Homework DL4CV

Team Name: Visionaire
Member 1: Yun Kui Alessandro Hu, 3225149, yun.hu2@studbocconi.it
Member 2: Tommaso Renso, 3093876, tommaso.renso@studbocconi.it
Member 3: Elena Rossi, 3110844, elena.r@studbocconi.it

Introduction to the problem

Within the context of changing financial environments, there is an ongoing investigation of novel approaches to improve investment decision-making.In the past, a number of researchers have mostly used machine learning methods to forecast future prices by feeding the models with “traditional” data. This project introduces a new data source—images—that deviates from these traditional methods. This change in approach is predicated on the idea that using computer vision to analyse image-based data might end up in better results since images are able to provide a wider range of insight from the same amount of available information.

The main objective of this project is to create a model that can accurately forecast future price trends using financial data that is visually represented. Although we recognise that attaining excellence in this field can be difficult since the financial market is subject to countless number of parameters, the project is committed to mitigate substantial losses coming from trading and foster more scientific based decision-making among traders since one of the biggest obstacles in the trading world is the effect of losses due to the lack of a clear plan, strategy, and proper knowledge.

Proposed solution

In a typical trading strategy that utilizes technical analysis, predefined patterns play a central role. Our approach aims to revolutionize this by employing Convolutional Neural Networks to learn these patterns entirely through deep learning, eliminating the need for manually crafted features. Unlike traditional methods, where patterns are specified in advance, CNNs allow the model to autonomously identify and comprehend relevant features.

Considering the CNN's prediction based on specific patterns involving only a certain number of candles, it becomes crucial to assess the placement of these patterns within the image; whether they appear on the left side, right side, or encompass the entire image with all the candles. This distinction is vital because elements on the left side may predict elements on the right side (the most recent data), which is not our focus. Our interest lies in recent patterns predicting future trends that have yet to unfold.

To address this, the Gradient-weighted Class Activation Mapping (Grad-CAM) emerges as a valuable tool, allowing us to visualize and interpret the regions of the image influencing the model's predictions.

Implementation of the solution and experiments

The solution is delineated in the following phases:

1. Data collection and preparation
2. CNN Model building
3. Experiments
4. Grad CAM implementation

Data Collection and Data Preparation

Instead of relying on pre-existing datasets with images and labels, we chose to construct our dataset from scratch. To achieve this, we scraped financial data from the Financial Modeling Prep website. This comprehensive data collection effort spanned various asset classes, including currencies, equities, and commodities. In particular, we collected data associated with the following tickers: "CL," "BZ," "NG," "SI," "PL," "ZS," "ZM," "KC," "SB," "CC," "LE," "GF," "HE," "DC," and "LB."

For each of these assets, our data extraction focused on key variables, such as the Opening price (O), Closing price (C), Highest price (H), Lowest price (L), and trading volume. With this dataset in hand, we proceeded to compute other crucial technical indicators, including the 5-day, 50-day, and 200-day moving averages, to gain deeper insights into historical price trends.

After gathering the necessary "traditional data," we transformed them into visual representations through black and white single-channel images. For OCHL data, we feeded the images using the candlestick representation, where each candle's body depicted the open and close prices, while upper and lower wicks showed the highest and lowest prices during the trading period.

Similarly, we generated separate black and white images for other variables providing unique insights into specific financial parameters. *(see image 1 in appendix for an example of generated images)*

Subsequently, we merged all these images into a single composite image with dimensions of 51x8x4 where the channels represent the following features: OCHL, wicks, 5-day moving average and volume. As a result we obtained a dataset composed of 49.944 images. From this data set we created three sub dataset corresponding to the train, validation and test set.

The next step involved the labeling for which we employed the following approach: we examined the closing price of the last candle in the image, compared it to the opening price of the candle 21 days later, and assigned a label of 1 (positive trend) if the difference was higher; otherwise, we assigned a label of 0 (negative trend).After having labeled the data we looked at the balanceness of the data, 56% of the images were labeled with 1 while 44% of them were labeled with 0.

1. Model building

The final model we have designed for this project is a hybrid of a VGG18-like architecture integrated with residual blocks, tailored specifically for processing image-based financial data. The use of residual blocks is a key point here, as they help in building deeper networks without succumbing to the vanishing gradient problem. This is crucial for our task since the complexity of financial data requires a model capable of learning subtle and intricate patterns without losing essential information across layers. In our preliminary experiments with alternative architectures, we observed a plateau in the loss function during training, where further epochs ceased to yield improvement in both training and validation loss. Notably, this stagnation was not attributed to overfitting, underscoring the necessity for a more sophisticated model design such as residual net blocks.

Moreover, the model incorporates batch normalization and dropout layers, which are essential for stabilizing and regularizing the learning process, ensuring that the model remains generalizable to new, unseen data. The use of a Global Average Pooling layer before the classification stage simplifies the network and reduces overfitting risk even more by minimizing the total number of parameters in the model.

The final layer is a dense layer with a sigmoid activation function, since this activation function is designed for binary classification and what we want to infer is the future price trend that can be positive or negative. We utilize the AdamW optimizer for its efficiency in managing both learning rate

and weight decay. And finally the choice of Binary Crossentropy as a loss function that aligns perfectly with our binary classification task.

By integrating these advanced deep learning techniques, the model is not only adept at learning complex patterns in financial data but is also robust against overfitting and efficient in terms of computational resources.

2. Grad CAM implementation

In the technical implementation of Grad-CAM, the function integrates seamlessly with our hybrid CNN model. It targets the last convolutional layer, extracting the feature maps and using the gradients of the target class (indicating price trend direction) to produce a weighted combination of these maps. This process generates a heatmap that visually represents areas of the input image most influential in the model's decision-making. The heatmap is then resized and superimposed onto the original candlestick images, highlighting the specific patterns and regions that the model identifies as key predictors of future price movements. This technique not only offers a clear visual explanation of the model's internal workings but also assists in fine-tuning the model by pinpointing any potential biases or misinterpretations. Grad-CAM thus plays a pivotal role in enhancing the reliability and accuracy of our financial forecasting model, directly contributing to its ability to make more informed and scientifically-backed trading decisions.

Delving into the analysis of specific Grad-CAM images enhances our understanding of the model's focus areas. The *heatmap image 1 (see appendix)* shows a concentration on the right side of the candlestick chart, indicating the model's emphasis on recent data for its predictions, aligning with our aim of forecasting future trends. The *heatmap image 2 (see appendix),* showcasing heatmaps from different convolutional layers, reveals a progression in feature recognition—from basic features in early layers to more complex patterns in deeper layers. Interestingly, these heatmaps predominantly highlight the candlestick regions over the volume indicators, signifying the model's reliance on price action data for forecasting.

In further analyzing the Grad-CAM results, *heatmap image 3 (see appendix)* we observed that some highlighted areas correspond to the start and end of each image. This observation led to a crucial insight: when patterns recognized on the left side of the image and patterns on the right predict different market movements, it creates ambiguity in the model's final prediction. This is because our primary interest lies in the patterns predicting unseen future trends, not the observed data predicted by the left pattern. In addition, in cases where these two sets of patterns suggest conflicting trends, it becomes challenging to discern which pattern predominantly influences the model's prediction.

3. Experiments

Prior to finalizing our model, we conducted extensive experimentation with various combinations of techniques and input data. In one such trial, we augmented our model's input with two additional channels representing the 50-day and 200-day moving averages, resulting in a network with an input size of 51x80x6 *(see image 2 in appendix)*. Contrary to expectations, this modification led to a decrease in model accuracy. We hypothesize that these additional features did not contribute relevant information for our predictive task but instead confounded the model. Another significant modification was the substitution of the fully connected layers with a fully convolutional neural network architecture, which is typically adept at maintaining the spatial hierarchy of features. However, when applying Grad-CAM to the last layer of this fully convolutional neural network, the resulting heatmaps lacked detail, reducing the interpretability compared to those generated from a network with fully connected layers. This outcome suggests that while fully convolutional networks

are proficient at detecting granular and location-specific features, they may not provide the same level of clarity in the context of highlighting influential features for financial trend prediction.

Data augmentation was deliberately omitted from our process for two key reasons. Our self-constructed dataset allowed for direct adjustments to address any imbalances, making augmentation unnecessary. Moreover, altering financial candlestick charts through augmentation techniques like flipping or rotating could misrepresent the crucial temporal and financial data, compromising the model's predictive accuracy.

Results

Upon deploying our tailored CNN model, we analyzed the test data spanning from January 2016 to November 2023. The model achieved an accuracy of 54.67% in predicting future price trends, surpassing our benchmark by 4.67%. The benchmark, set at 50%, represents the expected accuracy of a random coin flip, indicating no predictive power. The improvement suggests that our model has learned to identify patterns that offer predictive insights beyond random chance, albeit modestly.

In the practical application of our model to trading, we devised a strategy that began with a capital of $1,000 invested only on the crude oil (CL) for a naive trial, with the approach of reinvesting our entire balance in each transaction. We tested two distinct trading strategies. The first, a 5-day strategy, involved making buy or sell decisions based on our model's 5-day predictions. This approach yielded an annualized return of 0.795%, which, notably, aligns with the performance of our 'lazy portfolio' benchmark—characterized by a buy-and-hold tactic, suggesting consistent but conservative growth.

In contrast, our second strategy was based on daily predictions, where trading decisions were made every day, leveraging the most immediate insights from our model. This more active approach resulted in a significantly higher annualized return of 4.82%. Such an outcome highlights the potential of our model to capitalize on short-term market movements and substantially outperform the more passive investment strategy. These results underscore the efficacy of our model in a real-world financial setting, demonstrating its value as a tool in strategic investment decision-making.

Conclusions

Our project's journey, culminating in the implementation of a CNN model for financial forecasting, represents a promising intersection of technical analysis and machine learning. While our current focus has been predominantly on technical analysis, the potential for integrating this model with other analytical approaches is substantial. Incorporating elements such as sentiment analysis from news media or fundamental economic analysis could create a more comprehensive predictive framework. This synergy could address a wider array of market-influencing factors, offering a more nuanced and holistic view of potential market movements.

In our next steps, we plan to expand the model's capabilities by combining it with other types of financial analysis. This could include looking at how news affects market sentiments or considering fundamental financial data. By doing this, we hope to make our model even more accurate and useful across different market situations. Our goal is to create a powerful, well-rounded tool that can adapt to and make the most of the varied and often complicated patterns seen in financial markets.

Appendix

image1

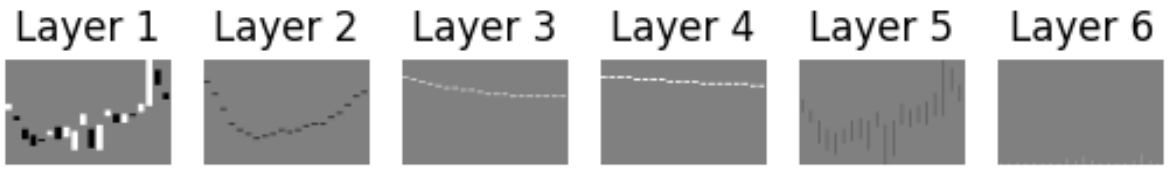
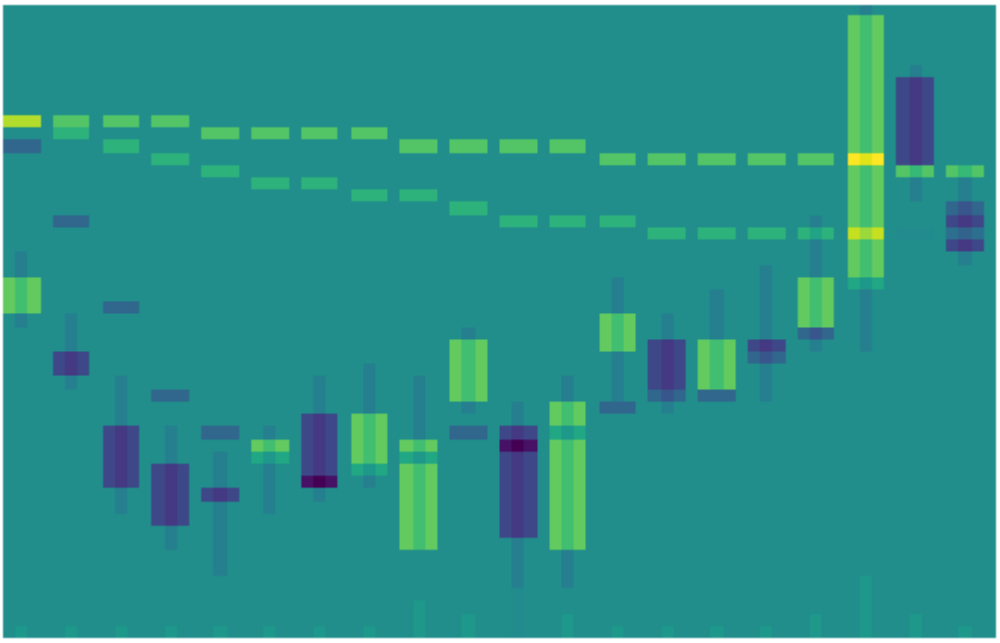
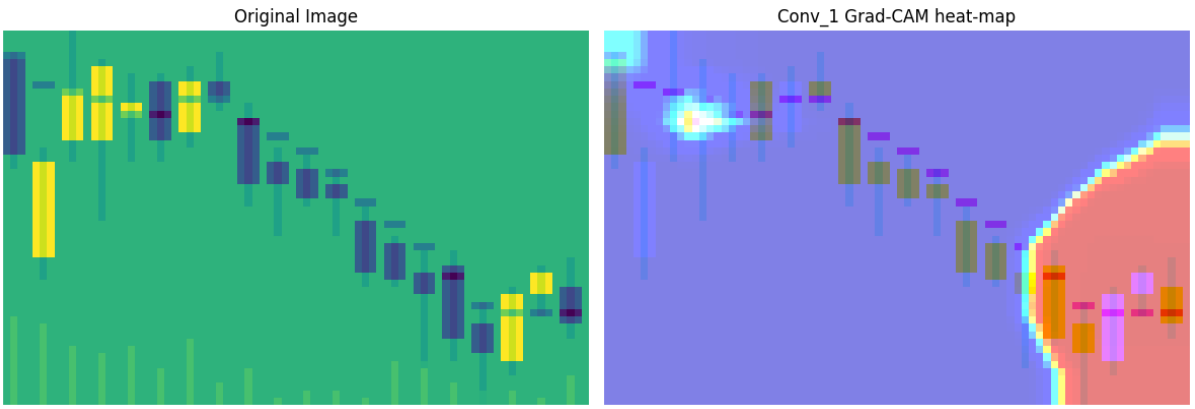


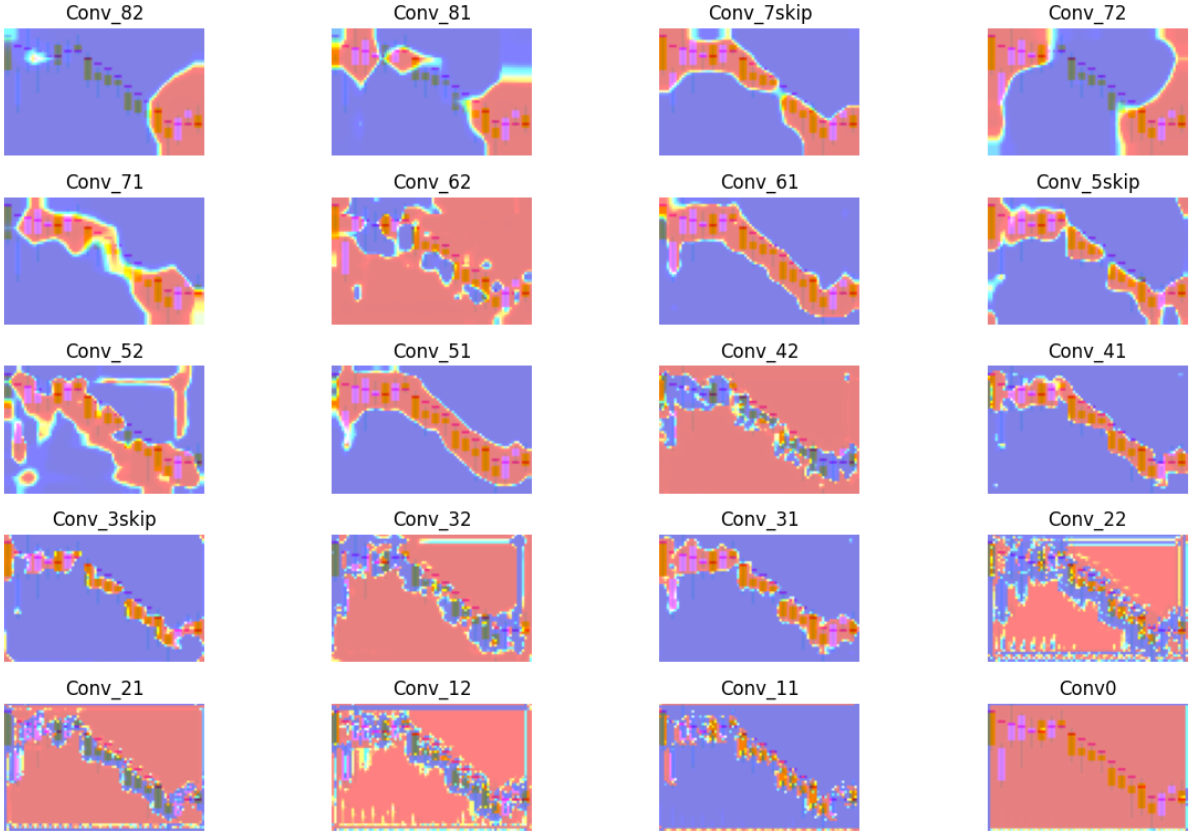
image2



heatmap image 1



heatmap image 2



heatmap image 3