Automatic Number Plate Recognition System using Machine Learning Techniques

Reuben Kurian

November 22, 2023

Abstract

Cyclists have been faced with adversity on the roads since the beginning of road transport. The recent rise in technology over the past few decades has also overlapped with the even more recent rise in greener modes of transport. As urban environments adapt to the increasing number of cyclists on the road, ensuring cyclists' safety becomes paramount. Traditional ANPR systems have been developed for use on motor vehicles, most commonly by law enforcement as well as some car parks. This paper explores the technologies behind ANPR, different machine learning methods as well as computer vision techniques. A solution to the problem statement is put forward, suggesting the use of ANPR with a GUI to allow cyclists to upload images or short videos of cars that may have been a threat to their physical safety.

I certify that all material in this dissertation which is not my own work has been identified.

Signed: Reuben Kurian

1 Introduction

Cycling is becoming an increasingly popular transport option for people in the modern day, as more people are shying away from cars powered by fossil fuels and turning more towards greener modes of transport, from electric-powered cars to cycling. Cycling, especially, is a great way to improve health, save costs and help the environment [13]. To accommodate for the increasing number of cyclists on the road, many changes have been made to the transport infrastructure, such as lower speed limits, cycle routes, and many other rules meant to benefit cyclists and be more of an inconvenience to fossil-fuel-powered car drivers to get more people to cycle.

With these changes, it is no surprise that problems will start to arise. With the increase in cyclists over the years, we have also been seeing an increase in cycle-related road accidents as well. A change to the road infrastructure that has helped mitigate this issue is the idea of cycle routes and roads built solely for cycle use. However, we have also seen road users trespass through these roads, putting all road users at risk, from pedestrians to cyclists to the drivers within the cars. This problem has been approached recently through the use of road cameras and using these cameras, as well as the use of law enforcement, with the aid of new laws, to help catch drivers and punish them. The use of road cameras has helped in reducing road traffic injuries [24] and helping prevent cars trespassing.

There are gaps in the current methods though. These road cameras may produce false negatives, which may allow vehicles that are not allowed to pass, to pass through, putting other road users at risk and allowing the offenders to continue without being punished with a penalty. The road cameras may also only cover certain angles on the road and may not cover parts of a road where someone may be trespassing, allowing offenders to continue without being punished.

The way I am approaching this problem is by creating automatic number plate recognition software for an app - that I will also create - that will allow cyclists to upload images or short videos of cars that cross through cycle routes and other unauthorised areas that may put any road users at risk. This app will be used to help with keeping roads safer for cyclists and by allowing members of the public to aid in keeping the roads safer by preventing road vehicles using cycle routes, it helps relieve some of the workload that would otherwise be put upon law enforcement.

We know that the problem is new, as the number of cycle users on the road has only become prominent over the past decade or two. As for that reason, there are not many solutions to this problem at the moment. Almost 1.3 million people die in road crashes each year, with 6% of them being cyclists [15]. As shown by the statistics, a small percentage of the amount of people who die in road crashes are cyclists, however, the number of deaths is large nonetheless, at roughly 78,000 of the 1.3 million people who die in road crashes each year being cyclists. The problem at hand may contribute to a percentage of accidents, and this number can be reduced by introducing stricter methods to prevent unauthorised vehicles passing through roads that are meant for cyclists.

I am approaching this problem by filling gaps and ensuring that my solution will address the issues with the current methods in use. To ensure that my solution will not produce false negatives or positives, I will train my model with large and diverse datasets to ensure that the results will stay consistent and accurate, regardless of what the image or video the user uploads may look like. Also putting my model through rigorous testing will ensure that the results stay consistent and accurate.
My hypothesis statement is that I believe the number of cars trespassing will reduce with the use of my solution and the number of trespassing cars will be inversely proportional to the number of users of my app. With more cyclists and civilians helping out law enforcement by having stricter measures

on ensuring cycle routes are only used by cycle routes, less cars will trespass.

I have found some tools that I will be using that will help me create my solution, I have found a few large datasets containing data that is diverse in nature, and I have found an open-source ANPR software that will allow me to take inspiration and guidance from when creating my own ANPR software.

2 Literature Review

In this section, machine learning and computer vision technologies are analysed. I will review machine learning, model selection and how machine learning is used in the development of ANPR software. In regards to computer vision techniques, I will be discussing OCR - the computer vision technology that is used in ANPR, as well as similar computer vision techniques that exist.

2.1 Machine Learning

How can one construct a computer system that automatically learns and gets better at tasks as they get more experience? The study of machine learning [12] is significant for addressing this question as well as contributing to the technologically advanced computer software it has fielded in many areas. Machine learning [8] is an evolving field of computational science that revolves around algorithms developed to mimic human intelligence by learning from its surrounding environment. It is a branch [12] within the artificial intelligence (AI) field that has emerged as the method of choice for developing software in areas such as computer vision, speech recognition, natural language processing, robot control, and various other applications.

2.1.1 Model Selection

The choice of model is important when it comes to developing machine learning algorithms. The most commonly used model with machine learning with the idea of ANPR in mind, is convolutional neural networks (CNNs), however, traditional machine learning models such as support vector machines (SVMs), K-nearest neighbours (K-NN) and template matching are also commonly used.

2.1.2 Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs) have shown tremendous performance [27] in numerous computer vision problems. CNNs work by taking an order 3 tensor as its input. An example would be an image containing H rows, W columns and 3 channels; H being the height of the image, W being the width of the image and 3 channels which signify as the image containing RGB colour channels (red, green, blue). The input then goes through a series of processing.

$$\boldsymbol{x}^1 \longrightarrow \boxed{\boldsymbol{w}^1} \longrightarrow \boldsymbol{x}^2 \longrightarrow \cdots \longrightarrow \boldsymbol{x}^{L-1} \longrightarrow \boxed{\boldsymbol{w}^{L-1}} \longrightarrow \boldsymbol{x}^L \longrightarrow \boxed{\boldsymbol{w}^L} \longrightarrow z \quad (5)$$

Figure 1: A breakdown of how CNNs work.

The above Figure 1 [27] encapsulates the sequential progression of a convolutional neural network (CNN) through its layers during a forward pass. x1 illustrates the input, which usually consists of an image of order 3 tensor. The parameters that influence the output of the first layer's processing is denoted conjointly as a tensor w1. x2 portrays the output of the first layer, which is utilised as the input for the succeeding layer's processing. This processing continues until all of the layers in the CNN have finished, which yields xL. A supplementary layer is added for the backward error propagation process, a technique that is employed to learn optimal parameter values in the CNN.

2.1.3 Support Vector Machines (SVMs)

Support vector machines (SVMs) are computer algorithms that learn from examples to assign labels to objects [19]. Some recent uses and advancements of SVMs include [5] pattern recognition and regression estimation. For pattern recognition, SVMs have been used for isolated handwritten digit recognition, object recognition, speaker identification, charmed quark detection, face detection in images, and text categorisation [5]. For regression estimation, SVMs have been used for benchmark time series prediction tests, the Boston housing problem, and (on artificial data) on the PET operator inversion problem [5]. The approach for SVMs [17] can be briefly outlined as follows: class separation, overlapping classes, non-linearity and problem solution. In class separation, the optimal separating hyperplane between the two classes is sought by maximising the margin between the classes' closest points [17], where the points lying on the boundaries are called support vectors and the middle of the margin is the optimal separating hyperplane. In the overlapping classes stage, data points on the incorrect side of the discriminant margin are assigned a lower weight to minimise their influence. This is also known as a "soft margin". The non-linearity stage suggests that in instances where it is not possible to find a linear separator, data points are transformed into a (usually) higher-dimensional space, frequently achieved through kernel techniques, which then allows the data points to become linearly separable. The problem solution stage signifies that the whole task can be expressed as a quadratic optimisation problem, where known techniques can be used to obtain solutions.

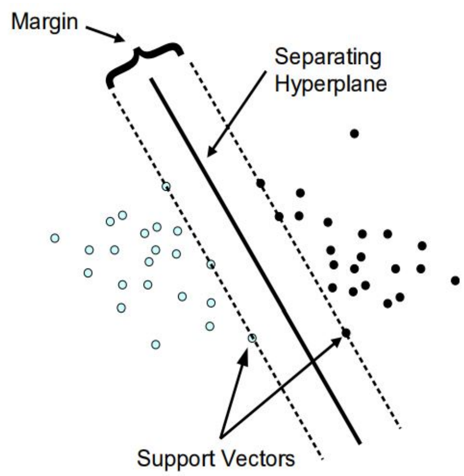

Figure 2: Classification in a linear separable case.

The above Figure 2 [17] visualises how classification occurs in SVMs. Maximising the margin size is crucial as it helps to optimise the SVM's ability to predict the accurate classification of unseen examples [19].

2.1.4 K-Nearest Neighbours (K-NN)

K-nearest neighbours (K-NN) is one of the most fundamental and simple methods used for classification [22]. It is a viable option when there is little to no knowledge about how the data is distributed. This rule involves retaining the full training set during the learning process and assigns to each query a class represented by the majority label of its k-nearest neighbours in the training set [11]. The workings of K-NN [6] can be explained in six steps. Firstly, the algorithm selects the number K of neighbours. The second step would involve the calculation of the Euclidean distance of K number of neighbours. Thirdly, the algorithm selects the K nearest neighbours according to the aforementioned Euclidean distances calculated in the previous step. The fourth step involves the counting of the number of the data points in each category among these K neighbours. For step five, the algorithm assigns the new data points to the category that has the highest number among its neighbours. The model is finally ready. Some characteristics of k-NN [22] include between-sample geometric distance, classification decision rule and confusion matrix, feature transformation and performance assessment with cross-validation.

2.2 Automatic Number Plate Recognition

Automatic number plate recognition, also known as ANPR, is a software that utilises computer vision and machine learning techniques to read number plates from an image or video. ANPR is most commonly used in car parks, where a mounted camera scans for the front number plate of a vehicle as it enters a car park [16]. An inductive loop detector triggers the ANPR system, which then extracts and decodes the image. The number plate is either compared to a database or stored for later use [16]. A conventional ANPR system is split into four parts, which consist of vehicle image capture, number plate detection, character segmentation and character recognition [21]. Other use cases for ANPR can include traffic safety enforcement and automatic toll text collection [21]. ANPR utilises optical character recognition, also known as OCR, to read licence plates [2].

2.2.1 Optical Character Recognition

Optical Character Recognition, also known as OCR, is a computer vision technology that utilises an optical mechanism, similar to how our eyes operate [18]. OCR is a key component in ANPR, helping to read the number plates from the inputted images and videos. The performance of OCR is highly dependent on the quality of the input, and in the case of ANPR, factors such as higher resolution images/videos, a more forward angle of the number plate, good lighting and a closer image/video will help produce more accurate results. OCR also has other use cases, such as converting scanned paper documents into PDF files [18].

2.2.2 Tesseract

Tesseract [25] is an open-source OCR engine that was created in 1994 by HP and has been released as open-source in 2005. The processing can be split into seven stages. The process begins with connected component analysis to identify and store outlines of components. Outlines are grouped into 'Blobs' purely on nesting. Blobs are organised into text lines, and both lines and regions are analysed to determine if it is fixed pitch or proportional text. The kind of character spacing can affect how the text lines are broken into words. Fixed pitch text is divided based on character cells, while proportional text is divided using definite and fuzzy spaces. Recognition is a two-pass process. The first pass is made to recognise each word, with results of good quality passed to an adaptive classifier as training data. The adaptive classifier is likely to recognise text lower on the page more accurately. A second pass is conducted to re-recognise words that were not initially recognized well, allowing for improved accuracy by using the adaptive classifier to make a contribution for text higher up on the page. The final phase resolves fuzzy spaces, and alternative hypotheses for x-height are checked to locate small-cap text.

2.2.3 Similar technologies to OCR

2.2.4 Intelligent Character Recognition

ICR [23], the acronym for Intelligent Character Recognition, is a computer vision technology that extends upon OCR by deciphering handwritten and machine-generated text. Due to ICR being able to decipher handwritten text, it is also a lot more difficult, as handwritten text is almost never identical. ICR systems prefer to analyse entire lines of text rather than small segments as the additional context can help understand handwritten text better, even when faced with the issue of recognising white space.

2.2.5 Scene Text Recognition

The need for text recognition in general settings (scene text) has become more prevalent with the rise of mobile imaging devices [26]. Scene text recognition [26] is a computer vision technology that works by utilising an OCR engine and a system rooted in generic object recognition. Although the scene text problem seems valuable, the computer vision community has shown limited interest in addressing this problem. There are many great use cases for scene text recognition, in helping automobiles equipped with cameras to navigate easier, and in assisting blind people in indoor environments, such as a grocery store. Scene text recognition [26] works in five steps. The first step is character detection. For this step, the potential locations of characters of an image are identified. Multi-scale character detection is utilised using sliding window classification to identify the potential locations; this approach has been incredibly successful in the use of face and pedestrian detection. A classifier would be optimal for detection of a large number of categories. Random ferns are an inviting option for a choice of classifier. Pictorial structures are used to detect words in the image. The final step is to operate non-maximal suppression over all detected words.

2.3 Image Preprocessing

Image preprocessing is a key stage [3] in image recognition. Some preprocessing techniques include mean normalisation, standardisation and zero component analysis [20]. It is an important step in image recognition software and is performed to improve the quality of an image so that it can be analysed more effectively [14].

[9] has used a wide range of methods to preprocess face images. The image preprocessing is split into distinct stages [9]. Colour normalisation methods such as intensity normalisation, grey world normalisation, comprehensive colour image normalisation, hsv hue and bgi hue are applied during preprocessing of images [9].

In [9], they proceed with using statistical methods to ensure that the brightness and intensity of the images remain constant. These methods include varying the brightness, horizontal brightness, vertical brightness, local brightness and local brightness mean, to ensure that the brightness and contrast remain constant across all face images.

The next stage involves convolution methods [9], and these would include varying the smoothness, blur, edge, contour, detail, sharpen and emboss. Convolution methods are utilised to Improve or diminish characteristics, reduce noise and extract edges [9].

Finally, [9] also performs method combinations to reap the benefits of multiple image processing techniques. These method combinations include "contour filtering followed by smoothing", "smoothing followed by contour filtering", "local brightness transformation followed by smoothing", "local brightness transformation followed by contour filtering", contour filtering combined with the use of local brightness transformation, "contour filtering followed by smoothing, summed with the Local Brightness transformation" and "Smoothing followed by the Local Brightness transformation, followed by contour filtering".

3 Analysis of Requirements

There are two major components to my project. The first being the machine learning model itself, which includes training and testing the model, and the second being the app itself. The machine learning model will be used in conjunction with an OCR engine to form the automatic number plate recognition (ANPR) software. The ANPR software will be used to scan for number plates inputted as images or short videos by the user. Once the number plate has been obtained, the app will input the number plate into an API that will return details about the vehicle to the user.

3.1 Automatic Number Plate Recognition (ANPR)

The fundamental purpose of the automatic number plate recognition (ANPR) software is to scan an image or short video and extract the recognised number plate from the image (or frame from the video). The ANPR software will be used in conjunction with a front-end that will take the outputted number plate that was recognised from the input, input it into a vehicle detail checking API and return details about the vehicle such as the make, model and the year it was manufactured. DVLA [1] offers an API that takes a vehicle registration number as input and returns details about the vehicle in JSON format. Some basic forms of projects could include using computer vision and machine learning libraries, such as using OpenCV, Tensorflow and Keras [7] in Python to develop the ANPR software. To simplify the process and remove some of the explicit programming that would have to take place, a viable option could be looking at an open source ANPR software that is available. OpenALPR [4] is an open-source C and C++ library which is formed on OpenCV and Tesseract OCR. Using OpenALPR for my project would make the process a lot simpler by providing me with pre-trained models which can save time and resources. However, there is a lack of flexibility as the models are already built and not custom-built to the project's needs.

3.2 Graphical User Interface

To allow the user to use the ANPR software in a simple and presentable manner, a graphical user interface, also known as a GUI, would be a useful tool that will help the app look and operate in a clean, user-friendly fashion. Some possible Python libraries that I could use include Tkinter, PyQt and Kivy. The user interface should allow functionality to use the ANPR software as intended, and should link the API to the ANPR software and return the vehicle details to the user upon recognition of the licence plate.

3.3 Machine Learning

If the idea of using an open-source ANPR software is not feasible, then having to train a model through machine learning techniques would be ideal to develop our own ANPR software. Deep learning techniques such as convolutional neural networks can be considered, as well as traditional machine learning techniques such as support vector machines and k-nearest neighbours. Convolutional neural networks would be the more favoured approach out of the methods I have researched as they have shown great success in pattern and image recognition projects [10] and seem to even outperform humans in rare cases.

4 Specifications

4.1 Functional Requirements

The functional requirements are the minimum amount of tasks to be met to satisfy a working product.

Automatic Number Plate Recognition: It should be able to take images and short videos as input and scan for a number plate within the image accurately. It should also be able to identify individual alphanumeric characters within the number plate flawlessly.

Graphical User Interface: The GUI should have a user authentication system to eliminate unauthorised access to the system. Users should be able to analyse details about the vehicle returned from the vehicle checking API. It should also have a configuration settings menu.

Machine Learning: Training data preparation is essential to producing a successful ML model, including necessary preprocessing and labelling.

4.2 Non-Functional Requirements

The non-functional requirements are listed below.

Automatic Number Plate Recognition: It should be able to recognise number plates with a high success rate in a short amount of time. The ANPR should have a high reliability rate and be able to adapt to images uploaded in different conditions. It should also be scalable to handle an increase in users. Compliance with data privacy regulations is essential.

Graphical User Interface: It should be user-friendly enough to the point where any user can access it, without needing a deep understanding of the system. The GUI should be responsive and have minimal latency. The user design should be the same and not be inconsistent.

Machine Learning: The model should be able to be trained within a reasonable amount of time. Resource optimisation is key to ensure peak performance and utilisation of computer resources as necessary. The model should also be able to adapt to different input data and evolve, maintaining or improving the performance as the system adapts.

5 Risk Assessment

In this section I will summarise what the risks are and how I will reduce them. The first and most evident risk is privacy. Some privacy issues that might arise may include the use and handling of the images and videos. To comply with data protection laws, I will only use the image and short videos for the necessary purposes and discard the data when we no longer have any purpose for it. Another risk includes user consent to processing images. To mitigate this, I will have the application explicitly ask for permission from the user to process the images and videos. False positives and negatives of the ANPR software are also a risk. Thorough training and testing of the model with diverse datasets will ensure the risk of false positives and negatives will be at a minimum. Data and app security is another risk, and this can be mitigated by implementing robust security techniques as necessary and limiting unauthorised access where possible. Scalability is another risk and to minimise this risk, it is important to implement scalability during the development of the ML model and the app to accommodate for such needs. Maintenance of the app is another risk as failure to maintain the app can cause issues, both minor and major. To mitigate this, regular maintenance and having users report bugs and issues is vital to ensuring that the app remains in a good working state.

6 Evaluation

As the components are tightly connected together, the evaluation would be based upon the testing of the system as a whole, i.e., integration testing. To test the entire system, some images as well as some short videos can be inputted into the ANPR software via the GUI, where the expected outputs can be compared with the actual outputs. A performance metric to see how well the number plate was recognised can be calculating the number of correctly recognised alphanumeric characters within the number plate as a percentage. The testing of the ML model itself can also be performed through means such as cross-validation, bias & fairness testing and assessing performance on unseen data.

7 Conclusion

This paper first examined machine learning and computer vision techniques, exploring the various types of machine learning and different computer vision methods. It then looked at image preprocessing, a critical element in ANPR systems. The requirements were analysed, exploring the major components of the project. The specifications of the project were then examined, looking at the functional and non-functional requirements. The potential challenges, risks and viable solutions that the project may encounter were then explored and finally the system evaluation was examined.

References

[1] Api developer portal. URL https://developer-portal.driver-vehicle-licensing.api.gov.uk/apis/vehicle-enquiry-service/vehicle-enquiry-service-description.html#vehicle-enquiry-service-ves-api-guide.

[2] Feb 2023. URL https://www.softwebsolutions.com/resources/number-plate-recognition-using-computer-vision.html.

[3] Sing T Bow. *Pattern recognition and image preprocessing.* CRC press, 2002.

[4] Elena Roxana Buhus, Daniel Timis, and Anca Apatean. Automatic parking access using openalpr on raspberry pi3. *Acta Technica Napocensis*, 57(3):10, 2016.

[5] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[6] Antony Christopher. K-nearest neighbor, Feb 2021. URL https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4.

[7] Abhinav Dadhich. *Practical Computer Vision: Extract Insightful Information from Images Using TensorFlow, Keras, and OpenCV.* Packt Publishing Ltd, 2018.

[8] Issam El Naqa and Martin J Murphy. *What is machine learning?* Springer, 2015.

[9] Thomas Heseltine, Nick Pears, and Jim Austin. Evaluation of image preprocessing techniques for eigenface-based face recognition. In *Second International Conference on Image and Graphics*, volume 4875, pages 677–685. Spie, 2002.

[10] Samer Hijazi, Rishi Kumar, Chris Rowen, et al. Using convolutional neural networks for image recognition. *Cadence Design Systems Inc.: San Jose, CA, USA*, 9(1), 2015.

[11] Sadegh Bafandeh Imandoust, Mohammad Bolandraftar, et al. Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International journal of engineering research and applications*, 3(5):605–610, 2013.

[12] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[13] Paraskevi Karanikola, Thomas Panagopoulos, Stilianos Tampakis, and Georgios Tsantopoulos. Cycling as a smart and green mode of transport in small touristic cities. *Sustainability*, 10(1), 2018. ISSN 2071-1050. doi: 10.3390/su10010268. URL https://www.mdpi.com/2071-1050/10/1/268.

[14] Barath Kumar. Image preprocessing - why is it necessary?, Feb 2021. URL https://medium.com/spidernitt/image-preprocessing-why-is-it-necessary-8895b8b08c1d.

[15] Loreta Levulytė, David Baranyai, Ádám Török, and Edgar Sokolovskij. Bicycles' role in road accidents a review of literature. *Transport and Telecommunication Journal*, 17(2):122–127, 2016. doi: doi:10.1515/ttj-2016-0011. URL https://doi.org/10.1515/ttj-2016-0011.

[16] R.A. Lotufo, A.D. Morgan, and A.S. Johnson. Automatic number-plate recognition. In *IEE Colloquium on Image Analysis for Transport Applications*, pages 6/1–6/6, 1990.

[17] David Meyer and FT Wien. Support vector machines. *R News*, 1(3):23–26, 2001.

[18] Ravina Mithe, Supriya Indalkar, and Nilam Divekar. Optical character recognition. *International journal of recent technology and engineering (IJRTE)*, 2(1):72–75, 2013.

[19] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[20] Kuntal Kumar Pal and KS Sudeep. Preprocessing for image classification by convolutional neural networks. In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 1778–1781. IEEE, 2016.

[21] Chirag Patel, Dipti Shah, and Atul Patel. Automatic number plate recognition system (anpr): A survey. *International Journal of Computer Applications*, 69(9), 2013.

[22] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[23] Raymond Ptucha, Felipe Petroski Such, Suhas Pillai, Frank Brockler, Vatsala Singh, and Paul Hutkowski. Intelligent character recognition using fully convolutional neural networks. *Pattern recognition*, 88:604–613, 2019.

[24] Katherine Pérez, Marc Marí-Dell'Olmo, Aurelio Tobias, and Carme Borrell. Reducing road traffic injuries: Effectiveness of speed cameras in an ..., Oct 2011. URL `https://ajph.aphapublications.org/doi/10.2105/AJPH.2006.093195`.

[25] Ray Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

[26] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464, 2011. doi: 10.1109/ICCV.2011.6126402.

[27] Jianxin Wu. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23):495, 2017.