(Re-)Imag(in)ing Price Trends - Replication and Extension

Alvin Lo, Ning Liu, Tony Tan, Ziqing Guo*
Hong Kong University of Science and Technology

Abstract

This report aims to replicate the model and findings by Jingwen, Bryan, and Dacheng (2021). We focus on the 20-days model for the replication. Our replication results closely match the result of the original paper. We also consider three different extensions: i) GradCam; ii) robustness check and iii) regression. Our regression result suggests that the extension from a binary classification model to a continuum regression model can be beneficial.

1 Introduction

Can we use CNN to predict the stock price?

The stock market prediction has been a main focus in the Finance literature. The debate of whether the stock price is predictable has been a hot topic, and many researches have been done to explore this question.[2]

The paper by Jingwen, Bryan, and Dacheng 2021 (JBD) contributes to this line of research. This paper not only supports the argument that stock prices are *predictable* but also proposed a new method of price predictions. Contrary to the previous study, which uses data-based analytic methods for predictions, this paper proposes to use the "image" of the stock price, combined with a trained Convolutional Neural Network (CNN) for stock price predictions. JBD first tests whether a trained CNN model can predict the future stock price movement. Then, they move on to test their investment performance and conduct a robustness check. Their result shows that a CNN model not only can predict the future price movement of individual securities but also outperforms the other trading strategies in terms of average return.

According to the paper, there are three main advantages of using the CNN model compared to the traditional data analytic method: First, the CNN model can generate investment signals automatically. With a well-trained model, we can use an image of the price plot as input, the model will then gives the prediction result automatically. The traditional method, on the other hand, usually requires a mathematical structured model for accurate prediction. Second, the image representation of data allows the model to 'visualize' the data trend and relationship as a human does. This can potentially provide prediction improvement. Third, the imaging process can transform historical data to a comparable scale, which alleviates the non-stationarity issue of most of the stock market data.

The goal of this report is to verify and replicate the result by JBD. We pick one model (the 20-day model) from JBD and conduct our analysis. Our replication result supports the findings by JBD. We also conduct three new experiments to understand more about the applicability of the proposed method. The three experiments are i) Gradient-weighted Class Activation Mapping (Grad-CAM); ii) Robustness and iii) Regression.

*This report is part of the requirement to the course MATH5470 in Spring 2022 from the Hong Kong University of Science and Technology. The names of author are arranged in alphabetical order.

[2] We will leave the literature review to the appendix.

The report is divided into three parts: First, we briefly introduce the model adopted in the paper, and also our replication result. Second, we will conduct several extensions of the model. Lastly, we will conclude our report by bringing some potential future research directions.

2 Baseline Model

This section briefly discusses the model adopted in JBD. The first task of this report is to replicate the result by JBD. The paper has a clear description of the model building blocks. For simplicity, we focus on the 20-day model since we can only obtain the 20-days input. We briefly describe the model in the main text and will leave the details of the model and the code in the appendix.

The 20-day model consists of three layers, each with an increasing number of filters. Since the images are all grey-scaled, the number of input channels is one. The input images are then passed to a convolutional layer, which consists of 64 filters, each with a size equal to 5×3. The activation function is the leaky ReLu. We also apply the maximum pooling with a pooling size equal to 2×1. This step effectively reduces the dimension of the feature map.

Since we are using the price plot images as input, we need to first 'imagine' the stock price time-series data. Each image contains two important pieces of information. First, the upper section of the graphs displays the OHLC bar, which represents the time-series movement of the security price. Second, the lower part of the graph represents the volume bar, which represents the daily trading volume.[3]

2.1 Evaluation

The model takes a historical price plot as input and feeds the input into a neural network. The model parameters are trained to minimize the cross-entropy loss.[4] The output of the model is a binary decision: up/down, which represents the prediction result, whether the stock price after 20 days will go up or go down.

For convenience, we define the accuracy of the model to be:

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{1}$$

Where *TP, FP, TN, FN* mean true positive, false positive, true negative, and false negative.

2.2 Replication Result

The model accuracy is 54% for the validation set, and 52% for the test set. We also present two replication results: i) the return and Sharpe ratio performance using the CNN model; ii) the CNN return in relationship with traditional predictors.

2.2.1 Performance Statistics

Using the architecture of the 'I20/R20' model[5] and the training method proposed by JBD, we trained the model using 5 different random seeds and averaged the predictions as our final output.

The summary of our replication result is displayed in **Table 1**.

Table 1 shows the performance by splitting the universe cross-sections by deciles monthly using both equally-weighted and market capitalization-weighted schemes.[6] As can be seen, the results are similar to that of JBD. The only notable difference is that the significance of the Sharpe ratio of long-short decile under the market capitalization-weighted scheme misses the 5% mark marginally.[7]

[3]More details of the image handling will be left in the appendix

[4]Details in the appendix

[5]I20R20 means the 20-days realized return base on model using 20-day image input.

[6]Equally-weighted means the investment consists of an equal amount of shares from each company, while value-weighted means the amount invested on one share depends on the share outstanding of that company.

[7]For JBD's result, please refer to table 3 and table 4, panel I20/R20 in their paper.

	Equally-Weighed		Value-Weighted	
Portfolio	Ret	Sharpe Ratio	Ret	Sharpe Ratio
1	-0.68%	-0.04	6.38%	0.38
2	5.08%	0.26	4.31%	0.26
3	8.55%	0.44	5.30%	0.32
4	9.20%	0.47	7.82%	0.49
5	10.72%	0.56	6.36%	0.39
6	12.36%	0.64	7.49%	0.47
7	12.20%	0.64	7.80%	0.48
8	13.85%	0.74	7.78%	0.52
9	15.48%	0.81	6.28%	0.39
10	19.96%	0.91	10.07%	0.64
H-L	**20.34%****	2.05	**3.69%***	0.35
p-value		0.00		0.06
turnover		174%		181%

Table 1: We divide the portfolio into 10 monthly deciles. Each has different probability of positive return (decile 1 has the lowest, while decile 10 has the highest). The H-L portfolio represents the portfolio constructed by long decile 10 and short decile 1 combination. Turnover represents the fraction of strategy that turns over on average. Our result is similar to the result reported by JBD. The only difference is the significance level of the H-L portfolio return. Our result shows that the return is significant in 10% level, while in JBD, the result is significant in 5% level.

2.2.2 Relationship with Traditional Predictors

In JBD, the authors ran a time series regression of portfolio returns on factors that are already known in academia (Mean Ret, Mkt-Rf, Momentum, STR, WSTR).[8] They do so to check: 1) how the CNN model is related to already known factors in the academia based on the slope of regressors, and 2) how much additional return CNN is generating on top of these factors from the magnitude of the intercept.

We are not able to replicate the time series regression due to data availability. Yet, we can test the usefulness of the CNN prediction using the alternative specification. Instead of using the regression structure in JBD, we choose the following three models for time series regression:

1. **Fama-French five-factor model**: [9].This is the most widely used benchmark model for asset pricing. As can be seen in **Table 2**, the factor is negatively loaded on the market factor, similar to the findings in JBD.

2. **Momentum factor** (past 12-month minus 1-month returns) **and Reversal factor** (past 1-month return). [10]They are the most well-known 'technical' factors.

3. **Low-Volatility (LowVol)**: Since Grad-CAM shows the images are most intensively activated amid high volatility, we also tested another well-known academic factor Low-Volatility, which is simply the inverse of stock volatility (exponentially weighted with 63-day half-life).

In particular, we use the above three linear regression models to identify the components of the CNN return which can be explained by these traditional linear models. **Table 2** summarise our findings. We can see from **Table 2** that the alpha intercept (i.e 'const') is statistically significant for equally-weighted portfolios and Fama-French 5 factor models. However for value-weighted strategies, alpha is no longer significant after considering momentum, reversal, or low volatility. This suggests this particular CNN model does not bring additional 'alphas' on top of well-known factors, especially for large-cap stocks.

[8]Table 5 in JBD shows the result of the regression. Appendix F provides more explanation of this subject matter.

[9]Returns are downloaded from Kenneth French Data Library [**Link**].

[10]Returns are downloaded from Kenneth French Data Library [**Link**].

	Equally-Weighted			Value-Weighted		
	Fama-French Five Factor	Momentum/ Reversal	LowVol	Fama-French Five Factor	Momentum/ Reversal	LowVol
const (alpha)	**0.02***	**0.02***	**0.02***	**0.00***	**0.00***	**0.00***
	(8.92)	(9.04)	(8.87)	(2.34)	(1.89)	(1.49)
CMA	-0.06			-0.20		
	(-0.49)			(-1.56)		
HML	0.08			0.02		
	(0.98)			(0.20)		
Mkt-RF	-0.19*			-0.18*		
	(-3.78)			(-3.25)		
RMW	0.15			-0.03		
	(1.69)			(-0.35)		
SMB	0.09			-0.06		
	(1.34)			(-0.79)		
Momentum		0.07*			0.08*	
		(1.86)			(2.19)	
Reversal		-0.08*			-0.20*	
		(-1.66)			(-3.84)	
LowVol			0.00*			-0.01
			(0.15)			(0.82)

Table 2: Loadings on Academic Factors, Regression on Returns (* indicates statistical significance at 5% level). The result shows that under the baseline CNN specification, the model predicted returns cannot be explained by using traditional models, indicating the potential benefits of adopting the CNN approach. However, the CNN model does not capture much additional information for the value-weighted portfolio.

3 Extensions

After replicating the baseline model, we now consider several extensions of the model. This session aims to explore whether incorporating new features to the model introduced by JBD can yield interesting new implications. In this report, we mainly consider three extensions: i) Grad-CAM; ii) Robustness, and iii) Regression.

3.1 Grad-CAM

One major drawback of a CNN is that it is difficult to interpret the result. All the parameters and model components are put together into a black box. Recent advancement in software (e.g. Tensorflow, Pytorch) enables the user to train the model without building the basic internal structures of each component of the neural network.

Grad-CAM aims to alleviate this issue, and provides a way to 'visualize' the model. Selvaraju et al (2017) suggested a way to display the 'heat-map' of each class and layer on CNN. The main idea is to use the gradient information flowing into a particular layer to understand the importance of each neuron.

To illustrate our trained CNN model, we randomly draw ten sample images in 2019 which are classified as "Up" by our CNN and ten images classified as "Down". We draw the Grad-CAM heatmap at each of our CNN's four layers in Figure 1 and 2. The row numbers in the figures stand for the depth of layers.

From these figures we have the following findings: i) The CNN is activated most heavily on days with high volatility, which is the same as JBD. If most days are low-volatility, CNN may be activated at the moving average line. ii) The opening and closing prices are brighter than in other areas, which means the model pays special attention to these prices. iii) In the fourth layer, the days with high volume are activated more intensively than the low-volume days, so high-volume days are more important when making final decisions by our model. iv) Compared to "Up" images, CNN concentrates more

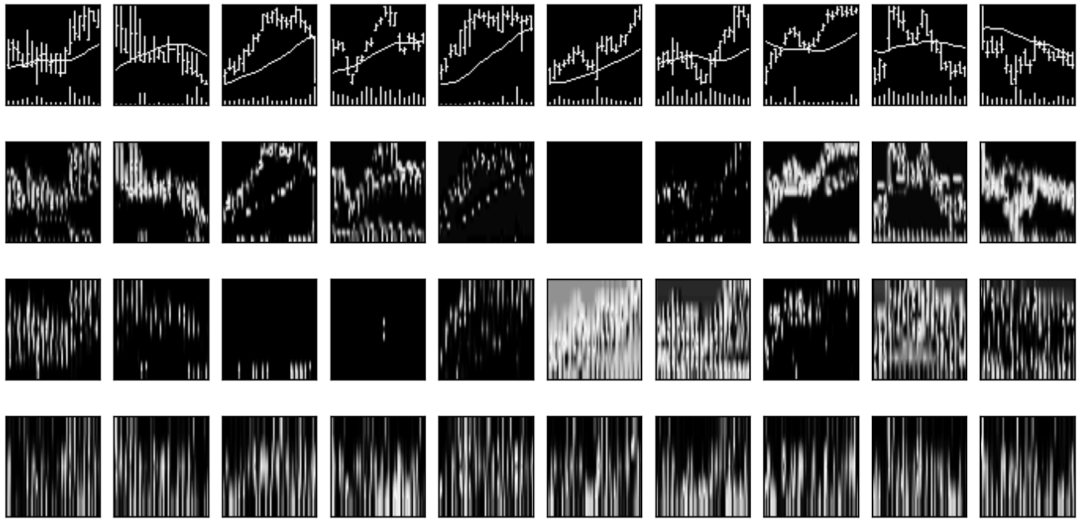

Figure 1: Images receiving “Up” classification

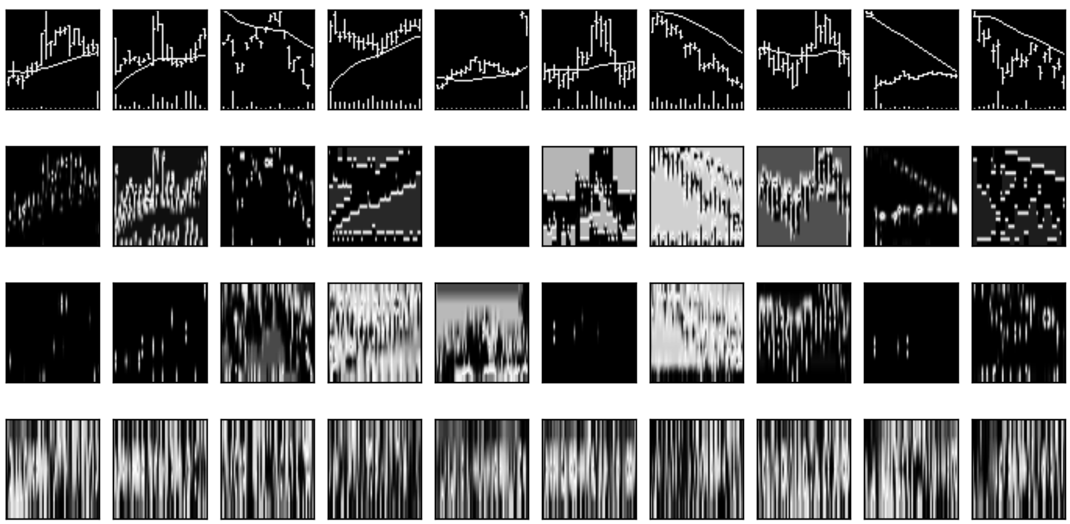

Figure 2: Images receiving “Down” classification

on decreasing moving average line and space in “Down” images. Also, CNN pays more attention to the bottom region in the last layer of “Up” images while the whole region for “Down” images, which means our CNN concentrates more on the volume when classifying images as “Up”. The last result, however, does not match the finding by JBD, who found that the "up" images show high activation in the upper regions, while "down" images are heavily activated in the bottom half of the image.

3.2 Robustness

Here we replicate the performance sensitivity to various dimensions of model structure and estimation choices. In Table 3 represents, we replicate performance sensitivity to various dimensions of model structure and estimation choices. We report sensitivity in terms of loss function value, classification accuracy, average cross-sectional correlation of forecasts with return realizations (both Pearson and Spearman rank correlation), and annualized Sharpe ratios for long-short decile strategies by following papers. We also add some new checking terms as shown in Table 4. The results show that the performance is insensitive to tuning the key parameters a little bit which proves our replication models are robust as well.

	Variations	Loss		Acc.		Correlation		SR	
		V	T	V	T	P	S	EW	VW
Baseline		**0.689**	**0.695**	**0.538**	**0.522**	**0.037**	**0.057**	**1.961**	**0.472**
Filter(64)	32	0.688	0.693	0.536	0.523	0.038	0.060	1.981	0.283
	128	0.689	0.695	0.540	0.524	0.028	0.045	1.904	0.486
Layer(3)	2	0.691	0.693	0.530	0.529	0.034	0.055	1.674	0.318
	4	0.689	0.696	0.539	0.520	0.033	0.053	2.029	0.372
Dropout(0.50)	0.00	0.725	0.724	0.515	0.523	0.023	0.034	1.644	0.097
	0.25	0.693	0.699	0.534	0.520	0.025	0.040	1.720	0.169
	0.75	0.688	0.693	0.540	0.520	0.037	0.059	1.904	0.063
BN(yes)	no	0.687	0.693	0.543	0.520	0.039	0.058	1.806	0.457
Xavier(yes)	no	0.694	0.695	0.527	0.528	0.037	0.053	1.943	0.379
Activation (LReLU)	ReLU	0.691	0.694	0.533	0.527	0.301	0.048	1.653	0.230
Max-pool Size (2×1)	(2×2)	0.689	0.694	0.534	0.523	0.029	0.050	1.678	0.471
Filter Size (5×3)	(3×3)	0.693	0.698	0.528	0.511	0.027	0.046	1.599	0.266
	(7×3)	0.692	0.697	0.532	0.515	0.028	0.045	1.854	0.307
Dilation/Stride (2,1)/(3,1)	(2,1)/(1,1)	0.696	0.704	0.527	0.501	0.031	0.052	1.759	0.155
	(1,1)/(3,1)	0.687	0.693	0.542	0.526	0.040	0.059	2.069	0.614
	(1,1)/(1,1)	0.703	0.711	0.518	0.492	0.040	0.063	1.707	0.557

Table 3: Sensitivity to Model Structure and Estimation, I20R20.
Note: it represents model performance sensitivity to model and estimation variations including the number of filters in the first layer (equal to 64 in baseline model), number of convolutional layers (baseline 3), dropout probability (baseline 0.50), use of batch normalization (BN, baseline yes), use Xavier initialization (baseline yes), activation function (baseline leaky ReLU), size of max-pooling layers (baseline (2,1)), filter size (baseline 5×3 pixels), and combinations of dilation and stride (baseline (2,1) and (3,1)). The columns show cross-entropy loss and prediction accuracy on the validation set (V) and the out-of-sample test set (T), out-of-sample Spearman(S) and Pearson(P) correlations,[12] and annualized decile spread Sharpe Ratio with equal and value weights.

3.3 Regression

One natural extension of the model would be regression. JBD uses binary output labels for the classification problem. This potentially loses some information about the cross-section of stocks and also creates imbalance problems during extreme bull/bear runs. Also, the performance of the long-short decile is driven by not the absolute performance, but the rather relative performance of stocks to their cross-section. Therefore instead we can consider training the model in a regression setting.

For the regression model, we revised the loss function to mean squared errors and the criterion for validation to (1 − R-squared). We tested future 1-month returns as well as 1-month Sharpe ratios as output variables. Given the images inherently scale the candlesticks by the volatility of a stock, we think Sharpe ratios (1-month return/volatility) is a more suitable target variable. To remove the excess impact of outliers, we normalize the target variables into relative rankings at each month and scale them to the range [-0.5,+0.5].

Table 5 shows the performance of regression models. The equally weighted portfolio achieves on-par performance with the baseline model. What is notable is the improvements in both Sharpe ratios and maximum draw-downs under the value-weight scheme.

Similar to the baseline model, we need to ensure that the predicted return using the CNN regression model cannot be fully explained by known model. We run regression on CNN regression return with respect to known factors as what we did to the baseline model. The result indicates that the CNN regression model can capture aspects which cannot be explained using known factors. We also try using the Sharpe ratio as the label to run regression. We leave the full analysis and model results to the appendix.

		SR	Vol			Max Draw-Down/Vol		
	Variations	MNC	EW	VW	MNC	EW	VW	MNC
Baseline		**1.832**	**0.100**	**0.111**	**0.031**	**0.660**	**2.005**	**0.975**
Filter(64)	32	1.782	0.107	0.119	0.032	0.716	2.990	1.168
	128	1.732	0.088	0.107	0.025	0.749	2.028	1.098
Layer(3)	2	1.697	0.113	0.103	0.028	0.673	2.673	1.178
	4	2.031	0.082	0.109	0.028	0.918	2.454	0.972
Dropout(0.50)	0.00	1.730	0.069	0.081	0.022	0.781	5.011	1.279
	0.25	1.778	0.079	0.091	0.025	1.073	3.123	1.073
	0.75	1.690	0.101	0.118	0.033	1.261	3.545	1.007
BN(yes)	no	1.879	0.131	0.111	0.029	0.287	2.459	1.272
Xavier(yes)	no	1.769	0.107	0.106	0.027	0.539	2.178	1.284
Activation (LReLU)	ReLU	1.605	0.106	0.104	0.028	0.523	3.972	1.292
Max-pool Size (2×1)	(2×2)	1.616	0.101	0.108	0.029	0.982	2.100	1.445
Filter Size (5×3)	(3×3)	1.322	0.095	0.107	0.031	0.657	3.380	2.030
	(7×3)	1.766	0.087	0.102	0.027	0.671	3.676	1.182
Dilation/Stride (2,1)/(3,1)	(2,1)/(1,1)	1.503	0.099	0.118	0.036	0.715	2.853	1.615
	(1,1)/(3,1)	1.861	0.110	0.114	0.026	0.454	2.056	1.273
	(1,1)/(1,1)	1.318	0.128	0.123	0.038	0.856	2.092	1.655

Table 4: Sensitivity to Model Structure and Estimation(continued), I20R20.
Note: The columns show annualized market neutral-volatility controlled (MNC) decile spread Sharpe Ratio, volatility and maximum Draw-Down divide volatility.

		Return Regression	**Sharpe Regression**	**Baseline Classification**
Loss	Validation	0.08	0.08	-
	Test	0.08	0.08	-
R-Square	Validation	0.01%	1.14%	-
	Test	−0.10%	−0.17%	-
Correlation	Pearson	0.05	0.04	0.04
	Spearman	0.07	0.06	0.06
SR	Equal-Weight	1.98	2.13	2.05
	Value-Weight	0.61	0.47	0.35
Vol	Equal-Weight	13.94%	10.84%	9.92%
	Value-Weight	12.68%	10.84%	9.92%
Max Draw-Down/Vol	Equal-Weight	**0.34**	0.40	0.64
	Value-Weight	**1.92**	3.03	3.05

Table 5: The table summarises the performance of using regression to predict the stock market return. In the regression result, the Sharpe Ratio result has similar performance as that in the Baseline classification, while the Max Draw-Down improves.

3.4 Alternative Architecture

Another extension would be a modification of architecture. The final layer of the baseline model consists of 46,080 fully-connected neurons. This could potentially lead to overfitting. We can help alleviate this by introducing more structure. Instead of using a fully-connected layer, we modify the structure by using a partially-connected layer. We let the output from each of the 256 filters of the final convolution layer feed into an individual network.[13] The output (256-dimension) then feeds into the final layer. We applied this feature to both regressions.

From **Table 6**, we can see that introduction of a partially-connected layer further improved both Sharpe ratios and maximum draw-downs under the value-weight scheme. However, the performance

[13] More details in the appendix.

of the equal-weight scheme is inconclusive. Further, we also see a pick-up in out-of-sample R-squared, indicating a better model fit of the test set.

		Baseline Classification	Regression: Sharpe	Regression: Sharpe + Partially Connected	Regression: Returns	Regression: Returns + Partially Connected
Loss	Validation	0.69	0.08	0.08	0.08	0.08
	Test	0.70	0.08	0.08	0.08	0.08
R-square	Validation	-	1.14%	1.02%	0.01%	0.88%
	Test	-	-0.17%	0.34%	-0.10%	0.31%
Correlation	Pearson	0.04	0.04	0.04	0.05	0.05
	Spearman	0.06	0.06	0.07	0.07	0.08
Sharpe Ratio	Equal-Weight	2.05	2.13	1.91	1.98	1.82
	Value-Weight	0.35	0.47	0.60	0.61	0.71
Vol	Equal-Weight	9.92%	10.84%	13.10%	13.94%	16.00%
	Value-Weight	10.64%	11.36%	11.88%	12.68%	15.40%
Max Draw-Down/Vol	Equal-Weight	0.64	0.40	0.44	0.34	0.41
	Value-Weight	3.05	3.03	2.53	1.92	1.28

Table 6: Performance of Regression + Sparse Linear Layers. This table shows that the Sharpe ratio and maximum draw-downs under value-weighted scheme are improved with partially-connected layers.

The result does not imply that a partially-connected network can outperform a fully-connected network. Rather, the alternative structure provides ensures that the model result is robust to the model architecture. This analysis provides insight to future research, highlighting the importance of considering alternative model settings while using neural networks in stock market estimation.

4 Conclusion

This report aims to replicate the findings by JBD, which suggested the usage of CNN in financial market prediction. Our replication result matches the findings by JBD. We also explore three extensions from their paper: i) Grad-CAM; ii) Robustness and iii) Regression.

We show that the regression result provides additional support to the usage of the CNN model in financial market modeling. Additionally, by introducing additional structure to the final fully-connected layer, model performance can be further improved during test time. Our result shows that CNN not only is useful in predicting the *direction* of the movement, but also the *magnitude* of the corresponding changes. This is important to our understanding of Machine Learning applications in the financial market, as it suggests that the signal generated by the CNN model can be used to make valid investment choices.

Future research can focus on the regression model. Our report shows that the regression result has strong predictive power to the future return. While our paper focuses on analyzing individual stocks, it can be beneficial to use the CNN architecture to build a diversified portfolio.

Appendix: (Re-)Imag(in) Price Trends - Replication and Extension

Alvin Lo, Ning Liu, Tony Tan, Ziqing Guo

A. Literature Review

This section discusses the relevant research in the field of machine learning application to stock market prediction. Research on the predictability of the stock market has been a major focus in the Finance literature. The Efficient Market Hypothesis (EMH), states that the stock return is unpredictable as the market is 'efficient', meaning that all prices should be fairly priced. This hypothesis has been a basic assumption used in both Finance and Economic research.

There is emerging research on applying modeling techniques in computer science to financial market prediction, the majority of them focus on applying neural networks to time-series data (Chen et al 2016, Kim and Kim 2019, Lee et al. 2019, Hoseinzade and Haratizadeh 2019). Gu et al. (2020) aimed at exploring the ability of different prediction models of Machine Learning in measuring the asset risk premium.[14] Hu et al. (2018) are the closest to this paper, who also use CNN to analyze price plots. Their paper, however, has a different focus. Their paper use CNN to identify the correlation of different stocks, and use the result to create a diversified portfolio to generate an alpha return. This paper, on the other hand, uses CNN to predict the price movement of individual stocks.

B. Details of the Baseline Model

This section provides a more in-depth explanation to the model in JBD. In the paper, they considers the 5/20/60 days model, while the focus of our paper is the 20-days model. The following figure shows the model building blocks, and the table shows the summary of all the ingredients in the 20-day model.

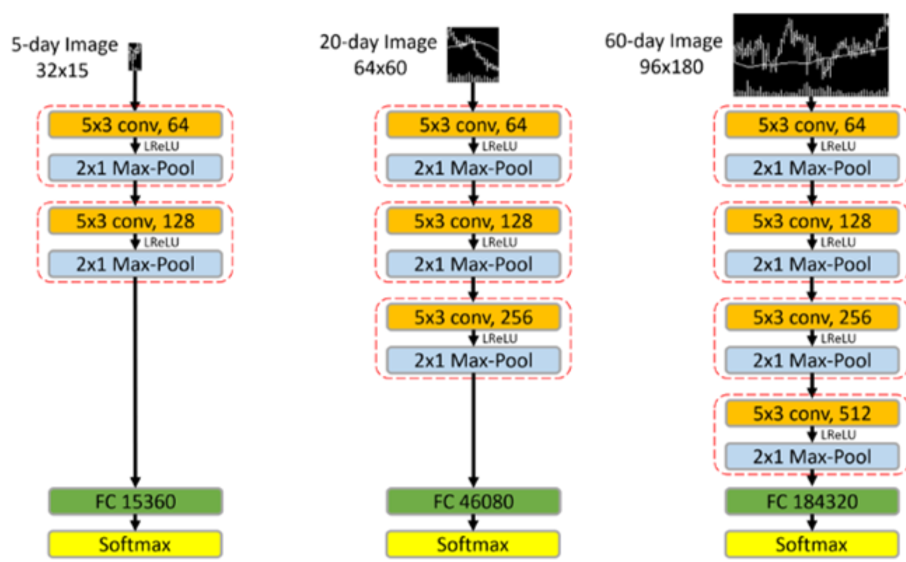

Figure 3: The graph is directly captured from JBD (diagram 7), which clearly outlined the building blocks for the 5/20/60- days model. Our report focus on the 20-days model.

The model is solved using Pytorch. There are in total 708,866 parameters, of which the parameters in the third convolutional layer constitute over half. The loss function we use is the Cross-Entropy. We train the model using 50 epochs. The training time is about one and a half hours using Google Colab and takes less than 30 minutes using GPU. The convergence result is shown in figure 4.

[14]The 'equity premium puzzle, which states that the asset return in the US market is disproportionally large compared to the underlying risk, is arguably the most famous problem in the Finance literature.

Components	Details
Input Channels	1
Number of layers	3
Filters in each layers	64, 128, 256
Filter size	(5,3)
Dilation	(2,1)
Strike	(3,1)
Parameter initialization	Xavier Initializer
FCC Dimension	40680

Table 7: Summary of the model building blocks.

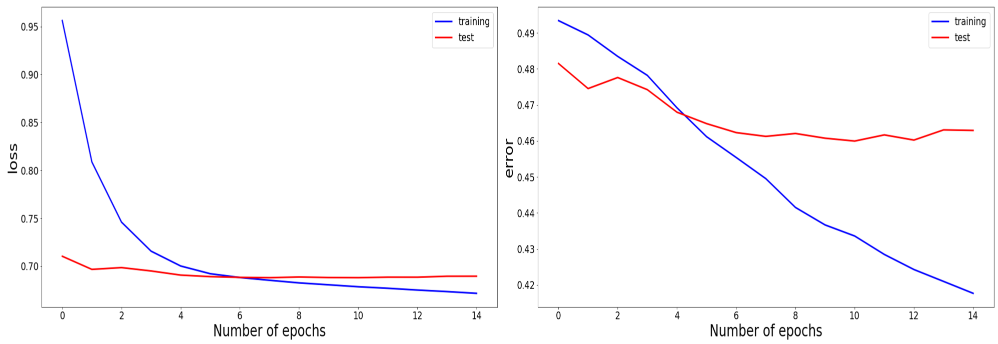

Figure 4: The graph shows the convergence result. The left panel shows the value of the loss function versus the number of epochs. The right panel shows the error of the model versus the number of epochs.

C. Input Image

The images used in this paper are provided by [**link**]. We omit the process of imaging the price series. The price plot is displayed in the OHLC chart, which shows the opening, high, low, and closing price of the stock on a given trading day. Figure 5 displays the input images we use as our model input. We also shows the OHLC graph which is normally used in the financial industry.

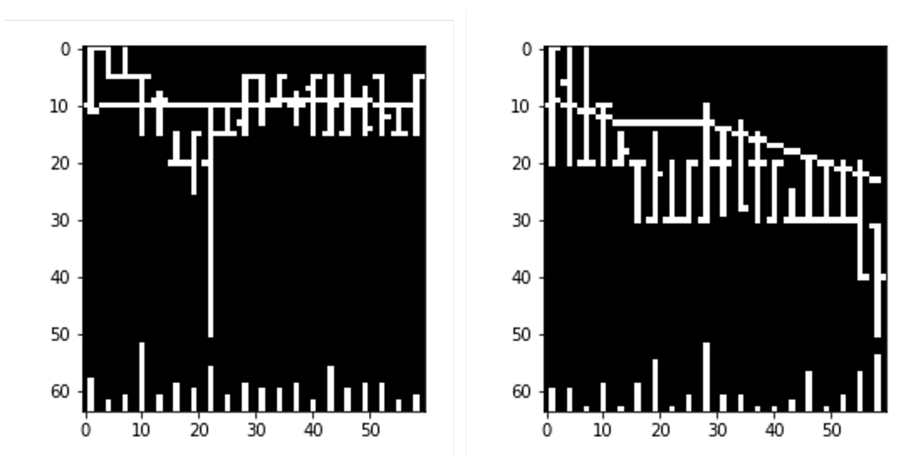

Figure 5: Examples of price image input. There are two parts in the graph: first, the upper portion of the graph shows the OHLC price bar, which contains the information of the opening, high, low and closing price. The lower portion of the image is the volume bars, which capture the trading volume data.

Figure 6: The figure shows the OHLC plot. In the paper and our report, we adopt the down OHLC (the left panel). The high-end and low-end of the bar represents the daily high and daily low price. The horizontal line on the left represents the opening price, while that on the right is the closing price.

The time-series data of the price are obtained from the Center for Research Stock Price (CRSP). The first step after obtaining the price series data is to transform them into price images. We omit this part in our replication, but we are will explain the procedure here.

Price Bars

First, we combine the OHLC bars over 5/20/60 days (depending on the model). The width of the resulting graph is increasing with a longer interval. We then normalize the data so that the maximum and minimum of each OHLC graph match the upper and lower end of the image. Hence, images with fixed interval lengths will have the same pixel.

It is important to stress the information captured in each image. First, the vertical part of the image describes the direction of movement. It shows how the price is moving over time. Secondly, the OHLC bars also capture the daily volatility of the stock price, as the high and low prices can be a good proxy for the daily fluctuation.[15]

Volume Bars

The second component of the image input is the volume bar. The trading volume is proven to be important in price prediction. In creating the figure, the upper fourth-fifth is the price plot, and the bottom one-fifth will be the volume bar plot.

D. Traditional Stock Predictors

There are several traditional predictors for stock return. One major evaluation of a stock price predictor is whether this new predictor can outperform the traditional predictor. Here I will introduce three important predictors used in the literature.

1. **Momentum (MOM)**: Measured as the average return over the two to twelve months prior to forecast.
2. **Short-Term Reversal (STR)**: Measured as the one-month return prior to the forecast.
3. **Weekly Short-Term Reversal (WSTR)**: Measured as the one-week return prior to the forecast.

While comparing the traditional predictors with the new CNN predictor, the new predictor can outperform these traditional predictors and generate real benefits. The result is shown in table 8.

[15] Parkinson (1980) shows that under the assumption that security price follows the Brownian motion, using the extreme value method (i.e. high-low estimation) can provide a good estimation of the diffusion.

	Return Horizon			
	20-day		60-day	
Image Size	Acc.	Corr	Acc.	Corr.
5-day	52.5%	3.1%	53.6%	2.3%
20-day	53.3%	3.4%	53.3%	2.4%
60-day	53.6%	2.9%	52.9%	3.1%
MOM	52.1%	1.8%	52.1%	1.5%
STR	50.4%	1.4%	49.8%	1.2%
WSTR	51.0%	2.8%	50.4%	2.6%

Table 8: The full result is taken from JBP. In their paper, they use out-of-sample analysis to test the accuracy of the prediction result. The result shows that the CNN model (top three rows) can outperform the traditional price predictor (bottom three rows).

	Equal Weight	Value Weight
Mean Ret	1.73%	0.45%
	(9.4)	(2.1)
Alpha	1.86%	1.31%
	(10.7)	(3.9)
Mkt-Rf	-0.32	-0.21
	(-7.3)	(-4.0)
Momentum	0.01	0.04
	(0.3)	(0.9)
STR	-0.07	-0.09
	(-1.3)	(-1.4)
WSTR	0.37	-0.92
	(1.2)	(-2.5)
R^2	26.8%	21.6%

Table 9: The table show the result from JBD (table 5 in their paper). The main message from the table is that the CNN classification model can explain stock return which cannot be explained by the traditional known factors. This can be justified by the significance of the Alpha in the regression model.

E. Finding the Alpha Signal

Finding the alpha signal has always been the focus of Finance research and also in the hedge fund industry. In the main context, we always use the Sharpe ratio to examine the 'power' of the alpha. The implicit logic behind this is that we are comparing the strategy of our investment to the risk-free investment. However, this is not what we should do in reality, as there are many ways we can construct a risk-free (or close to risk-free) portfolio that can generate a return higher than the risk-free rate.

The way we analyze the return generated from the CNN model is to understand *where the return comes from.* If the return generated from our model can be explained by other traditional models, then the CNN model is redundant, as we can construct another model which is solely based on the existing method and can replicate the result we generate from our 'new' methodology.

In JBD, they test whether the CNN can generate new alpha by running a regression on CNN returns with respect to some known return predictors. If the constant term is significant, there exist unexplained component of the CNN return. The five known predictors are i) mean return; ii) risk-free rate; iii) momentum; iv) short-term reversal and v) weekly short-term reversal. The result of the regression is shown in **table 9.**

$$\text{CNN Return} = \alpha + \beta_1 CMA + \beta_2 HML + \beta_3 Mkt - RF + \beta_4 RMW + \beta_5 SMB + \epsilon \quad (2)$$

Contrary to what is done in the paper, we explore alternative model to detect unexplained alpha due to lack of relevant dataset. There are some new existing ways to predict stock return, the most popular one is the 'Fama-French Five-Factor Model'. In their paper, they conclude that the stock return can be explained by five factors: i) size; ii) value; iii) quality; iv) momentum; v) volatility. Our goal is to test whether the return generated from the CNN model can be explained by the traditional method.

F. Grad-Gam

Grad-Gam was introduced by Selvaraju et al. (2017). The main advantage of using Grad-Cam is to help conceptualize the binary classification methodology used in CNN. More specifically, a Grad-Cam is a way to visualize the classification method by displaying the heatmaps for each layer of the CNN that illustrate the regions of the input most important for predicting a class. The Grad-Cam can be produced using the following equations:

$$L^c_{\text{Grad-CAM}} = ReLU\left(\sum_k \alpha^c_k A^k\right), \quad \alpha^c_k = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}},$$

where y^c is the score of class c, A^k is the k-th feature map, Z is the size of the k-th feature map. Note that α^c_k represents the "importance" of feature map k for a target class c. The final Grad-CAM $L^c_{\text{Grad-CAM}}$ is the ReLU of the weighted combination of forwarding activation maps.

G. Supplementary Result From the CNN Model

Table 10 and **Table 11** demonstrate statistically significant alphas under all three time series specifications, even for value-weighted portfolios. This is a significant improvement compared to the baseline classification model.

By capturing more information about the cross-section, the model becomes more additive in terms of its alpha contributions and more feasible to implement in larger caps.

	Value-Weighted			**Equally-Weighted**		
	Fama-French Five Factor	Momentum/ Reversal	LowVol	Fama-French Five Factor	Momentum/ Reversal	LowVol
const	**0.01***	**0.01***	**0.01***	**0.02***	**0.02***	**0.02***
	(3.34)	(3.30)	(2.87)	(8.72)	(8.96)	(8.90)
CMA	-0.08			0.03		
	(-0.50)			(0.21)		
HML	0.09			0.10		
	(0.82)			(0.89)		
Mkt-RF	-0.30*			-0.30*		
	(-4.57)			(-4.14)		
RMW	-0.04			0.08		
	(-0.34)			(0.66)		
SMB	0.11			0.14		
	(1.23)			(1.47)		
Momentum		0.19*			0.13*	
		(4.40)			(2.50)	
Reversal		-0.25*			-0.17*	
		(-4.12)			(-2.48)	
LowVol			-0.01			-0.10*
			(-0.56)			(-3.93)

Table 10: Loadings on Academic Factors, Regression on Returns (* indicates statistical significance at 5% level). The significance of the constant term indicates that the return generated from the CNN regression model is orthogonal to the known factors in the literature, implying the additional benefits of considering the image-based model relative to using the traditional model.

G1. Partially-Connected Network

In a partially connected network, not all nodes in the first layer are connected to all nodes in the second layer. While in a fully-connected network, all nodes in the first layer are connected to all nodes in the second layer. Figure 7 shows more comparison between the partially-connected and the fully-connected model.

	Value-Weighted			Equally-Weighted		
	Fama-French Five Factor	Momentum/ Reversal	LowVol	Fama-French Five Factor	Momentum/ Reversal	LowVol
const	0.01*	0.00*	0.00*	0.02*	0.02*	0.02*
	(2.79)	(2.21)	(2.13)	(9.79)	(9.49)	(9.37)
CMA	-0.12			0.04		
	(-0.84)			(0.30)		
HML	-0.02			0.10		
	(-0.26)			(1.18)		
Mkt-RF	-0.28*			-0.28*		
	(-4.79)			(-5.24)		
RMW	0.01			0.02		
	(0.14)			(0.21)		
SMB	0.10			0.10		
	(1.21)			(1.38)		
Momentum		0.18*			0.13*	
		(4.41)			(2.28)	
Reversal		-0.08*			-0.11*	
		(-1.37)			(-1.93)	
LowVol			0.00			-0.05*
			(-0.06)			(-2.41)

Table 11: Loadings on Academic Factors, Regression on the Sharpe ratio (* indicates statistical significance at 5% level). The result shows that the constant term is significant, meaning that using the regression specification can detect meaningful alpha.

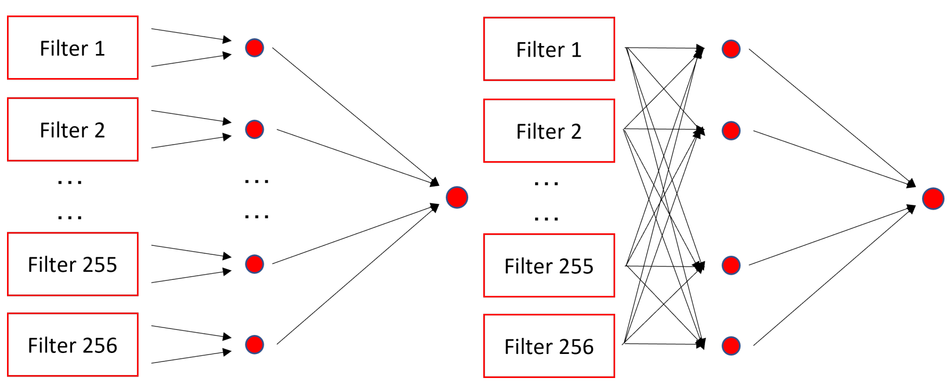

Figure 7: Architecture of the partially connected layer (left panel) and a fully-connected network (right panel). The partially-connected network does not requires all nodes from the previous layer to be connected to all nodes in the next layer.

Authorship Contribution Statement

- **Alvin Lo**: Discussed of project flow, designed and wrote the report, wrote the slides and do the presentation.
- **Ning Liu**: Discussed of project flow, made substantial contribution to the robustness check part of the project.
- **Tony Tan**: Discussed of project flow, made substantial contribution to the baseline model, regressions and sparse linear extensions.
- **Ziqing Guo**: Discussed of project flow, made substantial contribution to the Grad-Cam part of the project.

References

[1] Chen, Jou-Fan, Wei-Lun Chen, Chun-Ping Huang, Szu-Hao Huang, and An-Pin Chen, 2016, Financial time-series data analysis using deep convolutional neural networks, in 2016 7th International conference on cloud computing and big data (CCBD), 87 - 92, IEEE.

[2] Eugene F. Fama, Kenneth R. French, A five-factor asset pricing model, Journal of Financial Economics, Volume 116, Issue 1, 2015, Pages 1-22, ISSN 0304-405X, https://doi.org/10.1016/j.jfineco.2014.10.010. (https://www.sciencedirect.com/science/article/pii/S0304405X14002323)

[3] Shihao Gu, Bryan Kelly, Dacheng Xiu, Empirical Asset Pricing via Machine Learning, The Review of Financial Studies, Volume 33, Issue 5, May 2020, Pages 2223–2273, https://doi.org/10.1093/rfs/hhaa009

[4] Hoseinzade, Ehsan, and Saman Haratizadeh, 2019, Cnnpred: Cnn-based stock market prediction using a diverse set of variables, Expert Systems with Applications 129, 273-285.

[5] Hu, Guosheng, Yuxin Hu, Kai Yang, Zehao Yu, Flood Sung, Zhihong Zhang, Fei Xie, Jianguo Liu, Neil Robertson, Timpathy Hospedales, et al., 2018, Deep stock representation learning: From candlestick charts to investment decisions, in 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), 2706-2710, IEEE.

[6] Jiang, J., Kelly, B. T. & Xiu, D., (2020) (Re-)Imag(in)ing Price Trends. *Chicago Booth Research Paper No. 21-01*, Available at SSRN: https://ssrn.com/abstract=3756587 or http://dx.doi.org/10.2139/ssrn.3756587

[7] Kim, Taewook, and Ha Young Kim, 2019, Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data, PloS one 14, e0212320.

[8] Lee, Jinho, Raehyun Kim, Yookyung Koh, and Jaewoo Kang, 2019, Global stock market prediction based on stock chart images using deep q-network, IEEE Access 7, 167260-167277.

[9] Parkinson, M. (1980). The Extreme Value Method for Estimating the Variance of the Rate of Return. The Journal of Business, 53(1), 61–65. http://www.jstor.org/stable/2352357

[10] Selvaraju, Ramprasaath R, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, 2017, Grad-cam: Visual explanations from deep networks via gradient-based localization, in *Proceedings of the IEEE international conference* on computer vision, 618 - 626