angle variations as well as weather and light conditions are considered, and the recognition rate obtained by the proposed algorithm was 70%.” Moreover, in order to prove that the algorithm can contribute to better results, the proposed algorithm (TDNN) is also compared with the previous method (SVM).

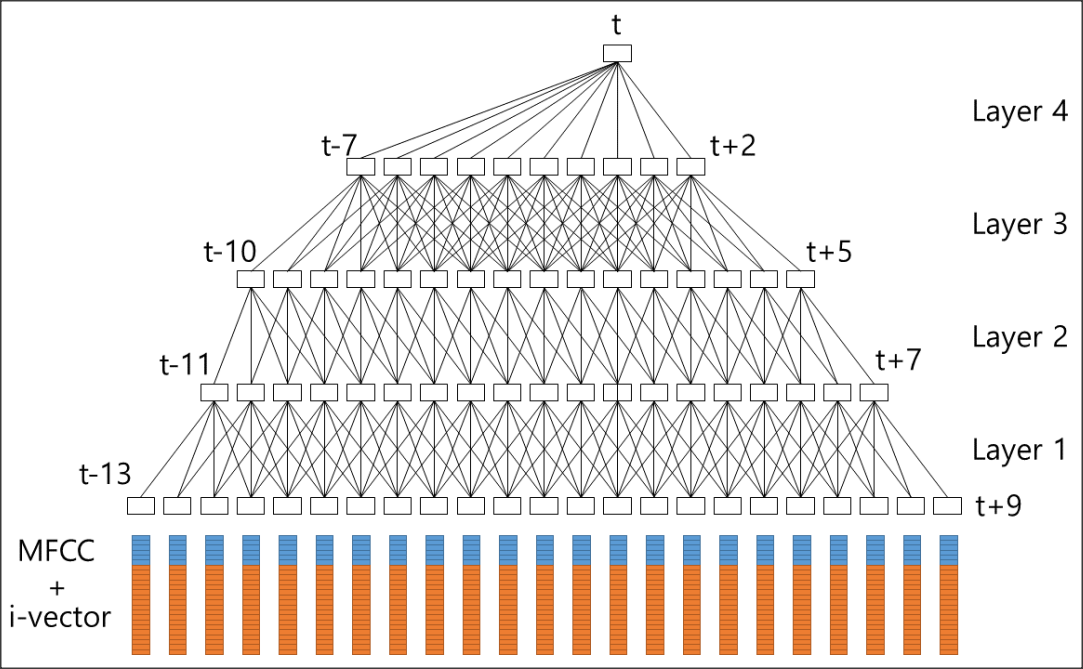


Fig. 4: TDNN architecture (Park, H., Lee, D., Lim, M., Kang)

At present, more research is invested in deep learning in the field of VLPR, especially the end-to-end training network method is extensively used. Glasmachers (2017) has shown that “it is trained in a holistic manner based on a single principle”. The overall architecture of the proposed end-to-end principle is shown in Fig. 5. And it approaches complicated problems by using the Deep Neural Networks (DNNs) model, which is made up of many layers. To end this, Qin and Liu (2020) proposed a strategy for VLPR, which is based on deep neural networks and end-to-end methods. And according to the research, it can decrease the error propagation effectively between the models of location and recognition network by means of common features, convolutional neural network, and multi-task learning strategy. Therefore, this basic structure and theoretical knowledge can be used as the basis for this project.

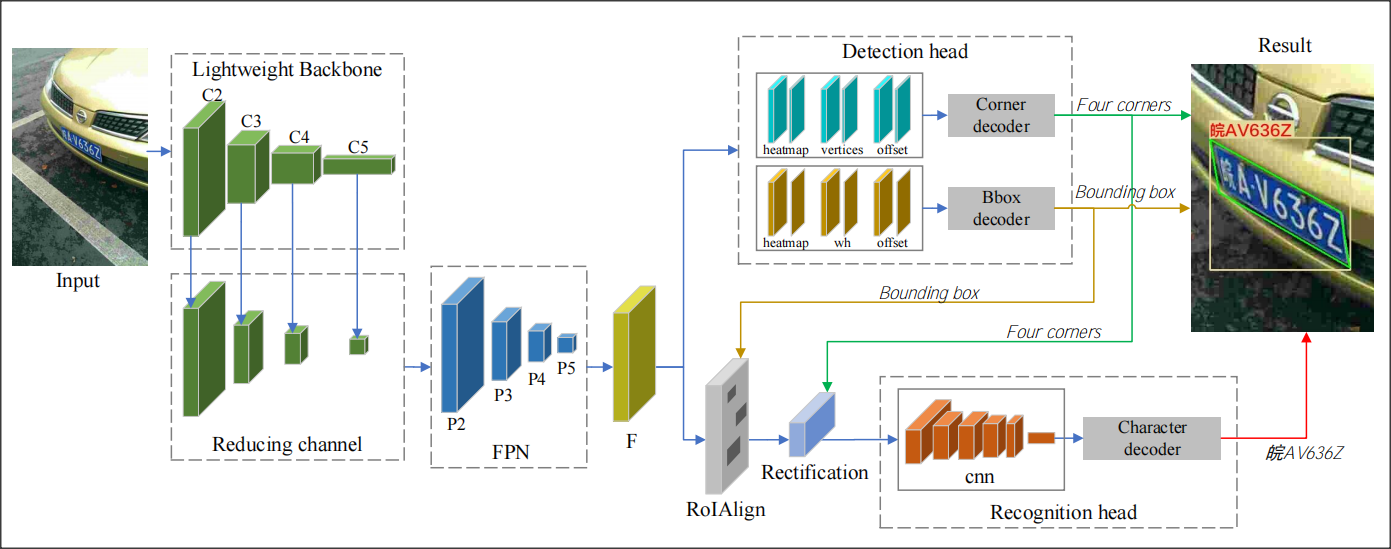


Fig. 5 The end-to-end network structure model for license plate location and recognition (Qin)

However, the end-to-end method is still under debate, and I believe it is not suitable and circumscribed in some cases. This discrepancy could be attributed to the following several reasons. Firstly, a large number of data samples are required to train the model. Secondly, it is difficult to optimize or modify the model system because the DNN model parameters must be replaced and retrained. Moreover, the method is difficult to interpret and validate due to the complicated neural network architecture, which point is significant for the automotive industry particularly. Therefore, the study by Glasmachers (2017) provided that “We have demonstrated that end-to-end learning can be very inefficient for training neural network models composed of multiple non-trivial modules. End-to-end learning can even break down entirely; in the worst-case none of the modules manages to learn. In contrast, each module is able to learn if the other modules are already trained and their weights frozen.”

To sum up, the technologies in this area generally concerns themselves with image processing and computer vision, also it can be divided in to more directions for development. [Although](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) some results would seem to suggest that the current VLPR technology has some limitations, it is against this background that license plate recognition technology, which focuses on non-traffic areas, is gaining more and more attention.

**Chapter 3**

# Methodology

This section will cover a number of aspects of your project where appropriate. **Not all projects will require every section though**. The key thing is that you demonstrate critical awareness of all of the processes that you have employed in your work and that for all sections needed in your report you are presenting a justification for the methods you adopted and not just presenting a list of methods.

## 3.1 Project Management

Some awareness of project management should be demonstrated in all projects. This section should outline the nature of your project and the specific characteristics that need to be considered in determining what project management methodology you should use. You should identify the specific demands of your project in terms of project management and support your rationale for the selection of a methodology with appropriate and recent academic references. Questions which may be relevant here are:

1. What are the guiding principles and processes in managing your project?
2. What project management methods may be useful for this project?
3. How can you exploit their advantages for your project and mitigate their drawbacks?

## 3.2 Software Development

There should be a methodological analysis of software development approaches used in your project. It is important to note that what is NOT required here is a pedestrian account of popular software development methodologies or a simplistic review of their strengths and weaknesses.

Where relevant, you should give serious thought to the proper design of research and requirements capture approaches. This may include surveys, questionnaires and interviews.

## 3.3 Tool-sets and Machine Environments

### 3.3.1 Tools-Sets

In this chapter, an overview and evaluation of the various development tools and software used in the development and management of this project are presented. In terms of project development, I chose Python as the programming language and Visual Studio Code (VS Code) is the integrated development environment (IDE).

**Python (3.9.2):** The vehicle integrated information collection system project is based on computer vision (CV) technology, and techniques from other disciplines related to CV are also used in the project, such as image processing, machine learning, deep learning, etc. The main programming languages currently used in the computer vision area include C++, and Python, they are compared and analyzed in Table 1. The benefit of developing in Python is that it has an excellent Python developer community, where developers can find a wealth of content about these libraries with a simple search. (Sarkar et al., 2018). Therefore, the Python 3.9.2 was finally chosen to develop the project.

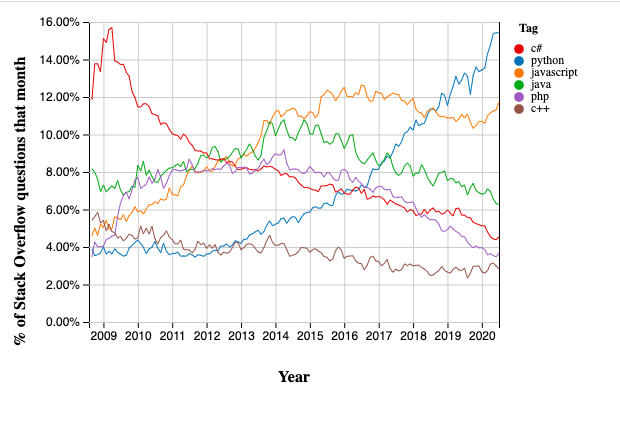


Table 1: The latest annual ranking of popular programming languages by IEEE Spectrum

Table....

|  |  |
| --- | --- |
| **The Comparison of Programming Languages** | |
| **Language** | **Summary of each Programming Language** |
| **Python** | As can be seen from Figure 1, Python is currently the most used and popular programming language, also in the field of machine learning and deep learning, it is currently the most widely used language, so there are many open source projects and courses for learning. Especially for image processing, OpenCV supports a Python interface; for scientific computing, Anaconda also provides a rich package interface. On the other hand, Python has a powerful and effective machine learning library - scikit-learn, and exhaustive resource (numpy, matplotlib).  However, Python is not suitable for embedded development because of inefficient execution. When using a large dataset to train a deep network model, the running speed is very slow and takes up too much memory for too long to do other work. |
| **C++** | In the field of vision and image processing, C++ favours underlying development, so suitable for larger projects and code execution with high efficiency. C++ is similar to Python, also supports interfaces to third-party tools such as OpenGL、EmguCV、OpenCV etc. But the inefficient development that is only suitable for commercial and industrial projects, does not apply to the development of this project. |

**Visual Studio Code:** Python has a variety of integrated development environment (IDE) for professional development, such as PyCharm, Visual Studio Code, and Vim etc. It is indicated in Table 2. That the differences between each software. According to the current computer configuration and project requirements, I have chosen VS Code as the IDE for my project development.

Table....

|  |  |
| --- | --- |
| **The Comparison of Integrated Development Environment** | |
| **Language** | **Summary of each IDE** |
| **PyCharm** | For PyCharm IDE, one of the most widely used IDEs for the Python programming language. It has many functions to help users improve their productivity when developing in the Python language, such as debugging, syntax highlighting, Project management, smart hints, unit testing, and version control. However, for a more full-featured IDE and to support web development you need to use the professional version of Pycharm, which is available for a fee. On the downside, PyCharm takes up a lot of RAM on your computer and is not ideal for low configuration computers. |
| **VS Code** | T[he](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) [other](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) [one](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) [is](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;) VS Code, which is a free open source IDE created by Microsoft for development in Python and other programming languages, with a rich ecosystem of plugins that can be easily searched for, installed and managed directly in the VS Code editor. Moreover, VS Code is more open source than sublime, faster than the Atom and more lightweight than Webstorm. |

**OpenCV (4.5.1):** A significant cross-platform computer vision library- OpenCV (Open Source Computer Vision Library) was also used in the development process. It can be used to develop real-time image processing, computer vision and pattern recognition programs because a variety of basic image processing functions and vision algorithms are encapsulated in the library. More importantly, OpenCV is open source and provides an interface to Python for project development efficiently.

**Tensorflow (2.5.0):** It is quite important to choose suitable Python Libraries for deep learning project because a suitable framework can play a positive role in the project. At [present](D:/Dict/8.9.6.0/resultui/html/index.html#/javascript:;), there are five most popular machine learning libraries, which respectively are TensorFlow, Keras, PyTorch, Caffe2, Scikit-learn. The two most mainstream frameworks, TensorFlow and PyTorch are compared and analysed in Table. In TensorFlow 2.0, Keras is the official high-level API for TensorFlow, allowing for fast and easy model design and training. And the Caffe2 incorporates all of its code into the PyTorch framework..

Although PyTorch is easier for beginners to learn and master, and I have never used any machine learning framework before, I have chosen TensorFlow as the machine learning framework for this project because it supports not only deep learning but also reinforcement learning and other algorithms. In addition, it has complete official documents and learning materials to better support the development of this project.

|  |  |
| --- | --- |
| **The Comparison of Machine Learning Frameworks** | |
| **Language** | **Summary of each framework** |
| **TensorFlow** | [TensorFlow](https://www.tensorflow.org/) is an end-to-end open source platform for machine learning that it performs regression, classification, neural networks model, etc., and [TensorFlow](https://www.tensorflow.org/) can run on both CPU and GPU. It is clear from the Fig. That TensorFlow is the most widely used machine learning framework. Because it plays a vital role in the fields of deep neural networks, natural language processing (NLP), image, text, and speech recognition. However, it also has some drawbacks, when running the program it uses up all of the GPU's memory by default, resulting in a slowdown. Since TensorFlow computational graphs are written in Python, large neural network models need to take a long time to compile. |
| **PyTorch** | PyTorch is a rival to TensorFlow that was created by Facebook AI Research. It has the advantage of supporting dynamic computational graph models, also known as define-by-run. Also PyTorch is easy to learn because of the large number of pre-trained models. |

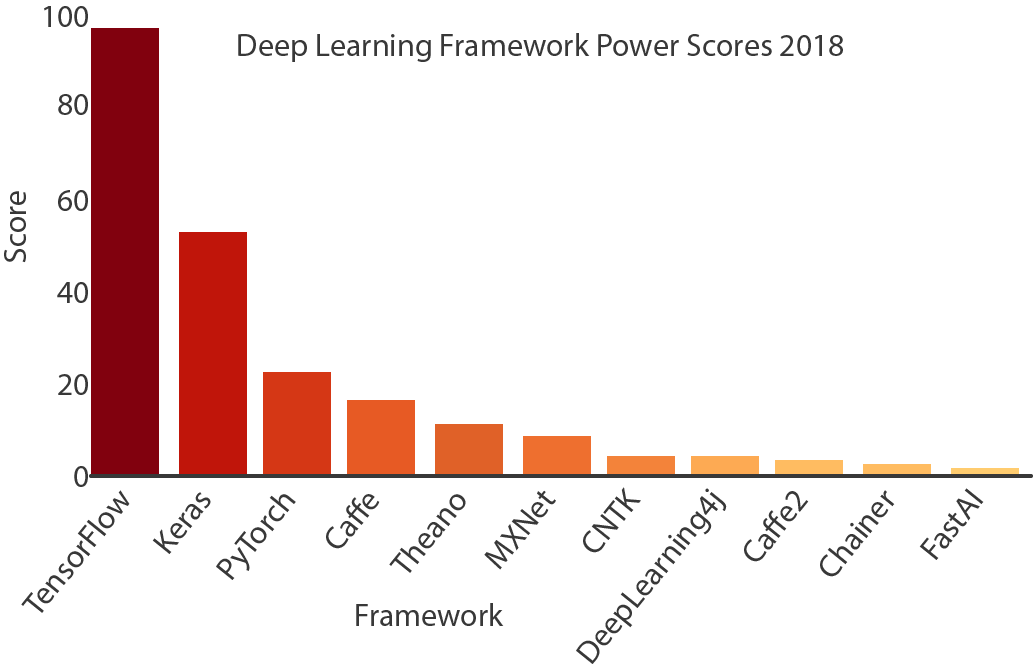


Fig. Top 5 Deep Learning Frameworks to Watch in 2018 - Towards data science

**Labelme:** Labelme is an image annotation tool developed by Massachusetts Institute of Technology (MIT), which can be used to create custom annotation tasks or perform image annotation. The main use in this project is for labelling semantic segmentation datasets of license plates. The image data is annotated in the project using lableme to generate the .json file corresponding to the image. And the reason why I chose this is that can execute the built-in function labelme\_json\_to\_dataset under Labelme to batch convert images to the corresponding x\_json file.

**Project Management - GIT:** GIT is a free and open source distributed version control system (VCS), which can handle software project management efficiently. A similar VCS to GIT is Apache Subversion (SVN), the main difference being that GIT is distributed, while SVN is not. And I selected GIT to manage the development of the project for several reasons. Blischak et al (2016) also indicated the same point that “By tracking your code development with a VCS and hosting it online, you are performing science that is more transparent, reproducible, and open to collaboration”.

* Repository localisation, offline commit support, relative independence without impacting collaborative development.
* GIT only generates a .git directory for each project, where all the project's version control information is located, unlike SVN which generates .svn directories under each directory.
* Supports quick branch switching for easy merging, and merges files faster than SVN.
* GIT's content storage uses the SHA-1 hash algorithm. This ensures the integrity of the code content.
* The ability to upload project code remotely to the GitHub platform, allowing efficient development in different environments and platforms

**3.3.2 Machine Environments**

本节将概述用于软件开发和项目管理过程的工具； 它将在可用的工具之间进行适当的比较，并根据度量标准（可能是矩阵图和其他标准）主张最合适的选择。 不要仅仅因为您对特定工具集和环境有充分的了解或已经掌握了技能，就为使用特定工具集和环境提供理由。

工具集既指软件开发和项目管理，所以覆盖范围应该解决这两个问题。本节将概述软件开发和项目管理过程的工具; 它将对可用的工具进行适当的比较，并根据指标、可能是矩阵图和其他标准进行最合适的选择。

该报告将讨论可能需要对人工制品进行操作的可能的机器环境，并通过分析、比较特性和可能的用户需求来确定所选择的环境。您应该在软件开发方面确定您的项目的具体需求，并使用适当的、最近的学术参考资料来支持您选择一种方法的理由。

## 3.4 Research Methods

You should investigate the types of research methods necessary to validly answer the research questions that your project addresses. You should cite relevant sources to justify your choices.

Interpretation overlooks much of the

much of the research up to now has been descriptive in nature

Consideration for the issue is presented in the second part of the report

Table 1: The latest annual ranking of popular programming languages by IEEE Spectrum

I have a other question about word count limited of each section, Is there a recommended word limit for each section?

**Chapter 4**

# Design, Development and Evaluation

This section of the report will vary significantly in both structure and content, depending on the type of project you are undertaking. For example, a Games design project may include a Game Design Document. However, it must be noted that if your project contains significant software development work, this should be presented in the structure expected of a formal development report. If your project involves an experimental evaluation – especially if that evaluation involved human participants – you are expected to write this work up in the format expected of a scientific research

report.

## 4.1 Software Development Projects

Include this section if you are undertaking a software development project. You should discuss:

1. Requirements elicitation, gathering, collection and analysis
2. Design
3. Building and programming
4. Testing
5. Operation

## 4.2 Research Projects

If your project includes primary research components it is expected that you present this work in a manner appropriate to a scientific report:

1. Participant recruitment
2. Evidence that ethical procedures have been followed
3. Study design (short summary of research methods section) – including hypotheses/research question as appropriate
4. A detailed description of the procedure
5. Results of experiment
6. Analysis of results. Consider the results of your work with respect to both your own specific hypotheses/research question and wider context identified in your literature review.

**Chapter 5**

# Conclusions

The results from this project indicate that ...

**Chapter 6**

# Reflective Analysis

The project went well ...

# References

Ascar, Davix.X., Judson, D. and Pittala, S.K. (2020) ‘SegNet Approach for Vehicle License Plate Localization.’ In: *2020 Seventh International Conference on Information Technology Trends (ITT)*, pp. 113-117, doi: 10.1109/ITT51279.2020.9320883.

Blischak, J.D., Davenport, E.R. and Wilson, G. (2016) A Quick Introduction to Version Control with Git and GitHub F. Ouellette (ed.). *PLOS Computational Biology*, 12(1) e1004668. Available from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4718703/> [Accessed 20 April 2021].

Chen, P., Bai, X. and Liu, W. (2014) Vehicle Color Recognition on Urban Road by Feature Context. *IEEE Transactions on Intelligent Transportation Systems*, 15(5) 2340–2346. Available from: <http://cloud.eic.hust.edu.cn:8071/~pchen/06777550.pdf> [Accessed 5 April 2021].

Chowdhury, P.N., Shivakumara, P., Raghavendra, R., Pal, U., Lu, T. and Blumenstein, M. (2020) A New U-Net Based License Plate Enhancement Model in Night and Day Images. *Lecture Notes in Computer Science*, 749–763. Available from: <https://link.springer.com/chapter/10.1007%2F978-3-030-41404-7_53> [Accessed 17 April 2021].

Glasmachers, T. (2017) Limits of End-to-End Learning. *arXiv:1704.08305 [cs, stat]*, Available from: <https://arxiv.org/abs/1704.08305> [Accessed 15 April 2021].

detectRecog (2020) *CCPD (Chinese City Parking Dataset, ECCV)* Available from: <https://github.com/detectRecog/CCPD> [Accessed 8 February 2021].

Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S. and Shet, V. (2014) Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. *arXiv:1312.6082 [cs]*, Available from: <https://arxiv.org/abs/1312.6082> [Accessed 5 April 2021].

Maglad, K. (2012) A Vehicle License Plate Detection and Recognition System. *Journal of Computer Science*, 8(3) 310–315. Available from: <https://core.ac.uk/download/pdf/25846945.pdf> [Accessed 22 December 2020].

Mithe, R., Indalkar, S. and Divekar, N. (2013) Optical character recognition. *Journal of Information & Communication Technology*, 2(1). Available from: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.673.8061&rep=rep1&type=pdf> [Accessed 13 February 2021].

Islam, N., Islam, Z. and Noor, N. (2016) A Survey on Optical Character Recognition System. *ArXiv*, 10(2). Available from: <https://www.semanticscholar.org/paper/A-Survey-on-Optical-Character-Recognition-System-Islam-Islam/6bdb84c02fd56c209c893efda7df083c9ae5c4df> [Accessed 10 February 2021].

Ronneberger, O., Fischer, P. and Brox, T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes in Computer Science*, 234–241. Available from: <https://arxiv.org/abs/1505.04597> [Accessed 16 March 2021].

Xu, Z., Yang, W., Meng, A., Lu, N., Huang, H., Ying, C. and Huang, L. (2018) ‘Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline’.In: *Proceedings of the European conference on computer vision* (ECCV) pp. 255-271.

Qin, S. and Liu, S. (2020) Towards End-to-end Car License Plate Location and Recognition in Unconstrained Scenarios. *arXiv:2008.10916 [cs]*, Available from https://arxiv.org/abs/2008.10916v1 [accessed 16 April 2021].

Siqi, S., Nanting, L., Yanjun, M. and Liping, Z. (2020) Robust Recognition of Truck License Plate in Mine Environment. *IEEE Xplore*, 36–42. Available from: <https://ieeexplore.ieee.org/abstract/document/9361802> [Accessed 19 March 2021].

Zhong, Y., Liu, Y., Luo, F. and Zhang, H. (2020) A Novel Integrated Neural Network for License Plate Detection And Recognition. *2020 Chinese Automation Congress (CAC)*, Available from: <https://ieeexplore.ieee.org/abstract/document/9326612> [Accessed 12 March 2021].

Jun-Wei Hsieh, Shih-Hao Yu, & Yung-Sheng Chen. (2002). Morphology-based license plate detection from complex scenes. *Object Recognition Supported by User Interaction for Service Robots*. doi:10.1109/icpr.2002.1047823

Koval, V., Turchenko, V., Kochan, V., Sachenko, A. and Markowsky, G. (2003) Smart license plate recognition system based on image processing using neural network. *Second IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2003. Proceedings*. doi:10.1109/idaacs.20031249531

Ghasrodashti, E.K. and Yazdi, M. (2010) Recognizing Persian license plates in digital zoom condition. *2010 2nd International Conference on Education Technology and Computer*, pp. V3-208-V3-212, doi: 10.1109/ICETC.2010.5529561.

Park, H., Lee, D., Lim, M., Kang, Y., Oh, J. and Kim, J.-H. (2018) A Fast-Converged Acoustic Modeling for Korean Speech Recognition: A Preliminary Study on Time Delay Neural Network. *arXiv:1807.05855 [cs, eess]*, Available from: <https://arxiv.org/abs/1807.05855> [Accessed 15 April 2021].

Rachmadi, R.F. and Purnama, I.K.E. (2018) Vehicle Color Recognition using Convolutional Neural Network. *arXiv:1510.07391 [cs]*, Available from: <https://arxiv.org/abs/1510.07391> [Accessed 9 March 2021].

Sarkar, D., Bali, R. and Sharma, T. (2018) *Practical Machine Learning with Python*. Berkeley, CA: Apress.