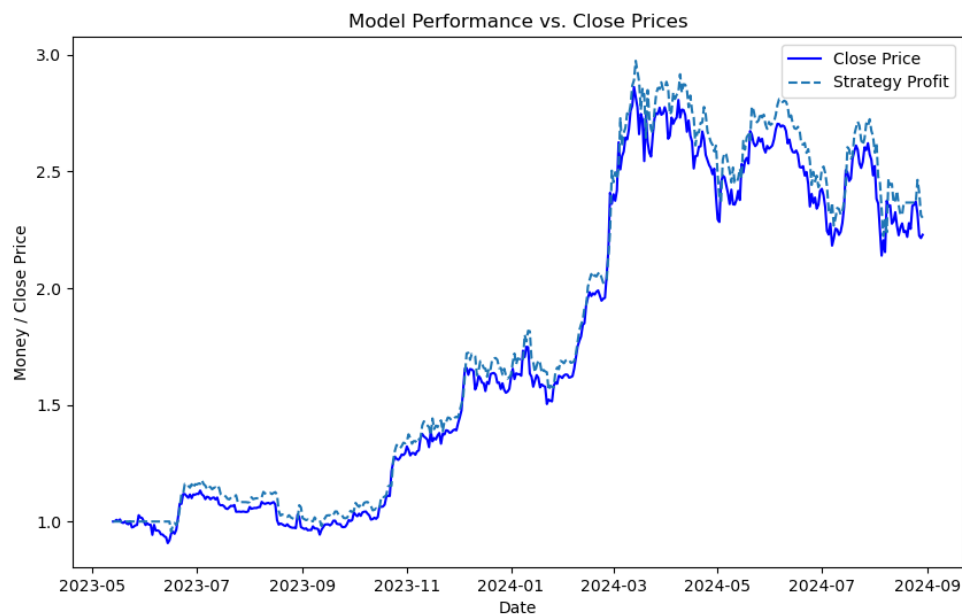


Report

- CNN (ConvMixer) 결과



```
○ (finance) PS C:\Users\kyuwo\Desktop\ViT_for_finance> python eval.py
Model: ./model_save/CNN.pt
{'USD': 0, 'KRW-BTC': 0.0002857183678134644, 'v_USD': 23020.04317636301, 'v_KRW-BTC': 0.0002857183678134644, 'buy_count': 1, 'hold_count': 459, 'sell_count': 0}
Profit percentage: 0.32550107940907524
Cumulative return: 1.3020043176362974
Volatility: 0.471391081436446
Sharpe ratio: 2.7620469900889217
```

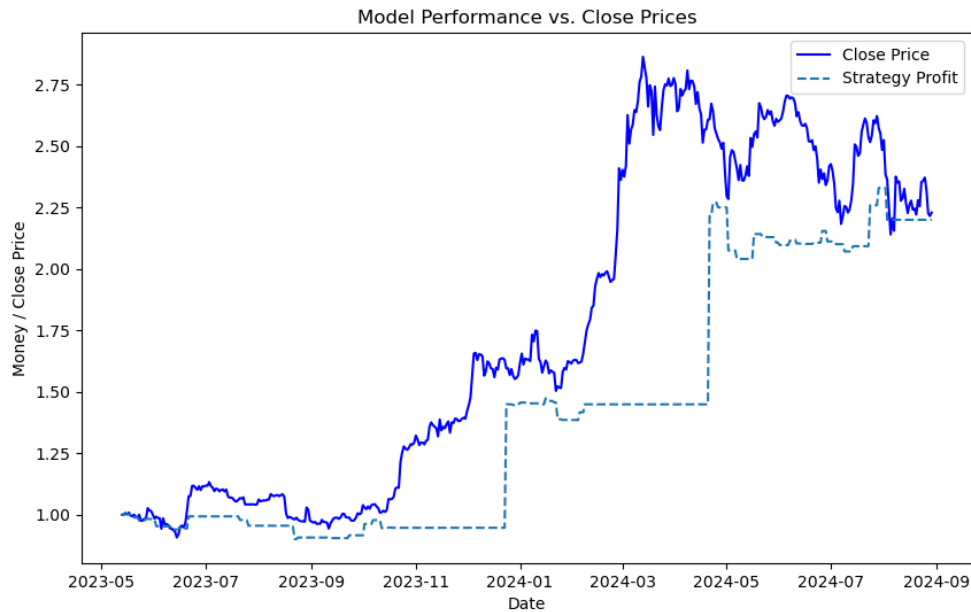
- description:

‘Recently proposed ConvMixer architecture has claimed that the main reason for the Vision Transformer’s success might be the use of patches in the input representation, rather than the transformer architecture.’이라는 문장에 따라 이미지를 patch로 나누는 것에 집중하여 구현했다. (stride를 kernel size와 같게 설정하면 patch로 나눈 것과 동일한 효과를 낼 수 있다.)

실제로 kernel size에 따라 sharpe ratio가 달라지는 것을 확인했는데 논문에서 제시한 것과는 다르게 kernel size가 5일 때 가장 높은 sharpe ratio가 관찰되었

다. kernel size가 크면 클수록 멀리 떨어진 공간 정보들이 더 잘 반영된다는 논문의 statement와는 상반되는 결과였다.

- ViT 결과



```
(finance) PS C:\Users\kyuwo\Desktop\ViT_for_finance> python eval.py
Model: ./model_save/ViT.pt
{'USD': 0, 'KRW-BTC': 0.0002715113193098006, 'v_USD': 21993.502909371087, 'v_KRW-BTC': 0.0002715113193098006, 'buy_count': 57, 'hold_count': 0, 'sell_count': 56}
Profit percentage: 0.29983757273427714
Cumulative return: 1.1995702479619057
Volatility: 0.7723886490096805
Sharpe ratio: 1.5530656095217568
```

- description:

논문에 제시한 것과 거의 동일한 구조로 구현했다. 이미지를 patch로 쪼갬 후 → CLS token과 concatenate하고 → positional embedding 값을 더해주었다. 이렇게 생성된 embedded patch들을 layer norm → (Multi-head self-attention + MLP) → residual connection이 여러 개 stack된 encoder에 대입한다. 마지막으로 single hidden layer MLP를 통과시키면 끝! 근데 sharpe ratio가 논문에 나온 것보다 훨씬 낮다,,,,, $\pi\pi$ dim이랑 patch size를 더 조정해봐야겠다. (어림짐작으로는 patch size를 현재보다 작게 설정해 더 세밀하게 볼 수 있다면 sharpe ratio가 개선될 것 같다.)