
ENCODING CANDLESTICKS AS IMAGES FOR PATTERN CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

A PREPRINT

Jun-Hao Chen

Department of Computer Science & Information Engineering
National Taiwan University
Taipei 10617, Taiwan

Yun-Cheng Tsai

School of Big Data Management
Soochow University
Taipei 11102, Taiwan
pecutsai@gm.scu.edu.tw

June 2, 2020

ABSTRACT

Candlestick charts display the high, low, opening, and closing prices in a specific period. Candlestick patterns emerge because human actions and reactions are patterned and continuously replicate. These patterns capture information on the candles. According to Thomas Bulkowski's Encyclopedia of Candlestick Charts, there are 103 candlestick patterns. Traders use these patterns to determine when to enter and exit. Candlestick pattern classification approaches take the hard work out of visually identifying these patterns. To highlight its capabilities, we propose a two-steps approach to recognize candlestick patterns automatically. The first step uses the **Gramian Angular Field (GAF)** to encode the time series as different types of images. The second step uses the Convolutional Neural Network (**CNN**) with the GAF images to learn **eight critical kinds of candlestick patterns**. In this paper, we call the approach GAF-CNN. In the experiments, our approach can identify the eight types of candlestick patterns with 90.7% average accuracy automatically in real-world data, outperforming the LSTM model.

Keywords Convolutional Neural Networks (CNN) · Gramian Angular Field (GAF) · Candlestick · Patterns Classification · Time-Series · Financial Vision.

1 Introduction

Financial market forecasts are critical research topics in commercial finance and information engineering. For example, the topics are predicting fluctuations or volatility forecasts for futures indices [1]. Market prices are susceptible to the expected psychological impact of the overall market. These prices are possible to develop predictive models of financial demand through particular pre-processing and complex model architectures.

Many tools are existing to help people predict stock price fluctuations and futures indices already [2]. For example, these tools are the neural networks, fuzzy time-series analysis, genetic algorithms, classification trees, statistical regression models, and support vector machines. However, these machine learning models are generic techniques and used for forecasting. They unusually combine with financial expertise [1]. Because the average person pursues profit in any transaction, the predictions of such models are not accurate enough for real-world operations. Investment forecasts and model predictions tend to have significant gaps, and investors are more inclined to find a good entry and exit point rather than merely predicting prices. Many studies focus on the accuracy of numerical predictions [3, 4, 5, 6, 7, 8], but investors only concern with the time of entry and exit (i.e., how much profit space they have). In other words, rather than blindly using machine learning or deep learning architecture to pursue unrealistic low-risk, high-accuracy profit models, it is better to combine these directly with a basic knowledge of transactions to create a reliable, applicable model [2, 9].

Candlestick pattern recognition is an essential tool for determining market conditions [10]. To make trading decisions, traders often make judgments based on much-complicated information, such as technical indicators, news, and candlestick patterns. Thus, candlestick pattern recognition is a crucial support for individual transactions [11]. Candlestick pattern recognition helps traders determine the current asset price in the market and establish whether the current buying pressure will continue or whether the current selling pressure will reverse. This information, along with other sources, assists traders to predict the future. Concerning price trends, the Morning Star and the Evening Star are examples of price reversal signals commonly. Candlestick pattern recognition requires a deliberate analysis of trader expertise rather than pure numerical analysis. This recognition requires traders to make visual judgments on images.

The Convolutional Neural Network (CNN) model is well-suited to image recognition [12]. CNN can update its convolution kernel by backward propagation and train the appropriate weights to extract excellent image features. The correlation between traits and images uses to help models make correct judgments. Further, the type of neural network suitable for image identification needs to carry out through a two-dimensional convolution. Principally, the financial time-series data representing uses a one-dimensional array. Therefore, we need to find a way to convert the time-series data into a consistent matrix form.

However, our datasets are always dynamic, and patterns in them are changing. Hence, we need to feature engineering to extract specific time-series features. For example, space transformation models are kinds of feature engineering. There are including Singular Value Decomposition (SVD), distance metric learning, Nyström methods, and Distance Metric Learning (DML) approach [13]. The process of Singular Value Decomposition (SVD) uses for investigation of the data. In these methods, linear algebra uses to construct a data matrix out of the collected data and to extract intrinsic features of that matrix. It is to separate elements that are similar between each subject and features that differentiate the items.

Instances with different labels are intertwined and often linearly inseparable. This issue brings new challenges to the CNN approach [14, 15]. The CNN approach considers unsuitable for directly encoding the time-series data as image pixels [16]. Hence, we need a method for transforming time series data into images.

The Gramian Angular Field (GAF) has the following advantages:

1. The GAF provides a way to preserve temporal dependency since time increases as the position moves from top-left to bottom-right.
2. The GAF contains temporal correlations because the Gramian Angular represents the relative correlation by superposition and difference of directions for the time interval.
3. The primary diagonal of the Gramian Angular Field matrix is the particular case.
4. The diagonal of the Gramian Angular Field matrix contains the original value and angular information.
5. From the main diagonal, we can reconstruct the time series from the high-level features learned by the deep neural network.

Hence, we use the **Gramian Angular Field (GAF) to encode the time-series data** [17] from a one-dimensional time-series array to the two-dimensional convolutional time-series matrix. The encoding data can improve the performance of the neural network in the two-dimensional convolutional time-series significantly. When the CNN model uses the GAF encoding as input, the LeNet [18] architecture can achieve outstanding results naively.

Therefore, we design a GAF-based CNN to emulate the trader to identify candlestick pattern characteristics in an experiment. We call our approach GAF-CNN. First, we use the Geometric Brownian Motion (GBM) model to simulate a volume of price data. According to Zhiguo, we set the same parameters to set the price, and its volatility is close to the real data [19]. Second, we choose eight candlestick patterns from The Major Candlestick Signals [20]. These eight types of pointers are Morning Star, Bullish Engulfing, Hammer, Shooting Star, Evening Star, Bearish Engulfing, Hanging Man, and Inverted Hammer. The difference between these eight candlesticks signals is subtle and will challenge a traditional CNN model.

To improve the traditional CNN model, we use the GAF-CNN to train the GBM simulation data. Our model produces outstanding performance in the simulation data. We also use real data to verify the viability of our GAF-CNN in the real-world. We expect that GAF-CNN enables the computer to look at the candlestick patterns with as much nuance as a trader. The results show a near-92% accuracy for the GBM simulation data. We use 2010-2017 historical data of the currency exchange rate for Euro (EUR) to US dollar (USD) to test our GAF-CNN model. The experimental results achieve a 90.70% accuracy. The simulation and experimental results show that GAF-CNN is suitable for shape identification in financial trading. Although this paper uses only eight of the most classical-type indicators, various morphological extensions that can be made based on GAF-CNN are feasible, such as the W-head M-bottom. We want to establish a financial vision field through this paper making computers can recognize candlestick as a human has seen.

The remainder of this paper organizes as follows. Section 2 provides a review of the literature, and Section 3 presents our methodology. Section 4 shows the result of our experiments. Section 5 describes the discussion of Section 4. Section 6 is the conclusion of our study, and Section 7 is the overall workflow of our experimental framework.

2 Preliminary

2.1 Candlestick

Japanese start using technical analysis to trade rice in the 17th century [21]. While this early version of technical analysis is different from the US version initiated by Charles Dow around 1900. Many of their guiding principles are similar. In this version, price action is more important than news and earnings. All happened information reflects in the price already. Buyers and sellers move markets based on expectations and emotions. The actual price may not reflect the underlying value. According to Steve Nison, candlestick charting first appears sometime after 1850 [22]. Much of the credit for candlestick development and mapping goes to a legendary rice trader named Honma from the town of Sakata [23]. His original ideas are likely modified and refined over many years of trading, eventually resulting in the system of candlestick charting used today.

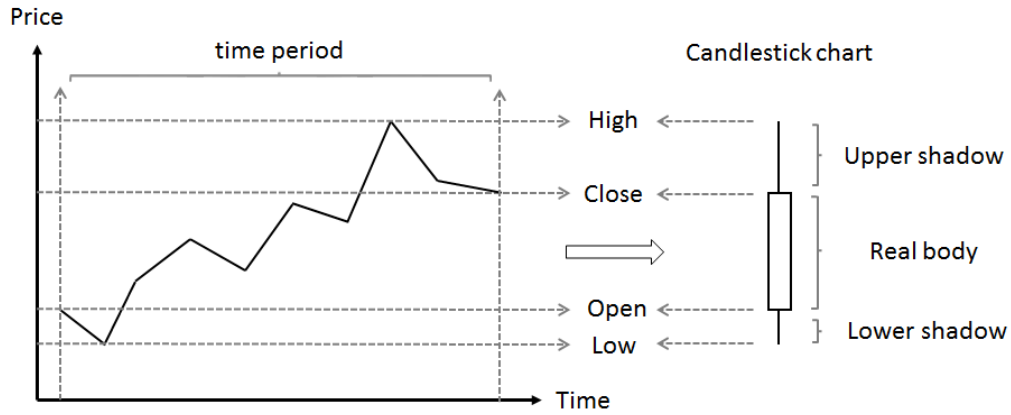


Figure 1: Candlesticks display all the market needed information, such as opening, closing, high, and low prices.

Figure 1 is the structure of a candlestick. The unit is the bar, which draws on the opening, high, low, and closing prices (OHLC) for a specified period. The real-body is the price difference between the opening and closing prices. The upper shadow is the price difference between the highest price and the real-body, and the lower shadow is the price difference between the lowest price and the real-body. The period of a bar can be arbitrarily customized, usually depending on the length of the transaction. If the open price is higher than the close price, the real-body is rendered in black, indicating that the price is falling during this time. If the close price is higher than the open price, the real-body is white, indicating that the price is rising during this time. If the close price is equal to the opening price, the real-body will be just a (horizontal) line.

From the above, the candlestick helps investors filter out much of the price noise. The bar only records the different price information of OHLC per unit time. When we put together multiple bar charts, we get a continuous market information map. Unique shapes call as a pattern.

Researchers focus on the topic of candlesticks for many years [22]. Many patterns use to identify trends summarized, such as trend continuation indicators or reversal indicators. Candlestick analysis is an approach to getting started with trading. However, some people think it is challenging to observe the trend by observing the candlestick. It cannot use as an indicator to predict direction [24]. Human begins to systematize the patterns generated from the candlesticks. They evolve into technical indicators of the system to form the candlestick patterns gradually. The indicators are also including the Average True Range (ATR), Relative Strength Index (RSI), Moving Average (MA), Moving Average Convergence and Divergence (MACD), Stochastic Oscillator (KD) [25] and so on.

2.2 Convolutional Neural Networks (CNN)

CNN models take advantage of the spatial properties of the data. According to Fukushima and Miyake, they propose a Neocognitron model. The model considers inspiring CNNs from the computational perspective generally [26].

Neocognitron is a neural network designed to simulate the human visual cortex [26], which consists of two types of layers. The first type is the feature extractor layers, and the second type is the structured connection layers. The feature extractor layers, also named S-layers, simulate the cell in the primary visual cortex and help human beings to perform feature extraction. The structured connection layers, also named C-layers, affect the complex cell in the higher pathway of the visual cortex, provide the model with its shifted invariant property.

The two most essential components of CNN are the convolutional layer and the pooling (Pool) layer. Figure 2 shows that the convolutional layer implements the convolutional operation, which extracts image features by computing the inner product of an input image matrix and a kernel matrix. The number of channels of the input image and kernel matrix must be the same. For example, if the input image is a red-green-blue (RGB) color space, then the depth of the kernel matrix must be three; otherwise, the kernel matrix cannot capture the information between different color spaces. The pooling layer, also called the sub-sampling layer, is mainly in charge of simplifying the task. Figure 3 shows that the pooling layer only retains part of the data after the convolutional layer. It reduces the number of significant features extracted by the convolutional layer and refines the remaining features.

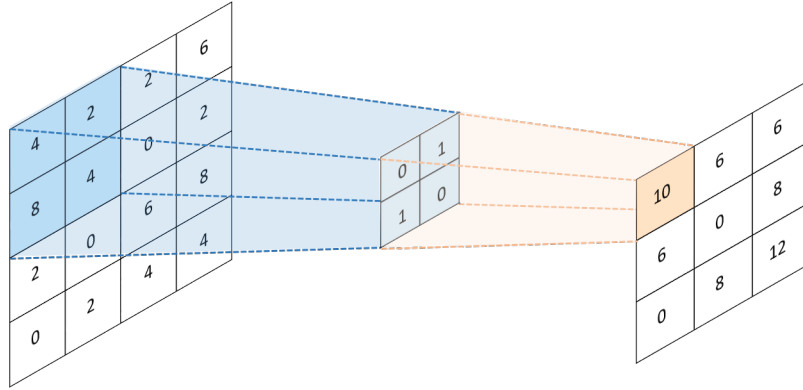


Figure 2: The **convolutional operation**.

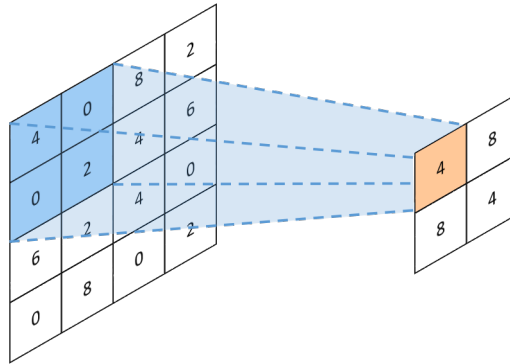


Figure 3: The **pooling operation**.

Only with these two components can the convolutional model be used to imitate human vision. In practical applications, the CNN model usually combines the convolutional layer and the pooling layer. The convolutional layer often extracts a significant number of features, and most of the elements may be noise, which could lead to the model learning in the wrong direction, also known as over-fitting. Furthermore, the fully-connected layers connect at the end of the sequence usually. The function of the fully-connected layer organizes the extracted features processed by the convolutional and pooling layers. The correlation between the extracted features learns in this layer.

Although the pooling layer can reduce the occurrence of over-fitting after convolution, it is inappropriate to use after the fully-connected layer. The other widely recognized regularization technique, called drop-out, designs to solve this issue. The drop-out technique randomly drops neurons with a specific probability, and the dropped neurons are not involved in the forwarding and backward computation. This idea directly limits the model's learning; the model can only update its parameters subject to the remaining neurons in each epoch.

The most general classic modern CNN model, LeNet inspires by Neocognitron and the concept of backpropagation [18]. The potential of the modern convolution architecture can be seen in LeNet [27], consisting of a convolution layer, a subsampling layer, and a full connection (FC) layer [28]. Figure 4 shows the LeNet model. As the concept of the rectified linear unit (ReLU) and drop out are presented in recent years, a new convolution-based model, AlexNet, proposed by Alex Krizhevsky and Hinton [29], appeared and beat the previous champion of the ImageNet Challenge, with 10M labeled high-resolution images and 10,000+ object categories.

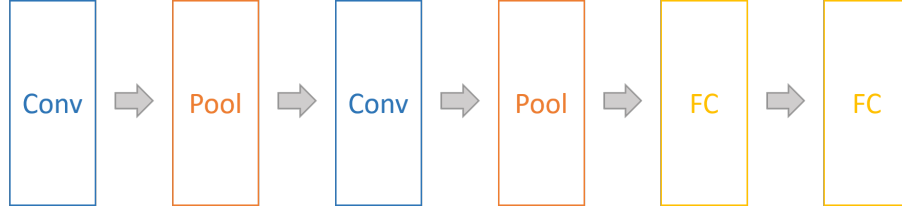


Figure 4: The classic LeNet model.

2.3 CNN for Patterns Classification

Human beings are visual creatures. The eyes are the most compact structure of all the sensory organs, and the visual intelligence of the human brain is rich in content. Exercise, behavior, and thinking activities all use visual sensory data as their most significant source of information. The more flexible and talented we become, the more we rely on visual intelligence. What general business and decision-makers desire after the analysis is not the data itself, but the value. Therefore, data analyses must be intuitive. In this way, the visualization of financial data more readily accept: decision-makers can see the story and interpret the data more efficiently.

Although visualization analysis can benefit decision-makers, many traditional statistical or machine learning methods for predicting currency movements use quantitative models. These methods do not consider visualization. We attempt to make good use of the advantages of display and comprehensively enhance the efficiency of intelligence analysis. For example, most traders use charts to analyze and predict currency movement trends, which carry apparent economic benefits. However, in this visualization, the analysis is artificial. We aim to teach machines to achieve the interpretation of visual information like a human brain. We then hope to use the tool to analyze robust financial data visually.

The CNN models use in pattern and image recognition problems widely. In these applications, the best possible accuracy has achieved using CNNs. For example, the CNN models have achieved a accuracy of 99.77% using the Modified National Institute of Standards and Technology (MNIST) database of handwritten digits [30], a accuracy of 97.47% with the New York University Object Recognition Benchmark (NORB) dataset of 3D objects, and a accuracy of 97.6% on over 5,600 images of more than ten objects. The CNN models not only give the best performance compared to other detection algorithms but also outperform humans in such cases as classifying objects into fine-grained categories, such as particular breeds of dogs or species of bird. The two main reasons for choosing a CNN model to predict currency movements are as follows:

1. The CNN models are good at detecting patterns in images, such as lines. We expect that this property can use to detect trends in trading charts.
2. The CNN models can detect relationships among images that humans cannot find easily. The structure of neural networks can help detect complicated relationships among features.

2.4 Gramian Angular Field (GAF)

GAF is a novel time-series encoding method proposed by Wang and Oates [17], which represents time series data in a polar coordinate system and uses various operations to convert these angles into symmetry matrix. Gramian Angular Summation Field (GASF) is a kind of GAF using the cosine function. Each element of the GASF matrix is the cosine of the summation of angles.

Our first step to making a GAF matrix is to **normalize the given time series data X into values between $[0, 1]$** . The following equation shows the simple linear normalization method, where notation \tilde{x}_i represents the normalized data.

$$\tilde{x}_i = \frac{x_i - \min(X)}{\max(X) - \min(X)} \quad (1)$$

After normalization, our second step is to represent the normalized time series data in the **polar coordinate system**. The following two equations show how to get the angles and radius from the rescaled time series data.

$$\phi = \arccos(\tilde{x}_i), -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} \quad (2)$$

$$r = \frac{t_i}{N}, t_i \in \mathbb{N} \quad (3)$$

Finally, we **sum the angles and use the cosine function to make the GASF** by the following equation:

$$\text{GASF} = \cos(\phi_i + \phi_j) = \tilde{X}^T \cdot \tilde{X} - \sqrt{I - \tilde{X}^2}^T \cdot \sqrt{I - \tilde{X}^2} \quad (4)$$

The GASF has two essential properties. First, the mapping function from the normalized time series data to GASF is bijective when $\phi \in [0, \pi]$. In other words, normalize data to $[0, 1]$ can transform the GASF back into normalized time series data by the diagonal elements. Second, in contrast to Cartesian coordinates, the polar coordinates preserve absolute temporal relations.

3 Methodology

This section begins with the overall experiment design, then illustrates the method of label creation, GAF-CNN model, feature selection, and neural architecture searching, respectively.

3.1 Experiment Design

Considering real-world data lacking and complexity, it starts with **simulation data** to ensure GAF-CNN model work and progress feature selection and neural architecture search. Further, it will adopt in the empirical research on real-world data.

The simulation data are including the 2000 training data, 400 validation data, and 500 testing data from the Geometric Brownian Motion (GBM) model. Furthermore, we use EUR/USD 1-minute price data from January 1, 2010, to January 1, 2018, to label the real-world data, including 1000 training data, 200 validation data, and 350 testing data.

3.2 Illustration of Label Creation

We select eight of the most classic candlestick patterns based on a classic candlestick patterns textbook, The Major Candlesticks Signals, as our training target. The eight candlestick patterns we chose are Morning Star, Bullish Engulfing, Hammer, Shooting Star, Evening Star, Bearish Engulfing, Hanging Man, and Inverted Hammer. All of these patterns are reversal patterns, which capture whether the price is going to change. The first four patterns detect the price from downtrend to uptrend, and the last four patterns detect the opposite. We illustrate Morning Star and Evening Star as examples below.

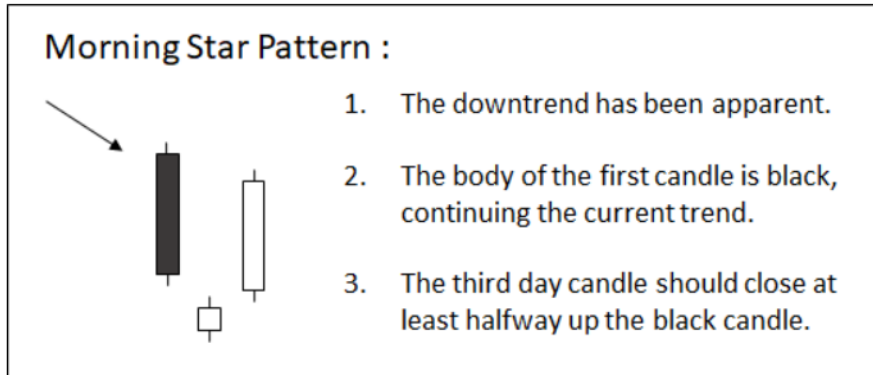


Figure 5: The left-hand side shows the appearance of the Morning Star pattern. The right-hand side shows the critical rules of the Morning Star pattern.

The Morning Star pattern detects a price changing from a downtrend to an uptrend. The description of this pattern has three stages. First, a downtrend must be confirmed, which means the whole market has an absence of confidence.

Second, the depressed atmosphere results in a big black bar. After a calm day, the third bar is a big white bar, which indicates that the investors expect the confidence of the market to reverse. Figure 5 shows the main appearance and rules of Morning Star in detail.

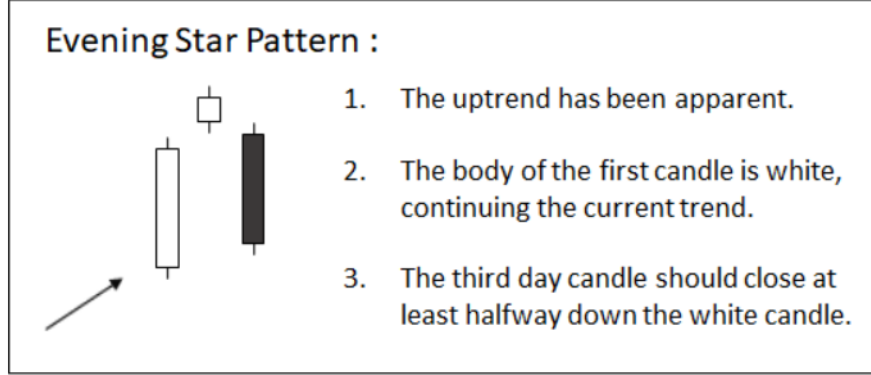


Figure 6: The left-hand side shows the appearance of the Evening Star pattern. The right-hand side shows the critical rules of the Evening Star pattern.

The Evening Star pattern detects the price changing from **an uptrend to a downtrend**. The description of this pattern also has three stages. First, an uptrend must be confirmed, which means the whole market is in a specific situation. Second, good days end with a big white bar. After a calm day, the third bar becomes a big black bar. These indicate that the investors expect the confidence of the market to reverse. Figure 6 shows the main appearance and rules of Evening Star in detail, and Figure 7 shows the difference between Morning Star and Evening Star patterns.

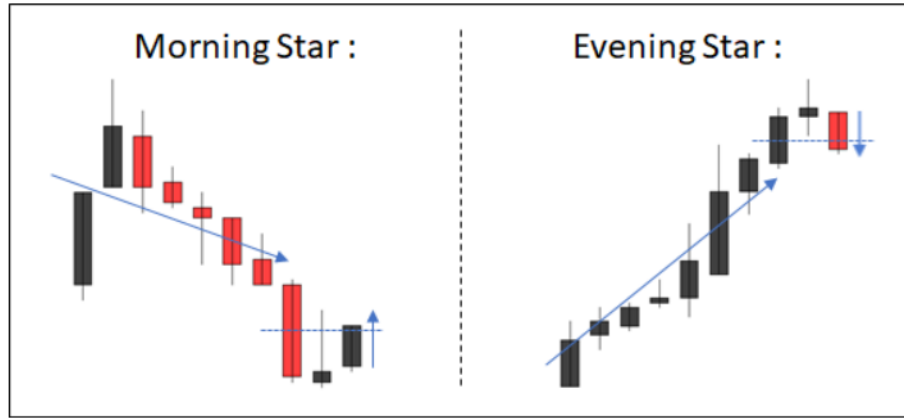


Figure 7: The morning star and evening star patterns recognize in real-world data.

The definition of our label bases on the rules given in The Major Candlesticks Signals, as shown in Figures 5 and 6. The downtrend and uptrend define from regression. **If the slope is higher or lower enough, the trend is confirmed**. The definition of slope in our implementation is as follows, Figure 8 has the entire illustration:

1. The slope value computes from the closing price among 7 bars.
2. Move a bar window to get another slope value.
3. Keep collecting positive and negative slope until 50 units, respectively.
4. If the current slope is over the 70th percentile of the group, then it will be defined as a positive or negative trend.

We must note that the other pattern rules are slightly different between the simulation and the real data. The rules from the simulation data are similar to the book. Nevertheless, the number of samples is insufficient in real-world data

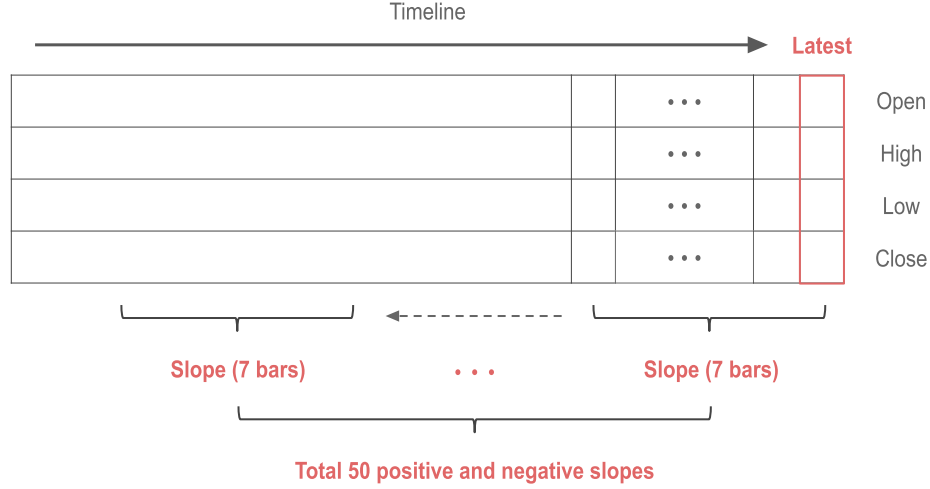


Figure 8: The flowchart of our slope definition.

PARAMETERS	VALUES
epochs	300
batch size	64
optimizer	Adam
learning rate	0.001
beta 1	0.9
beta 2	0.999
early stopping	20 epochs

Table 1: The parameters of our GAF-CNN model.

because of the strictness of the rules. Hence, we relax the rules to obtain sufficient data slightly. For example, the Bullish Engulfing pattern requires the opening price of the last bar to be lower than the closing price of the previous bar. If this rule is too strict, we relax the condition such that the opening price of the last bar only needs to be less than or equal to half of the real body of the previous bar.

3.3 GAF-CNN

We propose a two-step approach and call it the GAF-CNN model. The first step is the Gramian Angular Summation Field (GASF) time-series encoding, and the second step is the Convolutional Neural Networks (CNN) model. In the first step, we encode time series data based on opening, high, low, and closing prices (OHLC) to GASF matrices with the window size set to 10. After this step, the shape of the data matrices will be (10, 10, 4). In the second step, we train this 3-d matrices data with the CNN model. The architecture of our second step's CNN model is similar to LeNet, including two convolutional layers with 16 kernels and one fully-connected layer with 128 dense. Figure 20 illustrates the entire experimental architecture, and Table 1 shows the parameters used in our GAF-CNN model.

3.4 Features Selection

According to the previous section, the candlestick **patterns cannot judge from a single value** such as closing or opening price. Therefore, we need to combine opening, high, low, and closing prices (OHLC) and make the data features more reasonable. In order to close to humans have seen, we consider using the upper shadow, lower shadow, and real-body, which are more intuitive features for humans. Figures 9 and 10 are based on different features respectively of the Morning Star and Bearish Engulfing patterns through

1. the opening, high, low, and closing prices (OHLC); and
2. the closing price, upper shadow, lower shadow, and real-body (CULR).

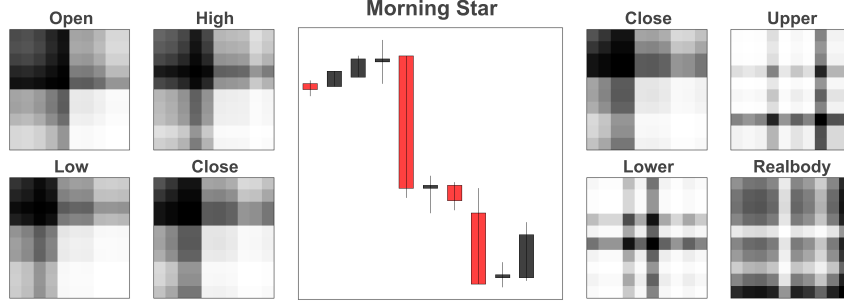


Figure 9: Examples are the Morning Star patterns. The left-hand side shows the GASF features using the opening, high, low, and closing prices (OHLC) and the right-hand side shows the GASF features using closing price, upper shadow, lower shadow, and real-body (CULR).

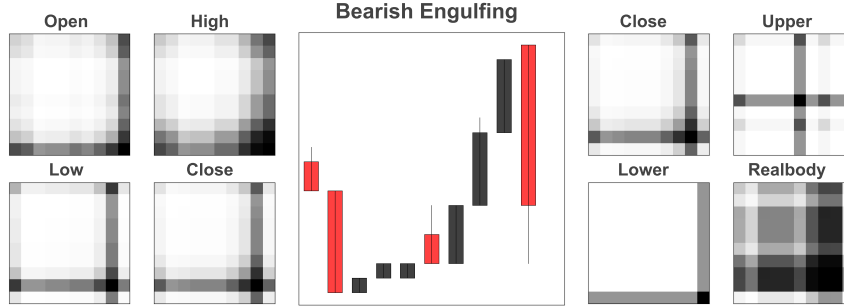


Figure 10: Examples are the Bearish Engulfing patterns. The left-hand side shows the GASF features using the opening, high, low, and closing prices (OHLC) and the right-hand side shows the GASF features using closing price, upper shadow, lower shadow, and real-body (CYLR).

Figures 9 and 10 show the visualization of the GASF matrix in two kinds transformation rules. Figure 10 shows more capable of extracting distinctive features observed than figure 9. Because the differences between the opening, high, low, and closing prices (OHLC) are generally small, resulting in high similarity among these four GASF matrices. If the model has too much repetitive information, this repeat information will reduce the convolutional model's effectiveness in learning critical features. Hence, we process the data into the features of the second transformation rule (CULR). When we use this transformation rule, the four features are not similar and pop out the significant 2-D features in the GASF matrix. From another perspective, this is a more intuitive approach that aligns with the observations of traders. Therefore, we design our experiments using

1. the opening, high, low, closing prices (OHLC); and
2. the closing prices, upper shadow, lower shadow, real-body (CYLR) features

in the simulation data. The better results are later applied to the real-world data.

3.5 Neural Architecture Searching

The GAF-CNN model works well with the simple neural architecture, two convolutional layers with 16 kernels, and one fully-connected layer with 128 denses. The max-pooling layer, which uses general picture classification, calculates the maximum value for each patch of the feature map usually. In other words, it may bring benefits about calculating cost-saving, but truncate the characteristics of the time series, which means discard information of data. Therefore, we design an experiment using a max-pooling layer or not in simulation data. Figure 11 illustrates where to use the max-pooling or not.

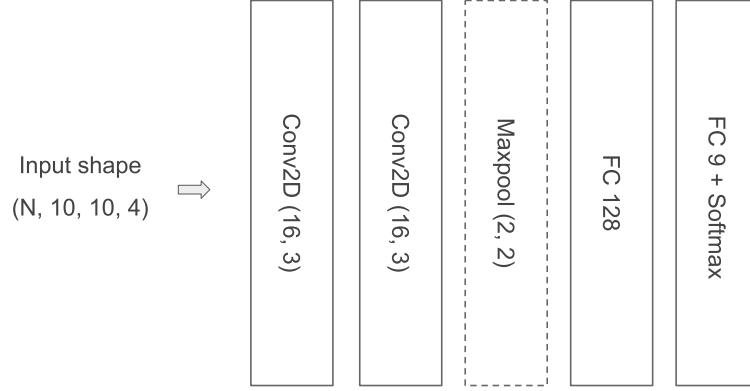


Figure 11: The max-pooling use-case.

4 Results

4.1 Baseline

Previous research on the candlestick with deep learning is about trading strategy but lack of pattern classification. It is hard to find the result from other studies to compare the GAF-CNN model, so we chose the Long Short-Term Memory model (LSTM) for reliable comparison since it is a standard method to accomplish the time series classification or regression tasks in the current year. Our goal is to achieve or surpass the performance of the LSTM model. The architecture used in this study include two hidden layer size of 128 LSTM layer and follow by a 128 dense layer [31]. More detail comparisons will discuss in sections 4-2 and 4-3.

4.2 Simulation Results

Figure 12 shows the result comparing between different features and neural architectures mention in section 3. Each experiment searches 100 times to find out the best model and predict testing data.

Architecture Features	LSTM	GAF-CNN with Max-pooling	GAF-CNN without Max-pooling
OHLC	88.58%	86.53%	88.78%
CULR	88.96%	90.40%	92.42%

Figure 12: The result is the difference between feature sets and neural architectures.

The GAF-CNN model without the max-pooling layer can achieve 92.42% accuracy, which is better than the LSTM model 88.96% accuracy in both feature sets. Figures 13 and 14 respectively show the confusion matrix of GAF-CNN model without max-pooling layer and with the different feature sets:

1. the opening, high, low, closing prices (OHLC); and
2. the closing prices, upper shadow, lower shadow, real-body (CULR).

The result of using (2) closing, upper shadow, lower shadow, and real-body (CULR) can achieve 92.42% average accuracy. If we focus on the result from class 1 to class 8, then the performance is 95.43% accuracy on average.

		prediction											
true value		0	1	2	3	4	5	6	7	8		precision	recall
	0	318	12	18	23	41	10	21	32	25		0.657	0.636
	1	7	478	0	6	4	0	5	0	0		0.945	0.956
	2	19	0	480	0	1	0	0	0	0		0.952	0.960
	3	24	4	0	464	8	0	0	0	0		0.926	0.928
	4	30	11	0	8	417	0	0	34	0		0.811	0.834
	5	9	0	1	0	0	489	0	0	1		0.978	0.978
	6	8	1	0	0	0	0	491	0	0		0.950	0.982
	7	52	0	4	0	43	0	0	388	13		0.834	0.776
	8	17	0	1	0	0	1	0	11	470		0.923	0.940

Figure 13: The confusion matrix is from GAF-CNN without using pooling layers with opening, high, low, and closing prices (OHLC) feature set. The accuracy is 88.78% on average.

		prediction											
true value		0	1	2	3	4	5	6	7	8		precision	recall
	0	342	11	19	17	29	16	24	27	15		0.794	0.684
	1	12	483	0	1	1	0	2	1	0		0.966	0.966
	2	17	0	481	0	0	1	0	1	0		0.958	0.962
	3	11	2	0	487	0	0	0	0	0		0.964	0.974
	4	13	3	1	0	442	0	0	41	0		0.870	0.884
	5	10	0	1	0	0	488	0	0	1		0.964	0.976
	6	5	0	0	0	0	0	495	0	0		0.950	0.990
	7	11	1	0	0	36	0	0	452	0		0.866	0.904
	8	10	0	0	0	0	1	0	0	489		0.968	0.978

Figure 14: The confusion matrix is from GAF-CNN without using pooling layers with closing, upper shadow, lower shadow, and real-body (CULR) feature set. The accuracy is 92.42% on average.

Figures 15 and 16 show the confusion matrix of LSTM model with two feature sets respectively. The accuracy of using (1) opening, high, low, closing prices (OHLC) is 88.58% on average, and using (2) closing, upper shadow, lower shadow, real-body (CULR) is 88.96% on average.

		prediction											
true value		0	1	2	3	4	5	6	7	8		precision	recall
	0	326	16	22	24	31	12	20	23	26		0.603	0.652
	1	15	471	0	6	6	0	2	0	0		0.955	0.942
	2	16	0	482	0	1	1	0	0	0		0.949	0.964
	3	25	1	0	470	4	0	0	0	0		0.923	0.940
	4	61	3	1	9	388	0	0	38	0		0.845	0.776
	5	7	0	0	0	0	493	0	0	0		0.974	0.986
	6	10	2	0	0	0	0	488	0	0		0.957	0.976
	7	62	0	3	0	29	0	0	393	13		0.854	0.786
	8	19	0	0	0	0	0	0	6	475		0.924	0.950

Figure 15: The confusion matrix is from the LSTM model with opening, high, low, and closing prices (OHLC) feature set. The accuracy is 88.58% on average.

		prediction											
true value		0	1	2	3	4	5	6	7	8		precision	recall
	0	330	15	17	24	32	13	19	30	20		0.626	0.660
	1	29	456	0	4	9	0	2	0	0		0.950	0.912
	2	22	0	472	0	0	3	0	3	0		0.963	0.944
	3	20	3	0	470	4	0	3	0	0		0.925	0.940
	4	41	4	1	10	406	0	0	38	0		0.846	0.812
	5	14	0	0	0	0	486	0	0	0		0.966	0.972
	6	4	2	0	0	0	0	494	0	0		0.954	0.988
	7	42	0	0	0	29	0	0	419	10		0.848	0.838
	8	25	0	0	0	0	1	0	4	470		0.940	0.940

Figure 16: The confusion matrix is from the LSTM model with closing, upper shadow, lower shadow, and real-body (CULR) feature set. The accuracy is 88.96% on average.

To explore more about the model training process, a comparison of the first 50 epochs under different conditions would help to realize the rate of convergence. Figure 17 and Figure 18 depict the difference of both feature sets and using max-pooling or not respectively.

4.3 Empirical Results

EUR/USD 1-minute price data from January 1, 2010, to January 1, 2018, are used in our real data framework, including 1000 training data, 200 validation data, and 350 testing data. Therefore, we used two times as much data in training set for class 0, which is the noisy data for the other classes. The purpose of this is to help the model clearly distinguish the patterns and increase the robustness.

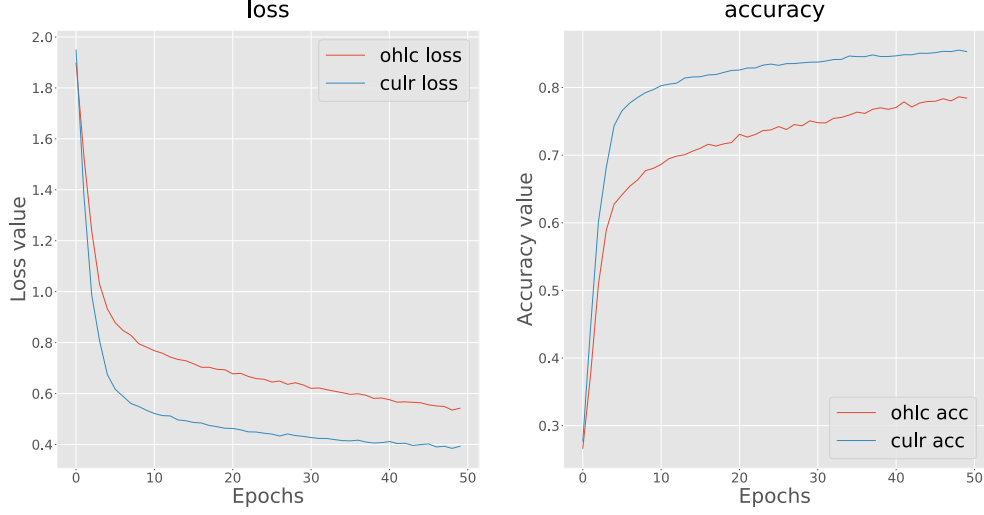


Figure 17: The loss and accuracy are from the first 50 epochs within two feature sets.

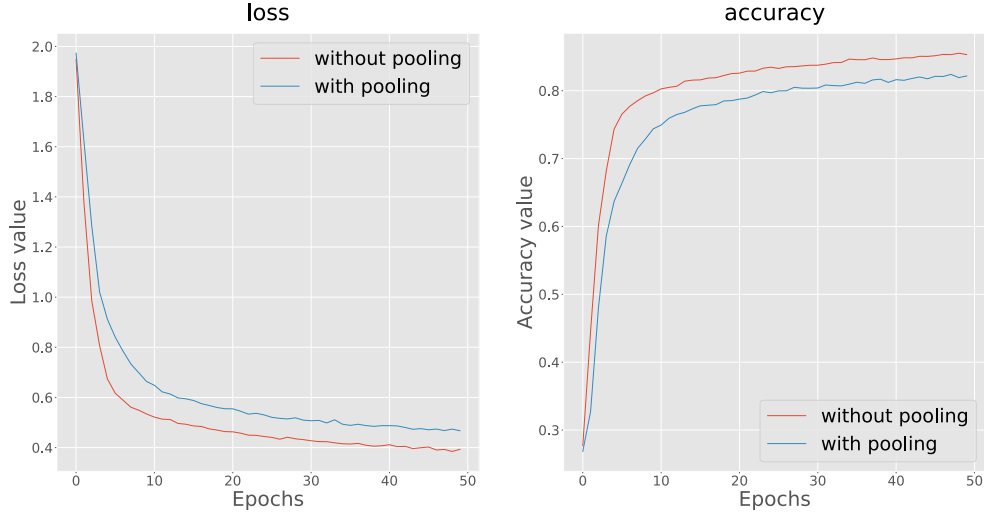


Figure 18: The loss and accuracy are from the first 50 epochs between using the max-pooling layer and without using the max-pooling layer.

Based on the results of the simulation data, we chose to use closing, upper shadow, lower shadow, and real-body (CULR) as our feature set, and to exclude the pooling layers in our model. Figure 19 shows the confusion matrix of the real-world framework. The GAF-CNN model achieves 90.7% accuracy on average in real-world data.

5 Discussion

5.1 Simulation Results

First of all, Figure 12 shows that using (2) closing prices, upper shadow, lower shadow, a real-body (CULR) feature set can significantly improve the accuracy in the GAF-CNN model than using (1) the opening, high, low, closing prices (OHLC) feature set. In Figure 17, the training process also converges significantly faster in the first 50 epochs, and end up with higher accuracy. This result is intuitive that this feature set is more close to trader way, observing the characteristics of the candlestick.

		prediction											
true value		0	1	2	3	4	5	6	7	8		precision	recall
	0	267	8	15	9	9	7	13	13	9		0.714	0.763
	1	8	341	0	1	0	0	0	0	0		0.958	0.974
	2	6	0	343	0	1	0	0	0	0		0.937	0.980
	3	11	7	0	315	0	0	0	17	0		0.924	0.900
	4	14	0	8	0	307	0	0	0	21		0.911	0.877
	5	20	0	0	0	0	328	0	2	0		0.973	0.937
	6	35	0	0	0	0	0	313	0	2		0.943	0.894
	7	7	0	0	16	0	2	0	325	0		0.910	0.929
	8	6	0	0	0	20	0	6	0	318		0.909	0.909

Figure 19: The confusion matrix is from the real data framework. The accuracy is 90.7% on average.

Secondly, the model also converges faster when using (2) without max-pooling layer than (1) with the max-pooling layer in Figure 18. In Figure 12, the GAF-CNN model without the max-pooling layer can achieve higher accuracy and lower loss value in both feature sets. The result can explain that the dependency on time series data contains many essential features. The complete time-series information will be truncated after the processing of the max-pooling layer, making it harder for the convolutional model to capture more detail features.

Lastly, the GAF-CNN model works well in both simulation and real-world framework. It achieves 90.7% accuracy on average in real-world data. Besides, our results show that class 0, which is the other class, has reduced precision and recall. The class does not affect the usability of the framework because, although class 0 does not perform well, as long as the accuracy of the other classes is high enough, the cost of misclassification is small.

5.2 Empirical Results

The result in Figure 19 shows that GAF-CNN can achieve 90.7% on average in the real-world data, outperforming the result of LSTM model. Therefore, our experimental results show that the GAF and the CNN framework are well-suited for candlestick pattern recognition for both simulation and real-world trading data.

6 Conclusions

Candlestick pattern recognition is an indicator that traders often judge with news, fundamentals, and technical indicators. However, even today, most traders decide by using their vision and experience. Although many people have directly drawn up rules to find patterns, the process is too cumbersome and hard to judge without the provision of soft scores. To better align with how traders identify patterns, we chose to use the two-dimensional CNN model. We used the GAF time series encoding with the traditional CNN model. Because of the direct use of images to train leads to underfit. We use GAF-CNN to process the GBM simulation and EUR/USD real word experiments.

In the simulation framework, we use eight candlestick patterns to test how the max-pooling layer and feature sets impact our model. The results indicate the following:

1. The max-pooling layer is terrible for the GAF-CNN model. We think that the time series are truncated and lead to the loss of practical information.
2. Using the feature set of closing price, upper shadow, lower shadow, and real-body (CULR) is better than using the simple feature set of opening, high, low, and closing prices (OHLC).

The model achieved an average accuracy of 92.42% in simulation data. Although the 0 class is prone to misclassification, the model is still available for practical work as long as the main pattern resolutions and recall are high enough.

In the real-world framework, we use the same model for the EUR/USD per minute data from January 1, 2010, to January 1, 2018 retraining, including 1000 training data, 200 validation data, and 350 testing data. The model obtained 90.7% average accuracy, outperforming the LSTM model. In real-world data, class 0 has more false positives than other types, but the main kind of recall is a certain extent. It can be considered a more conservative model. Finally, because the difference between these eight indicators is tiny, GAF-CNN has to extract subtle features. Now we only use the eight main candlestick patterns. Furthermore, future work could apply GAF-CNN to more candlestick patterns or technical indicators, such as W-head M-bottom. Thus, the entire architecture in finance candlestick, and the extensibility of the models is enormous.

7 Workflows

In this study, we find that the Convolutional Neural Network model can detect financial time series data effectively, and our **research workflow** is as follows:

1. Our experiments adopt simulation, and real-world framework, where the simulation data generates from Geometric Brownian Motion model and the real data is EUR/USD per minute data from January 1, 2010, to January 1, 2018.
2. Eight candlestick labels reference from The Major Candlestick Signals.
3. Use opening, high, low, and closing prices (OHLC) or closing, upper shadow, lower shadow, and real-body (CULR) feature sets. The data in this stage is still a 10 by 4 matrix, where 4 represents the features.
4. Encode time series data by Gramian Angular Summation Field. The data will become 10 by 10 by 4 in this stage.
5. Each framework of training, validation, and testing is with the Convolutional Neural Network model.

The first step is each experiment test in the simulation framework, then apply the result of feature sets and neural architectures to the real-world framework. In all experiments, the convolution model use only two convolutional layers with 16 kernels and one fully-connected layer with 128 denses. All these processes illustrate in Figure 20.

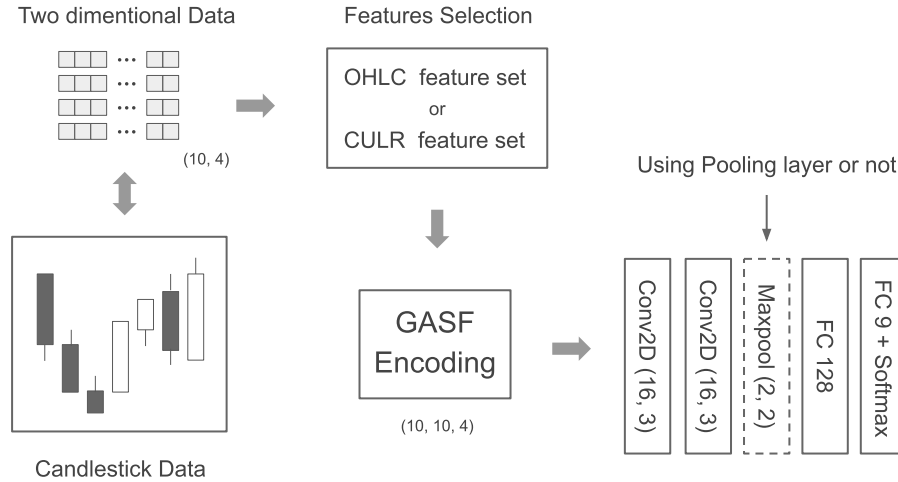


Figure 20: The workflow of the entire experiment.

8 Abbreviations

ATR	Average True Range
CDRs	Correction Detection Rates
CNN	Convolutional Neural Network
EUR	European Dollar

FC	Full Connection Layer
GAF	Gramian Angular Field
GASF	Gramian Angular Summation Field
GBM	Geometric Brownian Motion
KD	Stochastic Oscillator
LSTM	Long Short-Term Memory
MA	Moving Average
MACD	Moving Average Convergence and Divergence
MNIST	Modified National Institute of Standards and Technology
NORB	York University Object Recognition Benchmark
OHLC	Opening, High, Low, and Closing prices
CULR	Closing prices, Upper shadow, Lower shadow, Real-body
Pool	Pooling Layer
RSI	Relative Strength
USD	United States Dollar

9 Declarations

9.1 Availability of data and material

We provide an open source (<https://github.com/RainBoltz/Series2GAF>) Series2GAF which can be used to transform time series into Gramian Angular Field.

9.2 Funding

Jun-Hao Chen and Yun-Cheng Tsai are supported in part by the Ministry of Science and Technology of Taiwan under grant 108-2218-E-002-050-.

9.3 Authors contributions

Yun-Cheng Tsai conceived of the presented idea. Jun-Hao Chen developed the theory and performed the computations. Yun-Cheng Tsai and Jun-Hao Chen verified the analytical methods. All authors discussed the results and contributed to the final manuscript.

9.4 Acknowledgements

Thanks to Prof. Jane Yung-Jen Hsu for constructive discussion and great support.

10 Competing interests

Jun-Hao Chen and Yun-Cheng Tsai declare that we have no significant competing financial, professional or personal interests that might have influenced the performance or presentation of the work described in this manuscript.

References

- [1] Gang Kou, Yi Peng, and Guoxun Wang. Evaluation of clustering algorithms for financial risk analysis using mcdm methods. *Information Sciences*, 275:1–12, 2014.
- [2] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. *Twenty-fourth international joint conference on artificial intelligence*, 2015.
- [3] Emad W Saad, Danil V Prokhorov, and Donald C Wunsch. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on neural networks*, 9(6):1456–1470, 1998.
- [4] A-PN Refenes and Will T Holt. Forecasting volatility with neural regression: A contribution to model adequacy. *IEEE Transactions on Neural Networks*, 12(4):850–864, 2001.
- [5] Konstantinos N Pantazopoulos, Lefteri H Tsoukalas, Nikolaos G Bourbakis, Michael J Brun, and Elias N Houstis. Financial prediction and trading strategies using neurofuzzy approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(4):520–531, 1998.
- [6] Vasant Dhar and Dashin Chou. A comparison of nonlinear methods for predicting earnings surprises and returns. *IEEE Transactions on Neural networks*, 12(4):907–921, 2001.
- [7] Li-Juan Cao and Francis Eng Hock Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on neural networks*, 14(6):1506–1518, 2003.
- [8] Qiang Song and Brad S Chissom. Fuzzy time series and its models. *Fuzzy sets and systems*, 54(3):269–277, 1993.
- [9] Steven C Hall. Predicting financial distress. *Journal of Financial Service Professionals*, 56(3):12, 2002.
- [10] Ben R Marshall, Martin R Young, and Lawrence C Rose. Candlestick technical trading strategies: Can they create value for investors? *Journal of Banking & Finance*, 30(8):2303–2323, 2006.
- [11] Thomas N Bulkowski. *Encyclopedia of candlestick charts*, volume 332. John Wiley & Sons, 2012.
- [12] Y-lan Boureau, Yann L Cun, et al. Sparse feature learning for deep belief networks. *Advances in neural information processing systems*, pages 1185–1192, 2008.
- [13] Tie Li, Gang Kou, and Yi Peng. Improving malicious urls detection via feature engineering: Linear and nonlinear space transformation methods. *Information Systems*, page 101494, 2020.

- [14] Tie Li, Gang Kou, Yi Peng, and Yong Shi. Classifying with adaptive hyper-spheres: An incremental classifier based on competitive learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2017.
- [15] Rabia Aziz, CK Verma, and Namita Srivastava. Artificial neural network classification of high dimensional data with novel optimization approach of dimension reduction. *Annals of Data Science*, 5(4):615–635, 2018.
- [16] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.
- [17] Zhiguang Wang and Tim Oates. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. *Proceedings of the 2015 Association for the Advancement of Artificial Intelligence (AAAI) Workshops*, pages 40–46, 2015.
- [18] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.
- [19] Zhiguo He. Optimal executive compensation when firm size follows geometric brownian motion. *The Review of Financial Studies*, 22(2):859–892, 2008.
- [20] Stephen W. Bigalow. *The Major Candlesticks Signals*. The Candlestick Forum LLC, 2014.
- [21] Gary S Wagner and Bradley L Matheny. *Trading applications of Japanese candlestick charting*, volume 38. John Wiley & Sons, 1994.
- [22] Steve Nison. *Japanese candlestick charting techniques: a contemporary guide to the ancient investment techniques of the Far East*. Penguin, 2001.
- [23] Felipe Tudela. *The Secret Code of Japanese Candlesticks*, volume 402. John Wiley & Sons, 2008.
- [24] Y Goo, D Chen, Y Chang, et al. The application of japanese candlestick trading strategies in taiwan. *Investment Management and Financial Innovations*, 4(4):49–79, 2007.
- [25] Mark P Taylor and Helen Allen. The use of technical analysis in the foreign exchange market. *Journal of international Money and Finance*, 11(3):304–314, 1992.
- [26] Kuniyiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. *Competition and cooperation in neural nets*, pages 267–285, 1982.
- [27] Yann LeCun et al. Lenet-5, convolutional neural networks. URL: <http://yann.lecun.com/exdb/lenet>, page 20, 2015.
- [28] Haohan Wang, Bhiksha Raj, and Eric P Xing. On the origin of deep learning. *arXiv preprint arXiv:1702.07800*, 2017.
- [29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [30] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *2012 IEEE conference on computer vision and pattern recognition*, pages 3642–3649, 2012.
- [31] Denis Smirnov and Engelbert Mephu Nguifo. Time series classification with recurrent neural networks. *Advanced Analytics and Learning on Temporal Data*, page 8, 2018.