

Financial Trading with Deep Convolutional Neural Networks: Apple Stock

Gerend Christopher¹, Kevin Sean Hans Lopulalan²

^{1,2}Institut Teknologi Bandung

Email: ¹10120084@mahasiswa.itb.ac.id, ²10120074@mahasiswa.itb.ac.id

Abstrak

Pemanfaatan teknologi komputasi pada *financial trading* semakin berkembang dan menjadi populer. Deep Learning yang berkembang diminati dalam pengolahan citra (*image processing*) menggunakan Convolutional Neural Network (CNN). Dalam studi ini, model *algorithmic trading* untuk data saham Apple dikembangkan dengan menggunakan CNN dua dimensi. Data *time series* diubah menjadi gambar 2 dimensi dengan memanfaatkan 14 indikator teknikal. Setiap indikator menghasilkan data untuk periode 16 hari, sehingga terbentuklah citra dua dimensi berukuran 16 x 16 piksel. Setiap gambar juga diberi label *Buy*, *Sell*, atau *Hold* bergantung pada pola pergerakan saham.

Kata Kunci: *Deep Learning, Algorithmic Trading, Convolutional Neural Networks, Technical Analysis, Stock Apple*

I. Pendahuluan

Dalam era globalisasi dan kompleksitas pasar keuangan saat ini, pemahaman yang mendalam tentang saham, perdagangan keuangan, dan indikator teknis menjadi semakin penting bagi para akademisi dan praktisi di bidang keuangan. Pasar saham adalah salah satu instrumen keuangan yang paling diperdagangkan di seluruh dunia, menjadi fokus utama bagi investor, analis, dan peneliti. Untuk mengelola risiko dan mencapai keuntungan yang optimal, para pelaku pasar bergantung pada berbagai alat dan strategi, termasuk penggunaan indikator teknis.

Indikator teknis merupakan alat analisis yang digunakan untuk membantu para trader dalam membuat keputusan jual beli. Indikator teknis beroperasi dengan menganalisis data historis harga saham dan volume perdagangan untuk mengidentifikasi pola, tren, dan momentum pasar. Beberapa indikator teknis yang umum digunakan meliputi Relative Strength Index (RSI), Moving Averages (MA), dan Money Flow Index.

Namun, dengan semakin kompleksnya pasar dan ketersediaan data yang besar, para pelaku pasar mulai beralih ke pendekatan yang lebih maju seperti *algorithmic trading*. *Algorithmic trading* menggunakan algoritma komputer untuk membuat keputusan perdagangan secara otomatis dengan aturan yang telah diprogram sebelumnya. Salah satu pendekatan yang semakin populer dalam *algorithmic trading* adalah *deep learning*.

Deep Learning merupakan cabang dari *machine learning* yang menggunakan jaringan saraf tiruan yang terdiri dari beberapa lapisan (layers) untuk memahami dan menganalisis data yang kompleks. Dalam konteks jual beli saham, *Deep Learning* dapat digunakan untuk mengidentifikasi pola-pola yang lebih kompleks dan abstrak dalam data pasar, yang mungkin sulit atau bahkan tidak dapat dikenali oleh manusia atau metode analisis tradisional.

Dengan menggunakan *Deep Learning*, para peneliti dan praktisi dapat mengembangkan model prediktif yang lebih canggih untuk memperkirakan pergerakan harga saham di masa depan atau mengidentifikasi peluang perdagangan yang menguntungkan. Pendekatan ini mengintegrasikan analisis teknis dengan kecerdasan buatan untuk menciptakan strategi perdagangan yang lebih efisien dan adaptif.

Pengembangan algoritma *Deep Learning*, seperti *Convolutional Neural Network*, diterapkan pada proses jual beli saham. Indikator-indikator teknis saham dapat diterapkan pada *Convolutional Neural Network* sehingga diharapkan proses jual beli saham di pasar nyata menggunakan *algorithmic trading* menunjukkan kinerja dan keefektifan.

II. Metodologi

II.1 Data

Pada penelitian ini, data yang digunakan adalah data saham Apple. Data saham diperoleh dari [Yahoo! Finance](#). Data *time series* dipilih dari tanggal 1 Januari 2000 hingga 31 Desember 2023. Gambar 1 menunjukkan cuplikan harga saham tersebut.

	Date	Open	High	Low	Close	Adj Close	Volume
0	2000-01-03	0.936384	1.004464	0.907924	0.999442	0.846127	535796800
1	2000-01-04	0.966518	0.987723	0.903460	0.915179	0.774790	512377600
2	2000-01-05	0.926339	0.987165	0.919643	0.928571	0.786128	778321600
3	2000-01-06	0.947545	0.955357	0.848214	0.848214	0.718098	767972800
4	2000-01-07	0.861607	0.901786	0.852679	0.888393	0.752113	460734400
...
6032	2023-12-22	195.179993	195.410004	192.970001	193.600006	193.353287	37122800
6033	2023-12-26	193.610001	193.889999	192.830002	193.050003	192.803986	28919300
6034	2023-12-27	192.490005	193.500000	191.089996	193.149994	192.903839	48087700
6035	2023-12-28	194.139999	194.660004	193.169998	193.580002	193.333298	34049900
6036	2023-12-29	193.899994	194.399994	191.729996	192.529999	192.284637	42628800

Gambar 1 Harga Saham AAPL

Data ini memiliki 7 variabel yaitu *Date*, *Open*, *High*, *Low*, *Close*, *Adj Close*, *Volume*; serta 6037 baris yang merupakan data per hari dari 1 Januari 2000 hingga 31 Desember 2023.

Setelah dilakukan *preprocessing* untuk memperoleh *technical indicators* yang siap digunakan. Kemudian, data dibagi menjadi 60% data *training*, 20% data validasi, dan 20% data *testing*.

II.2 Technical Indicators

Ada 14 *technical indicators* yang digunakan dalam penelitian ini, yaitu

1. Relative Strength Index (RSI)

Relative Strength Index (RSI) adalah indikator analisis teknis jenis osilator yang menunjukkan kekuatan dan kelemahan historis harga saham. Nilai RSI berada pada antara 0 dan 100. Umumnya, nilai RSI di atas 70, saham dianggap berada di wilayah *overbought* dan jika nilai RSI di bawah 30, saham dianggap berada di wilayah *oversold*. Persamaan RSI adalah sebagai berikut

$$RSI_{step\ one} = 100 - \frac{100}{1 + \frac{average\ gain}{average\ loss}}$$

$$RSI_{step\ two} = 100$$

$$- \frac{100}{1 + \frac{(Previous\ Average\ Gain * period) + current\ gain}{(Previous\ Average\ Gain * period) + current\ loss}}$$

2. William %R

William %R adalah indikator analisis teknis berbasis momentum yang menentukan wilayah *overbought* dan *oversold* untuk harga saham. Nilai William %R berada pada antara -100 and 0. Umumnya, jika nilai William %R di

bawah -80, saham dianggap berada di wilayah *oversold*. Jika nilai William %R di atas -20, saham dianggap berada di wilayah *overbought*. Persamaan William %R adalah sebagai berikut

$$R = \frac{\max(\text{high}) - \text{close}}{\max(\text{high}) - \min(\text{low})} * (-100)$$

3. Money Flow Index (MFI)

Money Flow Index adalah indikator teknis yang menggunakan harga dan volume data untuk menentukan *overbought* atau *oversold* pada aset. MFI juga dapat digunakan untuk melihat divergensi yang memperingatkan adanya perubahan tren harga. Nilai MFI berada pada antara 0 dan 100. Persamaan Money Flow Index adalah sebagai berikut

$$\text{Money Flow Index} = 100 - \frac{100}{1 + \text{Money Flow Ratio}}$$

$$\text{Money Flow Ratio} = \frac{14 \text{ Period Positive Money Flow}}{14 \text{ Period Negative Money Flow}}$$

$$\text{Raw Money Flow} = \text{Typical} * \text{Volume}$$

$$\text{Typical Price} = \frac{\text{High} + \text{Low} + \text{Close}}{3}$$

4. Rate of Change (ROC)

Rate of Change adalah indikator teknis yang mengilustrasikan kecepatan perubahan harga dalam satu periode waktu

$$\text{ROC} = \frac{(\text{Latest Close} - \text{Previous Close})}{\text{Previous Close}} * 100$$

5. Chaikin Money Flow (CMF)

Chaikin Money Flow adalah indikator analisis teknis yang mengukur volume aliran uang dalam satu waktu periode. CMF berada pada antara -1 dan 1. Jika nilai CMF mendekati -1, tekanan penjualan semakin tinggi dan jika nilai CMF mendekati 1, tekanan pembelian semakin tinggi.

$$\text{Multiplier} = \frac{((\text{Close} - \text{Low}) - (\text{High} - \text{Close}))}{(\text{High} - \text{Low})}$$

$$\text{Money Flow Volume (MFV)} = \text{Volume} * \text{Multiplier}$$

$$21 \text{ Period CMF} = \frac{21 \text{ Period Sum of MFV}}{21 \text{ Period Sum of Volume}}$$

6. Chande Momentum Oscillator (CMO)

Chande Momentum Oscillator (CMO) adalah indikator teknis momentum yang serupa dengan RSI. CMO berada pada di antara -100 dan 100. Jika nilai CMO di atas 50, harga saham diinterpretrasikan pada daerah *overbought*. Jika nilai CMO di bawah -50, harga saham diinterpretrasikan pada daerah *oversold*. S_u adalah jumlah momentum pada hari naik dan S_d adalah jumlah momentum pada hari turun. Persamaan CMO adalah sebagai berikut

$$CMO = 100 * \frac{S_u - S_d}{S_u + S_d}$$

7. Simple Moving Average (SMA)

Simple Moving Average adalah rata-rata aritmatika dari harga pada suatu periode tertentu. Persamaan SMA adalah sebagai berikut

$$SMA(M, n) = \frac{1}{n} \sum_{k=a+1}^{a+n} M(k)$$

8. Exponential Moving Average (EMA)

Exponential Moving Average adalah rata-rata bergerak dari harga pada suatu periode tertentu yang lebih menekankan pada beberapa hari terakhir. Dengan kata lain, bobot pada beberapa hari terakhir lebih besar daripada hari-hari awal. Persamaan EMA adalah sebagai berikut

$$EMA(M, t, \tau) = \tau EMA(M, t - 1, \tau) + (1 - \tau)M(t)$$

9. Weighted Moving Average (WMA)

Weighted Moving Average (WMA) adalah rata-rata bergerak dari harga pada suatu periode tertentu dengan bobot tertentu serupa dengan EMA. Jika EMA memiliki bobot yang menurun secara eksponensial, pada WMA bobot menurun secara linier. Persamaan WMA adalah sebagai berikut

$$WMA(M, n) = \frac{\text{Sum of Weighted Averages}}{\text{Sum of Weight}}$$

10. Hull Moving Average (HMA)

Hull Moving Average adalah rata-rata bergerak yang mengurangi lag yang ada pada SMA, EMA, dan WMA. Persamaan Hull Moving Average adalah sebagai berikut

$$HMA(M, n) = WMA\left(\left(2 * WMA\left(\frac{n}{2}\right) - WMA(n)\right), \sqrt{n}\right)$$

11. Triple Exponential Average (TRIX)

Triple Exponential Average adalah EMA yang digunakan untuk menyaring pergerakan harga yang tidak signifikan atau tidak penting. Persamaan Triple Exponential Average adalah sebagai berikut

$$TRIX = \frac{EMA3(i) - EMA3(i - 1)}{EMA3(i - 1)}$$

$$EMA3(i) = EMA(EMA2, N, i)$$

$$EMA2(i) = EMA(EMA1, N, i)$$

$$EMA1(i) = EMA(Price, N, i)$$

12. Commodity Channel Index (CCI)

Commodity Channel Index adalah indikator yang membandingkan harga sekarang dan rata-rata harga dalam suatu periode waktu tertentu. Umumnya, nilai CCI berada pada antara -100 dan 100. Persamaan CCI adalah sebagai berikut

$$CCI = \frac{Typical\ Price - 20\ Period\ SMA\ of\ TP}{.015 * Mean\ Deviation}$$

$$Typical\ Price\ (TP) = \frac{High + Low + Close}{3}$$

13. Detrended Price Oscillator (DPO)

Detrended Price Oscillator adalah indikator teknis yang digunakan untuk menghilangkan tren harga dalam upaya memperkirakan panjang siklus harga dari puncak ke puncak atau dari lembah ke lembah. Persamaan DPO adalah sebagai berikut

$$DPO = Close\left(\frac{Period}{2} + 1\right) - SMA(Period)$$

14. Directional Movement Indicator (DMI)

Directional Movement Indicator adalah indikator teknis yang menunjukkan kekuatan tren dan arahnya. DMI terdiri atas tiga indikator: Average Directional Index (ADX), Plus Directional Indicator (+DI), dan Minus Directional Indicator (-DI). Nilai DM berada pada antara 0 dan 100. Persamaan DMI adalah sebagai berikut

$$+DI = 100 * EMA \left(\frac{+DMI}{Average True Range} \right)$$

$$-DI = 100 * EMA \left(\frac{-DMI}{Average True Range} \right)$$

$$ADX = 100 * EMA \left(Absolute Value of \left(\frac{+DI - -DI}{+DI + -DI} \right) \right)$$

II.3 Data Labelling

Data harga saham harian diberi label secara manual *Buy*, *Sell*, atau *Hold* berdasarkan titik puncak dan titik lembah pada *sliding window*. *Buy* dipilih jika harga terendah dan *Sell* dipilih jika harga tertinggi serta sisanya *Hold*. Algoritma 1 menunjukkan proses pelabelan data.

Algoritma 1. Pelabelan *Buy*, *Sell*, atau *Hold*

Input: Data, kolom harga, dan *windowSize*

Output: label

Langkah-langkah:

```
1. While (counterRow < numberOfDaysInData)
2.     counterRow ++
3.     if (counterRow > windowSize)
4.         windowBeginIndex = counterRow - windowSize
5.         windowEndIndex = windowBeginIndex + windowSize - 1
6.         windowMiddleIndex = (windowBeginIndex + windowEndIndex) / 2
7.         for (i = windowBeginIndex; i <= windowEndIndex; i++)
8.             number = closePriceList.get(i)
9.             if (number < min)
10.                 min = number
11.                 minIndex = closePriceList.indexOf(min)
12.             if (number > max)
13.                 max = number
14.                 maxIndex = closePriceList.indexOf(max)
15.             if (maxIndex == windowMiddleIndex)
16.                 label = "Sell"
17.             elif (minIndex == windowMiddleIndex)
18.                 label = "Buy"
19.             else
20.                 label = "Hold"
```

II.3 Model

Setelah ditentukan label, maka data saham telah menjadi data dengan variabel respon dan prediktor yang dapat dimodelkan dengan *machine learning*. Dalam kasus ini, tipe pembelajaran mesin yang akan dimodelkan yaitu *supervised learning*. *Supervised learning* melatih model untuk menggunakan data yang telah diberi label. Proses ini melibatkan dua fase utama, yaitu *training* dan *predicting*. Pada proses *training*, model dilatih untuk memetakan hubungan antara masukan dan keluaran sehingga dapat membuat prediksi yang akurat ketika diberikan data baru yang belum pernah dilihat sebelumnya. Pada tahap *predicting*, model yang telah dilatih digunakan untuk membuat prediksi pada data yang belum pernah dipelajari. Pembuatan model untuk saham ini termasuk juga *supervised learning* untuk regresi, yang mana pada akhirnya model harus memprediksi nilai kontinu dari saham.

Lebih lanjut, pada tugas ini akan digunakan *deep learning*, yaitu bagian dari *machine learning* yang menggunakan banyak *layer* untuk memahami data yang kompleks. Kemudian, arsitektur *deep learning* yang digunakan yaitu *Convolutional Neural Network* (CNN). CNN mengaplikasikan pembagian data menjadi grid atau matriks dalam *deep learning*. Contohnya seperti data *time series* yang dapat dianggap sebagai 1D *grid* pada interval waktu tertentu dan data gambar 2D *grid* pada *pixels*. Secara sederhana, *Convolutional Neural Network* adalah *Neural Network* yang menggunakan konvolusi sebagai ganti perkalian matriks setidaknya pada satu *layer* (Goodfellow, Bengio, & Courville, 2016).

Adapun struktur dan komponen utama dalam CNN adalah sebagai berikut.

1. *Convolutional Layers*

Convolutional layers terdiri dari:

- a. *Filter* atau *kernel*

Filter merupakan matriks kecil dengan ukuran tetap, untuk menggeser atau konvolusi seluruh data untuk mengekstraksi fitur pada data.

- b. *Stride*

Stride merupakan jumlah piksel yang digeser filter pada setiap langkah.

- c. *Padding*

Padding merupakan piksel tambahan di tepi data untuk mengontrol ukuran keluaran.

2. *Activation Function*

Fungsi aktivasi yang akan digunakan yaitu ReLU untuk *hidden layers* dan softmax untuk *output layer*. Fungsi ReLU dan softmax secara berturut-turut didefinisikan pada persamaan (1) dan (2).

$$ReLU(x) = \max(0, x) \quad \dots(1)$$

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad \dots(2)$$

3. *Pooling Layers*

Pada CNN, lapisan *pooling* merupakan komponen untuk mengurangi dimensionalitas *feature map* yang dihasilkan dari tahap *filtering* lapisan konvolusi. Tujuan dilakukannya adalah untuk mengurangi jumlah parameter dan komputasi dalam jaringan serta mempertahankan informasi penting dalam *feature map*. Adapun tipe *pooling* yang dapat digunakan pada tugas ini yaitu *max pooling*. *Max pooling* mempertahankan nilai maksimum untuk setiap blok yang digunakan dalam *pooling layers*.

4. *Fully Connected Layers*

Deep learning memiliki bagian yang disebut *neuron* untuk setiap *layer*-nya. Arsitektur *deep learning* sangat bervariasi sesuai dengan kebutuhan. Salah satu tipe arsitektur yang sering dimanfaatkan dalam CNN yaitu *fully connected layers*. Arsitektur ini menghubungkan semua *neuron* lapisan sebelumnya ke *neuron* lapisan berikutnya. Hal ini dilakukan untuk memungkinkan pembelajaran pola yang sangat kompleks dalam data, yang kemudian diberikan fungsi aktivasi sesuai dengan kebutuhan dari data. Tugas dari *fully connected layer* ini adalah melakukan *high-level reasoning*, seperti klasifikasi.

5. *Dropout Layers*

Kemudian, untuk mencegah fenomena *overfitting*, yaitu model terlalu spesifik pada data, CNN menggunakan *dropout layer*. Teknik *dropout layer* merupakan teknik regularisasi di mana *neuron* secara acak dipilih untuk diabaikan dalam proses training, yang artinya kontribusi neuron-neuron tersebut terhadap pembentukan model untuk sementara dapat dihapus dalam *forward propagation* dan bobot-bobot

yang telah diperbaharui tidak akan diaplikasikan pada mereka saat *backward propagation*.

II.6 Evaluasi Model

Dalam proses pembelajaran mesin, tahap *training* merupakan tahap di mana mesin mempelajari data beserta labelnya untuk membentuk suatu model untuk mempelajari jaringan fundamental yang membentuk suatu data. Kemudian tahap *testing* merupakan tahap di mana kualitas model diuji, apakah model tersebut sudah benar-benar paham fundamental dari data. Dalam penerapannya, model perlu dievaluasi dengan beberapa metrik khusus sebagai berikut.

1. Akurasi

Akurasi merupakan persentase rasio untuk mengukur seberapa sering model dapat membuat prediksi yang tepat sesuai dengan label. Semakin tinggi akurasi, semakin baik model mengenali data.

2. Matriks konfusi

Matriks konfusi memberikan gambaran tentang kinerja model yang menunjukkan berapa banyak prediksi yang benar dan salah untuk setiap kelas. Matriks konfusi secara umum dapat direpresentasikan sebagai berikut.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2 Matriks Konfusi¹

3. Skor F1

F1 *score* merupakan rata-rata harmonik dari *presisi* dan *recall*. F1 menjadi ukuran yang baik dalam mengukur akurasi model jika ada ketidakseimbangan dalam

¹ Narkhede, S. (2021, June 15). Understanding confusion matrix. Medium.
<https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

jumlah sampel antara kelas-kelas berbeda. Adapun formula untuk *precision* dan *recall* secara berturut-turut ditunjukkan oleh Persamaan (3) dan (4).

$$Precision = \frac{TP}{TP + FP} \quad \dots(3)$$

$$Recall = \frac{TP}{TP + FN} \quad \dots(4)$$

di mana

- TP (*true positive*) : jumlah prediksi yang benar positif
- FP (*false positive*) : jumlah prediksi yang salah positif
- FN (*false negative*) : jumlah prediksi yang salah negatif

Dari kedua persamaan tersebut, dapat dirumuskan skor F1 pada persamaan (5).

$$F1\ Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad \dots(5)$$

Skor F1 memiliki daerah hasil di antara 0 dan 1. Semakin tinggi skor, kinerja model dinilai lebih baik. Skor F1 yang tinggi menunjukkan bahwa model memiliki keseimbangan yang baik antara kemampuan untuk mengidentifikasi kelas positif (*precision*) dan kemampuan untuk menemukan semua instans kelas positif yang ada (*recall*). Lebih lanjut

4. Cohen's Kappa

Cohen's Kappa adalah ukuran statistik yang digunakan untuk mengevaluasi seberapa baik dua set data kategorikal yang berbeda dapat menghasilkan nilai yang sama. *Cohen's Kappa* dihitung menggunakan kesepakatan yang diamati (P_0) dan peluang kesepakatan yang diharapkan (P_e) dalam formula berikut.

$$\kappa = \frac{P_0 - P_e}{1 - P_e} \quad \dots(6)$$

di mana

- P_0 : proporsi yang teramati dari kesepakatan antara dua himpunan klasifikasi
- P_e : proporsi yang diharapkan dari kesepakatan jika tidak ada kesepakatan yang terjadi secara kebetulan.

Nilai *Cohen's Kappa* berkisar dari -1 hingga 1 di mana -1 menunjukkan kesetujuan sempurna antara dua set klasifikasi, 0 menunjukkan kesetujuan yang diharapkan secara kebetulan, dan 1 menunjukkan ketidakserujuan sempurna antara kedua set.

III. Hasil

Dari data harga saham harian AAPL, dilakukan penggunaan indikator untuk memproses data *adjusted close* dengan tujuan membentuk gambar yang dapat digunakan dalam CNN. Penghitungan indikator dilakukan untuk membantu model lebih mengenali pola-pola pada harga saham. Kemudian, pada bagian **II.3 Data Labeling**, data akan dilabeli dengan

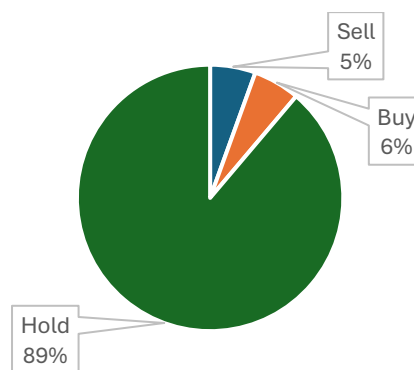
- 0 : *Sell*
- 1: *Buy*
- 2: *Hold*

Setelah itu data akan di-*preprocessing* menghasilkan variabel-variabel baru dari *technical indicator*. Cuplikan datanya ditunjukkan pada Gambar 3.

	Date	Open	High	Low	Close	Adj Close	Volume	rsi_6	rsi_7	rsi_8	...	dmi_15	dmi_16	dmi_17	dmi_18	dmi_19	dmi_20	dmi_21	dmi_22	dmi_23	Labels
30	2000-02-15	1.029018	1.070871	1.028460	1.062500	0.899512	485744000	72.696038	76.835396	79.793674	...	34.590150	36.285777	37.939623	39.545141	41.089363	42.597202	44.040927	45.429768	46.764635	0
31	2000-02-16	1.051339	1.054888	1.001116	1.018973	0.862862	378246400	48.159258	58.655606	64.345229	...	34.054820	35.853502	37.227624	38.787524	40.266775	41.721319	43.128830	44.488239	45.799386	2
32	2000-02-17	1.028460	1.031250	1.010045	1.025670	0.868332	289497600	50.313836	51.898792	61.359582	...	33.706805	35.196867	36.680701	38.145393	39.581938	40.964206	42.348190	43.671449	44.952689	2
33	2000-02-18	1.023438	1.030134	0.989955	0.993304	0.840931	233441600	45.938282	42.442376	44.199185	...	33.381991	34.768772	36.165950	37.557826	38.933145	40.283948	41.604724	42.891786	44.142805	2
34	2000-02-22	0.983259	1.044085	0.952567	1.016183	0.860300	422296000	51.509486	52.871429	40.850815	...	32.626843	33.967725	35.324437	36.681188	38.026413	39.351744	40.651227	41.920735	43.157519	2
...
6032	2023-12-22	195.179993	195.410004	192.970001	193.600006	193.353287	37122800	17.202025	43.670309	51.136163	...	20.569973	19.334348	18.231729	17.245871	16.379213	15.640682	14.972541	14.364833	13.809174	0
6033	2023-12-26	193.610001	193.889999	192.830002	193.050003	192.803986	28919300	14.443066	16.212729	41.828941	...	20.953421	19.713465	18.599896	17.598986	16.714012	15.954134	15.264906	14.636839	14.061832	0
6034	2023-12-27	192.490005	193.500000	191.089996	193.149994	192.903839	48087700	17.358911	16.032872	17.702440	...	21.311306	20.068887	18.946407	17.932484	17.031190	16.251914	15.543349	14.896481	14.303505	0
6035	2023-12-28	194.139999	194.660004	193.169998	193.580002	193.333298	34049900	29.985646	22.870982	21.175828	...	20.455621	19.375184	18.377922	17.461660	16.636358	15.915457	15.253537	14.644279	14.081905	0
6036	2023-12-29	193.899994	194.399994	191.729996	192.529999	192.284637	42628800	9.330794	23.599730	18.922148	...	19.656982	18.724836	17.842877	17.016993	16.262307	15.595823	14.977527	14.403542	13.869940	0

Gambar 3 Cuplikan data setelah preprocessing

Distribusi pembagian kelas ditunjukkan sebagai berikut.

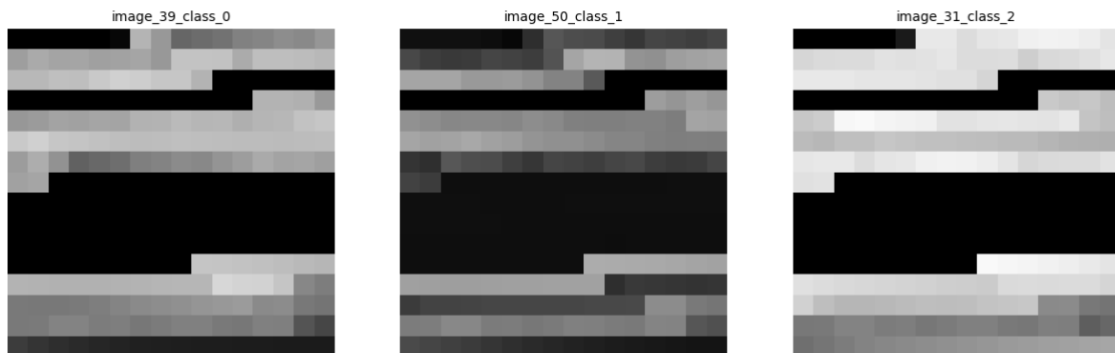


Gambar 4 Distribusi Pembagian Kelas

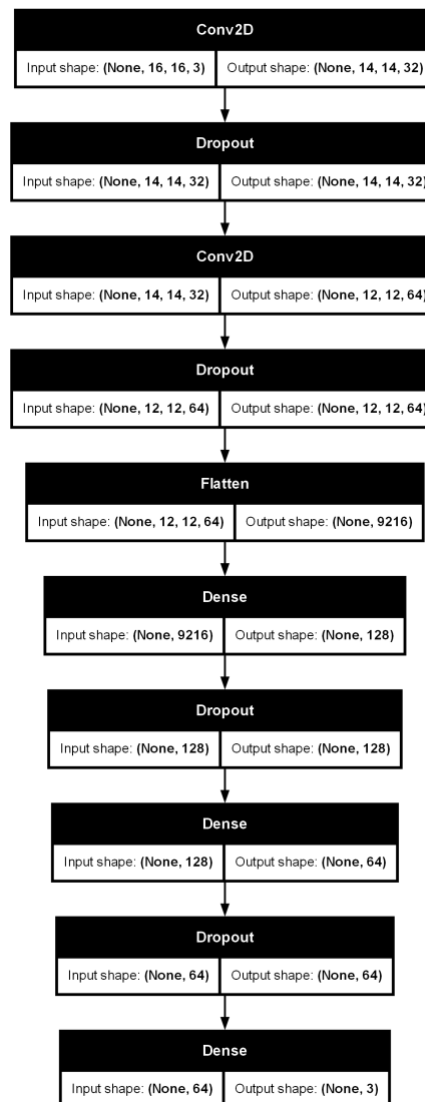
Kemudian, data baru akan diolah dengan menggunakan CNN. Pertama, model diberi masukan data berukuran (16×16) . Kemudian model menggunakan 2 *layer* konvolusi. *Layer* konvolusi pertama dengan filter berukuran (3×3) dan *stride* 1 menghasilkan *array* berukuran $(14 \times 14 \times 32)$. Lalu dilanjutkan dengan *dropout layer* sebanyak 20%. *Layer* konvolusi kedua dengan filter (3×3) dan *stride* 1 menghasilkan *array* berukuran $(12 \times 12 \times 64)$. Disusul dengan *dropout layer* sebanyak 30%. Gambar 5 menunjukkan sampel representasi hasil dari CNN.

Kemudian dilanjutkan dengan *fully connected layer* yang menggunakan 2 *layer* dan 2 *dropout layer*. *Layer* pertama berukuran 128 *neuron* dan fungsi aktivasi relu, lalu *dropout*

pertama sebanyak 20%, dan *layer* kedua berukuran 64 *neuron* dengan fungsi aktivasi ReLU lalu *dropout* sebanyak 20%. Terakhir, *output* layer diatur dengan ukuran 3 dengan fungsi aktivasi softmax. Arsitektur lengkap CNN dapat dilihat pada Gambar 6.



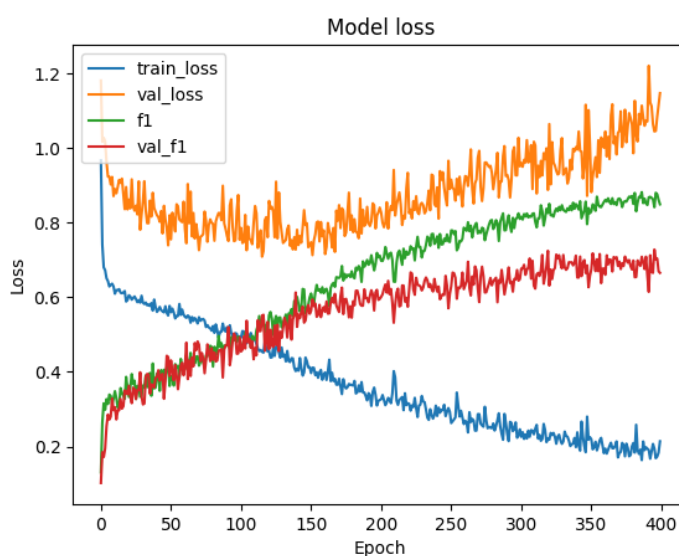
Gambar 5 Sampel representasi CNN dalam gambar piksel 16 x 16. Dari kiri ke kanan: Sell, Buy, Hold.



Gambar 6 Arsitektur CNN

IV. Diskusi

Simulasi dilakukan dengan *batch size* 80, 400 *epoch*, dan *learning rate* 0.001, serta menggunakan algoritma optimisasi *Adam*. Training dilakukan pada data *training* dan data *cross validation* (cv) menunjukkan model mampu untuk mempelajari pola dalam data dengan akurasi sebesar 0.8514 dan akurasi validasi sebesar 0.6664. Nilai *loss training* pada epoch terakhir sebesar 0.2136 dan *validation loss* sebesar 1.1462. Terakhir, pada training diperoleh skor F1 sebesar 0.8514 dan validasinya sebesar 0.6652. Plot dari proses training dapat dilihat pada Gambar 7.



Gambar 7 Plot Training Data

Dari hasil *training* tersebut, diperoleh matriks konfusi sebagai berikut.

Tabel 1 Confusion Matrix pada Test Set

		Prediksi			
Aktual		<i>Sell</i>	<i>Buy</i>	<i>Hold</i>	
	<i>Sell</i>	15	2	49	
	<i>Buy</i>	0	37	31	
	<i>Hold</i>	146	170	752	

Tabel 2 Evaluasi Metrik pada Test Set

Total Akurasi: 0.67			
	Sell	Buy	Hold
Recall	0.23	0.54	0.70
Precision	0.09	0.18	0.90

<i>F1-Score</i>	0.13	0.27	0.79
------------------------	------	------	------

Perhatikan bahwa pada Gambar 7, plot pembelajaran data baik itu akurasi maupun *loss*, memiliki sifat yang fluktuatif, tetapi seiring bertambahnya *epoch*, terdapat kecenderungan dari setiap metrik. Semakin tinggi *epoch* maka akurasi dan akurasi validasi model terus meningkat. Akan tetapi, semakin tinggi *epoch* pula, semakin jauh jarak dari akurasi dan validasi. Di sisi lain, semakin tinggi *epoch*, semakin rendah *loss*, tetapi tidak dengan *validation loss*. Nilai *validation loss* yang justru naik ketika *epoch* semakin tinggi. Kedua fenomena ini menjadi suatu indikator bahwa model yang telah dibuat *overfit*. Salah satu penjelasan untuk ini yaitu karena walau pada praktiknya kita telah menggunakan bobot untuk *fitting* model. Data saham ini sangat tidak berimbang seperti yang ditunjukkan oleh Gambar 4. Ketidakseimbangan yang sangat signifikan pada data menjadi salah satu faktor penyebab model *overfit*. Ketidakseimbangan ini memiliki implikasi yaitu data kelas minoritas memiliki kemungkinan untuk mengandung lebih banyak *noise* relatif terhadap jumlah data yang seringkali tidak terbaca dalam komputasi model.

V. Konklusi

V.1 Kesimpulan

Dalam penelitian ini, penulis membangun model *Convolutional Neural Network* (CNN) untuk prediksi pergerakan harga saham menggunakan indikator teknis. Dari simulasi, diperoleh kesimpulan bahwa model CNN dapat digunakan untuk memprediksi harga saham dengan akurasi yang cukup tinggi. Akan tetapi, data saham merupakan data praktikal dalam ilmu ekonomi yang sulit diprediksi. Salah satu dampaknya adalah ketidakseimbangan dari data saham yang ada untuk perilaku “sell”, “buy”, dan “hold”. Hal ini menyebabkan model CNN yang dibuat menjadi *overfit* terhadap data saham.

V.2 Saran

Pemodelan ini dilakukan untuk pembagian data yang sangat tidak seimbang untuk label “sell”, “buy”, dan “hold”. Penerapan ini mungkin cukup praktikal dalam kehidupan sehari-hari, tetapi dapat menyebabkan model menjadi *overfit*. Salah satu cara yang dapat dilakukan selanjutnya adalah mengeksplorasi metode supaya tidak *overfit*, salah satu caranya dengan melakukan augmentasi pada data supaya data kelas minoritas dapat lebih berimbang dalam jumlahnya.

VI. Referensi

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Narkhede, S. (2021, June 15). *Understanding confusion matrix*. Medium.
<https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>
- Nayak, A. (2020). Stock trading with CNNs: Time series to image conversion. Medium.
<https://towardsdatascience.com/stock-market-action-prediction-with-convnet-8689238feae3>
- Sezer, O. B., & Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, 70, 525-538.

VII. Kontribusi

Nama	Kontribusi
Gerend Christopher	<ul style="list-style-type: none">• Menyusun kode python• Melakukan simulasi model• Menulis abstrak, pendahuluan, dan metodologi• Membuat sebagian <i>powerpoint</i>
Kevin Sean Hans Lopulalan	<ul style="list-style-type: none">• Menyusun kode python• Melakukan simulasi model dan <i>tuning hyperparameter</i>• Menulis sebagian metodologi• Analisis hasil dan pembahasan• Membuat sebagian <i>powerpoint</i>