

# Trading via Image Classification

Naftali Cohen\*

J. P. Morgan AI Research  
New York, NY  
naftali.cohen@jpmchase.com

Tucker Balch

J. P. Morgan AI Research  
New York, NY  
tucker.balch@jpmchase.com

Manuela Veloso

J. P. Morgan AI Research  
New York, NY  
manuela.veloso@jpmchase.com

## ABSTRACT

The art of systematic financial trading evolved with an array of approaches, ranging from simple strategies to complex algorithms, all relying primarily on aspects of time-series analysis (e.g., Murphy, 1999; De Prado, 2018; Tsay, 2005). After visiting the trading floor of a leading financial institution, we noticed that traders always execute their trade orders while *observing* images of financial time-series on their screens. In this work, we build upon image recognition's success (e.g., Krizhevsky et al., 2012; Szegedy et al., 2015; Zeiler and Fergus, 2014; Wang et al., 2017; Koch et al., 2015; LeCun et al., 2015) and examine the value of transforming the traditional time-series analysis to that of image classification. We create a large sample of financial time-series images encoded as candlestick (Box and Whisker) charts and label the samples following three algebraically-defined binary trade strategies (Murphy, 1999). Using the images, we train over a dozen machine-learning classification models and find that the algorithms efficiently recover the complicated, multiscale label-generating rules when the data is visually represented. We suggest that the transformation of continuous numeric time-series classification problem to a vision problem is useful for recovering signals typical of technical analysis.

## KEYWORDS

finance, images, supervised classification

### ACM Reference Format:

Naftali Cohen, Tucker Balch, and Manuela Veloso. 2020. Trading via Image Classification. In *ACM International Conference on AI in Finance (ICAIF '20)*, October 15–16, 2020, New York, NY, USA. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3383455.3422544>

## 1 INTRODUCTION

Traders in the financial markets execute buy and sell orders of financial instruments as stocks, mutual funds, bonds, and options daily. They execute orders while reading news reports and earning calls. Concurrently, they observe charts of time-series data that indicates the historical value of securities and leading financial indices (see Fig. 1 for a typical workstation of a professional trader<sup>1</sup>). Many

<sup>1</sup>The photo was taken in a trading room at Rouen, Normandie, France, September 2015.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

ICAIF '20, October 15–16, 2020, New York, NY, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7584-9/20/10...\$15.00

<https://doi.org/10.1145/3383455.3422544>

algorithms have been developed to analyze continuous financial time-series data to improve a trader's decision-making ability to buy or sell a particular security (Murphy, 1999). Conventional algorithms process time-series data as a list of numerical data, aiming at detecting patterns as trends, cycles, correlations, etc. (e.g., De Prado, 2018; Tsay, 2005). If a pattern is identified, the analyst can then construct an algorithm that will use the detected pattern (e.g., Wilks, 2011) to predict the expected future values of the sequence at hand (i.e., forecasting using exponential smoothing models, etc.).

Experienced traders, who observe financial time-series charts and execute buy and sell orders, start developing an intuition for market opportunities. The intuition they develop based on their chart observations nearly reflects the recommendations that their state-of-the-art model provides (personal communication with J.P. Morgan's financial experts Jason Hunter, Joshua Younger, Alix Floman, and Veronica Bustamante). In this perspective, financial time-series analysis can be thought of as a visual process. That is, when experienced traders look at a time-series data, they process and act upon the image instead of mentally exercising algebraic operations on the sequence of numbers.



Figure 1: Typical workstation of a professional trader. Credit: Photoagriculture / Shutterstock.com.

This paper is written under the assumption that, given public knowledge, markets are efficient (e.g., Pedersen, 2019). That is, future market movements have almost no predictability. However, the way professionals trade is systematic (e.g., consistent, back-tested, and potentially profitable for a short duration) and can be characterized using a set of rules. We ask, can we build a system that identifies and replicates the way humans trade? For this, we create extensive financial time-series image data. We make use of three known label-generating rules following algebraically-defined binary trade strategies (Murphy, 1999) to replicate the way people trade. Using a supervised classification approach (e.g., Bishop, 2006; Goodfellow et al., 2016; Aggarwal, 2015), we evaluate predictions

using over 15 different classifiers and show that the models are very efficient in identifying the complicated, sometimes multiscale, labels.

## 2 RELATED WORK AND MAIN CONTRIBUTIONS

The focus of this work is on the representation of financial time-series data as images. Previous work on time-series classification suggests first transforming the data either locally using wavelets or globally using Fourier transforms and then compare the modes of variability in the transformed spaces (e.g., Wilks, 2011). Other methods apply similarity metrics such as Euclidean distance, k-nearest neighbors, dynamic time warping, or even Pearson correlations to separate the classes (e.g., Aggarwal, 2015). In addition to the above, other techniques focus on manual feature engineering to detect a frequently occurring pattern or shape in the time series (e.g., Bagnall et al., 2017).

More recently, it was suggested to approach time-series classification by first encoding the data as images and then utilize the power of computer vision algorithms for classification (Park et al., 2019). In an example, it was suggested to encode the time dependency, implicitly, as Gramian-Angular fields, Markov-Transition fields (Wang and Oates, 2015a; Wang and Oates, 2015b), or make use of recurrence plots (Souza et al., 2014; Silva et al., 2013; Hatami et al., 2018) as a graphical representation. Another work focused on transforming financial data into images to classify candlesticks patterns (Tsai et al., 2019).

In this paper, we examine the value of images alone for identifying trade opportunities typical for technical analysis. To the best of our knowledge, our work is the first that built upon the great success in image recognition and tries to systematically apply it to numeric time-series classification by taking a *direct* graphical approach and recency-biased label-generating rules. The contributions of this paper are as follows:

- (1) The first contribution is bridging the unrelated areas of quantitative finance and computer vision. The former involves a mixture of technical, quantitative analysis, and financial knowledge, while the second involves advanced algorithm design and computer science techniques. In this paper, we show how the two distinct areas can leverage expertise and methods from each other.
- (2) The second contribution is our understanding that, in practice, there are financial domains in which investment decisions are made using visual representations alone (e.g., swap trade) – relying, in fact, on traders’ intuition, experience, skill, and luck. Moreover, currently, numerous online platforms and smartphone applications (e.g., Robinhood) allow people to trade directly from their smartphones. In these platforms, the data is presented graphically, and in most cases, the user decides and executes his trade upon the visual representation alone. Therefore, it’s reasonable to examine the usefulness of visual representations as input to the model.
- (3) The third contribution is that we show that the concept of visual time-series classification is effective and works on real data. A large fraction of the artificial-intelligence research is conceptual and works only on synthetic data. As will be

shown, the concepts introduced in this paper are not only effective on real data, but they can be leveraged to deploy immediately as either a marketing recommendation tool and/or as a forecasting tool.

## 3 DATA AND METHODS

In this study, we use Yahoo finance to analyze the daily values of all companies that contribute to the S&P 500 index for the period 2010-2018 (hereafter SP500 data). These are large-cap companies that are actively traded on the American stock exchanges, and their capitalization covers the vast majority of the American equity market (e.g., Berk et al., 2013).



Figure 2: Converting continuous time series to images.

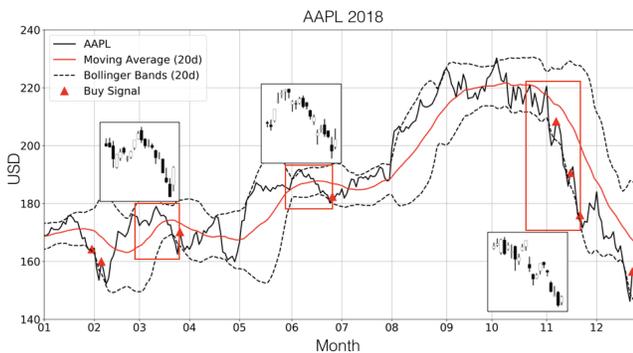
Trading is done continuously (during trade hours which usually span between 9:30 am to 4:00 pm, not including before- and after-market hours). However, we use a discrete form of the continuous data by accounting only for the start, max, min, and end values per stock per day. These values are denoted, as is common in finance, as the Open, High, Low, and Close (OHLC) values (e.g., Murphy, 1999). We visualize the data using a box-and-whisker (also called candlestick) diagram, where box edges mark the Open and Close price, while the whiskers mark the Low and High values (i.e., daily min and max). The color of each box reveals whether the Open price finalized higher or lower than the Close price for the same day; if  $\text{Open} > \text{Close}$  the box is filled in black indicating Bear’s market, whereas if  $\text{Open} < \text{Close}$  the box is filled in white indicating Bull’s market (e.g., Murphy, 1999). Figure 2 shows an example of this process by focusing attention on the AAPL ticker for Feb 19, 2019, and Feb 28, 2019. The left columns show the 1-minute continuous trading data during trading hours, while the right column detail the discretization process. Notice that the upper left time-series experiences a positive trend resulting in a white candlestick visualization, while the bottom left time-series data experiences a negative trend resulting in a black candlestick.

We compare three well-known binary indicators (Murphy, 1999), where each indicator is based on prescribed algebraic rules that depend solely on the Close values. Each indicator alerts the trader

only for a buying opportunity. If a trader decides to follow (one of) the signals they may do so at any point no earlier than the day after the opportunity signal was created. The three "buy" signals are defined as follows:

- **BB crossing:** The Bollinger Bands (BB) of a given time-series consists of two symmetric bands of 20-days moving two standard deviations (Colby and Meyers, 1988). The bands envelop the inherent stock volatility while filtering the noise in the price action. Traders use the price bands as bounds for trade activity around the price trend (Murphy, 1999). Hence, when prices approach the lower band or go below, prices are considered to be in an oversold position and trigger a buying opportunity. Here, the bands are computed using the (adjusted) Close values, and hence a buy signal is *defined* to trigger when the daily Close value *crosses above* the lower band.

Figure 3 shows an example of a Buy signal opportunities for the AAPL stock during 2018. In solid black, one can see the daily Close values for the ticker while the red line shows the 20-days moving average (inclusive) of the price line. The dashed black lines mark the two standard deviations above and below the moving average line. The BB crossing algorithm states that a Buy signal is initiated when the price line (in solid black) crosses above the lower dash black line. In this Figure, marked by the red triangles, one can identify eight such buy opportunities.



**Figure 3: Labeling time series data according to the Bollinger Bands crossing rule.**

- **MACD crossing:** Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator that compares the relationship between short and long exponential moving averages (EMA) of an asset (Colby and Meyers, 1988). As is common in finance (e.g., Murphy, 1999), we compute the MACD by subtracting the 26-days EMA from the 12-days EMA. When MACD falls to negative values, it suggests negative momentum and conversely, when the MACD rises to positive values, it indicates upward momentum. Traders usually wait for consistent measures, thus smoothing the MACD line further by computing the 9-day EMA of the MACD, known as the signal line. Here, the MACD buy signal is *defined* to trigger when the signal line *crosses above*.

- **RSI crossing:** The Relative Strength Index (RSI) is an oscillating indicator that summarizes the magnitude of recent price changes to evaluate the overbought or oversold conditions of an asset. As is common in finance (e.g., Colby and Meyers, 1988; Murphy, 1999), we compute RSI as the ratio 14-days EMA of the incremental increase to the incremental decrease in asset values. The ratio is then scaled to values that vary between 0 and 100: it rises as the number and size of daily gains increases and falls as the number and size of daily losses increases. Traders use RSI as an indication of either an overbought or an oversold state. An overbought state might trigger a sell order; an oversold state might trigger a buy order. The standard thresholds for oversold/overbought RSI are 30/70, respectively (Murphy, 1999). Here, the RSI buy signal is *defined* to trigger when the RSI line *crosses above* the value of RSI=30.

Figure 3 shows three positively-labeled images that correspond to the BB-crossing algorithm. These images are generated by enveloping 20 days of stock activity (the red rectangles) before and including the buy-signal day activity. It is also possible to create negatively-labeled images from this time-series by enveloping activity, in the same way, for days with no buy signal. Note also that these images tightly bind the trade activity and do not contain labels, tickers, or title, which is the essential input data standardization pre-process we apply in this study.

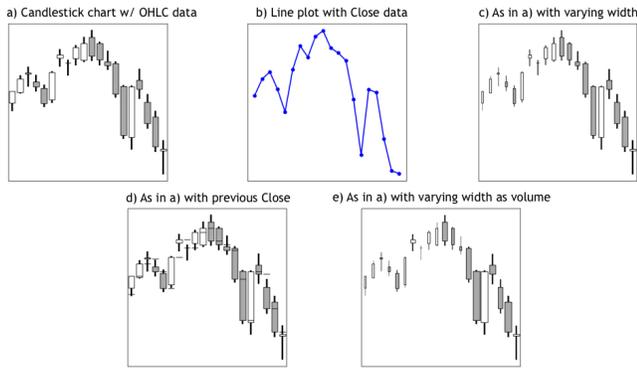
## 4 RESULTS

The objective of this study is to examine whether or not we can train a model to recover trade signals from algebraically-defined time-series data that is typical of technical analysis. We examine the supervised classification predictions of the time-series images that are labeled according to the BB, RSI, and MACD algorithms.

The data set is balanced, containing 5,000 samples per class per indicator. That is, for each of the S&P500 tickers, we compute all buy triggers for the period between 2010 and the end of 2017. We then choose, at random, 10 buy triggers for each ticker and create corresponding images. In the same way, we choose, at random, 10 no-buy triggers per ticker and create similar images. This process results in 10,000 high-resolution images per trigger.

A key difference between the three algorithms, besides their varying complexity, is the time-span each considers. While the BB algorithm takes into account 20 days of price action, RSI, which uses exponential-moving averaging considers (effectively) 27 days. MACD, which also uses exponential-moving averages, spans (effectively) over 26 days. For each of the triggers, we crop the images according to the number of effective trading days they consider. Thus, the BB images include information of 20 trade days, while RSI contains data for 27 days, and MACD, the most sophisticated algorithm that compares three time scales, contains 26 days of data. In other words, each sample has 80-108 features depending on the size of the window required to compute the label (i.e., 4x20 for the BB crossing, and 4x26, 4x27 for the MACD and RSI respectively).

Figure 4 depicts an example of the five different visual designs we use in this study. Panel 4a uses the OHLC data as in Fig. 2, while panel 4b uses only the Close values plotted as a line chart. The



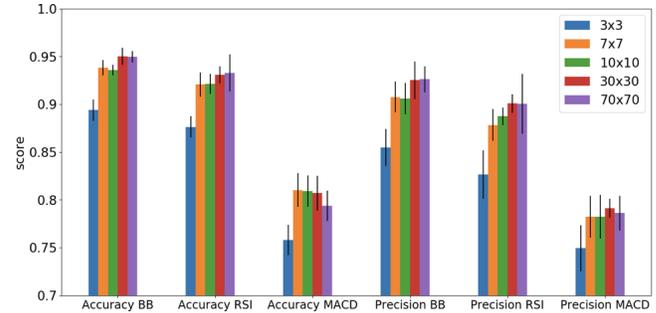
**Figure 4: Various visual representations of the same time-series data.**

design of 4b serves as a reference performance level, as will be discussed later.

In this study, a key element is how to express the direction and notion of time, or recency, in static images. A simple way of incorporating recency in the images is via the *labels*. Each image is labeled according to trade opportunities, which are defined by crossing above a threshold at the right part of the image. The labels are time-dependent and are tied to a specific region in the chart; thus, implicitly, they deliver the notion of time to the static images. Another way of incorporating recency in the images is to incorporate the notion of time directly in them. The designs at panels 4c and 4d aim at explicitly representing the direction of time by either linearly varying the width of the boxes towards the right (4c), or by overlaying the previous Close value as a horizontal line on each of the candlesticks (4d). Lastly, in panel 4e, we augment the OHLC data by incorporating the trade volume in the candlestick visualization by varying the width of each box according to the relative change of the trade volume within the considered time frame. Remember that all three label-generating rules consider only the Close value, but each Close value is influenced by its preceding daily activity, reflected in the candlestick’s visualization. We expect a trained model to either filter out unnecessary information or discover new feature relationships in the encoded image and identify the label-generating rule.

Following the above process, we create high-resolution images based on the discrete form of the data. Another question we have to address is what resolution do we need to maintain for proper analysis. The problem is that the higher the resolution, the more we amplify the (pixelated) feature space introducing more noise to the models and possibly creating unwanted spurious correlations. We examine this point by varying the resolution of the input images in logarithmic scale and comparing the accuracy score of a hard voting classifier over the following 16 trained classifiers: Logistic Regression, Gaussian Naive-Bayes, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Gaussian Process, K-Nearest Neighbors, Linear SVM, RBF SVM, Deep Neural Net, Decision Trees, Random Forest, Extra Randomized Forest, Ada Boost,

Bagging, Gradient Boosting, and Convolutional Neural Net<sup>2</sup>. The focus here is on comparing the models’ aggregated performance while changing the representation of the input space.



**Figure 5: The effect of varying the image resolution on the classification accuracy and precision scores for the three label-generating rules.**

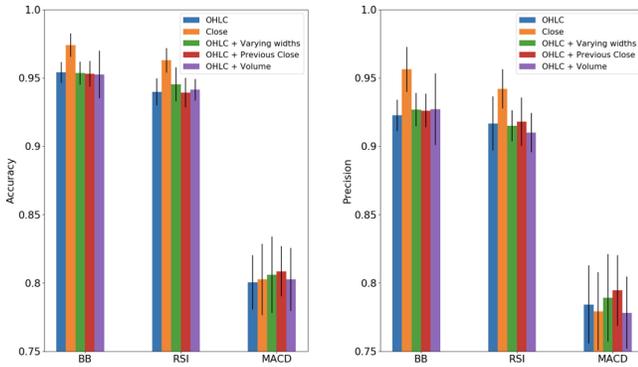
Figure 5 shows the results of the classification scores when we downscale the resolutions of the images that are labeled following the BB, RSI, and MACD algorithms. We use the Lanczos filter for downscaling, which uses sinc filters and efficiently reduces aliasing while preserving sharpness. To evaluate the models’ performances, we use the 5-fold cross-validation technique. This allows us to infer not only the mean prediction of the voting classifier but also the variability about the mean. (The vertical black lines in Fig. 5 show one symmetric standard deviation about the mean accuracy). Figure 5 shows that regardless of the labeling rule, the average accuracy and precision scores increase with finer resolutions but matures around 30x30 pixel resolution. For this reason, the following analysis is done using a 30x30 pixel resolution.

Figure 6 compares the predictability skill in the various image representations of the same input data for the three label-generating rules. All input representations perform remarkably well, and the predictability skill stands at about 95% for the BB and RSI label-generating rules, while at approximately 80% for the MACD labeled data. We were not surprised to see that the classifiers perform less efficiently on the MACD labeled data as this labeling-rule is the most complex involving multiple time-scales and smooth operations, all acting in concert.

The best performing input data is the one that uses the Close values exclusively as line plots, while the various OHLC representations fall only a little behind. However, the line plot serves only as a point of reference – the Bayesian performance level. This is because the label-generating rule depends exclusively on the Close values<sup>3</sup>. The key point is the fact that the various visual OHLC representations manage to achieve performance *comparable* to the Bayesian level. Most importantly, this finding is robust for the BB and RSI, as well as for the MACD algorithm.

<sup>2</sup>The Deep Neural Net uses 32x32x32 structure, while the Convolutional Neural Net (CNN) uses three layers of 32 3x3 filters with ReLU activations and Max Pooling of 2x2 in between the layers. The last layer incorporates Sigmoid activation. The CCN model is compiled with Adam optimizer, binary-cross entropy loss function and run with a batch size of 16 samples for 50 iterations

<sup>3</sup>Using the Close value alone is comparable to using the actual numerical data that the labeling rules are based upon.



**Figure 6: The supervised classification accuracy (left panel) and precision (right panel) scores for the various triggers as a function of the different input representations.**

Close examination of Fig. 6 shows that augmenting the OHLC input to include explicit time representation in the images by varying the bar widths linearly or by incorporating the previous Close values didn't add value. An exception to this is the MACD algorithm (a point we may explore further in a future study). On the other hand, encoding the irrelevant volume information in the candlestick images increased our uncertainty in predictions for all label-generating rules.

The precision score results are represented in the right panel of Fig. 6 and are almost identical to the accuracy scores on the left panel.

## 5 DISCUSSION

In this paper, we examine the supervised time-series classification task using large financial data sets and compare the results when the data is represented visually in various ways. We find that even at low resolutions (see Fig. 5), time-series classification can be achieved effectively by transforming the task into a computer vision task. This finding is in accordance with Cohen et al., 2019 who showed that classification of financial data using exclusive visual designs relates information spatially, aids in identifying new patterns and, in some cases, achieves *better* performance compared to using the raw tabular form of the same data.

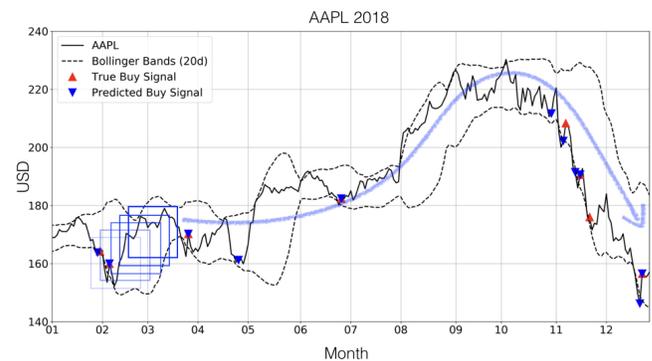
Visualizing data and time-series data in particular, is an essential pre-processing step. Visualization by itself is not straightforward, especially for high-dimensional data, and it might take some time for the analyst to find the proper graphical design that will encapsulate the full complexity of the data. In this study, we essentially suggest considering the display as the input over the raw information. Our research indicates that even very complex multi-scale algebraic operations can be discovered by transferring the task to a computer vision problem.

A key question in this study is whether time-dependent signals can be detected in static images. To be more explicit, if the time axis goes left to right, it means that data points to the right are more recent and therefore, more critical to the model than data points to the left. But how can we convey this kind of information without an independent ordinal variable (e.g., time axis)? We present two ways

to incorporate the time-dependency in the images: the first leverages labels to deliver the notion of time; the second augments the images with sequential features. Incorporating time-dependency via labeling is done throughout the paper. We label the candlestick images using three algorithms and each computes a time-dependent function. Thus, each image encapsulates implicitly, via its corresponding label, the notion of time. That is, the signal we seek to detect is located on the right-most side of the image; the cross-above trigger always occurs because of the *last* few data points. In an example, the BB crossing algorithm effectively yields images with suspected local minimum on the right-hand side of the image. Incorporating time dependency explicitly by image augmentation is considered in two ways, by varying the width of the boxed in the candlestick diagram linearly and by overlaying the previous Close value on each candlestick. It is noteworthy, however, that compared to the implicit label approach, we find the explicit augmentation to be less effective, as can be seen in Fig. 6.

In this study, we blended all S&P 500 stocks and did not cluster the data by season, category, or sector. We used specific window sizes that correspond to the total length of information required by each algorithm to compute its label. To isolate the effect of the various window sizes, we examined the classification results when all window sizes were set to include 30 days of information. We found that the performance decreased when the window size added unnecessary information. (In this instance, there was a decrease in accuracy scores by a few percentage points – not shown).

We see no need to account for the overall positive market performance during the 2010-2018 period as the analysis is done on a short times scales (about a month or less). One can complement this study by similarly analyzing for sell signals. We have repeated this analysis for sell signals and found that the overall quantitative results are very similar (not shown).



**Figure 7: Time-series forecasting using a 20-days rolling window.**

We end this paper by noting that the supervised classification task can be applied as a forecasting tool (e.g., Hyndman and Athanassopoulos, 2018). In Fig. 7, we take out-of-sample, daily trading data from 2018. (As noted, the previous training and evaluation were computed using data from the period between 2010 and the end of 2017.) We create 20 days of images for *every* day in the data. Next we feed these images to the voting classifier as a test set, and for

each image, we predict what the label will be. Figure 7 corresponds to Fig. 3 but also includes blue triangles showing the predicted buy signal. Clearly, at least five buy signals were correctly classified, but even the missed ones are incredibly close in the sense that there is *almost* cross-above the lower BB. Finally, depending on the use case, one can modify the binary probability threshold to achieve higher precision scores.

## 6 CONCLUSION

Visual object recognition and object detection using machine learning and deep neural networks have shown great success in recent years (e.g., Krizhevsky et al., 2012; Zeiler and Fergus, 2014; Szegedy et al., 2015; Koch et al., 2015; LeCun et al., 2015; Wang et al., 2017). In this paper, we follow up on these studies and examine the value in transforming numerical time-series analysis to that of image classification. We focus on financial trading after noticing that human traders always execute their trade orders while *observing* images of financial time-series on their screens (see Fig. 1). Our study suggests that the transformation of time-series analysis to a computer vision task is beneficial for identifying trade decisions typical for humans using technical analysis.

*Acknowledgments.* We would like to thank Jason Hunter, Joshua Younger, Alix Floman, and Veronica Bustamante for providing with insightful comments and crucial suggestions that helped in bringing this manuscript to completion.

*Disclaimer.* This paper was prepared for information purposes by the Artificial Intelligence Research group of J. P. Morgan Chase & Co. and its affiliates (“J. P. Morgan”), and is not a product of the Research Department of J. P. Morgan. J. P. Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

©2020 J. P. Morgan Chase & Co. All rights reserved.

## REFERENCES

- [1] Charu C Aggarwal. 2015. *Data mining: the textbook*. Springer.
- [2] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* 31, 3 (2017), 606–660.
- [3] Jonathon Berk, Peter DeMarzo, Jarrod Harford, Guy Ford, Vito Mollica, and Nigel Finch. 2013. *Fundamentals of corporate finance*. Pearson Higher Education AU.
- [4] Christopher M Bishop. 2006. *Pattern recognition and machine learning*. Springer.
- [5] Naftali Cohen, Tucker Balch, and Manuela Veloso. 2019. The effect of visual design in image classification. *arXiv preprint arXiv:1907.09567* (2019).
- [6] Robert W Colby and Thomas A Meyers. 1988. *The encyclopedia of technical market indicators*. Dow Jones-Irwin Homewood, IL.
- [7] Marcos Lopez De Prado. 2018. *Advances in financial machine learning*. John Wiley & Sons.
- [8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.
- [9] Nima Hatami, Yann Gavet, and Johan Debye. 2018. Classification of time-series images using deep convolutional neural networks. In *Tenth International Conference on Machine Vision (ICMV 2017)*, Vol. 10696. International Society for Optics and Photonics, 106960Y.
- [10] Rob J Hyndman and George Athanasopoulos. 2018. *Forecasting: principles and practice*. OTexts.
- [11] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, Vol. 2.
- [12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.
- [13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.
- [14] John J Murphy. 1999. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin.
- [15] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779* (2019).
- [16] Lasse Heje Pedersen. 2019. *Efficiently inefficient: how smart money invests and market prices are determined*. Princeton University Press.
- [17] Diego F Silva, Vinicius MA De Souza, and Gustavo EAPA Batista. 2013. Time series classification using compression distance of recurrence plots. In *2013 IEEE 13th International Conference on Data Mining*. IEEE, 687–696.
- [18] Vinicius MA Souza, Diego F Silva, and Gustavo EAPA Batista. 2014. Extracting texture features for time series classification. In *2014 22nd International Conference on Pattern Recognition*. IEEE, 1425–1430.
- [19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.
- [20] Yun-Cheng Tsai, Jun-Hao Chen, and Chun-Chieh Wang. 2019. Encoding Candlesticks as Images for Patterns Classification Using Convolutional Neural Networks. *arXiv preprint arXiv:1901.05237* (2019).
- [21] Ruey S Tsay. 2005. *Analysis of financial time series*. Vol. 543. John Wiley & sons.
- [22] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. 2017. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3156–3164.
- [23] Zhiguang Wang and Tim Oates. 2015. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- [24] Zhiguang Wang and Tim Oates. 2015. Imaging time-series to improve classification and imputation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.
- [25] Daniel S Wilks. 2011. *Statistical methods in the atmospheric sciences*. Vol. 100. Academic press.
- [26] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*. Springer, 818–833.