Image Recognition in Stock Prediction with Visual Explanations from Grad-CAM

Jou-Ying Lee[1], **Shin Ehara**[2], **Sohyun Lee**[3]
[123] Halıcıoğlu Data Science Institute, University of California San Diego, CA, 92093, USA
[1] jol067@ucsd.edu; [2] sehara@ucsd.edu; [3] sol107@ucsd.edu;

Abstract

Deep learning architectures are now publicly recognized and repeatedly proven to be powerful in a wide range of high-level prediction tasks. While these algorithms' modeling generally have beyond satisfactory performances with apposite tuning, the long-troubling issue of this specific learning lies in the un-explainability of model learning and predicting. This interpretability of "how" machines learn is often times even more important than ensuring machines outputting "correct" predictions. Especially in the field of finance, users' ability to dissect how and why an algorithm reached a conclusion from a business standpoint is integral for later applications of i.e., to be incorporated for business decision making, etc. This project studies similar prior work done on image recognition in the financial market and takes a step further on explaining predictions outputted by the Convolutional Neural Network by applying the Grad-CAM algorithm.

1. Introduction

With big data collected at an exponential growing speed today, the automated decision power this information is capable of providing has been recognized and has since been at the forefront of technological developments – especially in the area of Artificial Intelligence.

Throughout the past decade, several machine learning algorithms have been developed with distinct strengths and each different useful area of applications. While most traditional models have presented outstanding potentials in pattern recognitions for structured data, their computing power and learning abilities are however, not sufficient on large, complicated input dataset. To address this drawback, deep learning models are introduced, and have up to today – a proven record of impressive learning performances on large-scale data such as audios, images and even videos.

While prediction powers of these algorithms are not to be neglected, deep learning models however also come with a major problem that has been long troubling these algorithm users – that is the "un-explainability" of these algorithms.

In order to capture these complex patterns within datasets, deep learning models are by nature, very intricated in their architectures. This, however, makes it extremely difficult or most of the times, nearly impossible for users to manually track or inspect models' learning processes. Although having a nice portfolio of model accuracies might be beneficial in performing prediction tasks, access to this training procedure is equally integral to understand how a projected outcome is made. In context of this project's setting -- being able to tell why a model forecasted stock trends to behave in a specific way is crucial for users' reference when making investment decisions. Regardless of disciplines for application, common concern users have raised with respect to deep learning algorithms is the difficulty in trusting model outcomes. Due to the lack of interpretability in model functionality, many researchers in fact choose to forfeit model accuracies in exchange for more trust and certainties in outputted results.

This project is thereby carefully structured after attempt to address this problem. We aim to add and build trust into deep learning systems by introducing more explainability into machine learning. Specifically, we chose one of the most complicated application for both prediction and explanation – and that is the area of finance, i.e., this research works towards introducing interpretability into modeling in the stock market, to enable users of more trustable insights and references during investment decision makings. In specific, we investigate the key indicator in stock investment – the stock price. Additionally, we pick to base the project on a diversified stock index – NIFTY 100 – in hope of a more systematic and wholistic view into the financial market. By inspecting deep learning model's prediction on stock price behaviors, i.e., whether a stock price increases or decreases throughout daily trading period, we aim to provide an explainable view into such financial decisions made by computer algorithms.

2. Preliminaries

2.1. Stock Market Price Change (Label)

The stock market is undoubtedly one of the most unpredictable, yet most popular areas for financial investment. Through facilitating exchanges of securities between buyers and sellers, this marketplace creates opportunities of capital gain for participants ranging from small individuals to big entities such as banks or conglomerates.

While there are countless financial measures in security discussions, this study focuses on one of the most direct assessments, i.e., the closing stock price. "The closing price is considered the most accurate valuation of a stock or other security until trading resumes on the next trading day" and is defined as "the last price at which the stock traded during the regular trading day" (Kenton).

As we purposefully structured this project as a classification task, this target for investigation is therefore transformed to be introduced as a binary label for model learning. For our project's investigation purposes, we assign one class only to each day to represent trades that happen on that day. In specific, we compare the opening price to the closing price of a specific day in order to make a careful call on assigning an "increase" or "decrease" label to the combined daily stock entries.

The India's National Market Exchange market opens on 9:15 AM, and marks market closing on 3:30 PM across weekdays. We turned away from traditional considerations on pre-market hours and after-market hours for study, and adapted our target of investigation to be the price difference between the earliest opening price and latest closing price trading entries during a day. If this difference is of a positive output: that signals an "increase" in stock value, while a negative output suggests the opposite. Therefore, summarizing that described above, given a particular day, we have our binary labels represented as the following:

$$\begin{aligned} Class &= \Delta Closing_{Price} \\ &= Price_{Latest} - Price_{Earliest} \\ &= \begin{cases} 1, if > 0, increase \\ 0 , if \leq 0, decrease \end{cases} \end{aligned}$$

2.2. Methodology

While most prevalent approaches to stock prediction might base around modelling with time series data, this research purposefully structures this attempt as an image classification task – both to explore deep learning algorithms' capability on learning non-conventional images, and to inspect explainability of these neural networks.

Specifically, this is done by encoding time series data as images by using *Gramian Angular Field and* applying *Grad-CAM* algorithm over the learned CNN model to inspect generated class-activation maps for visual explanations of the deep neural network.

2.2.1. Gramian Angular Field (GAF)

Gramian Angular Field is an image obtained by transforming time series data. In GAF, time series is represented in a polar coordinate system by taking advantage of the Gram Matrix (Oates and Zhiguang). Specifically, the Gram Matrix has a key advantage of preserving the temporal dependency -- "Since time increases as the position moves from top-left to bottom-right, the time dimension is encoded into the geometry of the matrix" (Vitry).

Finally, steps to obtain each GAF is extracted from Zhiguang Wang and Tim Oates publication: *Imaging Time-Series to Improve Classification and Imputation*, and are summarized as follows:

Given a times series X = {$x_1, \ldots, x_n$}, we rescale X so that all values fall in the interval [-1, 1] by:

$$\tilde{x}^{i}{}_{-1} = \frac{(x_i - max(X) + (x_i - min\,(X))}{max(X) - min\,(X)}$$

We can then represent the rescaled time series $\widetilde{X}$ *in polar coordinates by encoding the value as the angular cosine and the time stamp as the radius with the equation below, where* t_i *is the time stamp and N is a constant factor to regularize the span of the polar coordinate system:*

$$\begin{cases} \phi = \arccos(\tilde{x}_\iota)\,, -1 \le \tilde{x}_\iota \le 1, \tilde{x}_\iota \in \tilde{X} \\ r = \dfrac{t_i}{N}, t_i \in \mathbb{N} \end{cases}$$

After this rescaling transformation, the angular perspective is then exploited by considering the trigonometric sum/difference between each point to identify the temporal correlation within different time intervals.

In particular, our project exploits Gramian Difference Angular Field (GADF) that is defined as follows, where "*I* is the unit row vector [1, 1, …, 1]" (Wang and Time Oates).

$$GADF = \left[\sin\,(\phi_i - \phi_j)\right] = \sqrt{I - \tilde{X}^2}' \cdot \tilde{X} - \tilde{X}' \cdot \sqrt{I - \tilde{X}^2}$$

Finally, Figure 1 is a cited illustration of the various steps of encoding time series as Gramian Angular Field images.

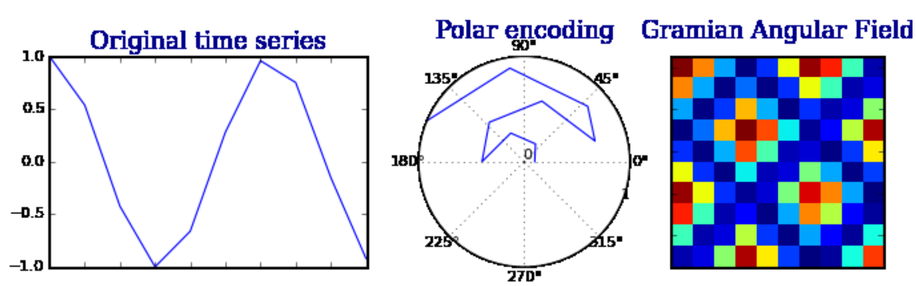


Figure 1: Various steps of the Gramian Angular Field Conversion (Vitry)

2.2.2. Gradient Weighted Class Activation Map (Grad-CAM)

Grad-CAM is an "Explainable AI" technique developed in 2016 by *Selvaraju et al*. It is introduced with a primary goal of boosting confidence in applying neural networks – making it possible for visual analysis on misclassified instances for detecting discrepancies. By "producing 'visual explanations' for decisions from large class of CNN-based models, making them more transparent", Grad-CAM helps people better understand a wide range of tasks, including image classification, image captioning, and visual question answering models, etc. (Selvaraju et al.).

Briefly summarizing the working process of Grad-CAM (*see Figure 2*): given a picture and a class as input, Grad-CAM forward-propagates the image through the network model to get raw class scores before the Softmax layer (Selvaraju et al.). A gradient signal with only the inputted class set to 1 and others to 0 is then back-propagated to the rectified Conv feature maps – where coarse localization is calculated and a heatmap is generated (Selvaraju et al.). Finally, the pointwise multiplications of this heatmap and guided backpropagation produces Guided Grad-CAM visualizations (Selvaraju et al.).

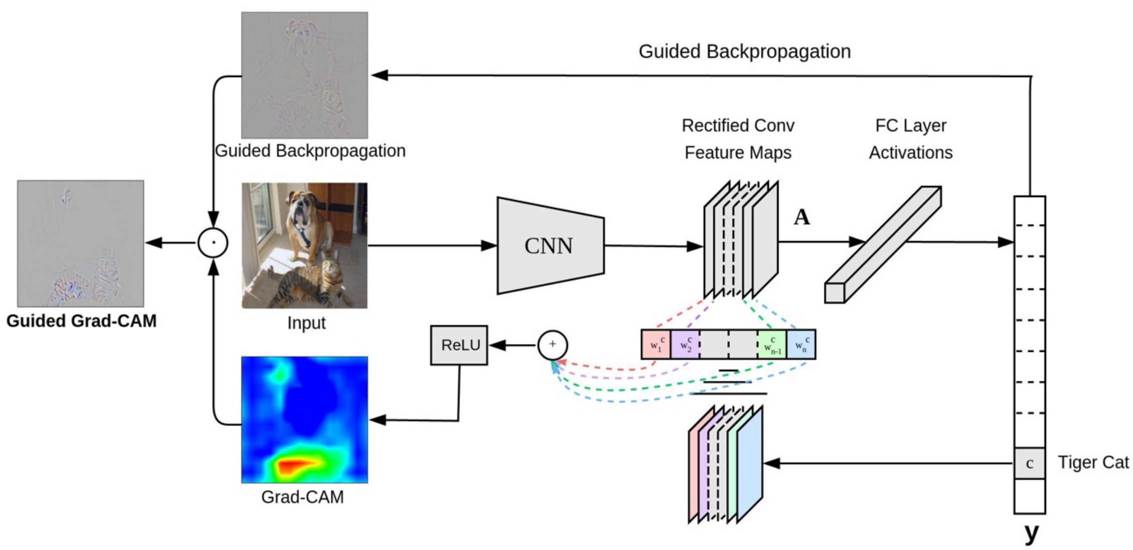


Figure 2: Mohamed Chetoui. "Grad-CAM Overview". *Medium*, Mohamed Chetoui.

3. Experiment

3.1. Dataset

The stock index NIFTY 100 is specially chosen for this study. NIFTY 100 is a stock index in India's National Stock Exchange and represents the major sectors of the country's economy. This index is chosen after careful investigation into the condition of its available dataset. Compared to many other datasets on financial markets, NIFTY 100 stands out by its rather complete and integral structure.

Made available by *Kaggle Competition*, the NIFTY 100 dataset covers abundant historical intraday minute-level transactions, ranging from January 2, 2017 to January 1, 2021. There are in total 988 days in this dataset. Trade information on this dataset include opening, closing, high, low prices as well as the transaction volume corresponding to each minute trade. The next section shows extracted information from our training dataset.

3.1.1. Stock Data in Time Series Representation

Transaction data of NIFTY 100 from January 2, 2017 to January 1, 2021 were obtained. Figure 3 is a time series representation of the *closing price* of the data throughout this period. Here, one interesting insight observed is how trading has been especially volatile since the coronavirus pandemic in the early 2020. Although the stock price has been increasing throughout those past years, the extreme price drop at the beginning of the pandemic illustrates the uncertainty associated with stock trading and the exceptional difficulty to predicting price movements.

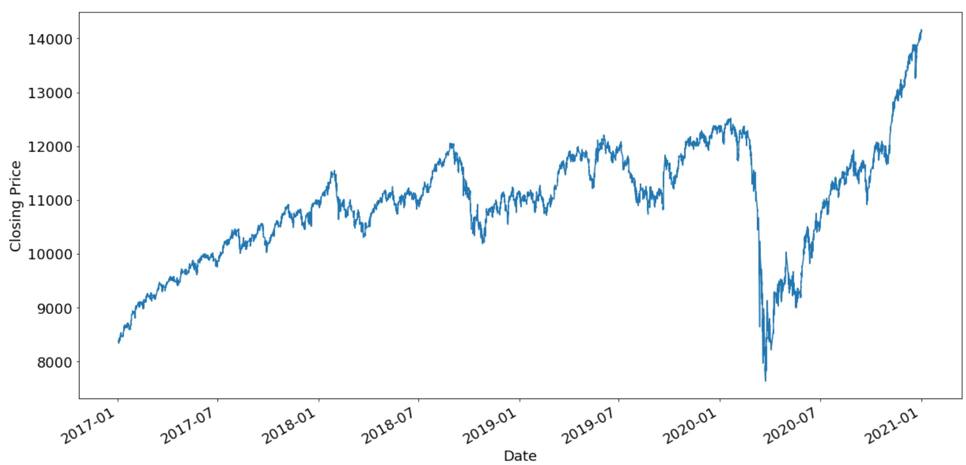


Figure 3: Daily Closing Price of *NIFTY 100* from 2017-01-02 to 2021-01-01.

The focus of our prediction is to explore whether it is possible to forecast if an index price increases or decreases thorough a day just by looking at the "first one hour of trading". As an example, the below Figure 4 represents the closing price of *NIFTY 100* on January 1, 2021. Here, we observe that the index price ended up Net-Positive that day. However, inspecting closely, the price decreased during the first hours of trading and then rose strongly later that day. Again, this observation too indicates that our prediction task is very difficult for a human without much experience in the market to accurately perform.

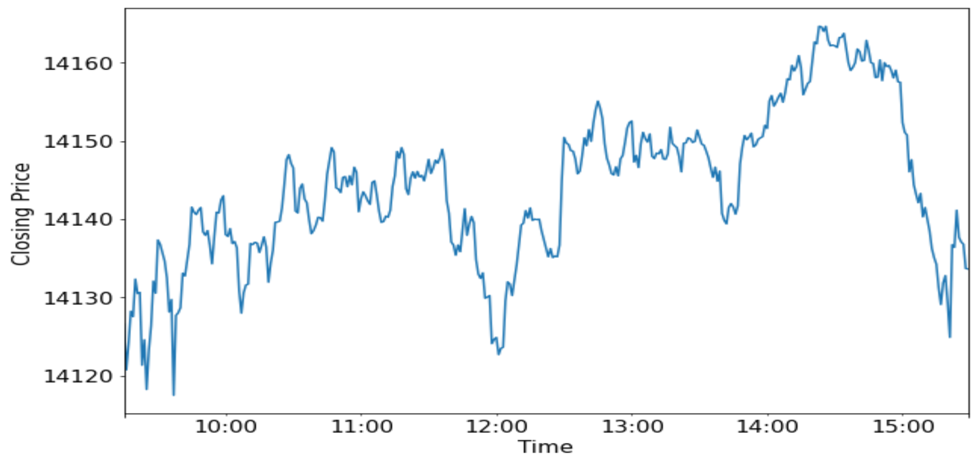


Figure 4: Minute-level Closing Price of *NIFTY 100* on 2021-01-01 - Full Market Hours.

3.1.2. Stock Price Data in Image Representation

As explained previously, in order to approach our *price change* prediction goal as an image classification task exploiting the CNN model, we leveraged the technique Gramian Angular Field to turn these time series representations into image data.

Continuing with the example we used for demonstration above, Figure 5 is a time series chart displaying the minute-level closing price of *NIFTY 100* during the first hour of trading (from 9:15 – 10:14 AM local) on January 1, 2021. This time series data totaling an hour (with 60 data points) corresponds to one image representation we are using as input to our CNN model.

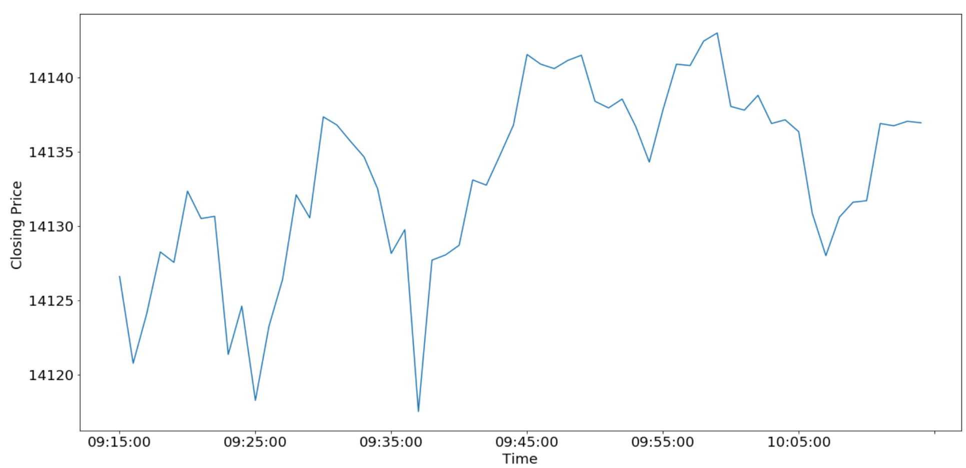


Figure 5: Minute-level Closing Price of *NIFTY 100* on 2021-01-01 – First Market Hour.

The following is the transformed coordinate data by Gramian Angular Field from *stock price change* on January 1, 2021, along with Figure 6 showing the actual image used for modeling converted from these polar coordinates. In total, 988 converted images for each of the 988 days of time series data are obtained.

```
array([[[ 0.,  0.,  2., ...,  0.,  0.,  0.],
        [ 0.,  0.,  2., ...,  0.,  0.,  0.],
        [-2., -2.,  0., ..., -2., -2., -2.],
        ...,
        [ 0.,  0.,  2., ...,  0.,  0.,  0.],
        [ 0.,  0.,  2., ...,  0.,  0.,  0.],
       [ 0.,  0.,  2., ...,  0.,  0.,  0.]]])
```

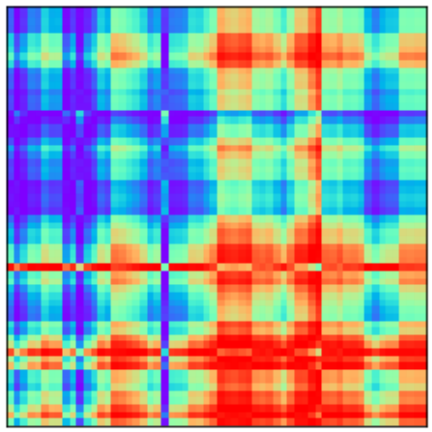


Figure 6: GAF for *NIFTY 100*'s Closing Prices (First Hour of Market Exchange on 2021-01-01)

3.2. Deep Learning

3.2.1 Market Prediction with CNN Model using FastAI Library

We used FastAI, a PyTorch-based deep learning library, to build the neural network. This enables us to figure out the relationship between input features and find hidden relationships within them. The input data is an image dataset with labels -- converted from time series with Gramian Angular Field as described in the previous section.

The entire dataset of 988 days (image instances) was divided into training and validation sets, with a 20% validation ratio. Our training procedure followed the following steps.

1. Create a baseline model, i.e., ResNet-34.
2. Find the optimal learning rate for the initial layers where the numerical gradients are minimized.
3. Train the model with the learning rate found in (1) with 10 epochs.

4. Unfreeze the model and find the optimal learning rate for all the layers where the numerical gradients are minimized.
5. Train the model with the learning rate found in (3) with 10 epochs.

For the CNN network, the pretrained ResNET-34 is utilized as the bottom layers. We added [1024, 2] dense layers on top and a simple linear activation node for the final regression as a custom head (See Table 1). Table 1 below shows the architecture of the top layers of the model. For the loss function, our final model utilizes Cross-Entropy loss, and for model performance measuring metric, we leverage the use of accuracy scores.

Table 1: Top Layers Summary on CNN Learner

Sequential
(0): AdaptiveConcatPool2d
(ap): AdaptiveAvgPool2d(output_size=1)
(mp): AdaptiveMaxPool2d(output_size=1)
(1): Flatten()
(2): Linear(in_features=1024, out_features=2, bias=True)

3.2.2 Model Performance

After the training procedure described above, the CNN model constantly achieves an accuracy score around 62% on the validation set. Figure 7 below shows the confusion matrix of our final model. We can observe that the model works quite equally for both of the two classes with the false positive rate being a little bit higher than the false negative rate.

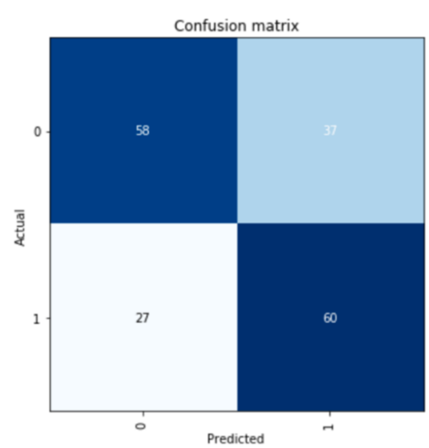


Figure 7: Confusion Matrix on implemented CNN Model

Figures below show sample instances from our validation results, where class “1” means the index price went up that given day and the class “0” indicates the opposite.

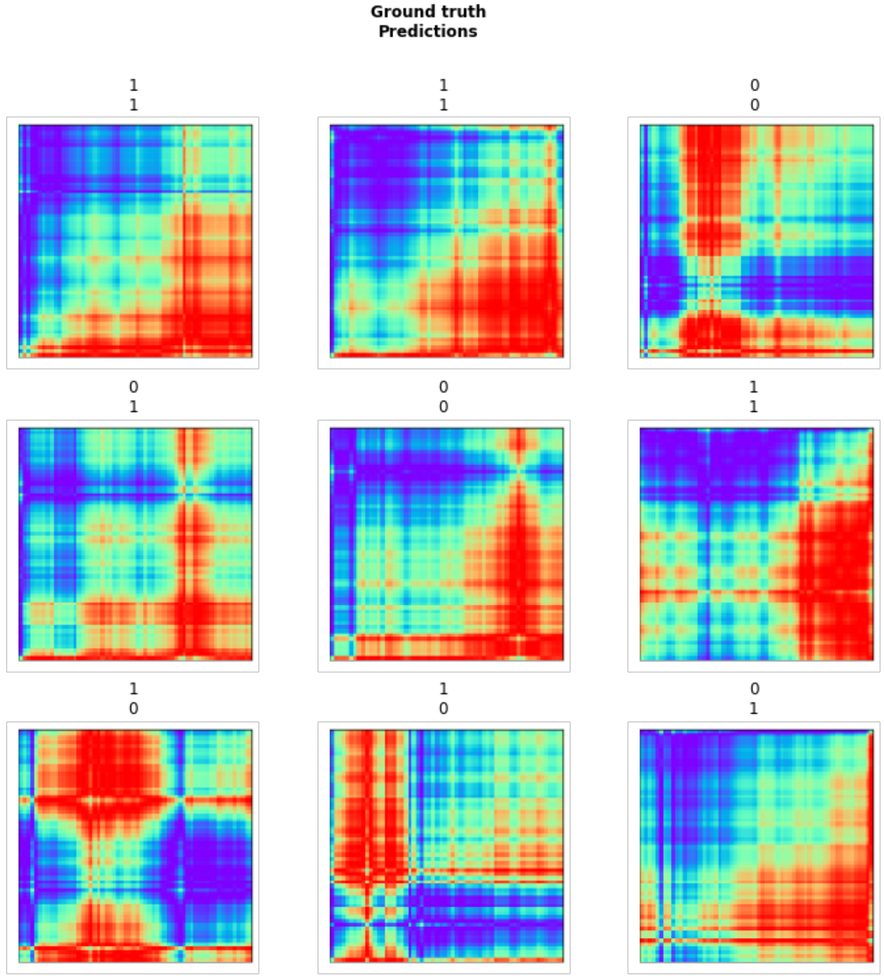


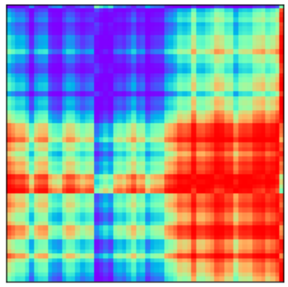
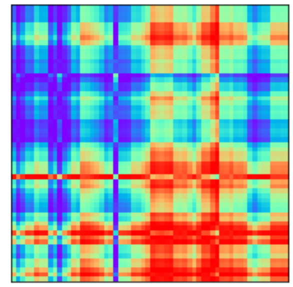
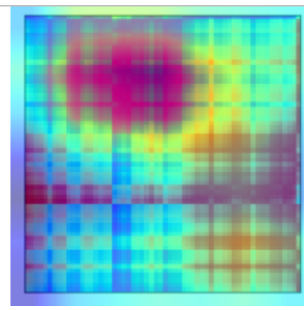
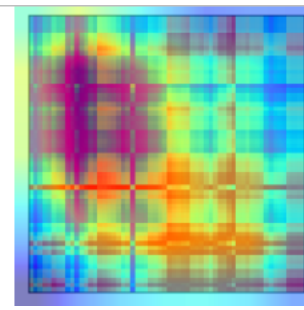
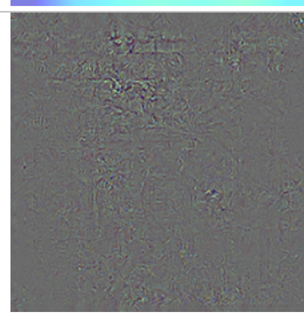
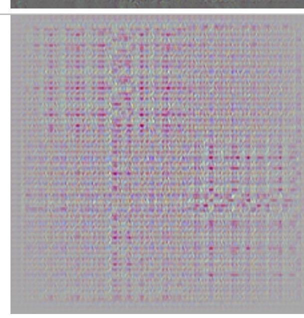
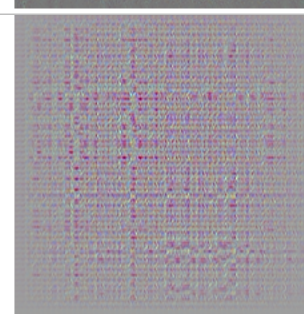
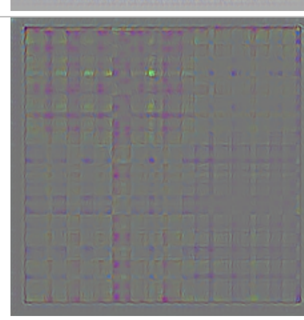
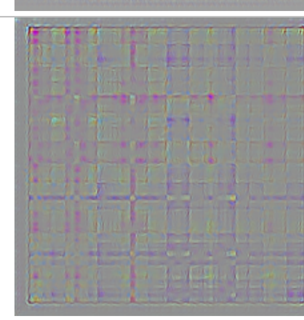
Figure 8: Training Results Sample

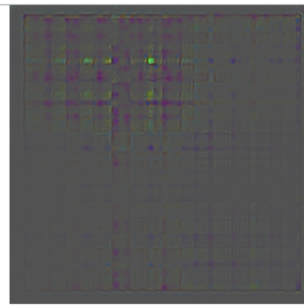
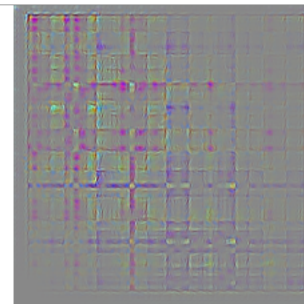
3.3 Grad-CAM inspection

This section summarizes results generated by applying the Grad-CAM Algorithm over our Gramian Angular Field converted time series data.

Before discussion on developed results, it is important to note that results shown below are made available by leveraging third-party Grad-CAM API instead of using Grad-CAM algorithm we trained ourselves. This interface is favored over ours due to it enabling additional visualizations for inspection – that include Guided Grad-CAM, gradients by vanilla backpropagation, gradients by guided backpropagation, and gradients by deconvnet. Trained on a different image database but sharing the same internal architecture of ResNet-34, we specify the target layer, i.e., *layer4* in the algorithm for visualization. For there are only two classes in our prediction task, outputs on only the *top 1* class of two sample images are generated.

Table 2: Visual Explanation from Various Algorithms

Predicted Class = 1	GAF Image: 2017-01-03	GAF Image: 2021-01-01
Original Image		
Grad-CAM		
Vanilla Backpropagation		
DeconvNet		
Guided Backpropagation		

Guided Grad-CAM		

Brief description of each visual explanation:

- Grad-CAM – "uses the class-specific gradient information flowing into the final convolutional layer of a CNN to produce a coarse localization map of the important regions in the image" (Selvaraju et al.).
- Vanilla Backpropagation – or Gradients, "commonly referred to as saliency maps' or 'backpropagation" (Draelos). It is a "visualization of an image in which the most salient/most important pixels are highlighted" (Draelos).
- DeconvNets – "DeconvNets are the same as the "Gradients" approach except for a difference in backpropagation through the ReLU nonlinearity" (Draelos).
- Guided Backpropagation – or Guided Saliency, "combines vanilla backpropagation and DeconvNets when handling the ReLU nonlinearity" (Draelos).
- Guided Grad-CAM – "This is an element-wise product of GradCAM with Guided Backpropagation" (Draelos).

While a first look at the generated Grad-CAM results may seem confusing, but it is important to notice how in the shown sample images -- the visual explanation maps seem to all be suggesting that information towards the top-left corner is comparably more important than that of the rest. This can be seen from patterns exhibited in results generated by Vanilla Backpropagation, DeconvNet, and Guided Backpropagation. The same is even more apparent from that in Grad-CAM: judging by the colors from heatmap mapped on top of the gramian angular fields, the *red* color -- signifying "visualizing 'the most important information'" – are both present towards the top-left corner. Guided Grad-CAM shows similar results with more enhanced presence.

To explain this result, we recall the key characteristic of Gramian Angular Field – it preserves temporal dependencies of time series as the position moves from the top-left corner to the bottom-right. Keeping this in mind while further inspecting visual explanations on all generated images, it is discovered that areas spanning the entire middle portion (from left to right) of Gramian Angular Fields seem to be most frequently highlighted. In relation to our original time series, this means that trades that happen during the middle of the hour of the trading period gives comparably more importance on predicting price "increase" or "decrease" than the rest of the daily entries.

4. Conclusion

Machine learning implementations in areas from pattern learning to target predictions have been at the center of computational research and development throughout the past decade. These mechanisms for label forecasting have today arrived at an unprecedented height with outstanding performances. The technology industry adapts machine learning practices in applications from hardware to software; The manufacturing industry has automated manufacturers made available by AI implants; The entertainment and sports industry take advantage of information that was never available before the recommending power of machine learning; And the healthcare industry leverages recognition abilities powered by these algorithms for early diagnoses and therapy provisions, etc.

Enabled by more mature hardware computational power, and fostered by the immense amount of information collected, artificial intelligence has expanded its learning power to applications on data larger than ever before. Developed with the special strength on pattern learning in audios, images and videos, deep neural networks has proven its powerful recognition power across these big data. However, with the commanding prediction potential comes along these deep learning algorithms' un-explainability.

Previously with traditional machine learning algorithm, models' prediction process are easily traced and tracked; However, due to the intricated architectural nature of deep learning models, this is no longer the case. Users commonly find it difficult to interpret how neural networks make their outputs, and this has therefore led to an issue of trust in systems like such. In sectors where interpretability of model learning progress is important, deep learning models cannot be deployed despite their guaranteed satisfactory performances. A representative example of such is the financial industry – an example of "implementing decision-making algorithms that cannot be understood on sensitive banking or market data" shall explain users' resistance on these practices.

This research project has therefore been structured specially to address this issue of "un-explainability" in deep learning models, i.e., CNN models in particular, and employs novel computational approaches in an attempt to introduce trust on using neural networks in the stock market prediction.

This project starts off with data pre-processing on stock data. Binary labels of "1" signifying closing price increase, and "0" representing the opposite were engineered as prediction targets. These time series data are then converted via use of Gramian Angular Field into image coordinates, and visualized for FastAI recognition task. Deployment of Grad-CAM is then introduced for visual inspections on the CNN model prediction process as our primary research output.

In particular, application of the Grad-CAM algorithm along with multiple additional visual explanation techniques including: Vanilla Backpropagation, DeconvNet, Guided Backpropagation and Guided Grad-CAM have been able to provide us with extra and novel insights into traditionally time-series represented stock-market data. In specific, we concluded that stock closing prices during the middle of trading periods during the day is more important than prices closer to market beginning or even market closing in predicting stock price increase. This is suggested by inspecting on pixel areas the Grad-CAM algorithm recognized and highlighted as significant in CNN model prediction of *label 1*, i.e. market price increase. This insight can be rather counter-intuitive to most financial investors, as one might expect the market condition closer to the end of a day's trading period being more important in drawing a conclusion on price change. However, given our best performing model's predictions, it is suggested of otherwise.

5. Discussion

Summarizing our work in general, while we have completed all above tasks as we have proposed, we also recognize and acknowledge that there are room for improvements and developments, we thereby describe these as follows:

Deep learning application in this project is structured purposefully as an image classification task, which is why we leveraged Gramian Angular Field for time series to image conversions. This approach is closely followed after prior research conducted on similar investigations, but none of these former studies took the additional step to apply Grad-CAM for visual explanations. It is discussed by us, researchers, that an alternative approach towards explaining the CNN model prediction may be suggested. This is because that images generated by Gramian Angular Field do not possess significant meanings, nor does there exist apparent objects for recognition. These two reasons combined makes explanations on results generated by Grad-CAM very difficult -- When mapping an image with its visual explanations, it is difficult to pinpoint the localization on the Gramian Angular Field. Going forward, this therefore makes it challenging to connect visual explanations with the financial market context. Founding on these discoveries and reflections, we believe it is appropriate for us to conclude that adopting explanatory techniques other than visuals is suggested.

Secondly, we believe that there is room for achievable improvements with implementing our CNN model. Given the limited research period we have, our best performing model has an accuracy score of 62%. Although given the highly unpredictable nature of the stock market, this performance may be appreciated in various cases, we also believe that the algorithm can be further enhanced to achive even better results. In specific, some potential ways to accomplish this can include:

- Changing the base model from ResNet-34 to other architectures.
- Further tuning model hyper-parameters.
- Switching Loss Function from Cross-Entropy Loss to other functions for optimization.
- Employing other model structures such as combining CNN with RSTM, etc.

Last but not least, we also believe that explainability of financial data in our problem setting may be better enhanced by improving our Grad-CAM implementation itself as well. This can be done by applying more suitable training on the Grad-CAM so it fits our training data. Additionally, although we are able to propose an explanation of which (or what kind of) stock market data serves an important role in CNN model learning with help of visuals, this conclusion may however, vary widely given different problem settings. Currently, with this project's setup, conclusions are drawn upon visual inspections, and therefore might not be as precise than a problem setup of scientific investigations. A solution to this is therefore proposed to be development of computable scores for justifying the importance and effect of Grad-CAM's application.

6. References

Bai, Chuan. "Image Recognition vs Other Techniques in Predicting the Financial Market." *Medium*, Towards Data Science, 30 Mar. 2020, towardsdatascience.com/image recognition-vs-other-techniques-in-predicting-the-financial-market-55548d4cda4.

Chetoui, Mohamed. "Grad-CAM Overview". *Medium*, Mohamed Chetoui, 15 Mar. 2019, https://medium.com/@mohamedchetoui/grad-cam-gradient-weighted-class-activation-mapping-ffd72742243a

Chen, Jun-Hao, and Yun-Cheng Tsai. "Encoding Candlesticks as Images for Pattern Classification using Convolutional Neural Networks." *Financ Innov 6, 26 (2020)*. doi: 10.1186/s40854-020-00187-0

Draelos, Rachel. "CNN Heat Maps: Saliency/Backpropagation". *Glass Box*, 21 June. 2019, https://glassboxmedicine.com/2019/06/21/cnn-heat-maps-saliency-backpropagation/

kazuto1011. grad-cam-pytorch, Github Repository, https://github.com/kazuto1011/grad-cam-pytorch/tree/fd10ff7fc85ae064938531235a5dd3889ca46fed

Kenton, Will. "Closing Price". *Investopedia*, 22 Aug. 2019, https://www.investopedia.com/terms c/closingprice.asp

Kuepper, Justin. "Volatility Definition." *Investopedia*, 13 Mar. 2020, https://www.investopedia.com/terms/v/volatility.asp

Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D. and Batra, D., 2016. Grad-cam: Why did you say that?. arXiv preprint arXiv:1611.07450.

Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D. and Batra, D., 2016. Grad-cam: Visual Explanations from Deep Networks via Gradient-based Localization. arXiv preprint arXiv: 1610.02391.

Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034. 2013 Dec 20.

Vitry, Louis de. "Encoding Time Series as Images." *Medium,* Analytics Vidhya, 15 Oct. 2018, https://medium.com/analytics-vidhya/encoding-time-series-as-images-b043becbdbf3

Wang, Zhiguang, and Tim Oates. "Imaging Time-Series to Improve Classification and Imputation." *Proceedings of the 24th International Conference on Artificial Intelligence,* Pages 3939-3945

7. Appendix

A. Project Proposal

With big data collected at an exponential growing speed today, the automated decision power these information is capable of providing has been recognized and has since been at the forefront of technological developments -- especially in the area of artificial intelligence.

Throughout the past decade, hundreds of machine learning algorithms have been developed with distinct strengths and useful areas of application. However, these models are commonly introduced as traditional baseline models because they do not generalize or perform well on “large datasets” such as images, audios, or videos, etc. Nowadays, tools applied for pattern recognition are advancing more towards prevalent uses of deep learning models that learn and recognize complex underlying data patterns. While these models have proven outstanding abilities in making predictions based on complicated input data, the problem lies in these algorithms’ “un-explainability”. Regardless of the problem being solved, let it be image recognition, object detection, or involved predictions, given the overly complex nature of these deep neural networks, users have commonly found it hard to trust the works of these outcomes -- because their learning process is practically untrackable, therefore resulting in un-explainability of these final predictions. Inspired by this phenomenon, this proposal presents a project that sets to address this issue in machine learning.

Hoping to resolve actual concerns arising from real-life practices, this project revolves around deep learning applications within the financial market. It is undebatable that the stock market is one of the most unpredictable and difficult areas of study, yet investigation into product price and return have always been at the center in quantitative analysis. Stock data are most commonly represented as time series, and therefore have previously primarily been modelled with traditional time series algorithms or conventional baseline machine learning models. While RNNs and LSTMs are now more frequently used to predict these time series trends, this project intends to also include CNNs to provide a comprehensive study on the performances of image recognition versus other techniques in predicting the financial market.

As the financial market is changing all the time, this project projects to work not only on historical stock data but also real-time daily and minute level financial market data. Therefore, Yahoo Finance, as one of the most reliable stock exchange sources, is to be deployed for analyses. There is readily available PythonAPI which enables remote data access to the market data, and will be pulled and utilized as the data source for this research.

Table 1 below shows the ending five entries of aggregated price data on a daily basis for S&P 500 ETF Trust stock market data between June 2020 to December 2020, and Figure 1 is a Time Series representation of the closing price through this whole period.

Date	Open	High	Low	Close	Adj Close	Volume
2020-11-23	357.279999	358.820007	354.869995	357.459991	357.459991	63230600
2020-11-24	360.209991	363.809998	359.290009	363.220001	363.220001	62415900
2020-11-25	363.130005	363.160004	361.480011	362.660004	362.660004	45330900
2020-11-27	363.839996	364.179993	362.579987	363.670013	363.670013	28514100
2020-11-30	362.829987	363.119995	359.170013	362.059998	362.059998	83872700

Table1: S&P 500 ETF Trust stock data pulled from *Yahoo! Finance*

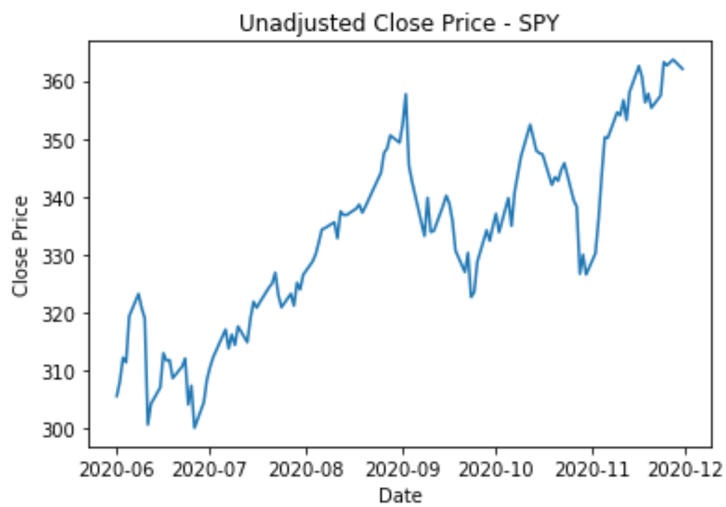


Figure1: S&P 500 ETF Trust stock data closing price plot (2020-06-01 -- 2020-12-01)

In specific, the volatility of a specific market during a specific time period will be inspected, and therefore this problem for prediction is purposefully structured as a regression-based prediction. The focus for this project will be image recognition based regression -- that is using CNN networks for time series prediction. Time series will be converted to image based polar coordinate relationships to assemble this approach as an image recognition task. Meta-Labelling technique will be applied to these time series in order to provide the outcomes of stock forecasting labels of "sell" or "buy" for classification. Similar research has shown that CNNs are able to achieve satisfying accuracy scores, yet none have explained the reasoning behind this. While this shall not be too big of a surprise, this project sets to address this particular problem by applying the Grad-CAM algorithm to this built CNN network. By inspecting the class-activation maps generated by this technique, and mapping them to the convolutional architecture, this project aims to study why the network made the prediction it did, as well as how parts in these images might have contributed to making the correct or incorrect decisions.

This project will require all members in the group to truly grasp model details regarding CNN models, and at the same time solidly understand conceptual and implemental components of the Grad-CAM algorithm. The outcome of this project is proposed to be a written paper with thorough descriptions on algorithm implementation details as well as justifications for model performances.

B. Initial Attempts to Use Different Datasets and Measurements such as Volatility

We were initially using Tesla Inc.'s stock (TSLA) obtained with the finance API AlphaAdvantage for the same purpose. However, this dataset contains many null values and many of the data points of specific minutes that we hope to manipulate on were unavailable. After a number of attempts with this dataset, we were finally unable to achieve a decent CNN model accuracy with it and decided to employ other datasets. After leaving this AlphaAdvantage dataset, there had been numerous attempts to achieve a decent model accuracy with different stock datasets from various sources, before eventually finding the NIFTY 100 dataset that we are using in this project.

We also initially used stock volatility as a measurement metric for this study, referencing to the prior study by *Bai*. We attempted to converting volatility time series data into image instance using GAF and built a deep learning model based on them. However, this attempt also did not result in successful prediction results, thus we eventually shifted our focus to recognition on pure stock prices instead of stock volatility.