

Transfer Learning with Pre-trained ResNet-50: Predicting SPX Option Prices

Zihao Lin

School of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong 510006, China

Abstract: Due to the volatility and complexity of the options market, it has exacerbated the difficulty of predicting option prices and attracted widespread attention. In recent years, many machine learning (ML) techniques have been explored for this purpose. This paper implements the advanced ML models to boost the accurate prediction of the option price, which includes the comparison between traditional ML techniques Random Forest and Support Vector Regression (SVR), as well as the Convolutional Neural Network (CNN) ResNet-50. Additionally, we further compare the performance of ResNet-50 pre-trained on ImageNet through transfer learning. These methods are used to predict the implied volatility, which is then transformed into option pricing using the Black–Scholes Option Pricing Model (BSOPM). Finally, we compare the performance of these models based on error analysis using RMSE and R^2 . The experimental results show that the pre-trained ResNet-50 reduces the error by 6% compared to the un-pretrained ResNet-50, by 51.2% compared to Decision Trees, and by 51.1% compared to SVR. The predicted option prices align well with the actual values, demonstrating the feasibility of applying pre-trained ResNet-50 on option price prediction.

Keywords: ResNet-50; CNN; Option Price; Transfer Learning.

1. Introduction

Options are derivatives of underlying assets and are financial products that allow trading on future market prices. There are two types of options: call options and put options. A call option is a financial contract that allows the buyer to purchase the underlying asset from the issuer at a specified price in the future, while a put option allows the buyer to sell the underlying asset to the issuer at a specified price in the future. Options play a crucial role in both the financial sector and everyday life. Companies may issue stock options to employees as part of incentive plans, enabling employees to buy company stock at a predetermined price in the future and thus share in the company's future growth [1]. Exchanges also issue options, providing standardized option contracts that allow investors to buy and sell in the secondary market. These option contracts typically have stocks as the underlying assets and come with preset prices and expiration dates. Through this mechanism, investors can purchase option contracts, aiming to profit from future fluctuations in stock prices [2]. The trading volume in the options market increased from 9.42 billion contracts in 2013 to 54.53 billion contracts in 2022, far surpassing the growth of the futures market during the same period. This demonstrates that options have become a highly liquid financial product with significant investment opportunities. However, the vast majority of options traders incur losses due to their lack of sufficient knowledge and tools. Therefore, an accurate prediction model is crucial as it can significantly enhance investors' success rates in various aspects. Our research focuses on options prediction and aims to evaluate the performance of different machine learning models in addressing this issue.

Currently, many researchers have made progress using pre-trained CNN networks for transfer learning in various fields. Hoo-Chang Shin et al. [3] utilized a CNN model pre-trained on ImageNet for transfer learning, applying it to Computer-Aided Detection problems with good results, which outperforms other machine learning models. Ali et al. [4]

studied the performance of pre-trained CNN models on tasks far removed from the original training set, reporting consistently superior results, and highlighting the universality of learned representations. Minyoung et al. [5] pre-trained models on subsets of ImageNet and evaluated their transfer performance across various standard visual tasks, finding that most changes in pre-trained data selection did not significantly affect transfer performance. They even argued that CNN training is not as data-hungry as previously thought, reaffirming its generalization ability. All the above studies have confirmed the good transferability and excellent performance of pre-trained CNN models. In the context of option price prediction, there are also several precedents for using machine learning methods. Omer et al.'s [6] literature review reported various neural network applications in financial forecasting, including Jonathan et al.'s [7] use of CNN models to predict price fluctuations, finding excellent performance. However, there have been no reports on using ImageNet pre-trained CNNs for transfer learning in volatility prediction. Boris et al. [8] proposed a method to visualize non-image data and then use CNN algorithms to solve learning problems. Ivascu [9] used machine learning methods for option pricing, showing better results than traditional methods, but did not mention pre-trained CNN methods.

Overall, the aforementioned research on applying neural networks to option price prediction has achieved significant results, and the use of pre-trained CNNs for transfer learning has also been proven feasible. However, the required dataset dimensions for different research subjects are diverse and extensive, and there have been no reports on using pre-trained CNN networks for option price prediction. This inspires and necessitates the exploration of transferring pre-trained CNN models to the problem of option price prediction, as better predictions can reduce investors' losses and help them achieve higher returns.

The contribution of this study lies in proposing a method to transform vector data into images during data preprocessing and comparing the performance of different algorithms in

option prediction. The results indicate that on a limited historical trading dataset, the pre-trained ResNet-50 outperforms the Random Forest and SVR algorithms by about 51% and the un-pretrained ResNet-50 by 6%. The structure of the paper is as follows: Section 2, Models and Methods, briefly explains the models used and the ResNet-50 transfer learning method; Section 3, Experiments and Discussion; and Section 4, Conclusion, which discusses the limitations of the study and suggests directions for future work.

2. Models and Methods

2.1. The Black-Scholes Option Pricing Model (BSOPM)

In this paper, we use the Black-Scholes Option Pricing Model (BSOPM) to convert the predicted implied volatility for the target date into option prices. For call options, the following equation (1) applies:

$$C(S, t) = S_t N(d_1) - e^{-rT} K N(d_2) \quad (1)$$

In this formula, C is the call option price, S_t is the current stock price, K is the strike price, r is the risk-free interest rate, T is the time to maturity, and σ is the implied volatility. N is the cumulative distribution function of the standard normal distribution, and d_1, d_2 are calculated as (2)(3):

$$d_1 = \frac{\ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} \quad (2)$$

$$d_2 = d_1 - \sigma\sqrt{T} = \frac{\ln\left(\frac{S_t}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} \quad (3)$$

2.2. Random Forest and Support Vector Regression

Random Forest is an ensemble learning method [10] that constructs multiple decision trees and averages their predictions as the output. Its key concepts include:

Bagging: Random Forest independently trains decision trees by drawing bootstrap samples from the training set.

Random Feature Selection: Random Forest splits nodes based on subsets of features to reduce the correlation between trees.

Ensemble Averaging: The final prediction is the mean of all the trees' predictions. The formula is as follows:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x') \quad (4)$$

Out-of-Bag (OOB): Each tree leaves out a subset of the training data, called out-of-bag samples, which are used to estimate the generalization error and adjust the number of trees accordingly.

Support Vector Regression (SVR) is a supervised learning algorithm adapted from Support Vector Machines (SVM) for regression tasks [11]. SVR introduces an ϵ -insensitive loss function that defines a tolerance margin, within which errors are not penalized. Only errors beyond the ϵ margin incur a loss. SVR aims to find the Optimal Hyperplane. In regression, this hyperplane is a function that best fits the data (support vectors) while maintaining a margin of ϵ around it. SVR also employs the Kernel Trick to transform the input space into a higher-dimensional space. This allows SVR to find a linear fit in the transformed space, which corresponds to a nonlinear fit in the original space. Its regularization parameter (C) controls the trade-off between a low error margin and a smooth regression function. A higher C value penalizes large errors more, resulting in a tighter fit, while a lower C value allows more error tolerance and results in a smoother function.

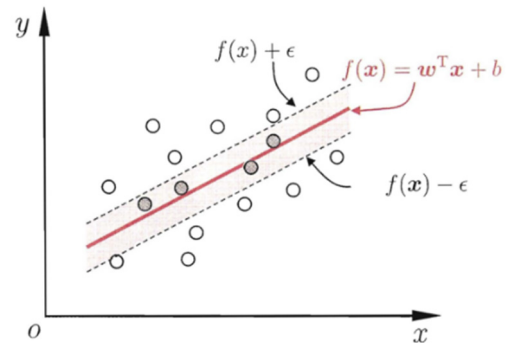


Figure 1. Support vector regression schematic diagram, red indicates the ϵ -margin band, samples falling into it do not calculate loss. [12]

2.3. ResNet-50 and Transfer Learning

ResNet-50 is a classical Convolutional Neural Network (CNN) commonly used for image classification and recognition. Figure 2 shows its network architecture. What sets ResNet-50 apart from other CNNs is its use of a residual structure introduced by the designers [13]. This structure allows some intermediate layers of the neural network to adapt to a residual mapping instead of directly fitting the desired underlying mapping. Assume the underlying mapping is $H(X)$, and the stacked nonlinear layers adapt to the residual mapping: $F(X) = H(X) - X$. The output of the convolutional layers is then added directly to the input, reformulating the original mapping as $F(X) + X$. This is achieved by adding a "shortcut connection" that skips one or more layers and directly adds the input to the output of the stacked convolutional layers. Figure 3 illustrates the structure of a residual block in ResNet-50, and Figure 4 provides a detailed breakdown of ResNet-50's composition.

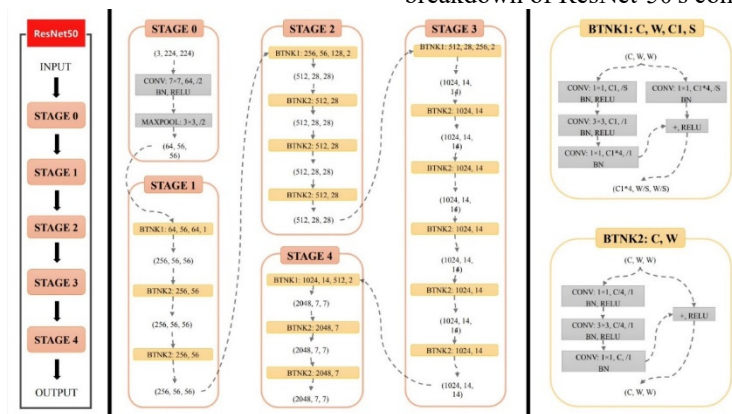


Figure 2. There are five main stages between the input and output

STAGE0 is the input layer, and each subsequent stage consists of two types of bottleneck blocks (special residual blocks): BTNK1, where the number of input and output channels differs, and BTNK2, where the number of input and output channels is the same.

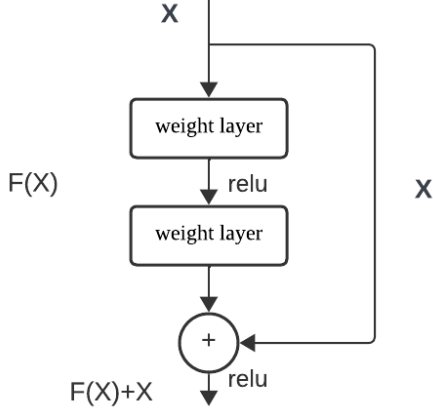


Figure 3. ResNet-50 Residual Block and Shortcut Connection Diagram

Layer name	Output size	50-layer
conv1	112×112	7×7, 64, stride 2
conv2.x	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 1 \times 1, & 64 \\ 3 \times 3, & 64 \\ 1 \times 1, & 256 \end{bmatrix} \times 3$
conv3.x	28×28	$\begin{bmatrix} 1 \times 1, & 128 \\ 3 \times 3, & 128 \\ 1 \times 1, & 512 \end{bmatrix} \times 4$
conv4.x	14×14	$\begin{bmatrix} 1 \times 1, & 256 \\ 3 \times 3, & 256 \\ 1 \times 1, & 1024 \end{bmatrix} \times 23$
conv5.x	7×7	$\begin{bmatrix} 1 \times 1, & 512 \\ 3 \times 3, & 512 \\ 1 \times 1, & 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax
FLOPS		3.8×10 ⁹

Figure 4. Composition of Convolutional Layer

To address potential discrepancies between the input and output sizes of residual blocks, ResNet-50 employs zero-padding. For issues where the number of channels in the feature maps might be inconsistent, a 1×1 convolution kernel is used to adjust them. This structure has a notable advantage in solving the vanishing gradient problem. During backpropagation of errors in the neural network, the presence of shortcut connections ensures that even very small errors can be effectively propagated back, thereby preventing the vanishing gradient issue.

2.4. Data Preprocessing

Next, we will explain how to preprocess the original vector data to fit the input of the pre-trained ResNet-50, enabling transfer learning. In the past, there have been related studies on solving non-image learning problems by mapping to images, such as [15]. Similarly, we use the following operations to preprocess data. Our design treats the ResNet-50 pre-trained on ImageNet as an effective feature extractor, then customizes the top layer: a global average pooling layer

and two fully connected layers for this task's transfer learning. To train these top layers, we convert the four feature dimensions of the input into RGB three-channel images. Specifically, the normalized values of the four dimensions are projected onto the range [0, 255] as grayscale, forming a 2×2 image. The standard input size for ResNet-50 is an RGB image with a height and width of 224 (224×224×3), but an input size of 56×56×3 is sufficient for this transfer learning task, greatly reducing computation while maintaining balanced scaling to avoid distortion. Therefore, we proportionally enlarge the 2×2 image, transforming each pixel into a 28×28 pixel block, and expand the grayscale x to an RGB three-channel (x, x, x). Finally, each sample is processed into an image as shown in Fig. 5.

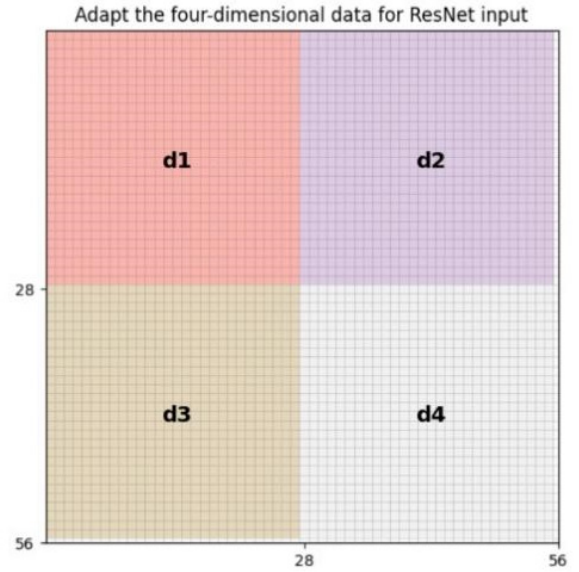


Figure 5. Convert the four-dimensional vector into an RGB three-channel image

After training the custom top layers with these images, the model can learn how to apply the high-level features extracted by the pre-trained model to predict option prices.

2.5. Error Assessment based on RMSE and R²

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (5)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \bar{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (6)$$

In formulas (5) and (6), n represents the number of samples. RMSE is a common metric used to evaluate regression, characterized by maintaining the same unit as the original data. This characteristic makes it very suitable for measuring price errors, as it brings the error metric back to the price scale [16]. R^2 is a metric used only in regression problems [16] to measure the amount of variation explained by the model as a proportion of the total variation, reflecting the model's ability to interpret fluctuations. Therefore, this paper uses these two indicators to measure errors. The smaller the RMSE and the closer the R^2 is to 1, the better the accuracy of the model. The error handling in this study is used in two ways: first, to measure the prediction accuracy of different models to determine the best-performing model; second, to adjust the internal parameters of the model (such as epoch, learning rate, etc.).

3. Experiments and Discussion

3.1. Data Description

The dataset used in the experiment is from the website www.DiscountOptionData.com. This website has recorded daily options data since 2005. We selected SPX call option contracts from December 1, 2023, to December 28, 2023,

with strike prices within $\pm 30\%$ of the underlying price as the dataset. Of these, 80% were used as the training set and 20% as the validation set. The data from December 29, 2023 (with strike prices within $\pm 30\%$ of the underlying price and with maturity between 100 and 500 days), was used as the prediction target. An example of the dataset information is as follows:

Table 1. Example of the raw dataset. Note: The implied volatility of some data points is very low, not missing.

Index	Symbol	Date	Maturity	StrikePrice	UnderlyingPrice	PutCall	AskPrice	AskSize	BidPrice	BidSize	LastPrice	ImpliedVolatility
1	SPX	2023/12/4	0.030136986	100	4569.78	call	4470.1	77	4460.2	77	4449.6	0
2	SPX	2023/12/4	0.030136986	1000	4569.78	call	3571.7	77	3561.6	77	3546.85	0
3	SPX	2023/12/4	0.030136986	1100	4569.78	call	3471.9	77	3461.8	77	3457.18	0
4	SPX	2023/12/4	0.030136986	1200	4569.78	call	3372.2	77	3360.3	30	3366.65	0
5	SPX	2023/12/4	0.030136986	1300	4569.78	call	3272.2	77	3262.5	77	3249.85	0
6	SPX	2023/12/4	0.030136986	200	4569.78	call	4369.9	77	4360.7	77	4353.7	0
7	SPX	2023/12/4	0.030136986	300	4569.78	call	4270.3	77	4260.3	77	4254	0
8	SPX	2023/12/4	0.030136986	400	4569.78	call	4170.5	77	4160.4	77	4157.1	0
9	SPX	2023/12/4	0.030136986	500	4569.78	call	4070.7	77	4060.6	77	4050.35	0
...												
93550	SPX	2023/12/28	5.98630137	6200	4783.35	put	1108.3	5	1028.3	5	0	0

This study selected 'maturity', 'StrikePrice', 'UnderlyingPrice', and 'PutCall' as the four input features based on the following considerations: Firstly, these features are core variables in BSOPM and theoretically have a direct impact on option prices and implied volatility. Secondly, related studies on the implied volatility surface show that maturity and StrikePrice significantly affect it, for example [17]. Additionally, preliminary analysis of historical data revealed that these features are highly correlated with implied

volatility. When using these features for training regression models, the model performance significantly outperforms the cases without these features. Therefore, selecting these features is not only theoretically sound but also empirically validated.

Even for the same option contract, trading prices on the same day can vary. Therefore, we define the weighted midprice (also known as micro-price) to describe the actual value of the option on that day,

$$\text{weighted midprice} = \frac{\text{asksize} * \text{bidprice} + \text{bidsize} * \text{askprice}}{\text{asksize} + \text{bidsize}} \quad (7)$$

The typical interpretation of formula (7) is that if the number of bids for an option exceeds the number of asks, it indicates greater buying pressure, thus the "true" price is closer to the ask price rather than the bid price. The weighted midprice not only takes into account the prices of the bid and

ask but also considers the imbalance in the number of bids and asks, providing a more accurate market price than the mid-price. This method has been widely applied [18].

For options that are not bidded or asked that day, we use lastprice for valuation:

$$\text{weighted midprice} = \frac{1}{2}(\text{askprice} + \text{lastprice}), \text{ if } \text{bidprice} = 0 \quad (8)$$

$$\text{weighted midprice} = \frac{1}{2}(\text{bidprice} + \text{lastprice}), \text{ if } \text{askprice} = 0 \quad (9)$$

$$\text{weighted midprice} = \text{lastprice}, \text{ if } \text{askprice} = 0 \text{ and } \text{bidprice} = 0 \quad (10)$$

3.2. Experiment Setting

We used SPX information from November 1, 2023, to December 28, 2023, as the dataset, with 80% of it used as the training set and 20% as the validation set, and the data from December 29, 2023, as the final test set. Figure 7 shows the OOB error of the Random Forest algorithm as the number of trees changes. Figure 8 is a heatmap of the SVR algorithm's cross-validation loss versus (C, epsilon) parameter pairs. The optimal parameters determined from the images are shown in Table 2. Figures 9 and 10 show the loss changes with epochs during cross-validation for Un-pre-trained ResNet-50 and Pre-trained ResNet-50 (learning rate=0.001). It can be seen that the loss for Un-pre-trained ResNet-50 almost stops decreasing at epoch 10, while Pre-trained ResNet-50 requires epoch 50. The final optimal parameters are shown in Table 3.

3.3. Results

After training the model with the optimized parameters from the previous section, we make predictions on the final test set. By substituting the predicted implied volatility into the BSOPM formula (1), we obtain the final option price prediction. The comparison of prediction errors among different models is shown in Table 4.

It is evident that the Pre-trained ResNet-50 has the smallest RMSE and R^2 , indicating the best performance. The scatter plot of the prediction results from the Pre-trained ResNet-50 is shown in Fig. 11, where the scatter distribution close to $y = x$ indicates that the predicted values are close to the actual values. The prediction result data is presented in the Table 5.

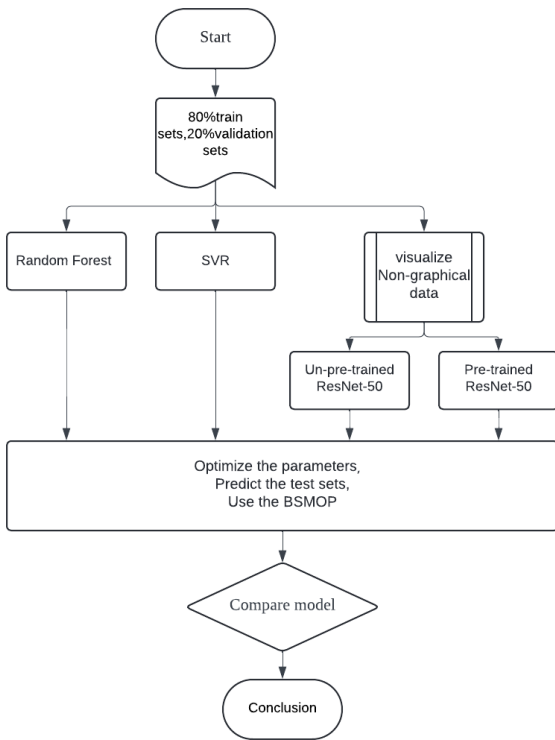


Figure 6. Architecture of Predicting option price and comparing the models

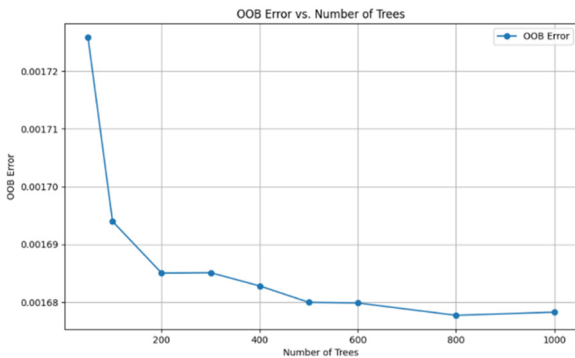


Figure 7. OOB error- $n_{estimators}$ line chart for Random Forest

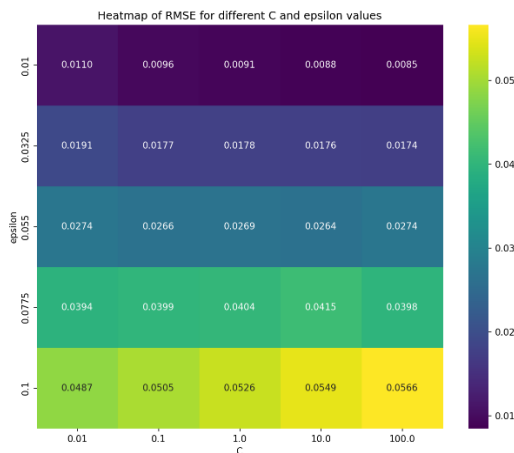


Figure 8. Heatmap of cross-validation loss with (C, epsilon) parameter combinations for SVR

Table 2. Optimized parameters of Random Forest and SVR

Optimized parameters of Random Forest		Optimized parameters of SVR	
n_estimators	800	C	100
max_depth	1000	epsilon	0.001
min_samples_split	50	gamma	scale
min_samples_leaf	20	kernel	rbf

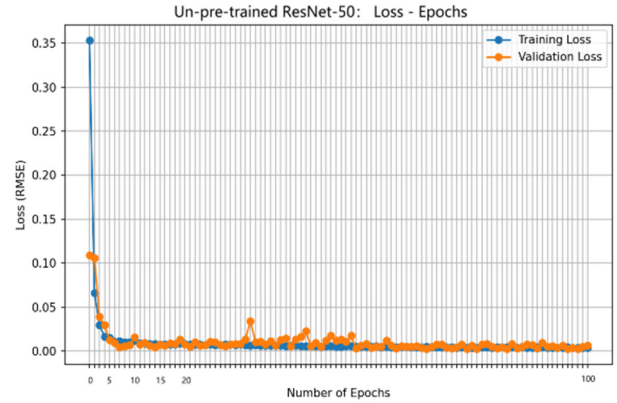


Figure 9. Line chart of Loss-Epochs for Un-pre-trained ResNet-50

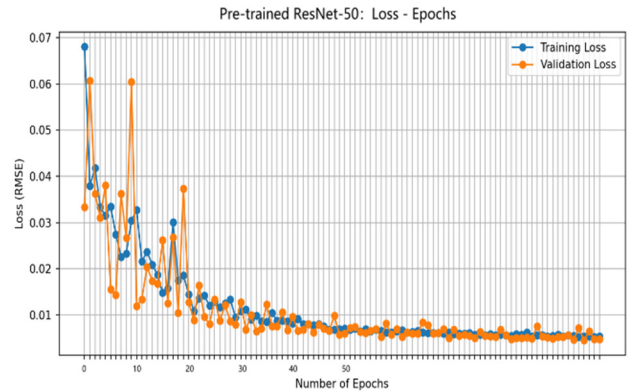


Figure 10. Line chart of Loss-Epochs for Pre-trained ResNet-50

Table 3. Optimized parameters of Un-pre-trained/ Pre-trained ResNet-50

Optimized parameters	Un-pre-trained ResNet-50	Pre-training ResNet-50
Epoch	10	50
Learning Rate	0.001	0.001
RMSE in Cross-Validation	0.0130	0.0071

Table 4. Error in the prediction results of four models.

Model	RMSE	R2
Random Forest	126.83811132344044	0.9191872682241727
SVR	126.57465092687552	0.9195226381221107
Un-pre-trained ResNet-50	66.26175925800771	0.9779325065386534
Pre-trained ResNet-50	61.878718575051955	0.9807662982367042

4. Discussion and Future Work

From the comparison of RMSE, Pre-trained ResNet-50 improved by 51.2% compared to Random Forest, by 51.1% compared to SVR, and by 6% compared to Un-pre-trained ResNet-50. This demonstrates that transferring the Pre-trained ResNet-50 to the problem of option price prediction is valuable, and it yielded a neural network to price options that

was quite accurate in terms of RMSE and R2.

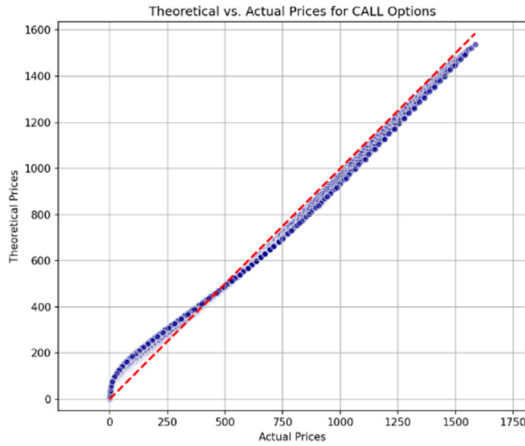


Figure 11. Distribution chart of the prediction results of pre-trained ResNet-50, where the red dashed line represents predicted values equal to actual values.

Table 5. Example of prediction results from pre-trained ResNet-50

Prediction results of pre-trained ResNet-50		
S/N	Predicted Price	Actual Price
1	1553.904	1585.25
2	1533.067	1567.05
3	1523.566	1556.05
4	1511.956	1544.7
5	1510.514	1544.45
6	1509.968	1540.8
7	1502.915	1534.05
8	1500.734	1533.3
9	1494.189	1524.957
10	1488.82	1522.35
11	1488.031	1521.9
12	1483.714	1515.075
13	1479.518	1510.8
14	1477.987	1510.55
15	1475.617	1501
...
1310	2.142062	0.140649

However, there may be some possible limitations in this study. Due to the constraints of computational resources and time, this paper focused only on the transfer learning of the ResNet-50 CNN model, and there may be other more promising models yet to be explored. Additionally, the dataset has a small-time span, which might introduce bias due to market uncertainties during that period. In future work, investigating more pre-trained CNN models and finding more reasonable data sampling methods are needed. Exploring how to transfer pre-trained CNN networks to more non-image problems is also a promising direction for future research.

5. Conclusion

This paper designed a method to transfer Pre-trained ResNet-50 to the option pricing problem. Treating the Pre-trained ResNet-50 as a high-level feature extractor, we trained custom top layers to fit the input by converting vector data into images. The final option price prediction was achieved with an error of RMSE = 61.879 and $R^2 = 0.981$, improving by 51.2% over Random Forest (RMSE = 126.838, $R^2 = 0.919$), by 51.1% over SVR (RMSE = 126.575, $R^2 = 0.920$), and by 6% over Un-pre-trained ResNet-50 (RMSE = 66.262, $R^2 = 0.978$). This demonstrates the effectiveness of transferring pre-trained CNN networks to the option pricing problem.

Acknowledgments

I sincerely thank my advisor Vipul Goyal for his guidance, support, encouragement, and assistance during the research period. His expertise and insights have been instrumental in shaping this work. I also want to thank my mentor Yuru Jing for the advice and help provided during the writing process. Lastly, I am grateful to all those who have indirectly contributed to this research. Thank you for your support and understanding.

References

- [1] Business Development Bank of Canada. (n.d.). How to use stock options to recruit, retain and motivate start-up employees. BDC.ca. Retrieved from <https://www.bdc.ca/en/articles-tools/employees/manage/how-to-use-stock-options-to-recruit-retain-motivate-start-up-employees/>.
- [2] Bullish Bears. (n.d.). Trading options for a living. *Bullish Bears*. Retrieved from <https://bullishbears.com/trading-options-for-a-living/>.
- [3] Shin, Hoo-Chang, et al. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning." *IEEE transactions on medical imaging* 35.5 (2016): 1285-1298.
- [4] Sharif Razavian, Ali, et al. "CNN features off-the-shelf: an astounding baseline for recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014.
- [5] Huh, Minyoung, Pulkit Agrawal, and Alexei A. Efros. "What makes ImageNet good for transfer learning?." *arXiv preprint arXiv:1608.08614* (2016).
- [6] Sezer, Omer Berat, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019." *Applied soft computing* 90 (2020): 106181.
- [7] Doering, Jonathan, Michael Fairbank, and Sheri Markose. "Convolutional neural networks applied to high-frequency market microstructure forecasting." *2017 9th computer science and electronic engineering (ceec)*. IEEE, 2017.
- [8] Kovalerchuk, Boris, Bedant Agarwal, and Divya Chandrika Kall. "Solving non-image learning problems by mapping to images." *2020 24th International Conference Information Visualisation (IV)*. IEEE, 2020.
- [9] Ivaşcu, Codruţ-Florin. "Option pricing using machine learning." *Expert Systems with Applications* 163 (2021): 113799.
- [10] Breiman, Leo. "Random forests." *Machine learning* 45 (2001): 5-32.
- [11] Pedregosa, Fabian, et al. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, vol. 12, 2011, pp. 2825-2830. Scikit-learn, <https://scikit-learn.org/stable/modules/svm.html>.
- [12] Zhou Zhihua. CSDN. (2018). Retrieved from https://blog.csdn.net/weixin_37160123/article/details/80963892.
- [13] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [14] Bing.ResNet50 Network architecture diagram and detailed structure explanation [EB/OL].(2021-02-26)[2022-05-31]. zhuanlan.zhihu.com/p/353235794.
- [15] Kovalerchuk B, Agarwal B, Kall D C. Solving non-image learning problems by mapping to images[C]//2020 24th

- International Conference Information Visualisation (IV). IEEE, 2020: 264-269.
- [16] Vitalflux. (2023). MSE vs RMSE vs MAE vs MAPE vs R-Squared: When to Use?. *Vitalflux*. Retrieved from <https://vitalflux.com/mse-vs-rmse-vs-mae-vs-mape-vs-r-squared-when-to-use/>.
- [17] Homescu, Cristian. "Implied volatility surface: Construction methodologies and characteristics." arXiv preprint arXiv:1107.1834 (2011).
- [18] Kumar, Santosh, et al. "The Six Decades of the Capital Asset Pricing Model: A Research Agenda." Journal of Risk and Financial Management 16.8 (2023): 356.