# Project2: Replication of **Introduction to "(Re-)Imag(in)ing Price Trends"**

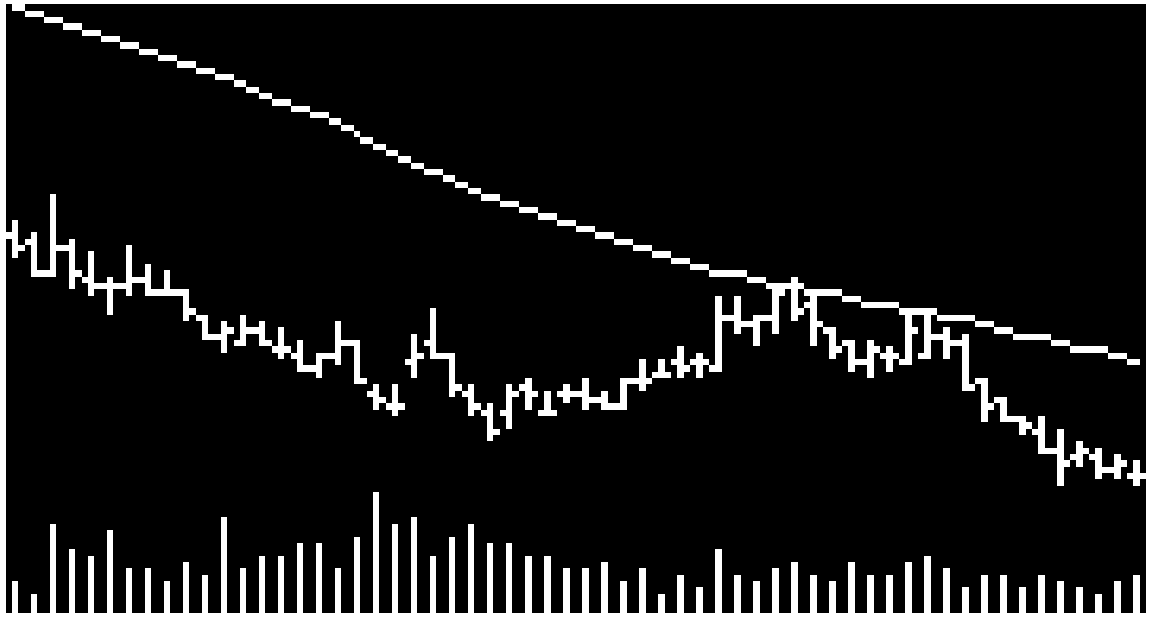
By

Jingwen Jiang
University of Chicago

Bryan Kelly
Yale University, AQR Capital Management, and NBER

Dacheng Xiu
University of Chicago Booth School of Business

# TLDR

- Encode market data as an **image** rather than in the more standard time series numerical format
- Do not require the researcher to pre-specify a set of technical patterns, simply present CNNs with historical market data in the form of an image.

# Highlights

Claims: we should use methods that flexibly learn price patterns that are most predictive of future returns to forecast future returns.

The raw predictor data are **images** from which authors model the predictive association between images and future returns using a convolutional neural network (CNN).

Empirical Result shows: via using CNN we can automatically identify **context-independent** predictive patterns which can give more accurate return predictions, translate into more profitable investment strategies and are robust to variations.

# Brief Intro about Practice

1. Authors first embed 1D time-series data into 2D images depicting price and volumes.

2. They feed each training sample into CNN to estimate the probability of a **positive** subsequent return over short(5-day), medium(20-day), and long (60day) horizons.

3. They use CNN-based out-of-sample predictions as signals in several asset pricing analyses.

4. They also attempt to interpret the predictive patterns identified by the CNN.

# Replication task

Mainly focus on:

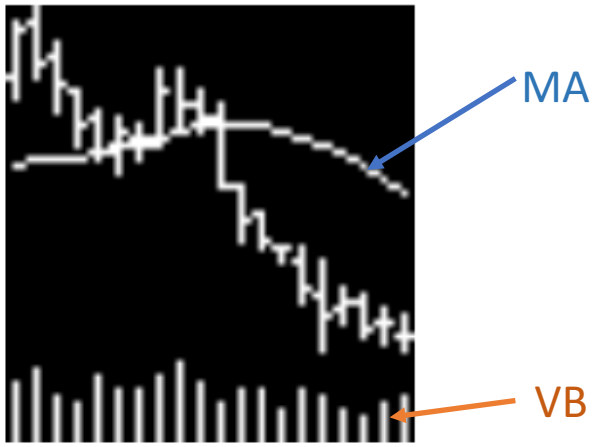
- Data Preparation
- Model Design
- Experiment Setting
- Ablation Study
- Interpretation

# Data Preparation

- The sample runs from 1993-2019(26 years) shows daily opening, high, low prices. The original paper constructs datasets consisting of three scales of horizons(5-day, 20-day, 60-day). Here we just collect the 20-day version. The total size of data is ~ 8.6G.
- We already transferred the OHLC charts into images following the same procedures. Current images have the same resolution (64 * 60) and added with moving average lines(MA) and volume bars(VB).


OHLC Chart

depict daily opening, high, low, and closing prices




processed

# Data Format

https://dachxiu.chicagobooth.edu/download/img_demo.html

'**Date**': The last day of the 20-day rolling window for the chart.
'**StockID**': CRSP PERMNO that identifies the stock.
'**MarketCap**': Market capitalization in dollar, recorded in thousands.
'**Ret_{t}d**': t=5,20,60, next t-day holding period return.
'**Ret_month**': Holding period return for the next month, from the current monthend to the next monthend.
'**EWMA_vol**': Exponentially weighted volatility (square of daily returns) with alpha as 0.05. One day delay is included.

| | Date | StockID | MarketCap | Ret_5d | Ret_20d | Ret_60d | Ret_month | EWMA_vol |
|---|---|---|---|---|---|---|---|---|
| 0 | 2017-01-31 | 10001 | 133078.0 | 4.370390e-07 | -0.000002 | -0.005954 | -0.000002 | 0.000450 |
| 1 | 2017-02-28 | 10001 | 133078.0 | 3.951997e-03 | 0.002795 | 0.009953 | 0.009953 | 0.000180 |
| 2 | 2017-03-31 | 10001 | 133604.0 | -7.874612e-03 | -0.015749 | 0.021723 | -0.015749 | 0.000064 |
| 3 | 2017-04-28 | 10001 | 131500.0 | 9.999880e-03 | 0.016001 | 0.038072 | 0.016001 | 0.000030 |
| 4 | 2017-05-31 | 10001 | 133604.0 | 4.370390e-07 | 0.021722 | NaN | 0.023703 | 0.000015 |

# Build Task Label

| | Date | StockID | MarketCap | Ret_5d | Ret_20d | Ret_60d | Ret_month | EWMA_vol |
|---|---|---|---|---|---|---|---|---|
| 0 | 2017-01-31 | 10001 | 133078.0 | 4.370390e-07 | -0.000002 | -0.005954 | -0.000002 | 0.000450 |
| 1 | 2017-02-28 | 10001 | 133078.0 | 3.951997e-03 | 0.002795 | 0.009953 | 0.009953 | 0.000180 |
| 2 | 2017-03-31 | 10001 | 133604.0 | -7.874612e-03 | -0.015749 | 0.021723 | -0.015749 | 0.000064 |
| 3 | 2017-04-28 | 10001 | 131500.0 | 9.999880e-03 | 0.016001 | 0.038072 | 0.016001 | 0.000030 |
| 4 | 2017-05-31 | 10001 | 133604.0 | 4.370390e-07 | 0.021722 | NaN | 0.023703 | 0.000015 |

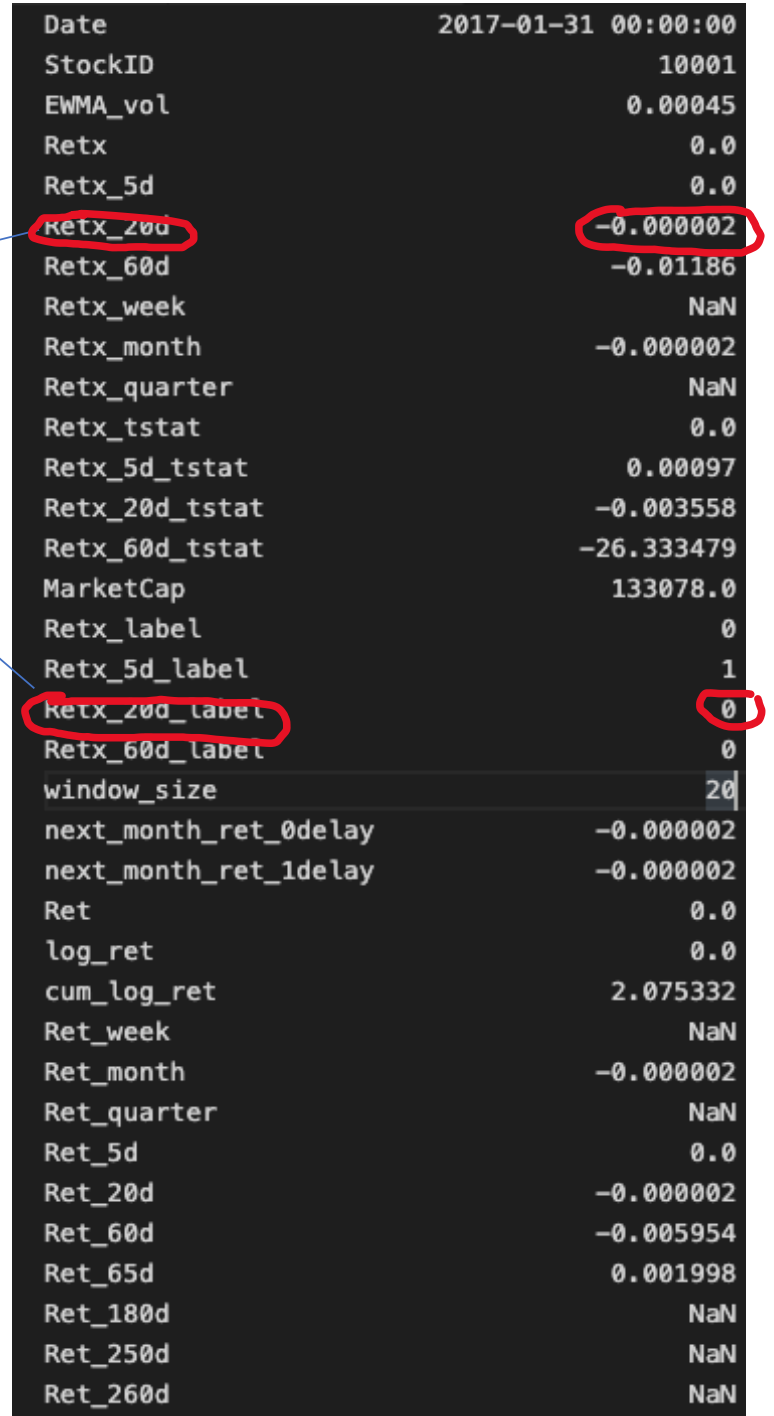
* Simplest implementation: build a classification task, predict 'up' or 'down' for the next_{t} days

Tip: How to convert to Classification Label:

1. If '**Ret_{t}d**' > 0, positive returns ('up'), marked as 1
2. Else, non-positive returns ('down'), marked as 0
3. Additionally, we should consider how to take existing 'NaN' values (maybe drop it?)

# Data: Label Format

Retx_20d: < 0
Retx_20d_label: 0

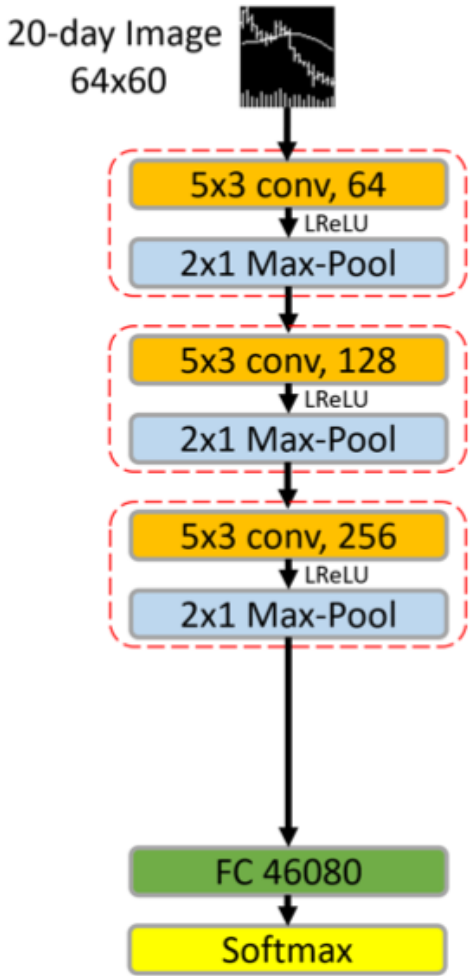


| | |
|---|---|
| Date | 2017-01-31 00:00:00 |
| StockID | 10001 |
| EWMA_vol | 0.00045 |
| Retx | 0.0 |
| Retx_5d | 0.0 |
| Retx_20d | -0.000002 |
| Retx_60d | -0.01186 |
| Retx_week | NaN |
| Retx_month | -0.000002 |
| Retx_quarter | NaN |
| Retx_tstat | 0.0 |
| Retx_5d_tstat | 0.00097 |
| Retx_20d_tstat | -0.003558 |
| Retx_60d_tstat | -26.333479 |
| MarketCap | 133078.0 |
| Retx_label | 0 |
| Retx_5d_label | 1 |
| Retx_20d_label | 0 |
| Retx_60d_label | 0 |
| window_size | 20 |
| next_month_ret_0delay | -0.000002 |
| next_month_ret_1delay | -0.000002 |
| Ret | 0.0 |
| log_ret | 0.0 |
| cum_log_ret | 2.075332 |
| Ret_week | NaN |
| Ret_month | -0.000002 |
| Ret_quarter | NaN |
| Ret_5d | 0.0 |
| Ret_20d | -0.000002 |
| Ret_60d | -0.005954 |
| Ret_65d | 0.001998 |
| Ret_180d | NaN |
| Ret_250d | NaN |
| Ret_260d | NaN |

- Images labels take value 1 for positive returns ('up') and 0 for non-positive returns ('down'). In addition, we use 2 to mark the NaN value.

# Model Design

- A core CNN building block consists of three modules:
  - Convolution layer
  - Activation layer
  - Pooling layer
- In the paper, for 20-day images, they build a baseline CNN architecture with 3 conv blocks and connected with a fully connected layer as a classifier head.




You should refer to the design of the conv block in the original paper, (refer to paper section 3)

- selection of the size of the convolution kernel,
- selection of the convolution method,
- selection of the pooling layer
- selection of the activation function,
- …

# Experiment Setting

- Data Split
  - Consider dividing the entire samples into training, validation and testing splits. (How, by time?)
  - In the original paper, they use the first seven-year sample (1993-1999) to train and validate the model, in which 70% of the sample are randomly selected for training and the remaining 30% for validation.
  - The remaining twenty years of data comprise the **out-of-sample test** dataset.

- Optimization Loss Design
  - You can simply treat the prediction analysis as a classification problem. Use Cross Entropy Loss

$$L_{CE}(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) log(1 - \hat{y})$$

- Evaluation Protocol Design
  - To measure the classification accuracy, a true positive (TP) or true negative (TN) occurs when a predicted “up” probability of greater than 50% coincides with a positive realized return and a probability less than 50% coincides with a negative return. False positives and negatives (FP and FN) are the complementary outcomes.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

    - For more evaluation metrics or methods, like **Sharpe Ratio**, please refer to the original paper.

- Training Process
  - **Over-fitting issue:** The author adopts several ways to combat with over-fitting and other tricks for efficient computation.
  - **Initialization:** They applied the *Xavier initialization* for weights in each layer, which guarantees faster convergence by scaling the initial weights. (How about other initialization choice?)
  - **What's more:** Other techniques like applying *dropout*, using *batch normalization* and *early stopping* also assists better performance.

❖We recommend referring to the training details mentioned in the section 3.3 when training the baseline model.

# Extensions

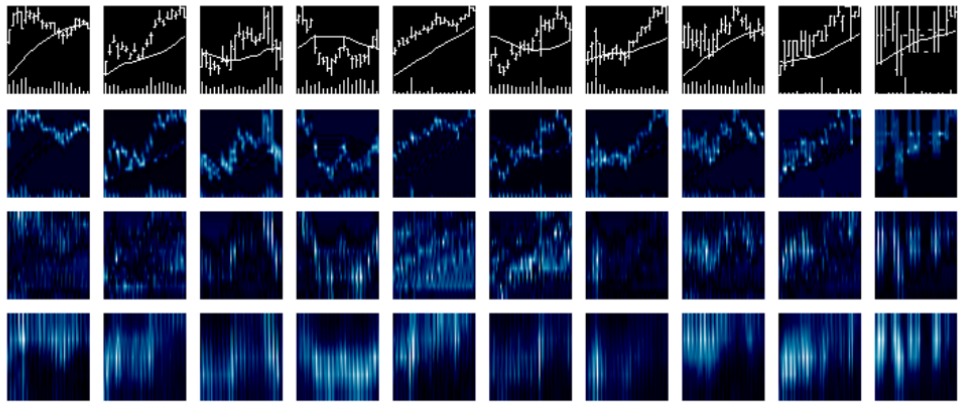
- Ablation studies
  - For example, you can perform the sensitivity analysis of the CNN prediction model to alternate choices in model architecture (e.g., varying the number of filters in each layer or varying the number of layers, like the paper shows in Table 18)

Table 18: Sensitivity to Model Structure and Estimation, I20R20

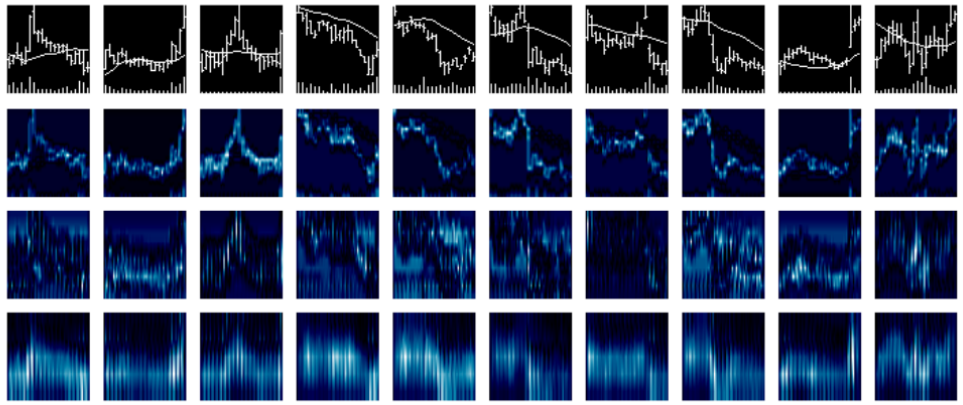
| | | Loss | | Acc. | | Correlation | | Sharpe Ratio | |
|---|---|---|---|---|---|---|---|---|---|
| | | V | T | V | T | Spearman | Pearson | EW | VW |
| Baseline | | **0.688** | **0.692** | **0.540** | **0.525** | **0.052** | **0.032** | **2.18** | **0.56** |
| Filters (64) | 32 | 0.688 | 0.691 | 0.541 | 0.526 | 0.053 | 0.032 | 1.91 | 0.43 |
| | 128 | 0.692 | 0.692 | 0.535 | 0.527 | 0.050 | 0.030 | 2.01 | 0.45 |
| Layers (3) | 2 | 0.688 | 0.692 | 0.540 | 0.525 | 0.047 | 0.028 | 1.61 | 0.19 |
| | 4 | 0.689 | 0.692 | 0.538 | 0.524 | 0.051 | 0.031 | 2.00 | 0.38 |
| Dropout (0.50) | 0.00 | 0.698 | 0.695 | 0.532 | 0.521 | 0.044 | 0.027 | 2.32 | 0.54 |
| | 0.25 | 0.691 | 0.693 | 0.536 | 0.525 | 0.050 | 0.030 | 2.11 | 0.56 |
| | 0.75 | 0.691 | 0.692 | 0.530 | 0.525 | 0.041 | 0.024 | 1.30 | 0.05 |
| BN (yes) | no | 0.685 | 0.691 | 0.549 | 0.528 | 0.059 | 0.037 | 2.44 | 0.58 |
| Xavier (yes) | no | 0.688 | 0.692 | 0.540 | 0.526 | 0.053 | 0.033 | 2.10 | 0.39 |
| Activation (LReLU) | ReLU | 0.689 | 0.693 | 0.538 | 0.520 | 0.052 | 0.031 | 1.71 | 0.35 |
| Max Pool Size (2×1) | 2×2 | 0.687 | 0.692 | 0.545 | 0.526 | 0.056 | 0.034 | 2.09 | 0.41 |
| Filter Size (5×3) | 3×3 | 0.689 | 0.693 | 0.538 | 0.522 | 0.050 | 0.028 | 1.39 | 0.28 |
| | 7×3 | 0.688 | 0.691 | 0.541 | 0.527 | 0.055 | 0.033 | 2.01 | 0.41 |
| Dilation/Stride (2,1)/(3,1) | (2,1)/(1,1) | 0.689 | 0.692 | 0.537 | 0.526 | 0.052 | 0.033 | 2.04 | 0.53 |
| | (1,1)/(3,1) | 0.689 | 0.692 | 0.538 | 0.526 | 0.049 | 0.030 | 1.66 | 0.17 |
| | (1,1)/(1,1) | 0.687 | 0.692 | 0.545 | 0.527 | 0.055 | 0.035 | 2.09 | 0.52 |

- Exploring of the **interpretability** of the CNN model
  - Using a visualization method (Grad-CAM) to understand how different image examples activate the regions of the CNN to trigger 'up' or 'down' return predictions.

Figure 12: I20R20 Grad-CAM for 20 Images from 2019



(a) Images Receiving “Up” Classification



(b) Images Receiving “Down” Classification

Note: Brighter regions of the heatmap correspond to regions with the higher activation. For each panel, the first row is the original images followed by the grad-CAM for each layer in the CNN.

- Association with Other Predictors
  - How unique are CNN forecasts compared to standard characteristics in the literature?
  - Compared with *recent price trends* (MOM, STR, WSTR), *risk* (beta and volatility), *liquidity* (bid-ask spread, dollar volume) etc.

Table 6: CNN Predictions and Standard Stock Characteristics

| | 5D5P | 20D5P | 60D5P |
|---|---|---|---|
| MOM | -0.10*** | 0.01** | 0.40*** |
| STR | -0.09*** | 0.27*** | 0.24*** |
| Lag Weekly Return | -0.85*** | -1.00*** | -0.89*** |
| TREND | 0.51*** | 0.46*** | 0.23*** |
| Beta | 0.11*** | 0.15*** | 0.22*** |
| Volatility | -0.09*** | -0.20*** | -0.24*** |
| 52WH | -0.05*** | -0.03*** | -0.09*** |
| Bid-Ask | 0.10*** | -0.11*** | -0.08*** |
| Dollar Volume | 0.20*** | 0.16*** | -1.22*** |
| Zero Trade | -0.09*** | 0.00 | 0.32*** |
| Price Delay | -0.01*** | -0.01** | 0.00 |
| Size | 0.21*** | 0.40*** | 0.44*** |
| Illiquidity | 0.08*** | 0.19*** | -1.43*** |
| McFadden $R^2$ | 8.20 | 8.56 | 9.78 |

Note: The table reports slope coefficients and $R^2$ from panel logistic regressions of CNN model forecasts on stock characteristics. Panel regressions are estimated during the test sample using CNN models estimated in the training sample. Coefficients accompanied by ***,**,* are significant at the 1%, 5% and 10% significance level, respectively, using Newey-West standard errors.

# What's more

- We encourage you not limited to simple binary classification tasks, since the annotation files we provided consist of more meaningful attributes, containing both categorical and numerical values.
  - For example, you can use the same 20-day horizon images to train your model to predict the return trend of different subsequent t-days even the detailed return values. (t can be 5, 20 even larger).