CNN-based Stock Price Trend Image Recognition

A Machine Learning Framework for Financial Time Series

Sun Tiantian, Liang Jing, Zhang Xinyi, Zhong Qi, Liu Sihui

Course Final Presentation

December 3, 2025

Team Members and Contributions

- **Sun Tiantian**: Designed the training pipeline and the implementation of optimization strategies.
- **Liang Jing**: Conducted model interpretability analysis using Grad-CAM to visualize predictive patterns.
- **Zhang Xinyi**: Performed robustness testing through systematic hyperparameter sensitivity analysis.
- **Zhong Qi**: Evaluated model performance and extended binary prediction to return value prediction.
- **Liu Sihui**: Wrote and organized the technical report.

Outline

Introduction & Motivation

The Problem

- Traditional asset pricing relies on hypothesis testing and ad hoc predictors.
- Time-series methods struggle to capture complex relational attributes.

Our Approach

- **Transformation**: Convert historical price/volume data into 2D images.
- **Model**: Use Convolutional Neural Networks (CNN) to uncover latent patterns.
- **Goal**: Standardize cross-asset scales and capture context-independent patterns.

- Validated against traditional indicators (Momentum, Reversal).
- Interpretability via Grad-CAM.

Pipeline Design: OHLC to Images

- **Data Source**: U.S. stocks (1993-2019).
- **Image Generation**:
  - 3-pixel width per trading day.
  - Normalized closing price.
  - Integration of Moving Averages and Volume bars.
- **Output**: Standardized sparse image matrices (Black background, white lines).

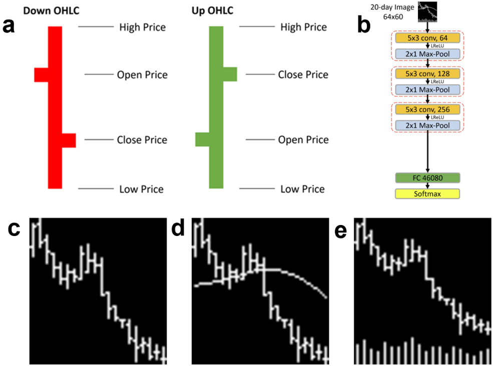


Figure: Data preprocessing pipeline and OHLC chart representation.

CNN Architecture Training Strategy

Model Architecture:

- Input: 64×60 images.
- Layers: Conv Blocks $\rightarrow$ Leaky ReLU $\rightarrow$ Max-Pool.
- Adaptive depth based on horizon (5, 20, 60 days).

Training Strategy:

- **Loss**: Cross-Entropy
  $L(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$.
- **Optimizer**: Adam ($lr = 10^{-5}$).
- **Regularization**: Dropout (50%), Batch Norm, Early Stopping.
- **Split**: 1993-1999 (Train/Val), 2000-2019 (Test).

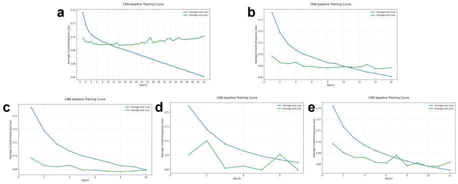


Figure: Training curves.

Classification Accuracy

We replicated the original paper's results and extended to 5-day predictions.

Task	I20R5		I20R20		I20R60	
	Acc.	Corr.	Acc.	Corr.	Acc.	Corr.
Original Paper	–	–	53.3%	3.4%	53.2%	2.4%
Our Work	**53.1%**	**5.3%**	**52.5%**	**3.0%**	**54.2%**	**2.1%**

Table: Performance Comparison

- Consistent performance with state-of-the-art.
- Strong correlation in short-term (5-day) tasks.

Portfolio Performance (Backtesting)

- **Strategy**: Sort stocks into Deciles based on predicted probability.
- **I20R20 (20-day)**:
  - Equal Weight (EW) Sharpe Ratio: **1.73**.
  - Monotonic return increase from Decile 1 to 10.
- **Limitations**:
  - Value Weight (VW) performance drops in long-term (60-day) models.
  - High turnover for monthly rebalancing.

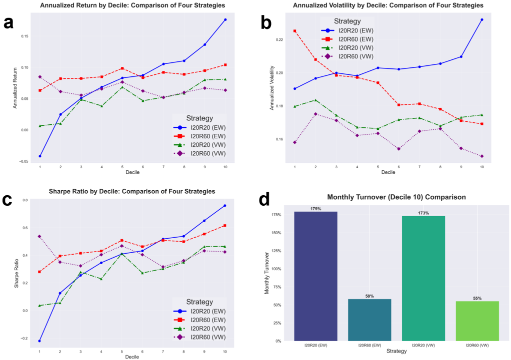


Figure: Portfolio Performance (Returns, Volatility, Sharpe, Turnover).

Sensitivity Analysis

"Macro robustness and micro precision"

- **Method**: Systematic perturbation of hyperparameters (Filters, Layers, Dropout, Kernel Size).
- **Results**:
  - Accuracy remains stable (51% - 55%).
  - Dropout is critical (Removal spikes validation loss).
  - Tolerant to micro-structural changes (Stride/Dilation).

Modification	Acc. (Test)	Corr. (Pearson)	Sharpe (EW)
Baseline	**0.530**	**0.044**	**1.73**
Filters (32)	0.523	0.037	1.31
Layers (2)	0.520	0.031	1.40
Dropout (0.00)	0.512	0.025	1.73

Table: Subset of Sensitivity Analysis Results

We used **Grad-CAM** to visualize decision patterns:

- **Shallow Layers**: Focus on local price extremes (candles, reversals).
- **Deep Layers**: Integrate trends and volatility regimes.
- **Pattern**:
    - "Up" predictions → Focus on higher relative price regions.
    - "Down" predictions → Focus on lower relative regions.

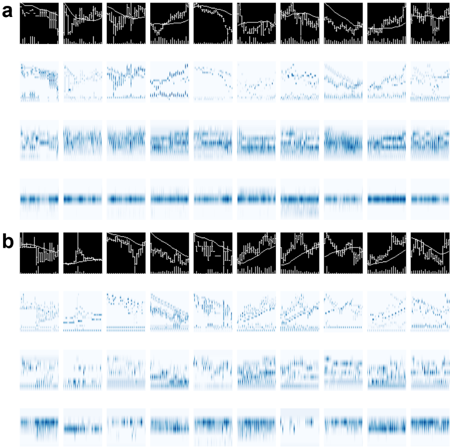


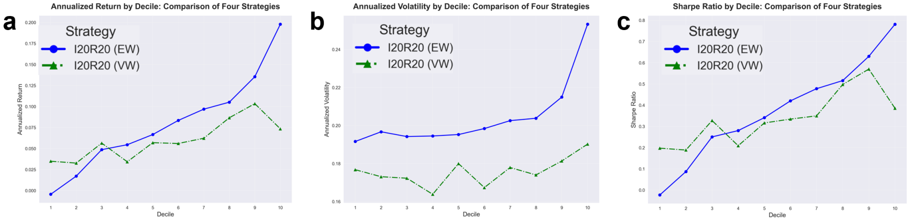
Figure: Grad-CAM Heatmaps (Up vs Down).

Extension: Return Value Prediction

Task Shift: Binary Classification → Regression (MSE Loss).

- **Objective**: Directly predict detailed return values.
- **Best Model**: Baseline with Early Stopping (Epoch 0).

Model	MSE	Corr.
Baseline (Best Epoch)	2.91%	2.7%
Baseline (Latest Epoch)	2.92%	2.6%



Conclusion

Summary

- Established a reproducible **CNN-based framework** for trend recognition.
- Demonstrated strong short-term predictive power (Sharpe 1.73).
- Verified robustness and interpretability via visualization.

Future Work

- Improve long-horizon performance for large-cap stocks.
- Implement Transfer Learning.
- Optimize regression architecture for precise return forecasting.

`https://github.com/Aaaaaant97/CNN-based-Stock-Price-Trend-Image-Recognition`

Thank You!

Q & A