THE UNIVERSITY OF ADELAIDE

MASTER THESIS

Deep Learning in Pattern Recognition and Stock Forecasting

Author:
Chao LU

Supervisor:
Dr. Javen SHI

A thesis submitted in fulfillment of the requirements for the degree of Master of Computer science

in the

School of Computer Science

June 14, 2018

Declaration of Authorship

I, Chao LU, declare that this thesis titled, "Deep Learning in Pattern Recognition and Stock Forecasting" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____________________________________

Date: ____________________________________

THE UNIVERSITY OF ADELAIDE

Abstract

Faculty Name
School of Computer Science

Master of Computer science

Deep Learning in Pattern Recognition and Stock Forecasting

by Chao LU

Stock market is a promising financial investment that can generate great wealth. However, the volatile nature of the stock market makes it a very high risk investment. Thus, a lot of researchers have contributed their efforts to forecast the stock market pricing and average movement. Researchers have used various methods in computer science and economics in their quests to gain a piece of this volatile information and make great fortune out of the stock market investment. This paper includes two parts. The stock pattern is a significant part for investment, and most investors find stock patterns through eyes which need a lot human resources. Therefore, in the first part, a head and shoulders (HAS) pattern recognition (current stock market analysis) model fed with candlestick charts is provided based on FASTER-R-CNN and resNet_50. To solve the overfitting problem brought by limited training images, two approaches, including data segmentation and data variation, are provided. Data segmentation uses time series segmentation method to remove noises on stock history data, leading to a much simpler chart. Data variation is an approach to change some values of the stock history data randomly to generate more HAS patterns. Finally, data variation method has a better result than data segmentation, and the AP@IOU is approximately 74% which is better than the best result of Yang (Yang, 2009)'s 70%. The final aim of pattern recognition is to predict the stock trend in the future. Therefore, the second part introduced a CNN model fed with 2D images to predict stock trend 10 days late (future stock market forecasting). The best result of this model is 58% which is better than Poulos's (Poulos, 2014) 3-layer LSTM (55.45%) and SVM (56.31%) model.

Contents

List of Figures

Chapter 1

INTRODUCTION

The stock market is a well-organized market where people can sell and buy in a public manner. When an individual buys a stock, he essentially becomes one of the owner of the company which sells the stock. The stock market is a community in which some people want to buy stocks while someone wants to sell. In finance and economics, there is a law of demand and supply which holds that when the demand increases or the supply decreases the price would increases. On the contract, if the demand decrease or supply increases the price will decreases. Stock also likes these common commodities to some extent in the stock market. The stock market analysis has become a popular topic in many areas.

Many researchers of computer science also did a lot of research in stock prediction. The main method of machine learning is a well-established method which is in a wide range of applications. Machine learning is the science of getting computers to take decisions without being explicitly programmed to do so. It can give satisfying results in some areas like image recognition, text recognition, and speech recognition. It can be used for almost anything. With the help of machine learning, a lot of complex problems can be solved as long as there are enough data and computation resources. While the improvement of the computing ability of computers in the 20th century, machine learning has been an important method of solving the complex problem. It can be divided into supervised, unsupervised, association and reinforcement learning. In supervised learning, input data will be labeled to represent the class of the data. This kind of learning algorithm includes Decision Tree, Regression, Support Vector Machines (SVM) and Naive Bayes. While the unsupervised learning algorithms use unlabeled data to draw inferences from beforehand. These algorithms include K-mans and Gaussian Mixture Models.

Pattern analysis in the stock market is a significant important part of stock analysis for all investors (Alkhateeb, 2016), it probes into the comparison between the strengths of the longs and shorts indicated by the stock curves. In a nutshell, stock pattern analysis is a technique for discovering both the undergoing and potential trends in stock price using various patterns displayed by the corresponding curves. There exists many stock patterns which are widely used to make stock predictions including short-term and long-term predictions. These data can be intraday, data and monthly history data, and the patterns exist in a period as small as one day or as long as many months. With those patterns, stock market professionals can easily predict the future market trend. Although there are some pattern recognition techniques, but their results are not satisfied for investors. Most investors still find stock patterns through eyes which need a lot human resources. In addition, because of the complexity of candlestick charts and stock market, it is easy to make mistakes during finding stock patterns through human eyes. Pattern recognition with computers can help investors save a lot of human resources and investment time cost. Also it has ability to achieve a more accruable result. The most used three stock patterns are Head and Shoulders (HAS), Inverse Head and Shoulders (IHAS) and rectangular patterns (RP), due to their reliable performance over the period of time. As data is visually represented, those patterns are more considered as shapes rather than numbers. Image classification and object detection in machine learning is a reasonable method to solve this problem. Therefore, pattern recognition with machine learning is one of the immediate needed techniques for stock markets. Stock pattern recognition is for current statement, however the final aim of it is to help investors predicting stock market trend in the future. Therefore, stock analysis includes two parts: pattern recognition in current stage, and stock forecasting in future stage.

In machine learning area, SVM and Artificial Neural Network (ANN) are mostly used for stock analysis. SVM was first provided by Vladimir N. and Alexey Ya Chervonenkis in 1963 and has become one of the most popular algorithms in machine learning with half-century development. While ANN possess attributes of learning, generalizing, parallel processing and error endurance, which makes ANN powerful in solving complex problems like the stock prediction (Azadeh, Ghaderi 2008)

1.1 GENERAL OVERVIEW

The main aim of this project is to using machine learning technique to help people to analyze current stock market (stock pattern recognition), and use machine learning technique to predict future stock market (stock prediction). In previous work, most of the methods for stock pattern recognition and stock prediction are based rule-matching and the algorithm based on template-matching, a few are base deep learning methods. However, both of the rule-matching based and template-matching based algorithms highly require the participation of domain experts, as well as their lacks of the learning ability, and for those with deep learning are most based with 1D data. In this project, the main machine learning technique is convolutional neural network (CNN) of deep learning and all input are 2D candlestick images.

This project includes two parts. In the first part, an object detection method based on the Faster Regression Convolutional Neural Network (Faster-R-CNN) is used to analyze current stock market (stock patterns recognition). The input of the model is candlesticks charts which are more acceptable for human. As a deep learning method, this pattern recognition method need huge number of labeled images as input, however it is impossible to label all these images by hands. To solve it, this work also presents 2 methods: a) using time series segmentation to reduce noise in candlesticks chart for a simple chart, b) automatically producing more labeled images as training data to improve the performance of the model. In the second part, the main aim is to predict future stock trend (stock forecasting) with machine learning techniques. Convolutional Neural Network (CNN) is used in this part. Different from previous work using number data or text data, the input in this part are images.

Chapter 2

BACKGROUND

2.1 PATTERN RECOGNITION (current stock market analysis)

Pattern analysis in the stock market helps people to predict the stock market through finding out those important patterns in historical data, according to these patterns people can easily make a prediction with previous experience. Stock pattern recognition is a technique to use various patterns displayed by the corresponding curves to discover the undergoing and potential trends in stock price. Stock patterns exist in yearly, monthly, daylily, even in intraday history data. With these patterns, stock market professionals can easily make stock predictions including short-term and long-term predictions. For example, the pattern shown in Fig. 2.1 is a typical stock pattern called Head and Shoulder (HAS). There are three peaks in this pattern. The first and third peak is similarly high, which means two shoulders. While the second one has the highest price, which means the head. That is why it is called head and shoulder. Normally, when this pattern exists in an increase of stock price, the stock would then follows a sharp decrease after the third peak. Therefore, it is a better trade strategy to sell the stock after the second peak when detected this pattern in an increased stock.

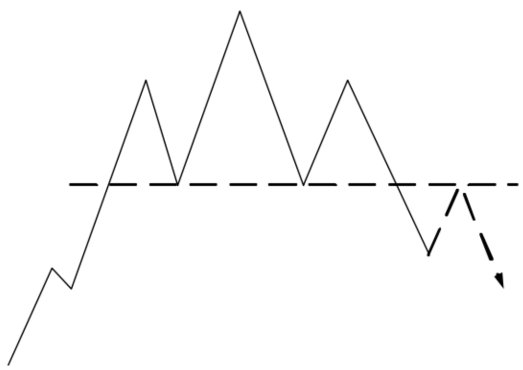

FIGURE 2.1: An example of Head and shoulder

2.1.1 PATTERN RECOGNITION WITH MACHINE LEARNING

It is unrealistic to recognize stock patterns from thousands of hundreds of stocks for finance practitioners. That is a huge work for human. Then researchers begin to use other techniques of computers for solving this problem since computers have much more powerful ability of computation. These methods can be divided into three parts: statistic and probabilistic approach, support vector machines and neural network approach.

2.1.2 STATISTIC AND PROBABILISTIC APPROACH

The probabilistic approach is a method widely used for pattern search in computer vision, and it has the ability for solving time series problem. Stock market prediction is one of the time series problems, therefore it can be solved with a probabilistic approach. Keogh and Smyth (Keogh and Smyth, 1997) have provided a method to find patterns in time series using probabilistic approach. In this work, authors use piecewise linear segmentation as the underlying representation and use a prior distribution on expected deformations from a basic template to define local features (like peaks, valleys, and plateaus). They use another prior to the relative location of the individual features to define the global shape. Then with an appropriately defined probabilistic model, the local and global features are integrated and overall distance between sequence patterns based on prior knowledge is measured. Finally, a search algorithm using these measured distance to find matches for a variety of patterns on a large number of data sets.
Hidden Markov Model (HMM) is another common method for stock pattern recognition. HMM is a statistical model which is a Markov process, but has hidden Markov parameters. The main problem is to find out these hidden parameters and then use these parameters for further analysis, like pattern recognition. In 2005, Hassan and Nath (Keogh and Smyth, 1997) provided a method using Hidden Markov Model to model a time series of multivariate observations for market forecasting. According to their results, this model has a similarly successful result as other models based on Artificial Neural Networks (ANN) in stock pattern recognition. However, this model has a significant weakness that is highly subjective to the training data. It has a worse adaptability, which means its trained model cannot adapt to another stock ticker if it is not trained again. This is because this model needs to be trained on a large dataset and it needs more exploratory to find optimal coefficients.

2.1.3 SUPPORT VECTOR MACHINES

Support Vector Machines (SVM) is also a good way for stock pattern recognition because It Is always used in the classification of different patterns. It divided data sets into different parts using a hyper plane. As shown in Fig 2.2, the hyper plane wx+b = 1 and wx-b = 1 separated the data points into two different parts.

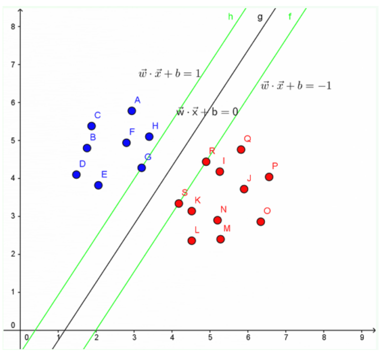

FIGURE 2.2: SVN

The main aim for SVM is to find the hyper plane:

$$f(x) = xw^T + b$$

To find this hyper plane, a loss function is given by Murphy (Robert, 2014):

$$L_\epsilon(y, \hat{y}) = \begin{cases} 0 & \texttt{if } |y - \hat{y}| < \epsilon \\ |y - \hat{y}| - \epsilon & \text{otherwise} \end{cases}$$

Where y is the know data point and y^ is the predicted data point. e is the accepted error. Then the aim is to optimize following equation (Dover, 2017):

$$J = C \sum_{i=1}^{N} L_\epsilon(y_i, \hat{y}_i) + \frac{1}{2}||w||^2$$

Here w is a linear combination of the inputs

$$\omega = \sum a_i x_i$$

However, this method for pattern recognition has a limitation on its speed and size during both training and testing, because training SVM takes two passes through the dataset. In 2009, Yang (Yang, 2009) presents an approach for stock pattern recognition with SVM. Their approach constructs the feature pattern vectors containing the characters on the market structure according to the profit unity approach. They believe market trend forecasting can be considered as pattern recognition. Their pattern vectors result from professional investing expertise. Then they use SVM to recognize stock patterns through mapping pattern vectors into class space of trend moving up and down. The result of this approach is about 70%.

2.1.4 ARTIFICIAL NEURAL NETWORK

In past few years, artificial neural network (ANN) technique has a great improvement and it is becoming more popular in stock pattern recognition. Kamijo's (Kamijo and Tanigawa, 1990) has presented an approach to detect stock patterns with an artificial neural network. In this work, the triangle stock pattern is used for detection. Triangle pattern indicates an important clue to the trend of future change in stock prices, but this pattern is not clearly defined by a rule-based approach (Kamijo and Tanigawa, 1990). They use 16 triangles extracted by an expert from Tokyo Stock Exchange. 15 of them are used for training and 1 for testing. The results show that it has a good accuracy, which was accurately recognized in 15 out of 16 experiments.

In 2007, Guo, Liang and Li (Guo, Liang, and Li, 2007b) have implemented a model for stock patterns recognition with ANN. They chose 2029 samples out of 508 stocks from Shanghai Stock Exchange for training while 4937 samples as testing samples. They chose only close price and plot it into 2D images for detection. Before training, they process the data with segmentation methods with the bottom-up method. The non-processed figure and processed figure are shown in Fig. 2.3 and Fig. 2.4. This can help the neural network extract the features much easier. Then those processed figures will be trained with a forward neural network.

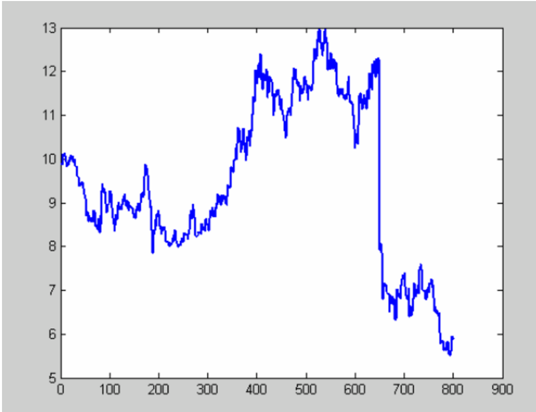

FIGURE 2.3: Without processed data

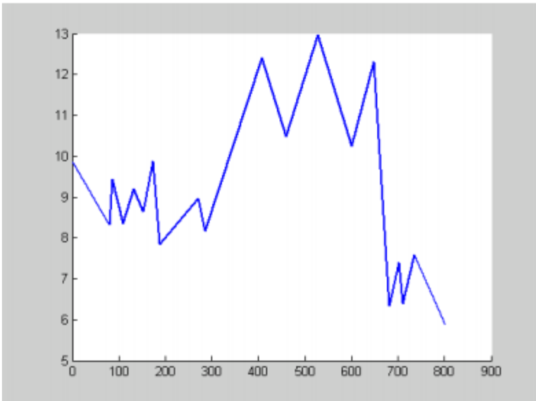

FIGURE 2.4: Processed data

In the same year, Guo, Liang and Li (Guo, Liang, and Li, 2007a) have also implemented another improved approach with Rival penalized competitive learning (RPCL) for stock patterns recognition. The results show this improved approach has a better result than basic artificial neural network (ANN).

2.2 STOCK PREDICTION (future stock market analysis)

The final aim of stock patterns recognition is stock prediction. Stock prediction has been an intriguing topic for a long time. In general, people would buy the stock before the price rising and sell it before drop. As shown in Fig. 2.5, green stick means the price increases while red stick means the price decreases. A good way for stock trade is to buy the ticker around the first valley and sell it before it reaches first peak. Stock prediction is to predict the stock trend or the stock price, which can help stock buyer to know when to buy and sell, to make more profit. In general a few percentage points of stock prediction improvement can increase profit by millions for those investors. Or help the stockholder against systematic risk. Because the stock market is subject to large price volatility. For common shares holders this large volatility means high risks. That is why the stock prediction is significant important to against systematic risk.

FIGURE 2.5: A sample of stock market

Unfortunately, stock analysis is not simple as it said in the law of demand and supply. Stock market is essentially dynamic, non-linear, nonparametric and chaotic in nature. Besides the demand and supply, it is also influenced by many macroeconomic factors such as political events, company's policies, general economic conditions, commodity price index, bank rates, investors' expectations, institutional investors' choices and psychological factors of investors. (Wang et al., 2011) In addition, according to Eugene Fama (1970)'s Efficient Markets Hypothesis (EMH), the stock always trade at a fair value on stock exchanges, making it impossible for investors to either purchase undervalued stocks or sell stocks for inflated prices. As such, it should be impossible to outperform the overall market through expert stock selection or market timing, and the only way an investor can possibly obtain higher returns is by purchasing riskier investments. The current price essentially satisfy the law of demand and supply and the prices movement is like random walking (RW) (Robert Brown). But the EMH is based on a questionable presumption that the investors in the market are sufficiently rational and can respond promptly to all market information. And in recent years many stock prediction methods has already achieved better results than the RW. Therefore it is possible to predict the stock price to some extent. Overall, predicting stock price movement accurately is not only interesting but also extremely challenging.

2.2.1 STOCK PREDICTION WITH MACHINE LEARNING

Previous works about stock forecasting can be divided into two parts according to input data, including stock prediction with historical data and stock prediction with news events.

Feed with history data

Most of the methods are feed with historical data. History data are the price history collected by some financial company like Yahoo Finance and Google Finance. These data mainly contain 6 columns, time, open price, close price, high price, low price, and volume. The opening price and close price is the price at which a security first trades and the last trades upon the opening of an exchange on a given trading day, while high and low represent the highest and lowest price among the time period. Since the stock prediction is one of the time-series-forecasting problems, many researchers use the history price with time to predict the stock movement. Also, according to machine learning method, these methods mainly include two parts, SVM and ANN.

ANN

Most researchers with machine learning prediction have focused on ANN, which is constructed with a series of interconnected nodes. This model simulates individual neurons which are organized into different layers based on function. The model assigns weights to connections and calculates the outputs with the weights. Preminger and Franck (Preminger and Franck, 2007) used a robust linear autoregressive and a robust neural network model predict stock, but this model is better than the RW. Yu and Huarng (Yu and Huarng, 2008) used a model called bivariate neural network-based fuzzy time series with substitutes to predict time series problem. Then Change et al (Chang et al., 2009) provided a method called CBDWNN which combines dynamic time windows, case based reasoning and neural network with other methods. They compared that method with some other main methods and CBDWNN had a better result.

In 2011, Erkam Guresen et al (Guresen, Kayakutlu, and Daim, 2011) combined the ANN with a genetic algorithm to build a new method for stock prediction. They use polynomials to build ANN, then use a genetic algorithm to estimate parameters of ANN, like starting polynomials and weights. This model is a regression model. However, the result of this model is not satisfied, but it is a good begin of a combination of ANN with other optimal algorithms. In the same year, Jianzhou Wang et al (Wang et al., 2011) provided an approach which can remove the noises in stock data to some extent. The method decomposed data into multiple layers by wavelet transform (WD). With this, input data would be divided into high-frequency and low-frequency data. Then establish back propagation neural network (BPNN) with those low-frequency data. One thing need mention is that they only use is the close price of history data. The final result is slightly better than normal back propagation neural network.

Kar (Kar, 1990) implement a RNN model to predict the stock market. In this model, a number of activation functions are implemented along with options for cross-validation sets. The input data contains the open, close, high and low price of history data. This model is a regression model and output is predicted close price. Average accuracy is 88%, seems much better than other works, but this accuracy is not the like other work. The accuracy here means the difference between predicted close price and actual close price. As price value sometimes is always more than thousands, therefore it is good but not as good as it looks like.

SVMs

SVM is powerful in tracing the stock market and helping maximize the profit of stock option purchase while having low risk. Compared with ANN May finding a local optimum, SVM has more chance to find a global optimum. Therefore, numbers of researchers have tried to using SVM to predict stock price. Kim's (Kim, 2003) method based on SVM can predict the stock price with a 57% accuracy which is significantly above 50% threshold. Shah (Shah, 2007) has compared some machine learning methods for stock prediction. Among those methods, SVM has achieved the best accuracy which is about 60%.

Instead of national stock data, Zhang (Shen, Jiang, and Zhang, 2012) used global stock data in association with data of other financial product as input features to SVM. Beside history data of the stock, they also use some related financial products, like currency price (USD, AUD, EUR and JPY), resource price (silver, platinum, and oil). He compared the prediction accuracy of single features. As expected the currency price has a significant influence on the prediction result. Also, he found the best term of prediction is 20-30 days.

Dai (Dai and Zhang, 2013) also compared the SVMs with other methods on the history data of Minnesota Mining and Manufacturing (3M). SVM had a 55.2% accuracy.

Madge (Madge and Bhatt, 2015) trained a SVMs model with 34 technology index. In this work, he predicted the price volatility and momentum for individual stock and overall sector. He calculated the average price changing rate of 'n' days to represent the price volatility while using the average label of n days to represent the momentum. When the price of next is higher than today, it is labeled as '1' otherwise '-1'. The prediction term 'm' is better to be between 20 and90. While increasing 'm', the model needs more data from this stock, and less form other stocks. This would lead to a higher accuracy on this particular stock but much lower on other stocks. Therefore, prediction term should be between 20 and 30 to keep a good accuracy on all the stocks.

Feed with news events

Different from those approaches feed with historical data, some approaches feed with news events also has been provided in recent years. In the most recent year, Ding (Ding et al., 2015) found a method to use news events to predict stock price with CNN in 2015. First, useful information would be extracted from 10 million finance news with Natural Language Processing (NLP). Second, a CNN model would be built with these vectors to predict price trend. With this model, he can achieve the best accuracy of 65%, which is a great improvement for stock prediction.

In the next year, Poulos (Poulos, 2014) used another RNN model to predict stock price. Similarly, he uses NLP for extracting information from news, but the difference is that Poulos uses news titles. And his RNN model is based on Gated Recurrent Unit (GRU). Totally, this model has 12 GRU with Dropout layers, then ending with 1 dense. With same datasets, the result is 55.94, slightly better than 3-layer LSTM (55.45%) but worse than SVM (56.31%)

Chapter 3

METHOD

3.1 STOCK PATTERN RECOGNITION WITH DEEP LEARNING WHILE FEEDING 2-D IMAGES

Stock chart patterns play an important role in the stock analysis and prediction technical and can be a powerful asset for traders at any level. It is a very basic level of price actions which happened in any time period: monthly, daily and intraday. Even for a beginner trader, if they can recognize these patterns early, they will gain a real competitive advantage in the markets. In this part, I implemented a deep learning model to recognize the common stock patterns which are helpful for any level of stock traders. More importantly, recognizing with computers is much quicker than finding out all patterns by humans. For example, it may only need a few minutes for computers to track intraday history data of all stock indexes while it may need a group of investors to work weeks. Besides saving time, a stock pattern recognition can also help the investment companies save a lot of human resources since a computer can do much more jobs than a human in a similar time period.

3.1.1 Stock patterns

There are 7 common stock patterns, including Ascending triangle, descending triangle, double top, double bottom, head and shoulders, Inverse head and shoulders and Cup and handle.

Ascending triangle

An ascending triangle exists in an increased movement of the stock, which is a con-

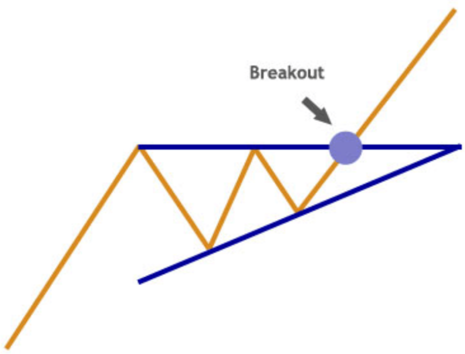

FIGURE 3.1: Ascending triangle

tinuation pattern. When the price breaks up the bound, it would normally increase. An ascending triangle is always bullish pattern whenever they occur.

Descending triangle
The descending triangle is an also a continuation pattern, but this is opposite to ascending triangle. This pattern always happens in a drop period in the stock market. When the price breaks down the bound, it would always decrease continually. The descending triangle is a usually a bearish pattern, and always be considered as a continuation in a downward trend. **Double top (M)**

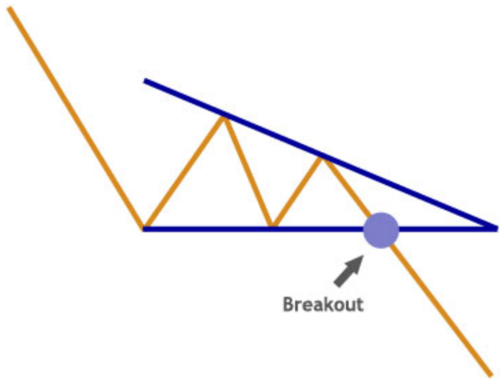

FIGURE 3.2: Descending triangle

A double top pattern is another very common and useful stock pattern. This pattern normally happens in an increased period. There are two peaks in this pattern, and two peaks are similar high with less than 3% difference (Investopedia, 2015). The valley between two peaks is less than 10-20% of the peak. Two peaks make a resistance line while the valley and two peaks make a support line (As the blue line in the figure). Once the price breaks through the support line, the price would decrease immediately, traders would better to sell after breaking through the support line. On the contract, if the price breaks the resistance line, it normally would increase continually. Therefore, it is a better trade strategy to buy after the price break the resistance line.

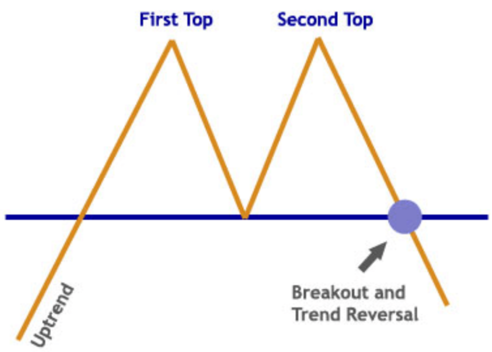

FIGURE 3.3: Double top (M)

Double bottom (W)
Similar to double top, a double bottom pattern is another very common and useful stock pattern, but it is an opposite of double top. This pattern looks like a 'W', so it is also called W pattern. This pattern normally happens within a drop period. There

are two valleys in this pattern, and two valleys are similar high with less than 3% difference. The peak between two peaks is higher than 10-20% of the peak (Investopedia, 2015). Two peaks make a support line while the valleys and the peak make a resistance line (or called neckline, as the blue line in the figure). Once the price breaks through the support line, the price would decrease immediately, traders would better to sell after breaking through the support line. On the contract, if the price breaks the resistance line, it normally would increase continually. Therefore, it is a better trade strategy to buy after the price break the resistance line.

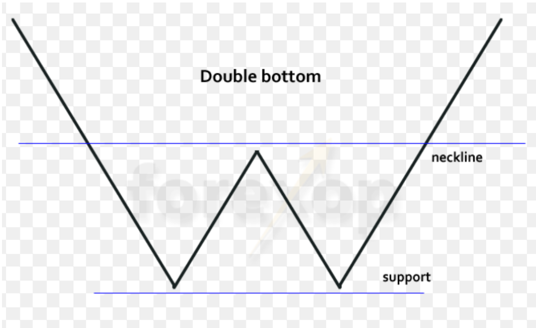

FIGURE 3.4: Double bottom (W)

Cup and handle

A cup and handle pattern gets its name from the shape of this pattern, as shown in Fig. 3.5. This pattern is curved u-shape, like a cup while the handle is in the right and is slightly downwards. In general the right-hand side of the diagram has low trading volume, and it would keep for a long time .

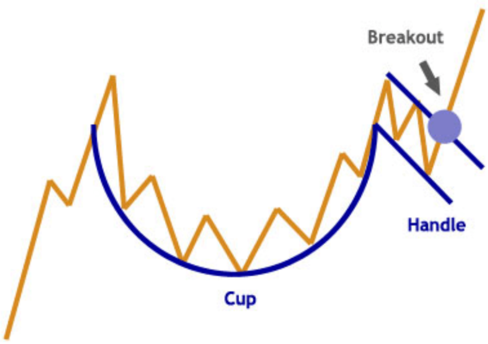

FIGURE 3.5: Cup and handle

Head and shoulders

Head and shoulders pattern is one of the most used patterns in stock analysis, because of its distinctive shape which is developed by two trend lines which converge. This pattern always exists in an increase period. Two valley makes the support line (or neckline) while first and third peak make the resistance line. Similar to 'M' and 'W' pattern, once the price breaks through the support line, the price would decrease, traders would better to sell after breaking through the support line. On the contract, if the price breaks the resistance line, it normally would increase continually. Therefore, it is a better trade strategy to buy after the price break the resistance line.

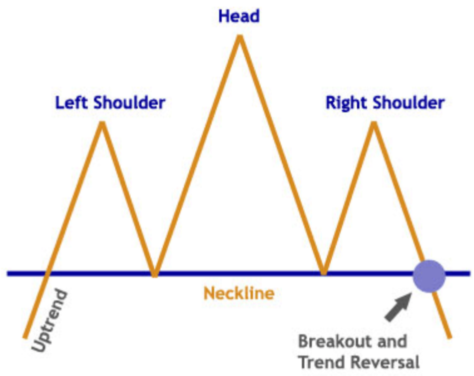

FIGURE 3.6: Head and shoulders

Inverse Head and shoulders
Inverse head and shoulders pattern is the opposite of head and shoulders pattern. This pattern always exists in a drop period. Two shoulders make the support line while first and second peak make the resistance line (or neckline). Similarly, traders would better to sell after breaking through the support line while buying after the price break the resistance line.

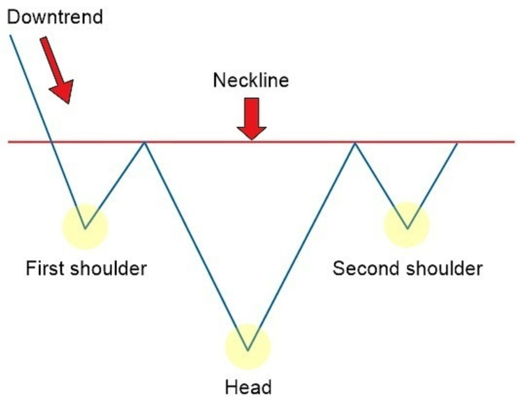

FIGURE 3.7: Inverse head and shoulders

In this project, the Head and shoulders (HAS) pattern are chosen as the pattern for recognition. This is because this pattern is important for pattern analysis, even for a beginner, it is easy to know the following trend after Head and shoulders (HAS) patterns. Another reason is that Head and shoulders have a distinctive shape, it is easier to be recognized with computers. This is the best choice for the first attempt at 2-D candlestick images as input. As said in the previous section, if a HAS pattern is detected, then traders can predict future price according to it. If the price breaks the support line, it will continually decrease. In contrast, if the price breaks the resistance line, it will then increases.

3.2 Object detection technique

Because only considered shapes of the patterns, object detection technique can be used to recognize stock patterns. The aim of object detection technique includes two steps. The first step is finding locations of all objects, and the second step is to judge classes of each object. The first step is the more difficult for a computer because objects can exist in many areas and their shapes and statements are variety.

3.2.1 R-CNN

In traditional object detection techniques, the most common algorithm is DPM (Deformable Part Model). In 2013, Girshick **\parencite** rosebrock2016intersection provided a deep learning object detection approach, R-CNN (Region-based Convolutional Neural Networks), based on Convolutional Neural Network (CNN). R-CNN can achieve a mAP of 48% on VOC 2007 test dataset. The providing of R-CNN means a great change from traditional object detection techniques to object detection techniques based on deep learning model. There are three main techniques in R-CNN, including selective search, CNN and SVMs.

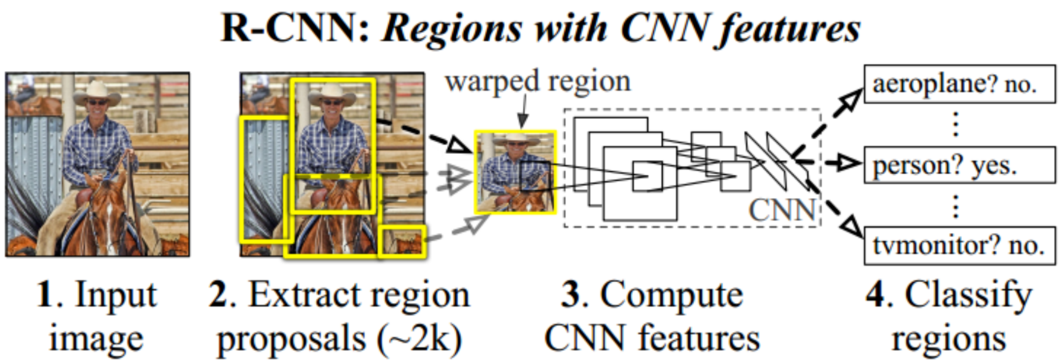

FIGURE 3.8: The structure of R-CNN (Girshick et al., 2014)

There are three steps as shown in Fig.3.8, including selective search (No.2 in the figure), extract features (No.3 in the figure) and classification (No.3 in the figure) (Girshick et al., 2014). In the first step, the selective search method is used to extract 2000 region proposals. Considering CNN needs same input size, 2000 region proposals are resized to 227*227. In the second step, the model for extracting CNN features need to be re-trained. To train the CNN pre-trained model, any region proposal is labeled as aim object even they only contain parts of aim region. The reason for it is training a CNN model needs a huge number of samples. If only keep those region proposals cover most parts of the aim object, training samples for CNN model is not enough. In this way, this pre-trained CNN model can be used to extract features. In the last step, using SVMs classification method to label those proposals. If only a region-proposal can cover all parts of ground truth area and its part not covered by ground truth is less than 5%, is labeled as aim object. Then train the SVMs classification with those proposals and their labels from SVMs to obtain new object detection model.

In the end, sort all proposals according to their score from SVMs and remove those proposals with low scores. R-CNN needs to do an extraction for all proposals extracted by selective search. Therefore, it has a great computation cost.

3.2.2 Fast-R-CNN

Based on R-CNN, Girshick (Girshick et al., 2014) provides another approach, Fast-R-CNN. With same CNN model VGG16, Fast-RCNN's speed is 9 times of R-CNN's. As shown in Fig **??**, It uses selective search method to extract several region proposals of the input images. Then generate feature map through some convolutional layers and the pooling layer. After that, a feature vector, roi_pool5, with a fixed length for each proposal is computed in the region of interest (RoI) layer. These vectors (roi_pool5) then are input into the fully connected layers for learning features and loss computation. The output of fully connected layers includes two parts. One of them is SoftMax Loss which represents the classification loss, and another one is regression loss which represents the location loss.

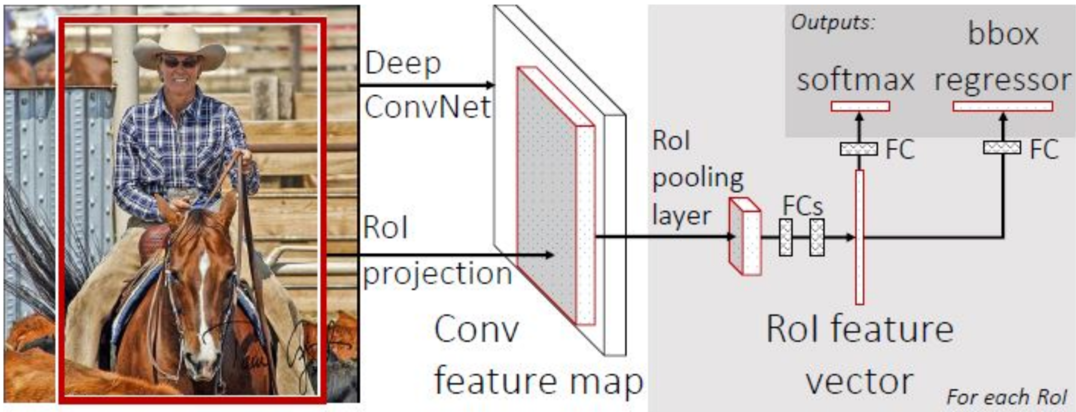

FIGURE 3.9: The structure of Fast-R-CNN. (Rosebrock, 2016)

RoI Pooling Layer is a simple way of using SPP-Layer. SPP-Layer is space Pyramid Pooling Layer, which has different sizes. RoI layers only have one size, they are all $7 * 7$. Therefore, the output of RoI Layer is r*c*7*7, because of for input (r, c, h, w) RoL Layer generate $7 * 7 * r * c * (h/7) * (w/7)$ blocks, then using Max-pool to compute the maximum value for each block.
Compared with R-CNN, Fast-R-CNN's accuracy is much better than R-CNN. In addition, this method does not need SVMs for classification because of using two different losses. Therefore, it can train the whole model. This makes Fast-R-CNN much easier to be trained.

3.2.3 Faster-R-CNN

Because of R-CNN need to use selective search to extract region proposals before extracting features, and this step needs much time. Girshick (Girshick, 2015) then provided another approach called Faster-R-CNN which uses Region Proposal Network (RPN) to extract proposals. RPN is a fully connected neural network, and it can extract region proposals by sharing convolutional features. Faster-R-CNN can extract proposals of an image with $1/20$ time that Fast-R-CNN spends. Faster-R-CNN includes two main parts: PRN for proposals extraction and Fast-R-CNN detection, as shown in Fig. 3.10
The main difference is RPN. Fig. 3.11 shows how RPN works. On the convolutional

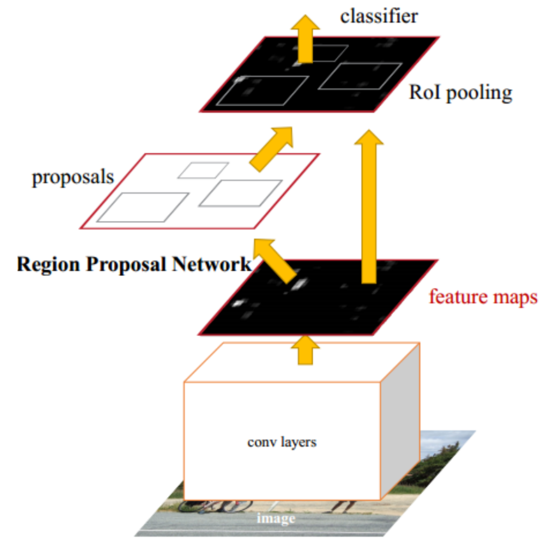

FIGURE 3.10: The structure of Faster-R-CNN (Girshick, 2015)

feature map, a 3*3 slide window is use to generate 256 or 512 dimensional fully connected feature. Then two fully connected layers are generated with this feature. First fully connected layer is reg layer which is used to predict the central anchor of the proposal and relevant coordinate (x, y), wide (w) and height (h). Second fully connected layer is cls layer which is used to judge it is background or an object. Sliding window ensures that both reg layer and cls layer are linked to all features spaces on convolutional feature map.

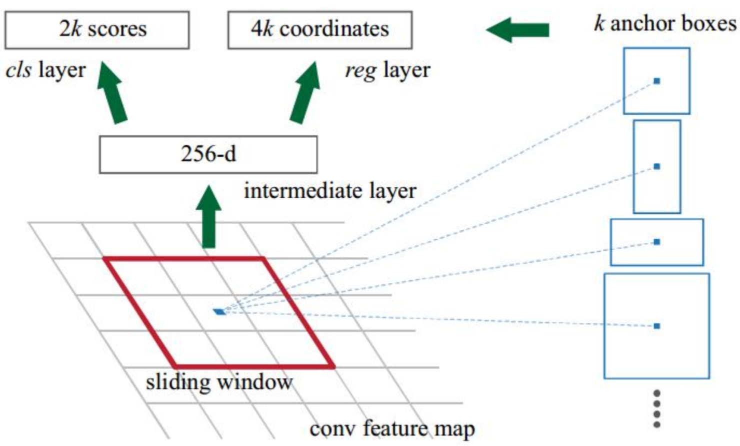

FIGURE 3.11: Region Proposal Network (RPN) (Girshick, 2015)

During training RPN, every Mini-batch is composed of 256 randomly picked proposal. Normally the ratio of positive to negative is 1:1, but when there are fewer positives, more negatives will be added to 256, otherwise positives will be added.

After proposals extraction, the author (Girshick, 2015) uses Fast-R-CNN to detect and recognize. RPN and Fast-R-CNN shared convolutional layers of CNN. To achieve this, alternating training is used for convolutional layers training. In the first step, train RPN. In the second step, train Fast-R-CNN with the extracted proposal from RPN. In the third step, initialize the convolutional layers of RPN. Alternating training iteratively executes these three steps. Through sharing weight to train RPN and Fast-R-CNN, Faster-R-CNN does not increase time cost but increases a lot on proposal quality.

The input of RPN can be any size, but according to the CNN, it has the minimum requirement. For example, the combine VGG16 with RPN, the input size is 228*228. In the initial stage of this work, a model based on VGG16 + Faster-R-CNN has been trained, but the result shows that VGG16 is not deep enough to learn all the features from stock patterns. Therefore, another much deeper convolutional network ResNet-50 is combined with Faster-R-CNN in this work.

3.2.4 Data collection

Candlestick chart

Similar to previous experiments, images represented history data would be used as input. Open, close, high and low price is used in the experiment, because of using a candlestick as input for feeding the Neural Network. The reason for using candlestick charts as input is that can candlestick charts are one of the most common ways for traders to analyze the stock market. It turns numeric data into a visualization form that can be understood by human easily. In a candlestick, there are at least 4 kinds of data: open, close, high and low, as shown in Fig. 3.12. High and low are represented with lines as upper shadow and lower shadow, while open and close are presented with sticks as real-body. When the close price is greater than open price, the real-body stick is colored with green to represent the increasing trend. Otherwise, it real-body stick is colored with red to represent a decreasing trend as shown in Fig. 3.13.

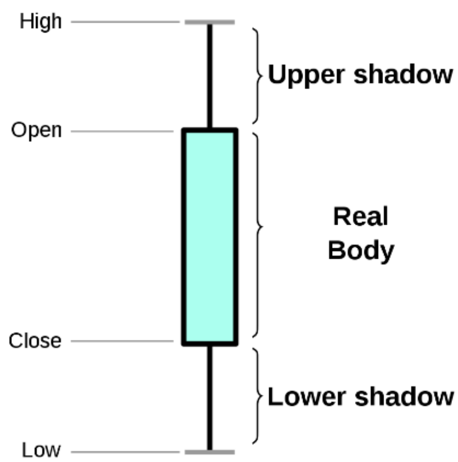

FIGURE 3.12: Candlestick

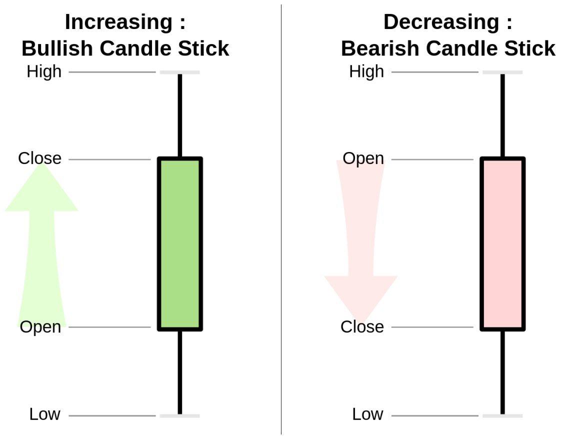

FIGURE 3.13: Increasing and decreasing candlesticks

Different window size has used in this experiment, including 30 days, 15 days, 7 days. 1 day. Some works show that there are most Head and Shoulder patterns when the window size is 2 to 3 months, which is more than 50%, while less when the window size is smaller than 1 month (Rajagopal, 2016). However, If setting window size to 2 months, the model would not be sensitive to intraday data, because data changes in hours or even in 1-2 days is not easy to be found compared with 2 months data. The aim of this project is to help traders to recognize the stock pattern in everyday stock price. The model needs to have the ability to find out the patterns timely. IF pattern recognition is too late. It does make any sense. Therefore, the smaller window size is better for investment, but too small will leading too few patterns can be found. Few patterns in datasets will also make the model useless. Therefore, the window size is set to 7 days and the step size is 1 day, which means previous seven days' history data is plotted on one image every day.

Intraday data is collected every 15 minutes, and every hour has 6 data points. Therefore in the image, there are about 300 data points in one image. It is easier to see both stock trend for the whole period and daily trend in a candlestick chart when there are 300 data points in one image, as shown in Fig. 3.14. When the Close price is greater than the open price, the daily trend is increasing and related stick is green. Otherwise, the color of the stick is red, which represents a decrease in daily trend.

Data Collection

Labels in this pattern recognition experiment need to be collected by hands, because it is impossible to general labels with mathematic computation as previous section. Therefore, datasets in this experiment is much smaller. 30 American stock indexes are picked, including 'AAPL', 'ABT', 'ABBV', 'ACN', 'ACE', 'ADBE', 'ADT', 'AAP', 'AES', 'AET', 'AFL', 'AMG', 'A', 'GAS', 'ARE', 'APD', 'AKAM', 'AA', 'AGN', 'MSFT', 'GOOG','ALXN','OMC', 'OKE', 'ORCL', 'OI', 'PCAR','DLPH', 'DAL', 'XRAY'. Their intraday data are collected from Google Finance. Time period is from Mar 2017 to Mar 2018.

Label with 'labeling'

In this pattern recognition work, 'labeling', one of the most common software for labeling images, is used in labeling all HAS patterns. In this way, a bounding box is drawn according to the position of HAS patterns, as shown in Fig. 3.15. This

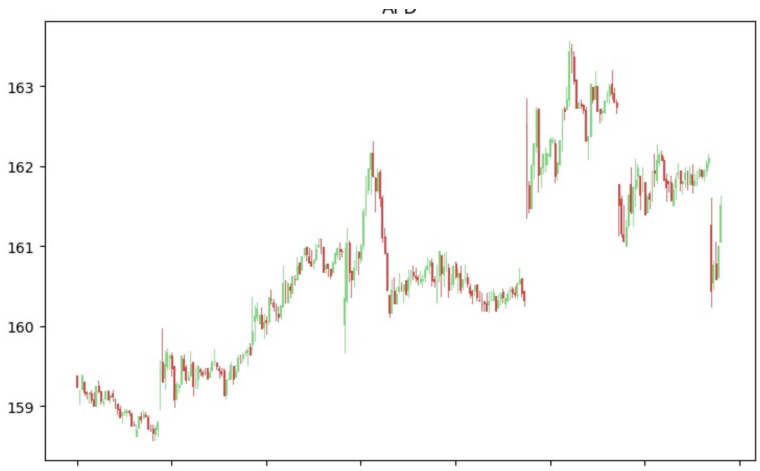

FIGURE 3.14: An example of input image

bounding box two points: left bottom point of the bounding box (Xmin, Ymin) and the right to point of the bounding box (Xmax, Ymax). Different from the previous section of directly stock prediction automatically labeling logarithm of the ratio between future price and current price, these patterns must be labeled by hands.

FIGURE 3.15: An example of labelled image

185 HAS patterns are collected from these 20 stock indexes. 150 them are chosen for training while 35 are used for testing.

3.2.5 Data process

According to experience, deep learning method needs a huge number of images for training. 150 images for training is not enough to get a good result in testing sets. As expected, with only 150 images, the overfitting is significant. The testing loss is much higher than training loss. Two ways of solving this overfitting problem are provided in this work. One of them is using data segmentation to remove noises in the chart and make the chart simpler to be recognized by the model. Another way is to generate more variation data as training data through variation based on those 150 original training images.

Data segmentation

Stock price chart contains a lot of points and noises, it is a time-consuming job to detect the patterns from the stock price chart directly. Is it easier to detect patterns after reducing these noises? But it is difficult to remove those noises on the candlestick chart. Therefore, it should be changed to line chart before fed into the network for training. This process can be described in Fig. 3.16. After changing into line chart, the label is same, therefore it can be trained directly without labeling again. After training, trained model can be used tested with a tested line chart. Then a bounding box would be obtained. Finally, the same bounding box can be drawn, because the bounding box on the line chart and candlestick chart are same.
After obtained line charts, data segmentation method can be used to reduce noises

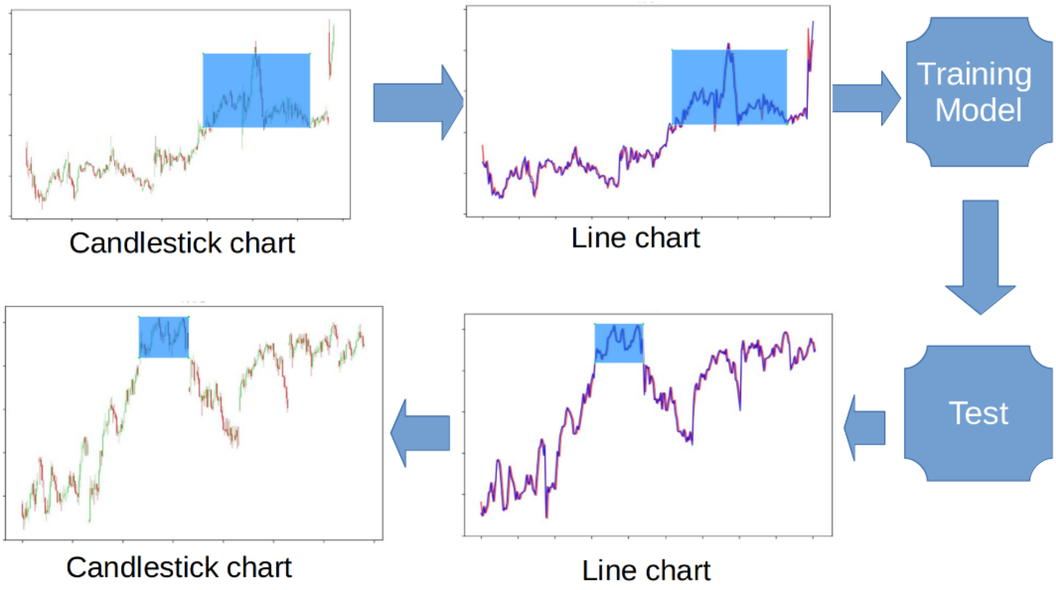

FIGURE 3.16: The processes of using data segmentation approach

online charts. Data segmentation method is normally to be used for a time serious, and stock price problem is a time series problem essentially.

The process of the time series segmentation method in this work can be described in Fig. 3.17. a) From beginning to end, chose the point which has the maximum total distance to begin and added to the segmentation chart. b) Then Chose one segmentation in the segmentation chart. c) Chose another point not in the segmentation chart from original line chart, which has the maximum total distance to begin and end of chosen segmentation. d) Continually do step b) and step c) until got enough points in the segmentation chart. In this way, some noises can be removed and a much simple line chart will be obtained.

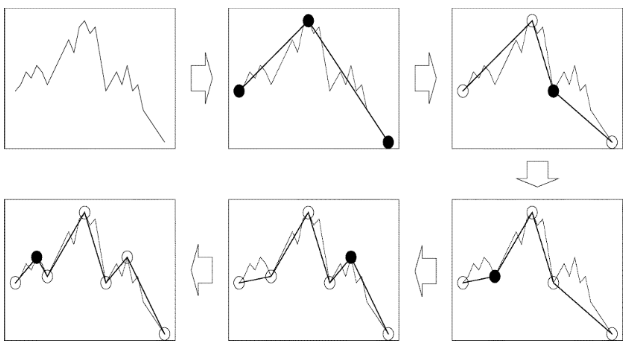

FIGURE 3.17: Segmentation method for time series (Chung et al., 2004)

Finally, much more simple line charts can be obtained without those small noises, like Fig. 3.18. The green line represents the original stock price line, and the blue line represents the new price line after using segmentation method.

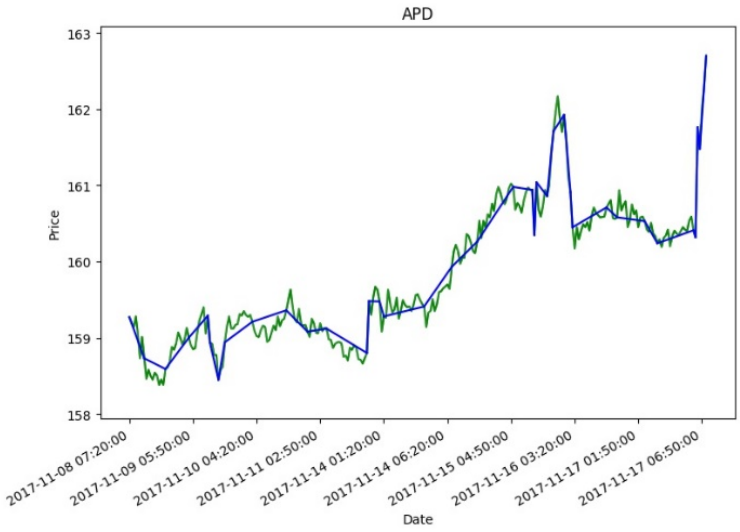

FIGURE 3.18: The result of data segmentation

The pseudo-code of this Segmentation method is:

```
Function PIPlocate (P,Q)
    Input: sequence P[1..m], length of Q[1..n]
    Output: pattern SP[1..n]
Begin
    Set SP[1]=P[1], SP[n]=P[m]
    Repeat until SP[1..n] all filled {
      Select point P[j] with maximum distance
      to the adjacent points in SP
      Add P[j] TO SP }
    Return SP
End
```

Therefore, the whole process of data segmentation approach can be shown in Fig.3.19. a) Change to line chart. b) Remove noises with data segmentation method. c) Training with segmentation charts. d) Testing on segmentation charts. e) Draw bounding box on candlestick chart with output bounding box on segmentation charts.

Generate variation data

Another way for solving to add more training data and this is the most common way in deep learning. However, it is impossible to label thousands or millions of images from the actual datasets that would take too much time. Therefore, this work provides an approach to general some variation data to instead of real stock data for training. This is because even the stock market is volatile, stock pattern recognition is based on the shapes of patterns. The recognition model only considers the shape of the pattern, therefore no matter the shapes from variation data or real data, they are all can be used for training. As shown in Fig.3.20, green circle represents all HAS patterns in the real-world, blue circle means 150 labeled HAS patterns in original images, and the red circle is generated variation data. It is obvious that those 150 images cannot cover all HAS patterns in the real-world, which leads to the overfitting problem. Although generated variation data cannot cover all real HAS patterns and has some data that not included in real data, it has the ability to increase the covered area of training data.

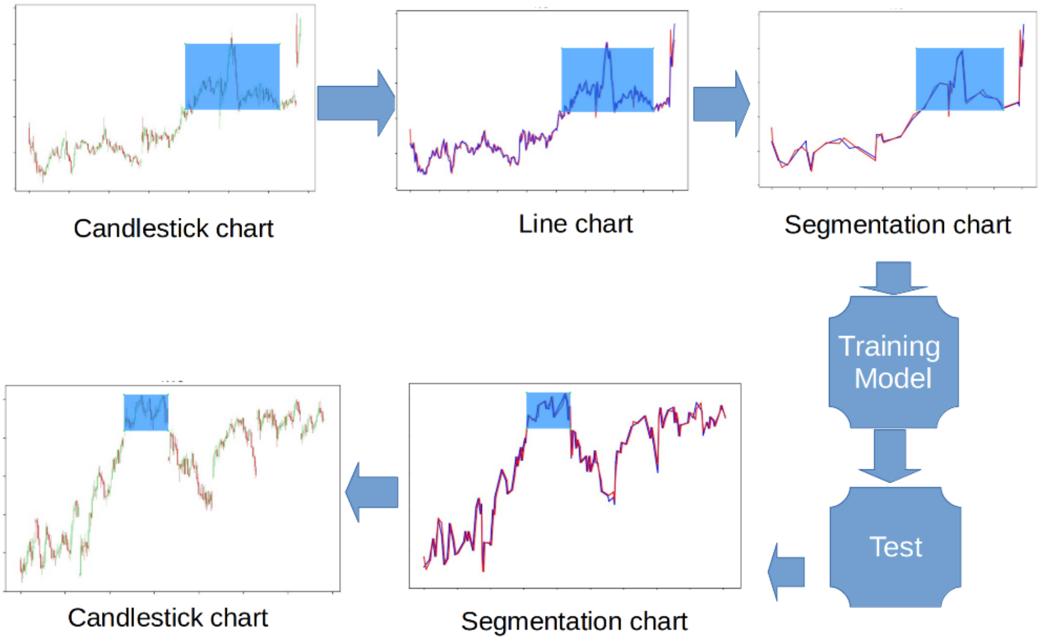

FIGURE 3.19: The whole processes of using data segmentation approach

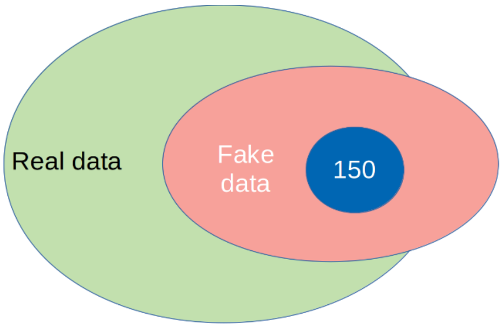

FIGURE 3.20: The relationships between real data and variation data

The generation includes two-step: the generation on X-axis and generation on the Y-axis. In general, variation in this work means to change some values of the HAS pattern to make a difference but still, a HAS pattern. During generation, the positions of HAS does not move, therefore related label can be reused without labeling by hand again. 30 images which are significantly different from each other are chosen from those 150 images. Each one is variated for 30 images on the X-axis and then variated for 10 images on the Y-axis. Therefore there are totally about 9000 variation images.

A variation on the X axis

For a variation on X-axis, the HAS pattern is cut into three parts: first peak, the second peak, and the third peak. According to the definition of HAS pattern, the second peak is the highest peak among those three parts, any variations that leading to the highest peak is not on the second peak (the head) will be removed.

Before variation, 2 random values (r1, r3) are generated with a normal distribution. The reason for using normal distribution random number is that numbers are not distributed uniformly. Normal distribution random numbers are likely to distributed on those areas close to μ, here μ is set as 0 3.21. This means there is a high probability to change prices slightly, but still has some probability to change a lot for prices. This is more close to the real stock market, which does not change too much every time but may have great changes. In addition, a negative number means to delete according to the number, while a positive number means add according to the number.

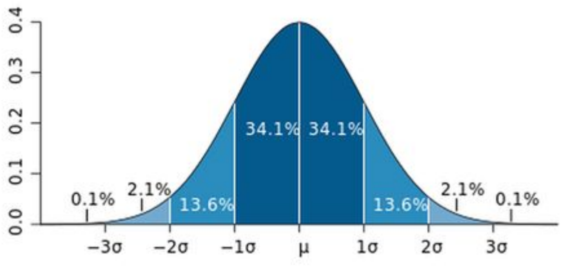

FIGURE 3.21: Normal Distribution

Variation can be divided into 2 steps:

a) P1 = P1 + P1 * r1, P3 = P3 + P3 * r3

b) P2 = P2 - (P1 * r1 + P3 * r3)

The first step means the variation of the first peak and third peak. R1 and r3 mean the percentage that the first peak and third peak will change. P1, P2, and P3 mean each peak. These values are between -0.7 and 0.7, which means each peak can remove or insert at most 70% of their data. If the number is negative, the related peak will remove some data points according to the number uniformly. If the number is positive, related peak first needs to insert data points. Finally, the second peak will insert or delete some data points according to the total changed point. Fig. 3.22 shows the process of variation on X-axis.

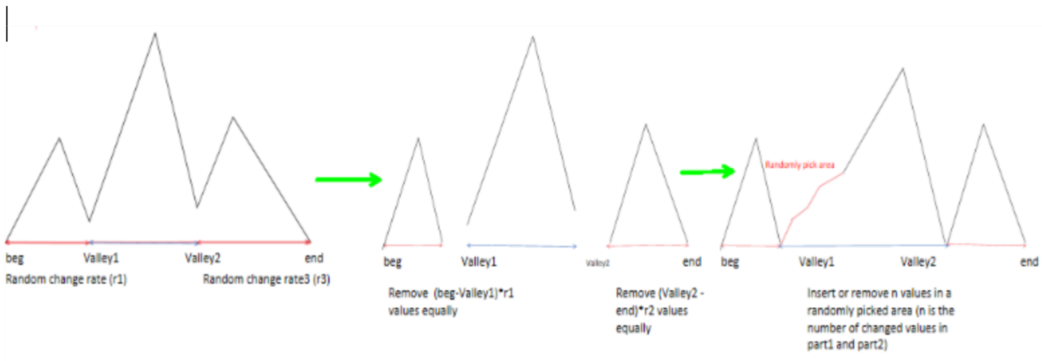

FIGURE 3.22: Variation on X axis

In the process of variation on X-axis, there are two types of variation process: remove and insert. Remove is easy to understand, remove process removes data

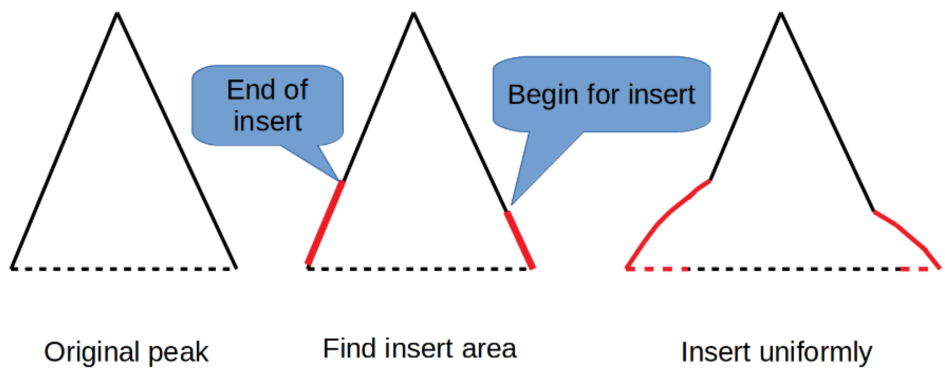

FIGURE 3.23: Insert process of variation on x axis

points uniformly. For example, r1 is -0.3, and there are 30 data points in the first peak. Then 9 data points need to be removed. Data points in the first peak are removed every 3 (30/9) points until 9 data points are removed.

Compared with remove process, there is an additional process before insert. The peak is considered as a circle, then a random position on this peak is chosen as begin for the insert. When the index is out of the range of the peak, it will then back to the beginning of the peak, until inserted enough data points. The way of choosing the beginning position of the insert is a uniform random number. Each point in this has the same probability to be beginning of insert. In addition, the length of the insert is also randomly picked, but the length is always not greater the length of the peak. In other words, the end of the insert is also randomly picked. Then insert values uniformly in a selected area, until inserted enough data points. In this way, many different HAS patterns can be generated. As shown in Fig.3.23, before insert it randomly selects a point as begin, then randomly selects a length or the end position. Finally, insert data points uniformly like remove process in the selected area.

Fig. 3.24 shows some examples of variation on X axis. The first image is the HAS pattern in original image while others are the HAS patterns on generated images. **Variation on the Y axis** Compared with X-axis, a variation on the Y axis is much

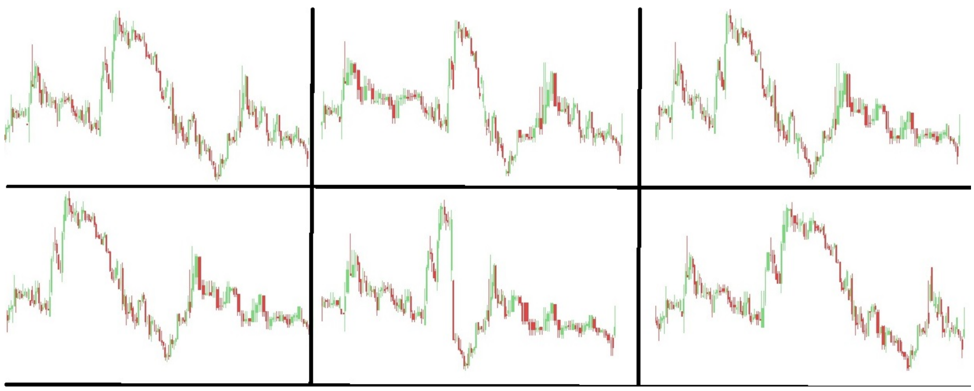

FIGURE 3.24: HAS pattern on original figure and variation figures

easier. It contains two steps:

a) Add a change to Open, Close, High and Low prices equally.

b) Add a different small change to Open, Close, High and Low prices.
The first step represents the total trend change of four prices, while the second step represents the difference between those four prices.

In the first step, all four prices will add a same random value to each point. This random value includes two part: a random number of a normal distribution which can be a positive number and a negative number, and the difference between the maximum value and minimum value in this pattern. This random number cannot be too large or too small. When it is too large, the price on two consecutive days are much different from each other, this would lead to a gap between two consecutive days. When the random is too small it is hard to generate diversity for the patterns. After testing, when setting sigma to 0.05, the variation is under control and there is no space between two consecutive days.

$$\begin{gathered} R_i = random(normal, \sigma = 0.05) \\ x_{i,j} + R_i * (max_x - min_x) \\ (i = 0,1,2 \ldots n, j = open, high, low, close) \end{gathered}$$

In the second step, a smaller random number is added to different prices.

$$\begin{gathered} R_{i,j} = random(normal, \sigma = 0.025) \\ x_{i,j} + R_i * (max_{x_j} - min_{x_j}) \\ (i = 0,1,2 \ldots n, j = open, high, low, close) \end{gathered}$$

After variation on X and Y, the label needs to be changed to adapt to new patterns. If the highest peak in new pattern is lower than original one, the label needs to be narrowed according to the difference between new highest peak and original highest peak. Similarly, if the minimum value in the new pattern is different from previous one, the label also need to be adapted according to the difference. In this way, thousands of images can be generated while labeled automatically.

3.2.6 Evaluation

In this work, AP@0.5IOU is used to evaluate the pattern recognition model. AP@0.5IOU means the average precision when Intersection over Union (IOU) is 0.5. Intersection over Union is a common way to evaluate objective detection method. IOU needs a ground-truth bounding box which is the labeled bounding box area of the pattern. IOU also needs another bounding box which is the predicted bounding box from the model. Fig.**??** shows an example the Ground-true bounding box and predicted bounding box.

IOU score is computed by dividing the area of overlap of two bounding boxes by area of the union, as shown in Fig.3.26. @0.5IOU means when IOU score is greater than 0.5 it is a positive sample, otherwise, it is a negative sample.

AP@0.5IOU shows the accuracy of the model when IOU is 0.5. Generally, there are four results for a prediction: true positives, true negatives, false positives and

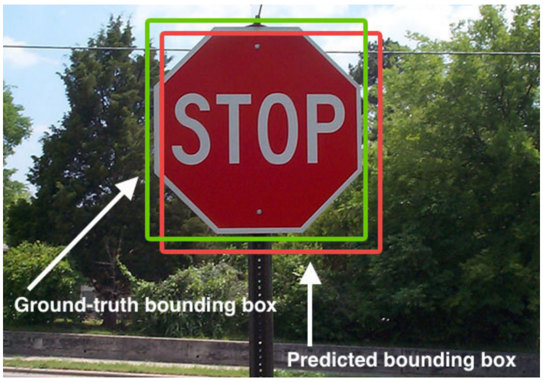

FIGURE 3.25: A ground-truth bounding box and a predicted bounding box (Rosebrock, 2016)

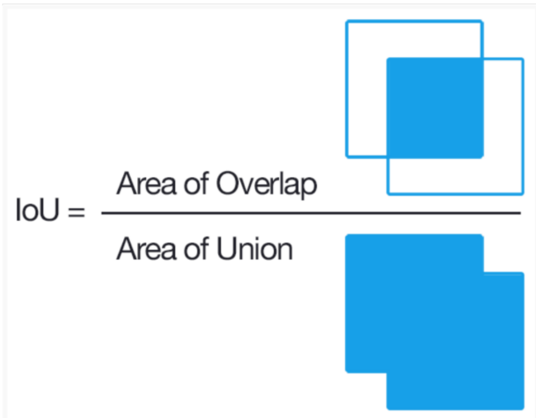

FIGURE 3.26: IOU

false negatives. In this work:

True positives mean HAS pattern is recognized as HAS pattern correctly.

True negatives mean there is no HAS pattern detected in those charts does not contain HAS
patterns.

False positives mean those are not HAS patterns but detected as HAS pattern.

False negatives mean HAS pattens is not recognized by the model

The precision is the ratio of true positives to all recognized images, while recall means the ratio true positives to all patterns.

$$\text{precision} = \frac{tp}{tp + fp} = \frac{tp}{n}$$

Then the average precision is used to represent the performance of the model, and it is calculated with followed formula:

$$\int_0^1 p(r)dr$$

In this formula, p represents the precision and r means the recall.

3.3 Future STOCK PREDICTION WITH DEEP LEARNING WHILE FEEDING 2-D IMAGES

Now it can be been seen that most of the approaches for stock prediction are fed with 1D vectors, no matter history data or news events. For a stock investor, he would look at the images of stock price for more time than the historical data. And for those who want to buy shares, pictures like candlestick chart is more convenient to obtain and easier to understand. Therefore, in this project, a model fed with images about price information would be implemented with machine learning. As it is an image classification model essentially, Convolutional Neural Network (CNN) would be used since its power on image classification problem.

3.3.1 Convolutional Neural Network

Convolutional Neural Network (CNN) was proposed in the 1960s by Hubei and Wiesel when they were studying the neurons of local sensitive and directional selection in the cortical cortex. They found that their unique network structure can effectively reduce the complexity of the feedback neural network. Inspired by this, they proposed CNN. Now CNN has been developed in recent years and becomes a high-efficiency identification method. CNN has also become one of the hotspots in many fields of science, especially in image classification.
For 1D vector, convolution can be represented as follow:

$$y[n] = x[n] * h[n] = \sum_k x[k]h[n-k]$$

x[n] is input signal, h[n] is the unit response while Y[n] is output signal. For 2D signal, the convolution is:

$$y[m,n] = x[m,n] * h[m,n] = \sum_j \sum_i x[i,j]h[m-i,n-j]$$

Therefore, for image processing, convolution is to find the weighted sum for the pixels of the image with given convolution kernel. The main aim of convolution processing is to reduce space cost with the time cost. In general, the more convolution kernels gave, the more feature maps would be extracted and the results should be more accurate, but more parameters would be needed. Pooling Layer In the pooling layer, a value would be used to represent an area. This value can be the average, maximum, minimum value of the area of some other values. The main reason to add this layer is to reduce the parameters.

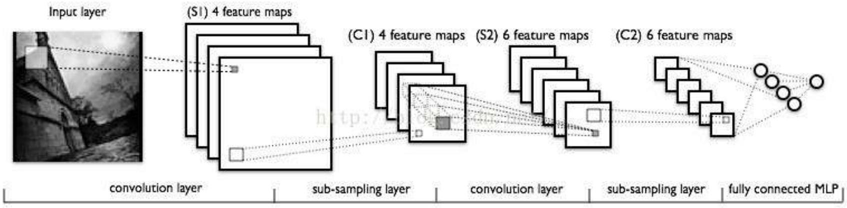

FIGURE 3.27: Convolution for images

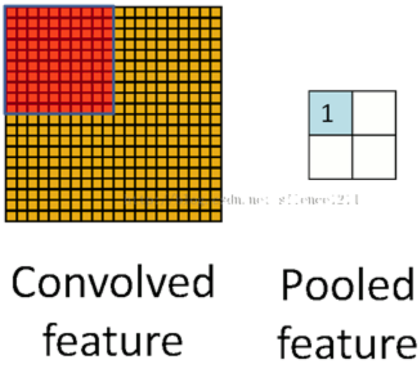

FIGURE 3.28: Pooling layer

3.3.2 Activation layer

Activation functions are used to ensure that the convolved results are in a fixed range. This can keep the value range is controllable after several times convolution. Some common activation functions include sigmoid, tanh, ReLu. The value range of them are [0, 1], [-1, 1] and [0, Infinity]

3.3.3 CNN model

In this paper, two models are used, including Alexnet and VGG19.

Alexnet

This network is provided by Alex (Krizhevsky, Sutskever, and Hinton, 2012) on the imagenet image classification challenge, which won the champion in 2012. This is a classic model of image classification. Therefore, this model is chosen to do the first attempt for stock prediction with images. As showed in Fig.3.29, Alexnet has 5 convolutional layers, 3 fully connected layers and a softmax at the end. The activation function used in this model is ReLU. Kernel sizes are 11, 5 and 3. Fully connected layers' sizes are 4096, 4096 and 1000.

The feature map size is calculated by:

$$new\_feature\_size = (Img\_size - filter\_size + 2 * padding\_size)/Stride + 1$$

For example, for a 224*224 input image, it will be per-processed to 227*227. Then the feature map size after first convolution layer is (227-11)/4+1 = 55. The output number is 96. Therefore, there will be 96 feature maps with size 55*55. The reason why it is 48 in Fig.3 is that they are divided into 2 parts when using 2 GPU. During

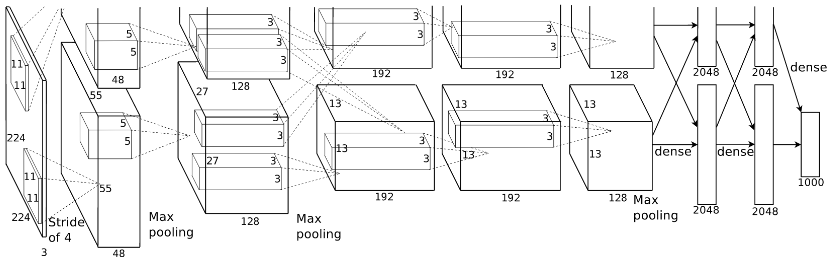

FIGURE 3.29: Alexnet network structure (Krizhevsky, Sutskever, and Hinton, 2012)

the first pooling processing, the output is 96 feature with size (55-3)/2+1 = 27. Similarly in the second convolution layer, the feature size is (27-5+2*2)/1+1 = 27 and the output number is 256. After the last convolution layer and pooling, the output size is 6*6, output number is 256. These neurons would be connected with 4096 neurons in next fully connected laysers.

In the first convolution layer, there are 95 output feature maps. Each of them has 11*11*3 (kernal_size*channel) weights and 1 bias. Therefore in this layer, there are 96*363 + 96= 37824, and 60M parameters in total. Similarly the total parameters of Alexnet is about 60M

VGG19

Then another deeper network called VGG19 is also be implemented. Compared with Alexnet, VGG19 has 16 Conventional layers which is much deeper than Alexnet, but similarly they both use 3 fully connected layers. Another difference is that VGG19 uses fixed kernel size (3*3). They both use same activation ReLU.

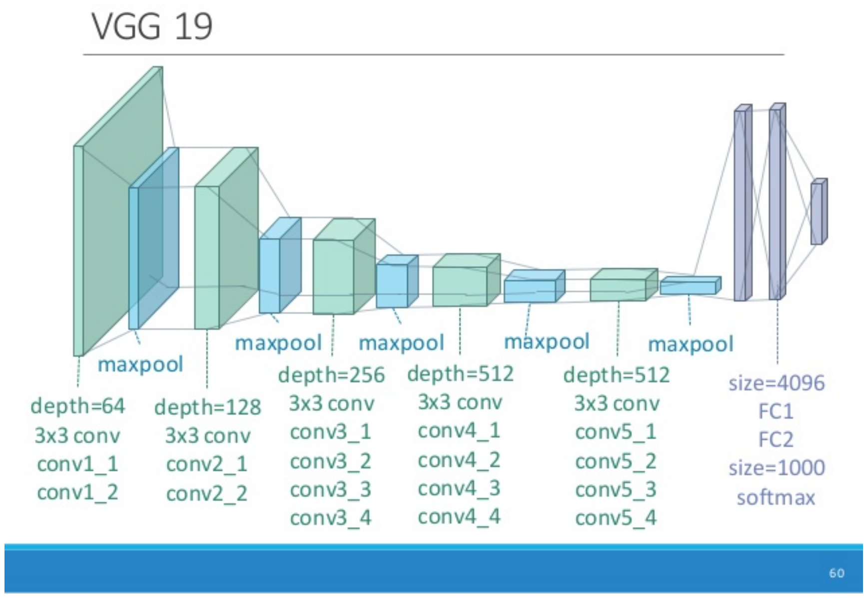

FIGURE 3.30: VGG19 network

The output size and output number can be calculated similarly as introduced in Alexnet. The output number of each layer is much bigger than the previous model. As expected, the parameters of this model are much more than Alexnet. The total parameters of VGG19 are about 144 million.

3.3.4 Data Collection

In this experiment, images represented history data would be used as input. First, history data need to be collected. 3000 stock indexes are randomly chosen from Yahoo Finance. The history data of them in 15 years before 2017 were downloaded. Unfortunately, because of data loss or network issues, only 2397 indexes were downloaded. Then during pre-processing, about 400 indexes were removed because of some 'NAN' value. Before removing them, the closed value had been used to represent these 'NAN' value, but the results are unsatisfied. Therefore, all these datasets with 'NAN' values were removed from the datasets.

Open, close, high and low price is used in the experiment. At first, only the high and low price were used since high and low contains more information according to Siripurapu(Siripurapu, 2014). High and low price are bound of daily stock price, which also contains the open and close price. Open and close is in a sense of statistical artifacts, which are the prices sampled by the Google or Yahoo. However, after tried with only open and close price, the result was not good. Information on input images only with open and close price is not enough for the deep network to extract enough useful features. Therefore, in next experiment, four kinds of the price of the stock are input into the network, then the network can automatically choose which one is more useful.

Then images of stock price would be poled as RGB with size 5*5 kb. The windows size of an image is n 180, 90, 60, 30, which means using n days history data pictures to predict future m 30, 20, 10, 5 days. An example input image is showed as Fig 3.31:

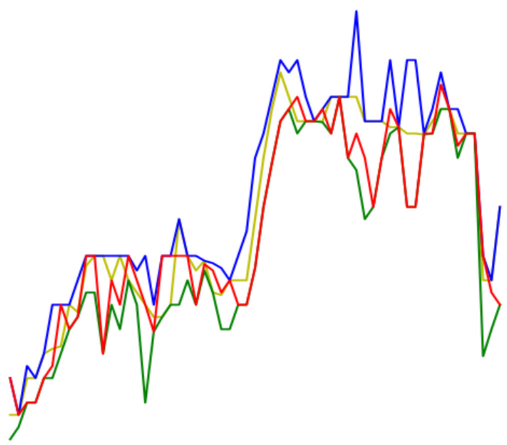

FIGURE 3.31: Input example of n = 60

Then this picture would be stored into Tfrecord and its size would be resized as 224*224.

3.3.5 Prediction strategy

As said previous, prediction strategy is using previous n day's history data pictures to predict future m days. 4 combination of n and m are used including using 180 days history data to predict 30 days price, using 90 days history to predict 20 days price, using 60 days history data to predict 10 days price and using 30 days history data to predict 5 days price.

These pictures would be labeled by the return of the price after m days. The return here is logarithmic return.

$$r_1 = \log\left(\frac{p_2}{p_1}\right).$$

$$p_1 = p(t), p_1 = p(t+m)$$

Price is mean of open, close, high and low price. 't' is current time and 't+m' is the future price. When 'r' > 0, it will be labeled '1' which means 'up', otherwise it is '-1'.

The image numbers for training and testing are always about 8:2. The image number also increases with the increasing of model's depth. When using the Alexnet, the image number is from 1000 to 10000, while using at least 50000 to feeding the VGG19 model.

Cross-entropy is chosen as loss function and optimal is Gradient Descent Optimal (SGD). Learning rate is from 0.01 to 0.00025 while Dropout rate is 0.5.

Chapter 4

Results

4.1 Current stock pattern recognition

In the first step, a model based on Faster-R-CNN and resNet_50 is trained with 150 collected and labeled images. This model can recognize more than half of the HAS patterns on 35 testing images. Fig. 4.1 shows the AP@0.5IOU of HAS pattern on 150 training images, the finally AP value is about 0.64 and the accuracy on test data is 54%. Fig. 4.2 shows some results of test images.

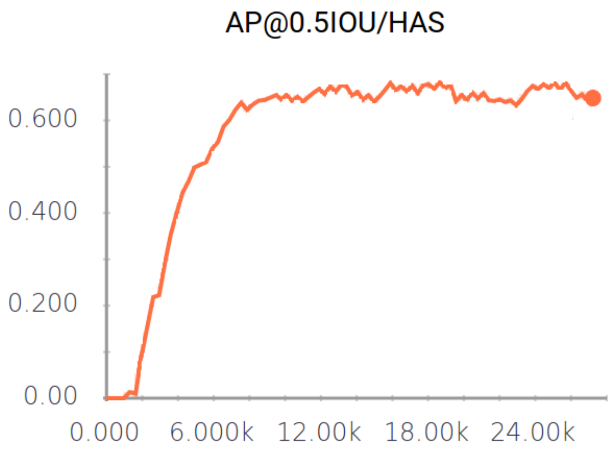

FIGURE 4.1: AP@0.5IOU of 150 training images

However, this first pattern recognition model still exists an over-fitting problem, 150 training images are not enough to train a satisfied model. That is why two other approaches are used to solve this problem. They are segmentation approach and variation data approach. Unfortunately, the performance of the second model with segmentation approach is not improved, it even becomes worse. It can detect more true positives, however, it can also detect more false positive. As shown in Fig.4.3, there is only one true positive at the right corner. Others are all not HAS patterns, but they are recognized. After analysis, this is because after using segmentation approach, noises are removed, a few new noises are also added to the chart. This segmentation approach makes other patterns that are not HAS patterns more similar to HAS patterns. Because to make the chart more simple, a lot of features are also removed. Therefore, the segmentation approach makes it more difficult for the model to detect HAS patterns. That is the reason for a worse performance in the second model with the segmentation approach.

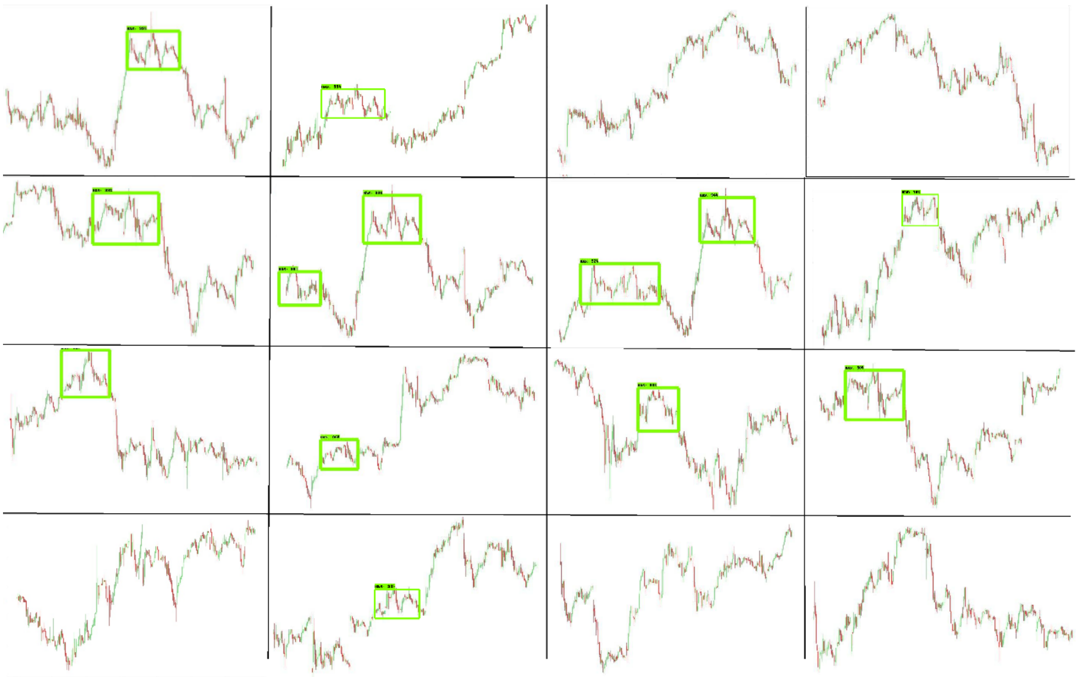

FIGURE 4.2: Some of the testing results of 150 training images

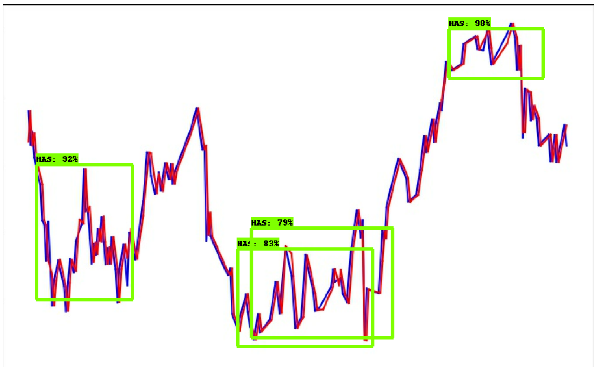

FIGURE 4.3: Results of the segmentation approach

To solve the over-fitting problem and improve the performance of the model, another approach variation data is used in this work as said before. In this work, 8000 variation images based on 30 original training images are used to train the third model. As expected, the performance has increased a lot. The final AP@0.5IOU is approximately 74%, which is a 10% increase. Fig 4.4 shows the AP@0.5IOU HAS pattern on 8000 variation images.

Compared with the model with only 150 original images, the accuracy of model trained with variation images is 74%, which is a great increase. Fig. 4.5 shows the results of the model fed with variation images.

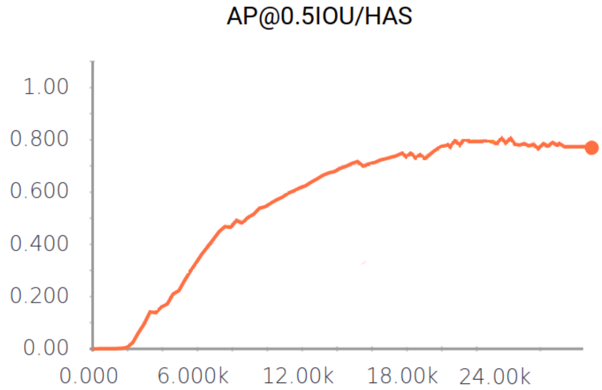

FIGURE 4.4: AP@0.5IOU of 8000 variated images

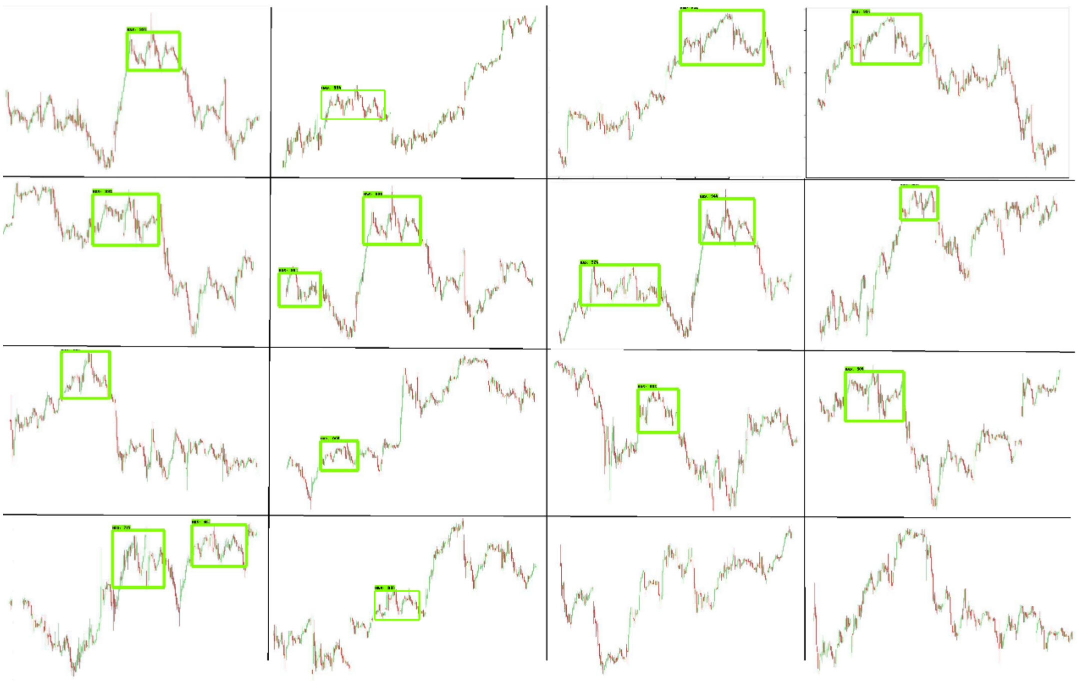

FIGURE 4.5: Some testing results of 8000 variated images

Obviously, the new model with variation data can recognize some patterns which the first model cannot. This is because training dataset can cover more HAS patterns which exist in testing data. For example, the third figure and fourth figure in the first row of Fig. 4.2 are not recognized. The HAS pattern is shown In Fig **??**. This HAS pattern is not covered with 150 original training images.

However, after variation, some variation patterns like Fig. 4.8, and Fig.4.10 are very similar to that test image, although the original images are significantly different from the test images. As shown In Fig 4.7, and Fig. 4.9, they are the original HAS patterns and related variation patterns are shown in Fig. 4.8 and Fig.4.10. Through variation, the training dataset can cover this pattern finally. This the reason for the model fed with variation data has better performance.

FIGURE 4.6: A pattern in figures cannot be recognized with first model

FIGURE 4.7: An image in original datasets

FIGURE 4.8: Variation patterns of Fig. 4.7

FIGURE 4.9: Another image in original datasets

FIGURE 4.10: Variation patterns of Fig. 4.9

4.2 future stock prediction

In the second part, we simply implemented two models with Alexnet and VGG19 for future stock forecasting. After compared the combination of n and m, I found prediction term m should be 5 or 10 days, while history data used to predict should be 30 or 60 days. Using 60 days to predict 10 days has the higher accuracy in this experiment. Then followed experiment is based on 60 days to predict 10 days. With 8 layers model, training an Alexnet model need only a few time which is less than 2 hours. This model converges quickly even with a learning rate of 0.01. However, no matter how to adjust the parameters or change input data (like change n and m, or use different datasets), the model is always overfitting. The accuracy of this model is close to the RW, which is around 50% and the loss is always more than 0.69 as shown in Fig 4.11. However, when I try to increase the training data number, the gradient disappears quickly even with the smallest learning rate 0.00025.

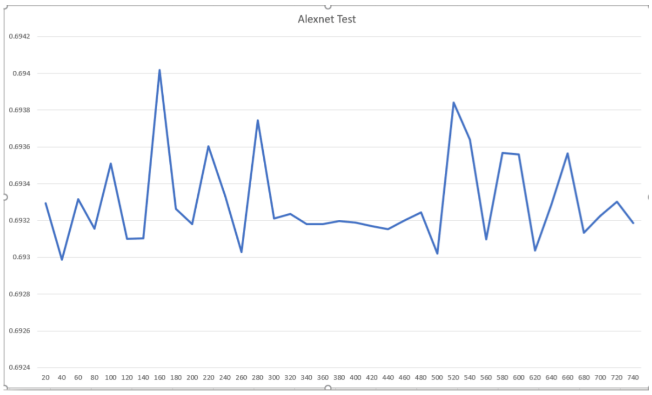

FIGURE 4.11: Test loss of Alexnet

As expected, with much more layers network, VGG19 need much more time to train. Because this mode takes more memory, however, this model does have much better results than the Alexnet. As shown In Fig 4.12 it can achieve the smallest testing error of about 0.61 and with the best test accuracy of about 58%. Unfortunately, with the training goes on, the test loss would increase again, since this model is also overfitting. But it is still better than most previous works, like Poulos (Poulos, 2014)'s 3-layer LSTM (55.45%) and SVM (56.31%) model.

FIGURE 4.12: Test loss of VGG19

Chapter 5

Discussion

In the first part, test dataset is too small. 185 images are collected. 150 of them are used for training, and 35 left images are used for testing. Among these test, some patterns are similar. Because of the limited test dataset, there may exist coincidence in the test data. Therefore, the final test result cannot really represent the performance of the model.

In this work, only one pattern, head and shoulders (HAS) is used for pattern recognition. The reason for it is difficult to find this pattern in stock history data may be HAS pattern appears infrequently. All efforts are put on this pattern, pattern recognition for other patterns are not tried. Patterns like double bottom and double top appear more frequently. Maybe in the future, a multi-classes classification for different patterns can be implemented.

Besides the data segmentation method represented in the paper, another bottom segmentation method (Guo, Liang, and Li, 2007b) is also tried, however, the result still does show any improvement. Beside data segmentation method, there are still some other approaches that can remove noises on lines chart, leading to a simpler chart, like smooth methods. These methods can be tried in the further research.

In the second part of future stock prediction, there still exists a lot of aspects need to be improved. First is about the data collection. 100,000 images for tanning is still not enough, and 1 million pictures are needed for training. History data used in this paper are from 3000 randomly picked stock indexes. That's maybe one issue. As shown in other's work, most of them use similar stocks or a single stock. Therefore, in the future work, stocks in the same sector, like technology sector, would be used for training and testing. Because in the same sector, stocks have more similar features, which can help the model to predict easier.

To avoid all lines being entangled with each other, channels would represent different lines instead of RGB. Then input pictures would be converted into gray images. Therefore the input of model would be 224*224*5. 5 channels mean open, close, high, low and average price. This can be more if the current price is also added in future work.

Also, besides close, open, high and low price, more feature values should be plotted into the pictures, like the average price of all indexes. Currency price is also another significant feature, like AUD, USD, EUR, and others. Currency price represents the economic policy to some extent, while the economic policy is a great influence on stock price.

In addition, this work does not combine current stock patterns recognition and future stock prediction together. Stock patterns recognition is used for stock prediction. Therefore, a stock forecasting approach based stock patterns recognition can be implemented in the future.

Chapter 6

CONCLUSION

In conclusion, this paper represents two approaches based on deep learning techniques for stock market analysis, including head and shoulders (HAS) pattern recognition in current stock market and future stock prediction. Both two approaches use 2D images as input instead of 1D vectors to predict the stock trend. The reason for using 2D images is that images about the stock price like candlestick chart are more often used for stock investors and easier to understand. Compared with feeding with 1D vector, this approach almost does not need any pre-processing, and the model can feed with raw pixels. In addition, images can contain more than one time-series. Another reason is that Convolutional Neural network (CNN) is more stable. In the first patter a pattern recognition for current stock market approach is provided. This approach can help traders at all levels to analyze stock market without much experience in the stock market. It can help investors find HAS patterns from the stock market quickly without many human resources. 185 head and shoulders (HAS) patterns are collected and labeled from 20 stock indexes. FR-CNN is used to train a model with 150 of those images. As expected, with only 150 stock images, AP@0.5IOU is 64%. There still exists the over-fitting problem in this model, because of 150 images are not enough for training. To solve these over-fitting two methods are provided, including data segmentation method and data variation method. Data segmentation aims to remove noises from a simple chart. However, this method leads to a significant increase in false positives. Data variation method is a better way to solve this problem. The model fed with 8000 variation images based on 30 original images has an AP@0.5IOU of 74% which increased 10% compared with the model with 150 original images, and it is better than the best results of previous work, Yang (Yang, 2009)'s 70%. The final aim of stock pattern recognition is future stock forecasting. Therefore, in the second part, a stock forecasting approach based on Alexnet and VGG 19 is implemented. As expected, the model with VGG 19 gets a better prediction accuracy which is 58%. It still exists over-fitting problem, since it is a simple implementation, but is still better than most previous work, like Poulos's (Poulos, 2014) 3-layer LSTM (55.45%) and SVM (56.31%) model.

Bibliography

Alkhateeb, Sara (2016). "Cyber Crimes". In: *International Journal of Scientific & Engineering Research* 7.4, p. 918.

Chang, Pei-Chann et al. (2009). "A neural network with a case based dynamic window for stock trading prediction". In: *Expert Systems with Applications* 36.3, pp. 6889–6898.

Chung, Fu-Lai et al. (2004). "An evolutionary approach to pattern-based time series segmentation". In: *IEEE transactions on evolutionary computation* 8.5, pp. 471–489.

Dai, Yuqing and Yuning Zhang (2013). *Machine Learning in Stock Price Trend Forecasting*.

Ding, Xiao et al. (2015). "Deep learning for event-driven stock prediction." In: *Ijcai*, pp. 2327–2333.

Dover, Kathryn (2017). "Pattern Recognition in Stock Data". In:

Girshick, Ross (2015). "Fast r-cnn". In: *arXiv preprint arXiv:1504.08083*.

Girshick, Ross et al. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Guo, Xinyu, Xun Liang, and Nan Li (2007a). "Automatically Recognizing stock patterns using RPCL neural Networks". In:

Guo, Xinyu, Xun Liang, and Xiang Li (2007b). "A stock pattern recognition algorithm based on neural networks". In: *Natural Computation, 2007. ICNC 2007. Third International Conference on*. Vol. 2. IEEE, pp. 518–522.

Guresen, Erkam, Gulgun Kayakutlu, and Tugrul U Daim (2011). "Using artificial neural network models in stock market index prediction". In: *Expert Systems with Applications* 38.8, pp. 10389–10397.

Investopedia, LLC (2015). *Investopedia."*

Kamijo, Ken-ichi and Tetsuji Tanigawa (1990). "Stock price pattern recognition-a recurrent neural network approach". In: *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*. IEEE, pp. 215–221.

Kar, Abhishek (1990). "Stock prediction using artificial neural networks". In: *Dept. of Computer Science and Engineering, IIT Kanpur*.

Keogh, Eamonn J and Padhraic Smyth (1997). "A Probabilistic Approach to Fast Pattern Matching in Time Series Databases." In: *Kdd*. Vol. 1997, pp. 24–30.

Kim, Kyoung-jae (2003). "Financial time series forecasting using support vector machines". In: *Neurocomputing* 55.1-2, pp. 307–319.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*, pp. 1097–1105.

Madge, Saahil and S Bhatt (2015). "Predicting Stock Price Direction using Support Vector Machines". In: *Independent Work Report Spring*.

Poulos (2014). "Predicting stock market trends by recurrent deep neural networks". In: *Pacific rim international conference on artificial intelligence*. Springer, pp. 759–769.

Preminger, Arie and Raphael Franck (2007). "Forecasting exchange rates: A robust regression approach". In: *International Journal of Forecasting* 23.1, pp. 71–84.

Rajagopal, Suresh (2016). “Pattern Extraction in Stock Market data”. PhD thesis. University of Colorado at Colorado Springs.

Robert, Christian (2014). *Machine learning, a probabilistic perspective*.

Rosebrock, A (2016). “Intersection over Union (IoU) for object detection”. In: *Online] http://www. pyimagesearch. com/2016/11/07/intersection-overunion-iou-for-object-detection*.

Shah, Vatsal H (2007). “Machine learning techniques for stock prediction”. In: *Foundations of Machine Learning | Spring*, pp. 1–19.

Shen, Shunrong, Haomiao Jiang, and Tongda Zhang (2012). “Stock market forecasting using machine learning algorithms”. In: *Department of Electrical Engineering, Stanford University, Stanford, CA*, pp. 1–5.

Siripurapu, Ashwin (2014). *Convolutional networks for stock trading*.

Wang, Jian-Zhou et al. (2011). “Forecasting stock indices with back propagation neural network”. In: *Expert Systems with Applications* 38.11, pp. 14346–14355.

Yang, Yiwen (2009). “Pattern recognition based on support vector machine: Computerizing expertise for predicting the trend of stock market”. In: *Computer Science and Information Engineering, 2009 WRI World Congress on*. Vol. 6. IEEE, pp. 60–66.

Yu, Tiffany Hui-Kuang and Kun-Huang Huarng (2008). “A bivariate fuzzy time series model to forecast the TAIEX”. In: *Expert Systems with Applications* 34.4, pp. 2945–2952.