(Re-)Imag(in)ing Price Trends:

Cao Bokai,Luo Zhuang,Shi Jie,Lai Fujie

November 14, 2021

Abstract

In this paper reproduce report, we mainly focus on five parts. First is the data preparation, we figure out the reason why we use images instead of time series data. Second is we design a CNN model that is very similar to the original CNN architecture in the paper and give our reasons for choosing the model. Thrid is that we design workflow and model details including each tunning parameter and then train our model. After that, we give performance evaluation and interpretation of the result based on our model, particularly in the testing set performance, portfolios performance, and CNN visualization methods. We make our conclusion that the CNN constructs image-based forecasts that in general outperform (and are largely distinct from) traditional price trend signals in the asset pricing literature.

Workload: Cao Bokai finished the main code, tuned the parameters, trained and saved the model, and tested its performance as well as controlled classification accuracy.Luo Zhuang participated in part of the model building, performance evaluation and Grad-GAM code writing, and wrote the corresponding report. Lai Fujie does partly model design and constructs job, also manage portfolio reproduce and writes the report. Shi Jie finished data preparation and model design parts

Part I: Data preparation

We use the same data as the original paper, including the stock data of 26 years from 1993 to 2019. The most important feature of the original paper and our report is to convert the stock data from a simple one-dimensional time series to a two-dimensional image. There are three main purposes. First, the data format that CNN can process is image rather than time series, secondly, converting the time series into images may be more helpful for the model to predict the stock data, finally, putting the price and trading volume data in the same image may be more helpful for the model to understand the data relationship and improve the prediction effect, and the ultimate purpose of this treatment is to predict the trend of stock price. The original paper also includes the moving average price of 5, 20 and 60 days, our report only includes the moving average price of 20 days, "OHLC" bars that decimal daily opening, high, low, and closing prices, as well as the daily trading volume of stocks. We selected "OHLC" bars rather than a candlestick chart, because OHLC bars use fewer pixels to convey the same information. The important feature of the image constructed in the original paper is that the conjunction 20-day interval of the daily data, and unified the height of all images is, scaled the vertical axis. We use the following figure to represent different price indicators in order to convey information more concisely.

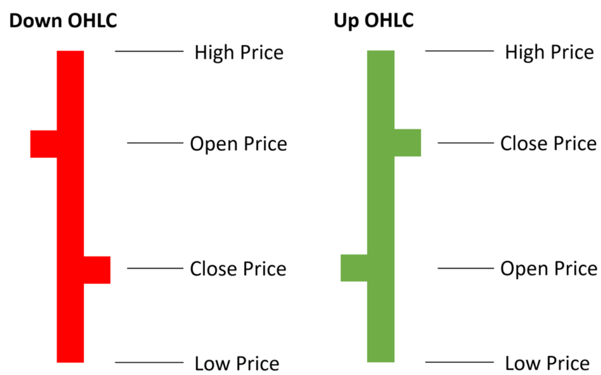

Figure 1: OHLC chart

The vertical dimension of the image shows two important points, The first is the trend of stock price, which directly reflects the trend of stock price, and the second is the volatility of price and trading volume. Our data does not contain stocks that IPO or delisted during the data window, but we allow missing data if it occurred in the middle of the stock's history. If there is missing data, the columns of pixels corresponding to the missing days are left blank. We use black as the background color and white as charts to minimize the memory occupied by pictures.

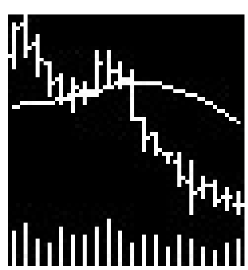

Figure 2: Final picture

The final picture contains a wealth of information, including not only the stock price and trading volume data, but also the efforts made to fit to the CNN model.

We have added 20-day moving average to the data to better show whether the current price deviates from the long-term price trend, which is more conducive to discovering the potential change trend of stock price. We also added the daily stock trading volume and display it at the bottom of the picture, the height of trading volume does not exceed one fifth of the picture height, the maximum trading volume height is set to one fifth of the picture height, and the other trading volumes are scaled accordingly. Each picture is expressed as matrix of pixel values in our dataset, where 0 for black pixels and 255 for white pixels.

Part II: Model Design

One of the most important features of the original paper and our report is the choice of CNN model. The advantages of choosing CNN are obvious, because CNN requires much less parameters than traditional neural networks, so it requires much less data. At the same time, traditional neural networks are inherently sensitive to position, scale, and orientation of an object in the image. CNN model can well solve the above two problems, finally, a better prediction result is obtained.

A core building block concerns of three operations: convolution, activation, and pooling, We also choose a variety of different filter to extract different features. The existence of filters helps to identify some specific shapes in the picture. After using the filter, we go to the "activation" step. We select "leaky ReLU." as the activation function, takes the max of the filter output value and zero. The last step is "Max pooling", in this step, we continued to use the filter and select the maximum value in the range selected by the filter. Max pooling has two main functions, firstly, Max pooling can reduce the dimension, secondly, the method of selecting the local maximum also effectively avoids the disturbance of unimportant data to the final result. Within a building block, the output from all convolutional filters are fed element-wise through the leaky ReLU activation function. The output of ReLU activation function continues to the next building block. After processing, we can finally get the linear combination of vectorized pictures, after these linear combinations are processed by softmax function, we can get the probability of the rise of stock prices in the future.

We now discuss specific choices for our models. Since our images are largely sparse in the vertical dimension, we use 5×3 convolutional filters and 2×1 max-pooling filters. We use the same filter sizes in all layers for convenience. We use horizontal and vertical strides of 1 and 3 and vertical dilation rates of 2 for 20-day images, respectively, only on the first layer because raw images are sparse. The number of CNN building blocks in our model is based on the size of the input image. We use 3 blocks for 20-day images. The number of filters for the first building block is 64. According to the original paper and the requirements of Zeiler and Ferguson (2014)[1] , we increase the number of filters after each revolutionary layer by two, finally, the fully connected layers have 46, 080 neurons for 20-day models, specifically, the total number of parameters are 708, 866.

So why do we choose to convert 1D time series into 2D images? Because 2D images help the convolutional filter to obtain some nonlinear spatial relationships of prices, while the 1D kernel filter cannot do. Similarly, CNN can automatically process pictures and extract information from them, while the traditional time series requires us to manually find different models to calculate various indexes and relationships in data, so the final effect may not be so ideal.

In short, transforming 1D time series into 2D images can enable CNN model to find some hidden data relationships. Just as human beings are more sensitive to graphics rather than numbers, CNN model may be able to obtain some effective information from graphics for prediction.

Part III: Work Flow Design

In the original paper, authors'workflow from training, to model tuning, and finally to prediction follows the basic procedure outlined by Gu et al.So we mainly did the same things. First, we divide the entire sample into training samples from 1993-1999, among which we randomly pick thirty percent as validation samples and out-of-samples testing data from 2000-2019. To realize training we need images data and their labels so that we can treat the prediction analysis as a classification problem.In

[1] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks[C] European conference on computer vision. Springer, Cham, 2014: 818-833.

particular,let y represents label,then y = 1 if the subsequent return is positive and y = 0 otherwise.The overall training step aims at minimizing the cross-entropy loss defined as

$$L(y, \hat{y}) = -ylog(\hat{y}) - (1 - y)log(1 - \hat{y})$$

and we use the Adam algorithm to renew network weight. To realize training we read in all the images data which already become digital data,i.e each image stored as 64*60 memmap data type, and then their labels as the target output. For the faster training process, we use a batch size of 64, which is a little different from paper's 128 batch size, to deal with out of memory problems. We set the learning rate as 10^{-5}as the paper said and train for 100 epochs. If the loss function doesn't improve in the validation set in consecutive two epochs then we stop training and save the model as our final model. In this way, we deal with the overfitting problem as what's paper said.

The small difference between our model and the papers is that we did not use 50 % drop out in our final linear layer when do I20R20 reproduces but we did add drop out layers after CNN layers when do I20R60 reproduce, just expect to see what's the difference.

After training and validation, we can use the testing set to our trained-well model and give a prediction. Results are shown below.

Part IV: Performance evaluation

Classification Accuracy

We use the trained CNN model tested in a test sample from 2001 to 2019 and calculate the precision year by year. Accuracy follows the definition in the paper as Accuracy = $(TP + TN)/(TP + TN + FP + FN)$

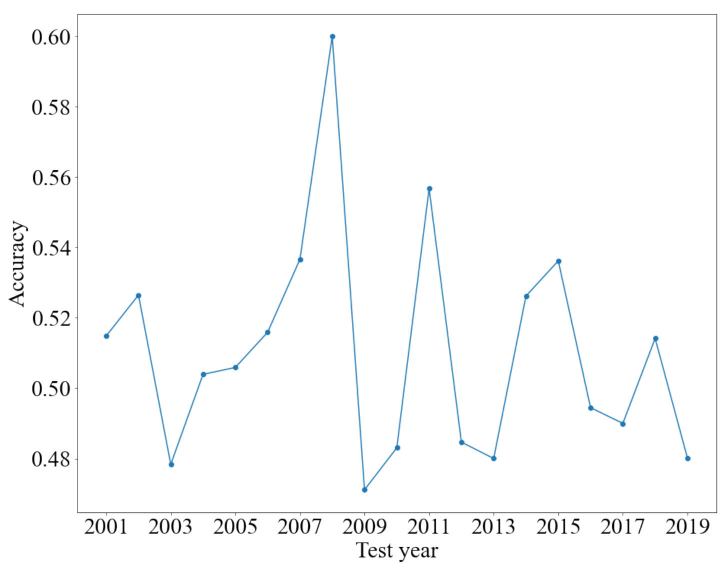

Figure 3: Out-of-Sample Classification Accuracy

Figure 3 summarizes the out-of-sample classification accuracy of the CNN at the stock level. It can be seen that the accuracy of CNN slightly exceeds that of random guesses (0.5) and performs best in 2008, probably due to the drastic fluctuations caused by the financial crisis.

Portfolio Performance

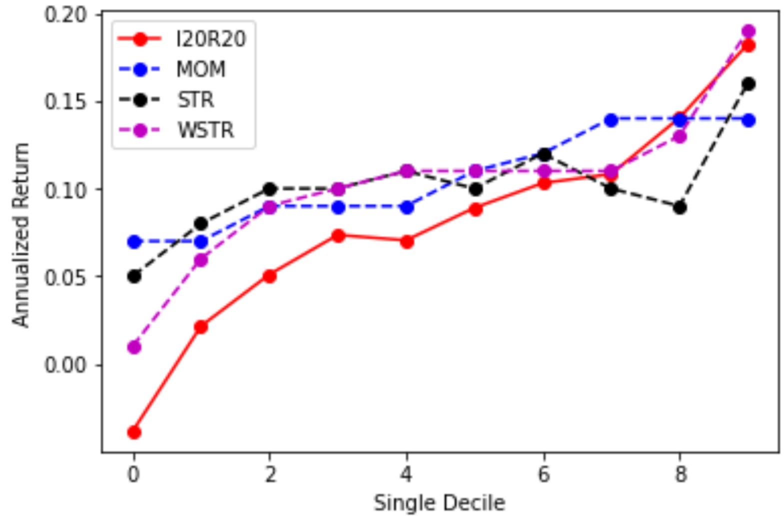

Figure 4: Reproduce:Realized returns in each decile of I20R20 CNN model forecasts and for each decile of the benchmark signals

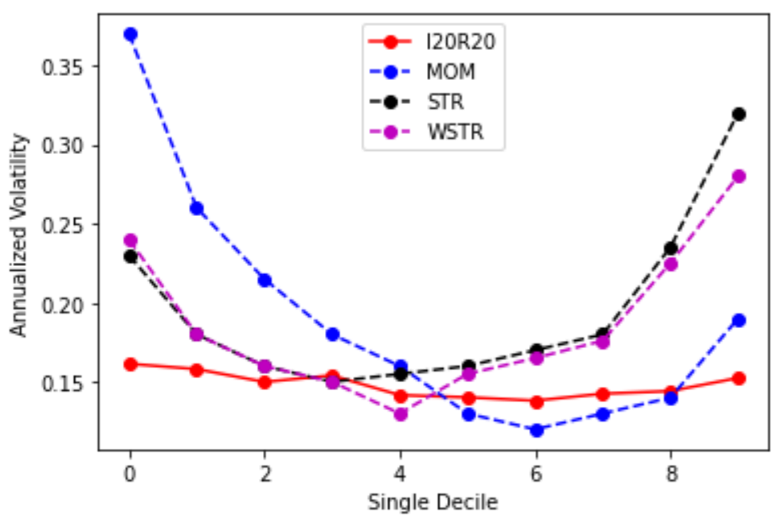

Figure 5: Reproduce:Time series volatility of decile returns

In this part, we produce some portfolios performance generated on testing data based on our reproduced I20/R20 and I20/R60 models. "Ix/Ry" to denote that the model uses x-day images to predict subsequent y-day holding period returns. Particularly we show decile portfolios sorted on out-of-sample predicted up probability. Decile 1 corresponds to stocks having the lowest probability of a positive future return and Decile High is related to the highest.

We can see from Figure 1 and Figure 2 that the trend and data of our portfolio are very similar to the original paper, but not totally the same. There are several reasons to explain that . The first is that our model's struct is not totally the same as the original one, e.g we give up 50% drop out on our I20/R20 model. The second is that machine learning methods are statistical methods and they may have some statistical error in that point. Third is that training is on computers that have different

Table 1: Reproduce:Performance of Equal Weight Portfolios

	I20/R20 Ret	I20/R20 Vol	I20/R60 Ret	I20/R60 Vol
Low	-0.038436	0.161422	0.244186	0.250674
2	0.021509	0.158073	0.226058	0.249356
3	0.050951	0.149941	0.215140	0.248701
4	0.073544	0.154072	0.236076	0.258125
5	0.070454	0.141639	0.222883	0.246025
6	0.088964	0.140229	0.230534	0.253513
7	0.103112	0.138123	0.218860	0.255481
8	0.108225	0.142517	0.234777	0.251809
9	0.140338	0.144171	0.244452	0.254395
High	0.192321	0.152488	0.237496	0.251448

Table 2: Reproduce:Performance of Value Weight Portfolios

	I20/R20 Ret	I20/R20 Vol	I20/R60 Ret	I20/R60 Vol
Low	0.010633	0.247295	0.157925	0.377766
2	0.035356	0.236314	0.186158	0.388291
3	0.031114	0.231220	0.167615	0.392990
4	0.039861	0.239669	0.187282	0.381606
5	0.051332	0.225950	0.172423	0.385467
6	0.063910	0.220664	0.194993	0.380871
7	0.081608	0.222273	0.151255	0.387423
8	0.053996	0.219794	0.172302	0.370160
9	0.078436	0.219458	0.163200	0.375164
High	0.093453	0.210918	0.170970	0.383456

properties, which hardly make sure the same experiment environment.

Details of equal weight portfolio and value weight portfolio data are shown in the table. In the table, each panel reports the average annualized holding period return and volatility of return, which is different from the original paper's Sharp ratio, but not crucial. More positive CNN forecasts translate more or less monotonically into higher returns on average, which are the same conclusions as the papers. At the same time, CNN models have more stable annualized volatility compared with volatilities of MOM, STR, and WSTR.

Part V: Interpretation of result

Controlled Classification Accuracy

Since the classification is not satisfactory, we wonder whether the prediction accuracy becomes better when the difference between the two probabilities of the model output is larger, i.e., the model is more confident. Defining the entry signal as $\max(\text{Up prob}, \text{Down prob}) > \text{Parameter}$, we test it on test set. The results are shown in Figure 6. As the parameters increase, the more confident the model is in its prediction results and the more accurate it is on the test set.When the entry signal comes to 0.6, there is a significant increase in classification accuracy to 55%.

Grad-CAM

Understanding and interpreting the decision made by CNN is of central importance for humans since it helps to construct the trust of CNN models. Especially for stock trading, a completely incomprehensible system is unlikely to allow users to invest their money with confidence.

Since the innovation of the paper is to use CNN to identify patterns that cannot be reached by traditional technical analysis, the results can be more directly observed using CNN visualization

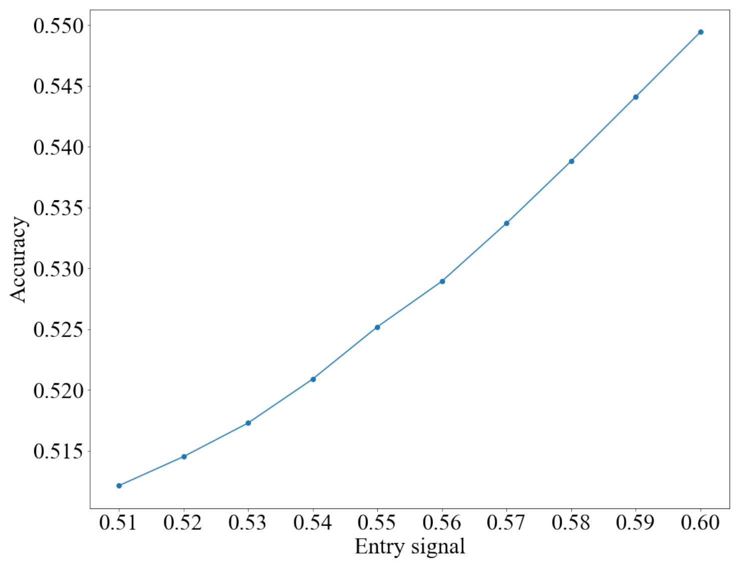

Figure 6: Controlled Classification Accuracy

methods. In the area of computer vision, one critical technique is generating intuitive heatmaps that highlight regions, which are most related to CNN's decision. Towards better explanations of CNN, Gradient visualization, Perturbation, Class Activation Map (CAM) are three of the widely adopted methods.

Gradient-based methods backpropagate the gradient of a target class to the input layer to highlight image region that highly influences the prediction. These maps are generally of low quality and still noisy. Perturbation-Based approaches work by perturbing original input and observe the change of the prediction of model. To find minimum region, these approaches usually need additional regularizations and are time-costing.

CAM-Based explanation provide visual explanation for a single input with a linear weighted combination of activation maps from convolutional layers. CAM creates localized visual explanations but is architecture-sensitive, a global pooling layer is required to follow the convolutional layer of interest. Grad-CAM and its variations, e.g. Grad-CAM++ , intend to generalize CAM to models without global pooling layers and finally adopted widely in the community. This is also the author's approach to visualization in the paper. For each class ("up" or "down" returns in our setting), Grad-CAM produces heatmaps for each layer of the CNN that illustrate the regions of the input most important for predicting a class.

We followed the authors' steps and randomly drew 5 images from the last year of the sample (2019) that were classified as "rising" by the CNN and 5 images that were classified as falling by it. Figure 5 and Figure 6 shows the original images and the grad-CAM heat map for each of the four layers of the CNN.

The bright areas of the image are the patterns identified by the CNN. As seen in the image, in the first layer the model focuses on the noteworthy opening and closing prices. The higher volume days in the second and third layers highlight the brightness of the price series. The volume area itself is activated only on days with particularly high volume, in which case the volume bars are mixed with the price bars to form wide vertical activation areas.

However, in general, Grad-CAM does not give us a direct feeling, it is difficult to infer specific technical analysis tools from the results. The impact of historical price movements and trading volume on future price movements seems to remain vague.

Figure 7: Images Receiving “Up” Classification

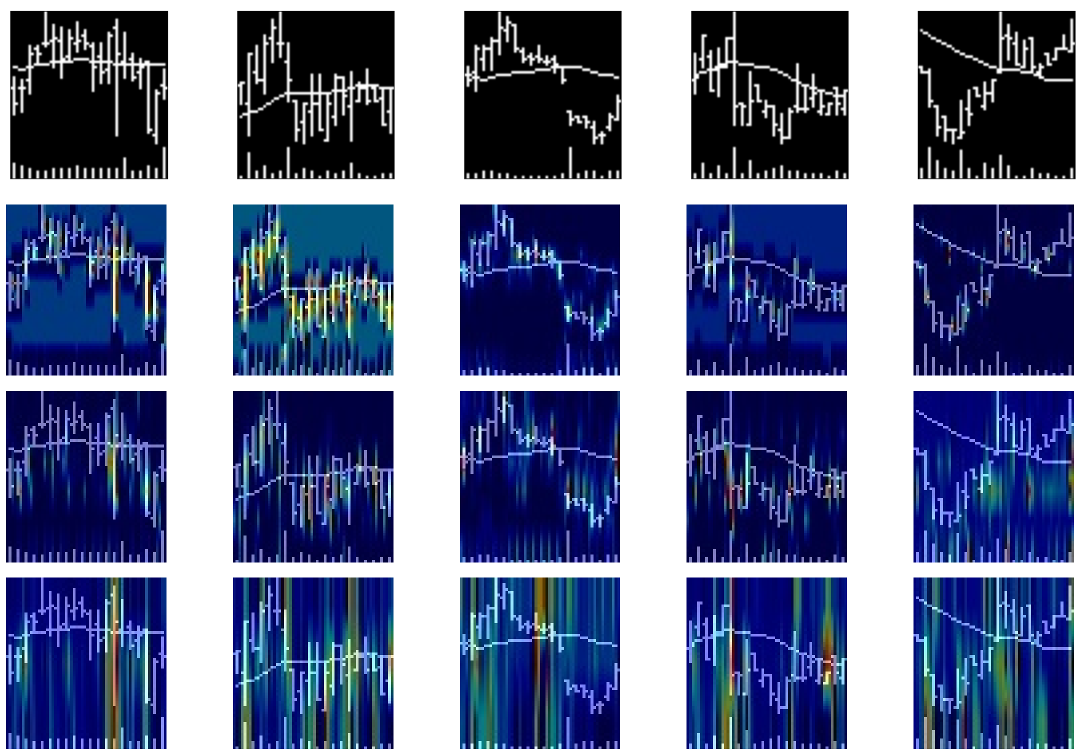

Figure 8: Images Receiving “Down” Classification