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 $\tilde{X}$ 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:

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

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.

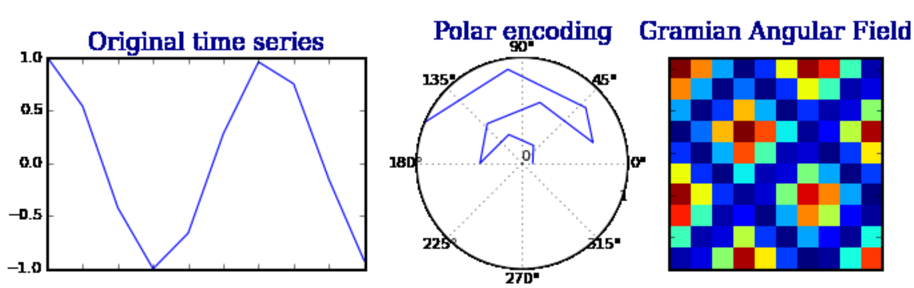


Figure 1: Various steps of the Gramian Angular Field Conversion (Vitry)

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 the 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 backpropagated 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.).

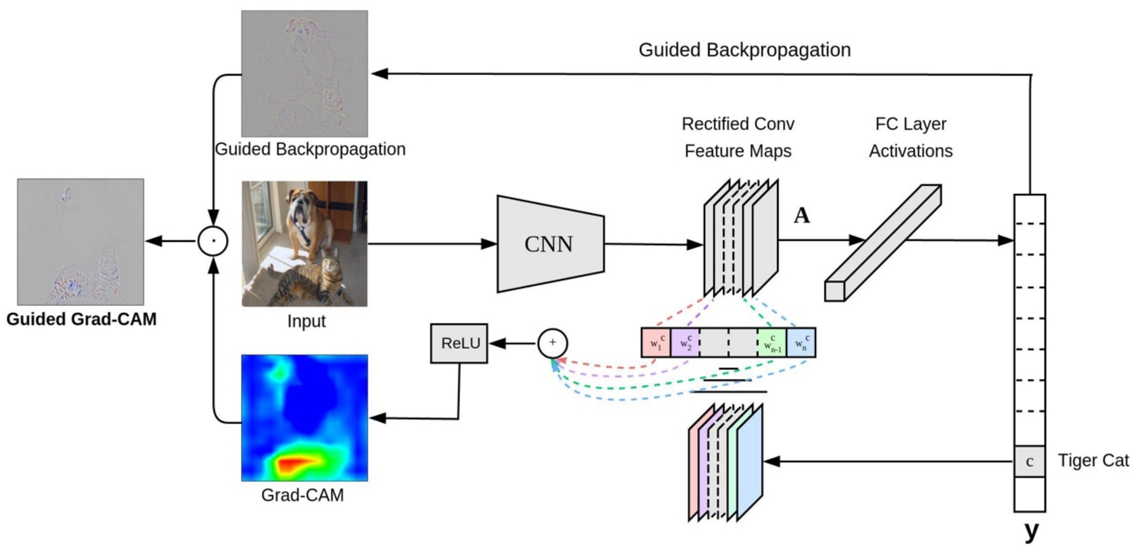


Figure 2: Mohamed Chetoui. "Grad-CAM Overview". *Medium*, Mohamed Chetoui.

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 following is a sample excerpt 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 2 is a time series representation of the *closing price* of the data throughout this period. Here, one thing you can easily observe is that the 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 predict the price movements.

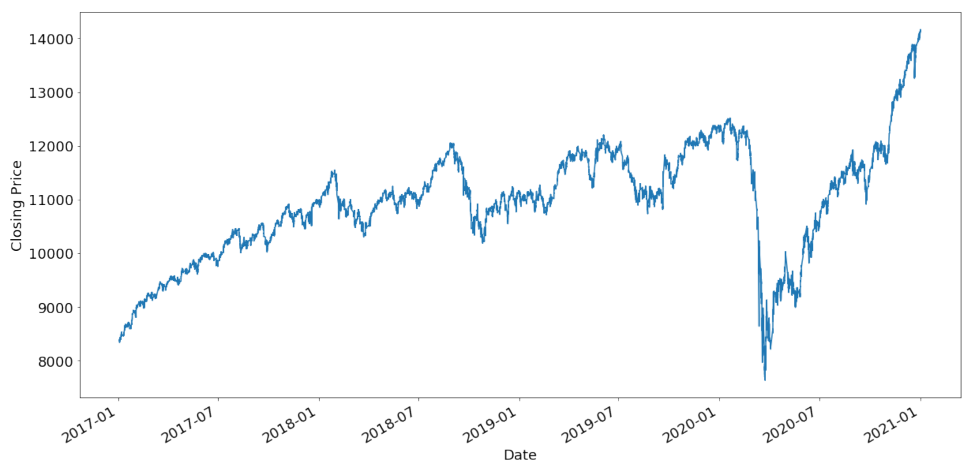


Figure 2: Daily Closing Price of *NIFTY 100* from 2017-01-02 to 2021-01-01.

The focus of our prediction is to explore whether it's possible to forecast if the index price increases or decreases thorough a day just by looking at the first one hour of trading. The below Figure 3 represents the closing price of *NIFTY 100* on January 1, 2021. Here, we can observe that the index price ended up net-positive that day. However, inspecting closely, the price decreased during the first hours of trading that day and then the price later rose strongly. Again, this observation too indicates that our prediction task is very difficult for a human without much experience in the market to accurately perform.

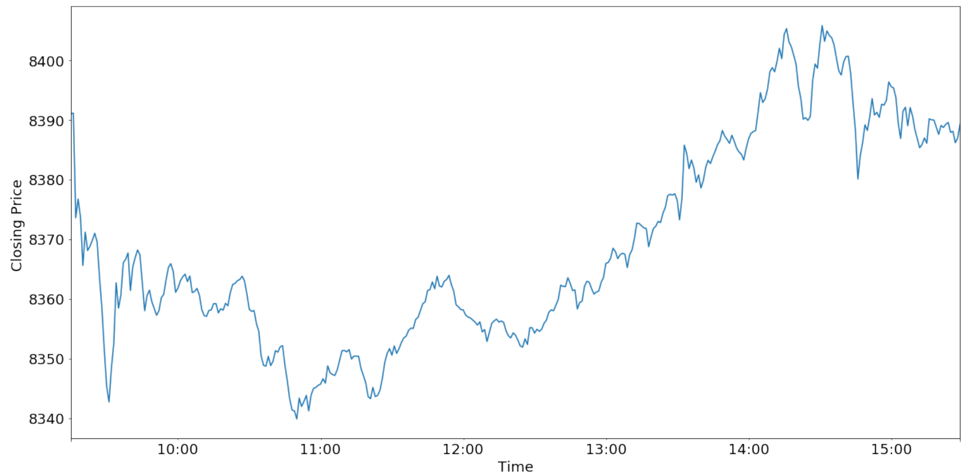


Figure 3: Minute-level Closing Price of *NIFTY 100* on 2021-01-01 - Full Market Hours.

3.1.2. Stock Volatility 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, the following is the time series chart representing the minute-level closing price of *NIFTY 100* during the first hour of trading (from 9:15 – 10:15 AM local) on January 1, 2021. These time series data of an hour with 60 data points in total corresponds to one image representation we used as an input to our CNN model.

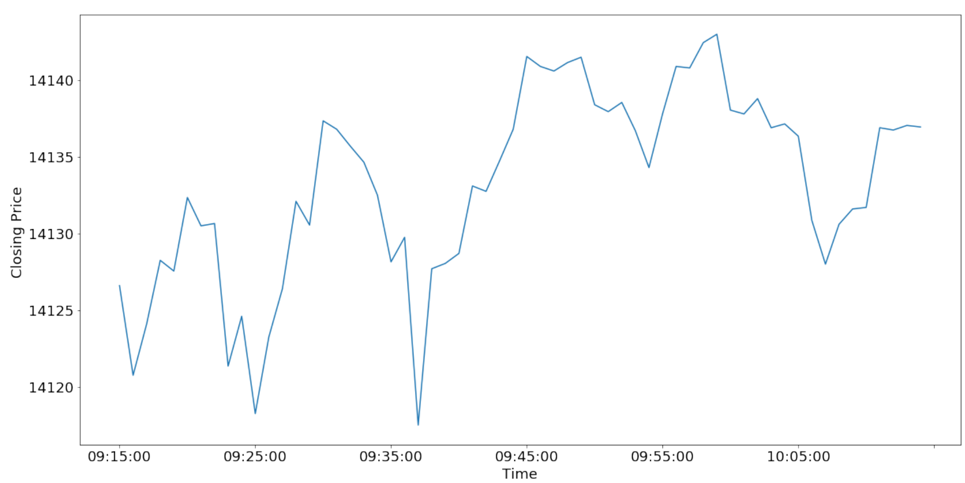


Figure 4: Minute-level Closing Price of *NIFTY 100* on 2021-01-01 – First Market Hour.

The following is the transformed coordinate data by Gramian Angular Field from *stock price change* on January 1, 2021, along with Figure 5 showing the actual image used for modeling converted from these polar coordinates.

```
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.]]])
```

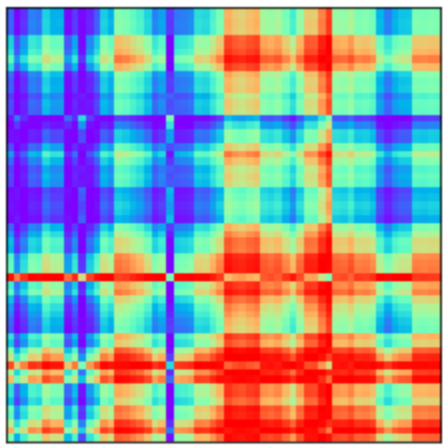


Figure 5: Gramian Angular Field Image Representation of *NIFTY 100*'s Closing Prices during the First Hour of Market Exchange on 2021-01-01

There are in total 988 converted images for each of the 988 days of time series data obtained.

3.2. Deep Learning

3.2.1 Market Prediction with CNN Model (FastAI)

We used FastAI as a PyTorch-based deep learning library to build the neural network, which is able to figure out the relationship between input features and find hidden relationship with them.

The input data is an image dataset with labels, which is converted from time series with Gramian Angular Field algorithm.

For the CNN network, ResNET34 is utilized as the bottom layers, a [1024, 2] dense layers on top, and use a simple linear activation node for the final regression.

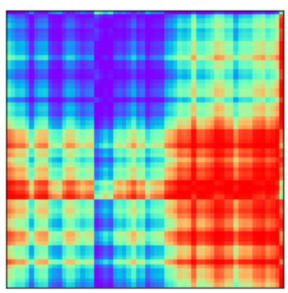
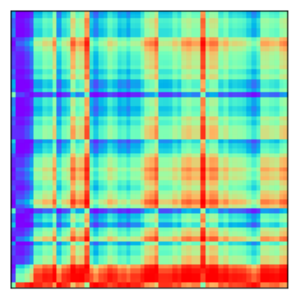
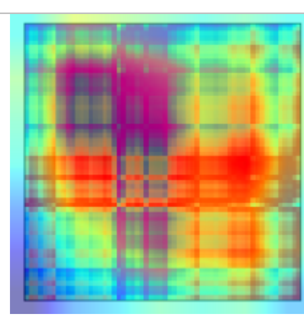
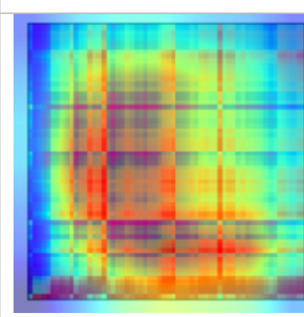
```
learner = cnn_learner(data,
                      models.resnet34,
                      pretrained=True,
                      custom_head=head,
                     metrics=[error_rate, accuracy]
                     ).to_fp16()
learner.model[1]
```

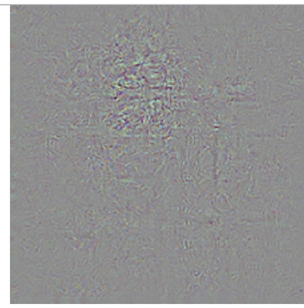
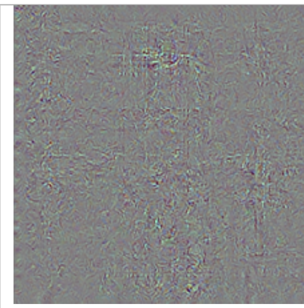
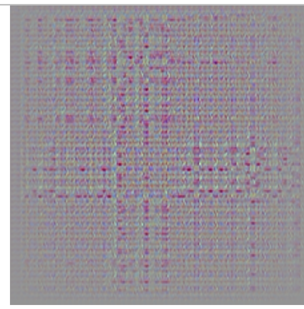
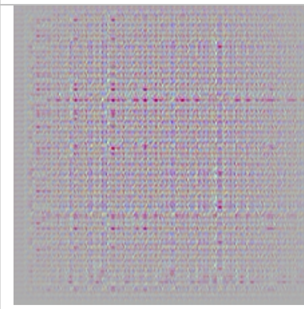
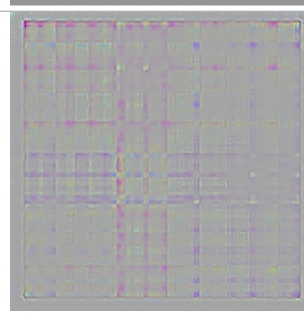
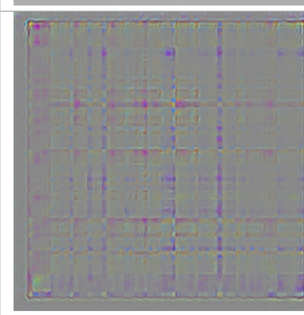
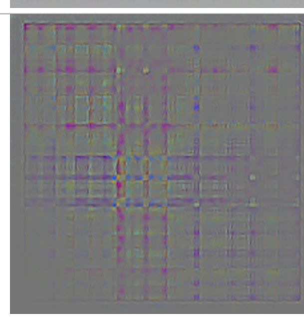
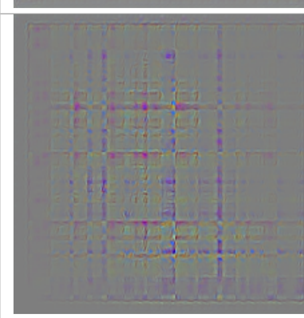
```
Sequential(
  (0): AdaptiveConcatPool2d(
    (ap): AdaptiveAvgPool2d(output_size=1)
    (mp): AdaptiveMaxPool2d(output_size=1)
  )
  (1): Flatten()
  (2): Linear(in_features=1024, out_features=2, bias=True)
)
```

Figure 6: Our CNN Learner

3.3 Grad-CAM inspection

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

	Predicted Class = 1	Predicted Class = 1
Original Image	GAF Image: 2017-01-03 	GAF Image: 2020-12-31 
Grad-CAM		

Vanilla Backpropagation		
Deconvnet		
Guided Backpropagation		
Guided Grad-CAM		

4. Discussion

[Analysis on Grad-CAM Results]

One room for improvement is the CNN model's accuracy. Our final model achieved around 62 percent accuracy for the two-class classification task. The figure below shows the confusion matrix of final model. Considering the highly unpredictable nature of the stock market, we believe this performance is quite solid and certainly superior to predictions made by regular humans. Yet, it is also true that the model can accomplish an even better performance with different approaches. Some potential ways that can be considered for improvements are changing the base model from ResNet34, further tuning parameters, and employing other model structure than CNN such as CNN + RSTM.

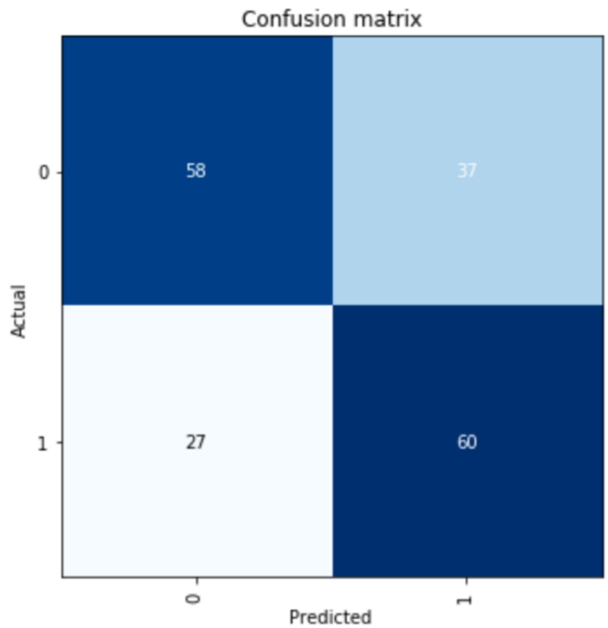


Figure 7: The Final Model Confusion Matrix

In addition, one thing we couldn't achieve during this study was to decode GAF images. Although the image conversion algorithm contributed to a better prediction of the stock price movements, since the GAF images consist of color patterns that we cannot intuitively understand, even the use of explainable AI technique like Grad-CAM did not fully allow us to interpret why a specific GAF representation resulted in a correct (or wrong) prediction by the CNN model. We used a public library to turn our time series data into image representations and it was very difficult for us to fully decode the innerworkings of the library in the given timeline for this capstone project. Of course, we attempted to understand how time series data are converted to polar coordinates data, then finally GAF image representations. Yet, we were not able to fully "explain" each of our image instances. In the future study, this facet should be addressed for the better understanding of the image recognition algorithm.

5. **Conclusion**

[connect to grad-cam]: what grad cam allows us to do

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

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.

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*

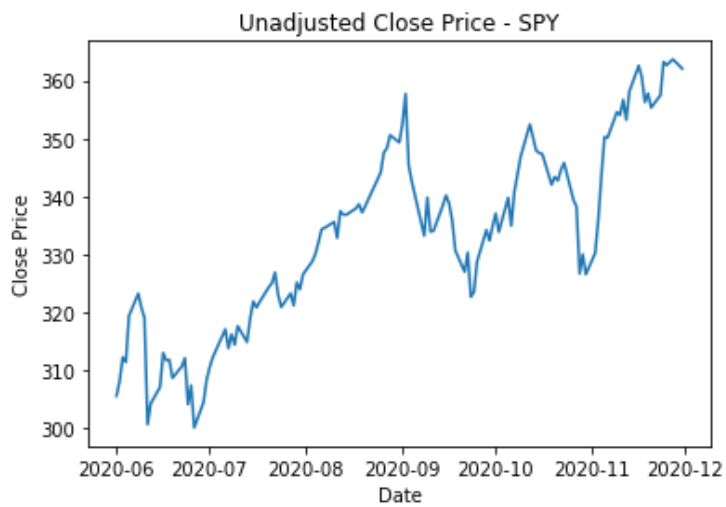


Figure1: S&P 500 ETF Trust stock data closing price plot (2020-06-01 -- 2020-12-01)

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

We were initially using Tesla Inc.'s stock (TSLA) obtained with the finance API AlphaAdvantage for the same purpose. However, this dataset contained quite a number of null values and many of the data points of specific minutes that we wanted to use 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 were 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.