LINKER: TJCT

Journal of Computing and Technology

Research Article **Open Access**

Spikes-Eye: Stock Market Price Spikes Detection in Philippine Stock Exchange Composite Index (PSEi) Using Deep Learning

Julie Ann B. Susa[1], Helcy D. Alon[2], Cherry Casuat[3]

(Department of Computer, College of Engineering and Architecture, Technological Institute of the Philippines, Philippines)[1]
(Graduate School, College of Architecture and Engineering, Technological Institute of the Philippines, Philippines)[2]
(Department of Computer, College of Engineering and Architecture, Technological Institute of the Philippines, Philippines)[3]

Corresponding author: e-mail address: jsusa.cpe@tip.edu.ph

ABSTRACT

The financial sector is the best way of making money because of its stability during some sort of economic turmoil and significant events such as a pandemic. Despite this solidity, values on the share market can not be entirely forecasted; rather, the force of chance plays a significant role in how investors spend their profit. Assessing these uptrend measures will be useful for potential stock market investors' references, as statistical data is being used to assess the likelihood of the direction in which the stock price will move. Using object recognition is a way to accurately recognize these upward movements since the background of the stock market stretches back at least 20 years at most. To assess uptrends, specifically the upward spikes of historical stock market charts of indices on the Philippines stock market, this research focuses on object recognition using transferred learning. Also, this analysis will use only the transferred learning of YoloV3 object recognition and will not recognize any other artifacts in the PSEi other than the upward spikes and clear upward trend price movement. To ensure consistency in the tests, the charts will have to be uniformly configured; the black background, the yellow positive candles, the red negative candles, and the light yellow borders for the wicks and the candles. Overall, in the fundamental principle of evaluating the upward spikes in the Philippines Stock Exchange, the project is reliable. The research will successfully assess the uptrend spikes of the images with an accuracy of 80 percent based on the 30-image test and accuracy of 93.5745 percent to 94.3409 percent based on the design training.

Keywords *– Deep Learning, Prediction, Psei, Spike, Stock Market, YoloV3,*

Introduction

The stock market is a forum in which the value is pushed up or down by buyers and sellers who exchange stocks. Stock prices are changing every day by market forces (Vlastakis & Markellos, 2012). Because of the law of supply and demand, these changes occur. The price moves up if more customers choose to purchase a stock (demand) than sell it (supply) (Torab, 2018). If, on the other hand, more consumers decided to sell a stock than purchase it, supply would be higher than demand and the price would decline (Macfarlan, 2016).

It's easy to grasp these processes. What is hard to understand is what allows people to purchase a specific market and sell some stock, keeping the stock market extremely volatile and inherently unpredictable (Marszk & Lechman, 2019). Consequently, with intention of getting a return, investors are still taking chances. As per their behavior, people have invested in the stock markets

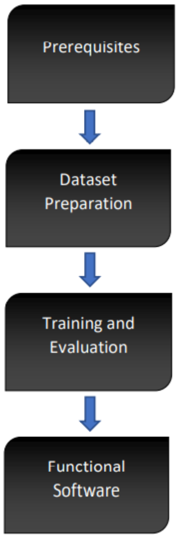
In exchange for their savings, they gamble their appetite and predict profits (Griffin et al., 2015). It is almost unlikely. Because of the volatility of variables that play a significant role in the prediction of stock prices, the market movement (Choudhry et al., 2016).

The stock market prediction has been a difficult field to estimate globally (Gandhmal & Kumar, 2019). There are many variables that domestically and universally influence the stock market. There is no precise procedure and concept for this purpose in the research published up to this period. Stock market prices are very hard to anticipate by using mathematical models (M et al., 2018). This is the price, if perceived to adjust for a short time due to news or media factors, differ in the long term due to firm circumstances and consumer demands and supply. However, a convolutional neural network model may provide powerful stock market predictive values from historical data (Hoseinzade & Haratizadeh, 2019). Generally, stock prediction methods are classified into three major groups, namely: technological research, basic analysis, and developmental analysis. A trading strategy is a most frequently used approach, which is based on the research analysis. It is possible to learn and interpret these statistics associated with the input data. Many external factors, however, such as the state of the business to which the shares belong, public order the share market pattern is influenced by universal and universal occurrences. For this reason, the time series on the stock market is typically a vibrational and dynamic operation (Anbalagan & Maheswari, 2015). At the same time, stock-market fluctuations are very serious. The data is, therefore, very noisy. Keeping the light is very essential when designing a prediction for the market.

This study focuses on object identification using transferred learning to determine uptrends, specifically uptrends of historic stock-market index charts on the Philippine Stock Exchange.

The determination of these spikes will provide insights into the potential movements of the stock price. Continuous upward spikes, for example, would appear to retain momentum and provide entry rates for investors where they can purchase stock shares and sell when the spike hits its saturation. This measure would not only evaluate the higher likelihood of maintaining momentum, however, a reference or a research guide may also be used, along with other indicators for future transactions. While upward spikes are considered to be accurate price indicators in the upper bracket, they do not provide for a definite future movement of stocks. It is within the discretion of the investor and his expertise that he should decide to engage in the purchasing of shares of that stock.

Methods



Prerequisites have included set-up and essential specifications of this report, such as Google Account, including access to Google Drive and Google Colab, a working computer/laptop with an internet browser, general knowledge in python programming, experience in developing an environment. Python and the import/installation of libraries, the demand for simple information stocks, and a secure internet connection.

Dataset preparation will resolve the characteristics of managing vast quantities of the picture and how to use it effectively just annotations.

Training and evaluation is the focal point of this report. In which the algorithm for Deep Learning will be used and it will employ translated learning.

Applying the model to a model after training and evaluation running program, which is fitted with a graphical user the analysis will end with the interface.

A. Dataset Preparation

Since the transmitted training is used, only around 300-500 are used.

Images are required, as opposed to deep learning, starting from a scratch that would take a few thousand checks, and iteration, guy. Screenshots are to be gathered and put accordingly, the layout of the YOLOv3 directory file.

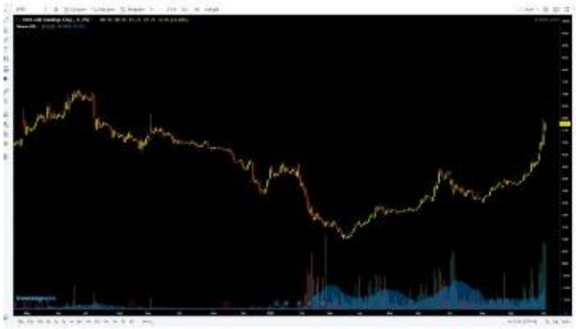


Fig b1: Sample screenshot of DITO stocks chart

Fig. b1 is a screenshot of 1920x1080 and is located. In line with the YOLOv3 architecture. There are 240 screenshots of various charts for this review. There are 60 screenshots in the verification folder in the train folder and that would reflect an 80 percent -20 percent split in the dataset.

- <project folder>
  - ➢ Train
    - * Images
    - *annotation
  - ➢ Validation
    - *images
    - *annotation

The < project folder > can have any name, but sub-folders can have any name. The strict classification as stated above has been followed within. This, this,
It will guarantee that the transferred learning h5 can be used right. The 240 screenshots for training are put in the /train/images while the 60 validation screenshots are put in the subfolder, Subfolder of /validation/images. The YOLOv3 calls for photos in
Format for JPG.

The photos should be annotated, choosing all possible artifacts in conjunction with the way the spikes will appear. LabelImg.exe is used for this review to annotate the spikes inside the screenshots. To preserve the quality of the dataset and generate acceptable results, rectangular annotations should be performed with precision.

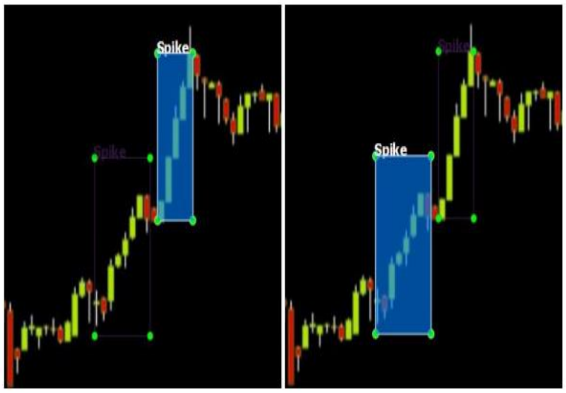


Fig b2: Proper “tight” annotation (left); Imprecise annotation (right)

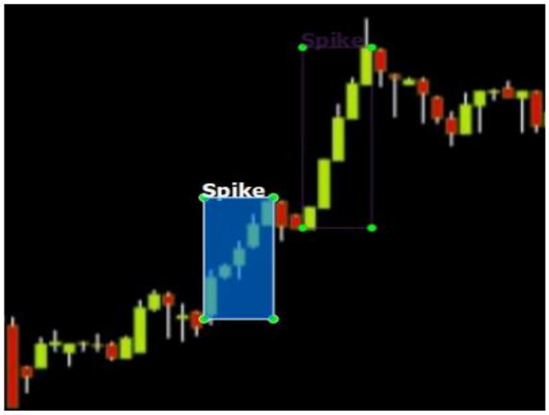


Fig b3: Corrected “tight” annotation

The annotation file must be saved in the /train/annotations and /validation/annotations subfolders in the XML format of PascalVOC.

B. Training

Ideally, the computer to be used as a GPU and is powerful enough to process an image within a decent length of time. A secure way to access such efficient GPU features is the Google Colab virtual machine. Since the analysis will be cloud-based, it is important to upload the prepared datasets to but still keeping the folder architecture, Google Drive. Python 3 is the programming language used, and the notebook should also be placed on Google Drive. It is vital to connect Google Drive and Google Colab and it can be accomplished with the following code:

```
from google.colab import drive
drive.mount('/content/gdrive')
```

After navigating to the < project folder >, the configuration of the required libraries and extensions becomes a critical phase in machine training.

```
!pip install opencv-python==4.1.2.30
!pip install keras==2.3.1 !pip install
!tensorflow==1.14.0 !pip install
!tensorflow-gpu==1.14.0 !pip install
!imageai --upgrade
```

Effective installations of libraries and extensions will be accompanied by the development of a pre-trained-yolov3.h5 file that can be found in [https://github.com/OlafenwaMoses/ImageAI/release s/downlo ad/essential-v4/pretrained-yolov3.h5] and downloaded to Google Drive using the!wget function.

The training session of the datasets using the pre-trained yolov3 h5 file will be executed using the following 6-liner. Make sure the patches and directory and file names return zero errors.

```
from imageai.Detection.Custom import
DetectionModelTrainer trainer =
DetectionModelTrainer()
trainer.setModelTypeAsYOLOv3()
trainer.setDataDirectory(data_directory=)
trainer.setTrainConfig(object_names_array=["S
pike"], batch_size=, num_experiments=,
train_from_pretrained_model=)
trainer.trainModel()
```

data_directory: set the path to the project folder *object_names_array:* the name of the annotated object

batch_size: the size of the batch per experiment/run *num_experiments:* the number of experiments the AI is preferred to undergo, the greater the number, the “smarter” the AI.

train_from_pretrained_model: the path to the pretrained yolov3 h5 model.

The training program can take a significant amount of time, even with just a few hundred data sets. In this research, when set up with 50 experiments, it took about seven hours for the training to end. A workout will produce a 'model' in the Google Drive model folder in an H5 file format as long as it has better accuracy than the previous one. A detection file system will be generated as well, in JSON format.

C. Evaluation

To identify the mean accuracy of the statistical average of the generated models, evaluations are required. Throughout this case, to test the generated models, the following 5-line code is used.

```
from imageai.Detection.Custom import
DetectionModelTrainer trainer =
DetectionModelTrainer()
trainer.setModelTypeAsYOLOv3()
trainer.setDataDirectory(data_directory=)
trainer.evaluateModel(model_path=,
json_path="", iou_threshold=0.5,
object_threshold=0.3, nms_threshold=0.5)
```

data_directory: set the path to the project folder *model_path:* the path to the model file (/models) *json_path:* path to the json file (/json)

Each model developed in the learning will generate an assessment with its associated average precision (mAP) percentage value. The closest it is to 100%, the better.

D. Model Selection

Using MS Excel or some other charting program, the accuracy is then correlated with its Fig. number (epoch). Accuracy is calculated by subtracting 100 percent of the loss and Val loss that will result in two-line charts. After observing the line map, the chosen model must have a 'fitting' behavior; the accuracy of the train and the accuracy of the validation should be upward. This model is also supposed to provide a high valued MAP.

E. Graphical User Interface (GUI) and functions

After regression models, the features of the cloud-based computer are no longer needed, so the h5 model file and the JSON file generated on Google Drive can be imported to the local desktop/laptop.

A new anaconda platform with the libraries and extensions listed above is to be developed and built. After that, a python file is generated using any anaconda IDE to construct a graphical user interface using three key functions: Image Detection, Video Detection, and Livestream Detection. The selected model of the h5 file and its accompanying configuration file for JSON.

For this specific study, with PyQt5 library and imageai detection library, the graphical user interface and its features are developed. To provide better data storage, CSV files should also be generated alongside its three primary functions if the project is to be used in combination with other applications or solutions. With the identification code, the detection codes are straight forward. Instruction packets, and the model path, and the JSON configuration path just need to be injected.

Results

A. Training and Validation Results

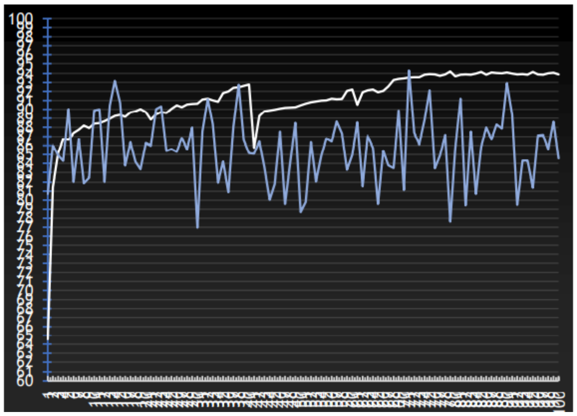
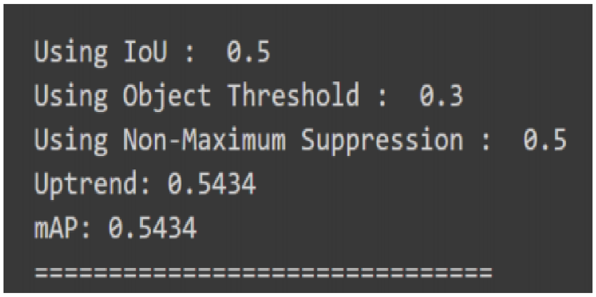


Fig. c1: Train Accuracy (white); Validation Accuracy (light blue)

In Fig. c1, throughout 100 experiments, there are three ‘fitting’ behavior; epoch 32, epoch 38, epoch 71. The best among the three is the 71st epoch with a training accuracy of 93.5745% and a validation accuracy of 94.3409%. Naturally, this 71st epoch is to be chosen as the model for the Deep Learning AI.

B. Evaluation of Model



```
Using IoU :  0.5
Using Object Threshold :  0.3
Using Non-Maximum Suppression :  0.5
Uptrend: 0.5434
mAP: 0.5434
===============================
```

Fig. c2: Evaluation result of the chosen model

For this particular model, its mAP is low at only 54.34%

C. Graphical Interface

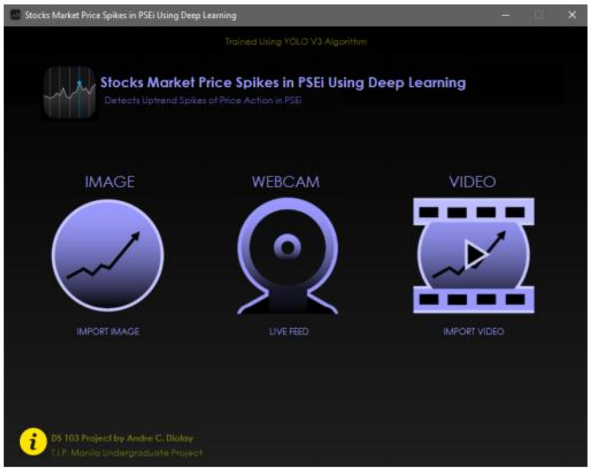

Stocks Market Price Spikes in PSEi Using Deep Learning
Trained Using YOLO V3 Algorithm
Stocks Market Price Spikes in PSEi Using Deep Learning
Detects Uptrend Spikes of Price Action in PSEi
IMAGE
WEBCAM
VIDEO
IMPORT IMAGE
LIVE FEED
IMPORT VIDEO
DS 103 Project by Andre C. Diolay
T.I.P. Manila Undergraduate Project

Fig. c3: Graphical User Interface

In Fig. c3, as a supplementary visual effect, the dark-themed graphical interface is compatible with the stock market table.

D. Testing Results (30 tests)

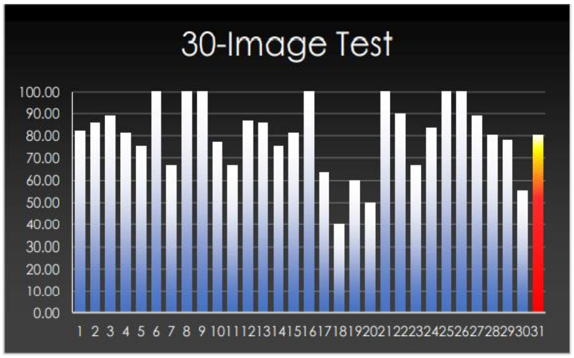


Fig. c4: 30-Image Test Accuracies and Average (31st column)

The 30-image test was performed using the software's image detection feature. The precision was determined by the number of spikes detected minus false-positive spikes detected over the total spikes current

$$Accuracy = \frac{detected\ spikes - false\ positives}{total\ spikes\ present} \times 100\%$$

In Fig. c4, the first 30 columns represent the accuracy of the 30 pictures, while the 31st reddish column reflects the average accuracy of all images, which is 80%. Image detection is processed in approximately 10 seconds.

Individual testing of information retrieval and live feed functionality has been checked and is working well with outputs in AVI video file format included with the previewed annotated items. The video detection production results in around 15 minutes for a 10-second video, while the live feed yields results.

A 5-second fast-forwarded video file is available in 5 minutes.

Conclusion and Future Works

The research will successfully assess the moving average spikes of the images with an accuracy of 80 percent based on the 30-image test and accuracy of 93.5745 percent to 94.3409 percent based on the model training. Overall, the project is credible in terms of the basic definition of the upward trend in the Philippines Stock Exchange. But since charts in the PSEi are modified in real-time, video detection and even live feed can be used to automate the entire process.

Acknowledgment

The author would like to express deep and sincere gratitude to his Data Science Class of 1st SEM SY20-21. Additionally to MR-SUAVE Lab and the Data Science Lab of the Technological Institute of the Philippines.

Authors Bio note

JULIE ANN B. SUA is a graduate of the Master's Degree in Engineering in Education major in Computer Engineering. She is currently an Assistant professor in the Department of Computer Engineering, College of Engineering and Architecture, Technological Institute of the Philippines. Her current research interests include Signal Processing and Embedded Systems.

HELCY D. ALON currently a Master of Engineering student from the Technological Institute of the Philippines. Her current research interests include Data Science and Embedded Systems.

CHERRY D. CASUAT currently a Doctor of Engineering student from the Technological Institute of the Philippines. She is currently an associate professor in the Department of Computer Engineering, College of Engineering and Architecture, Technological Institute of the Philippines. Her current research interests include Employability Signals and Machine Learning.

References

Vlastakis, N., & Markellos, R. (2012). Information demand and stock market volatility. *Journal Of Banking & Finance*, 36(6), 1808-1821. https://doi.org/10.1016/j.jbankfin.2012.02.007

Torab, E. (2018). A law or just a hypothesis? A critical review of supply and demand effect on the affordable residential markets in developing countries. *Alexandria Engineering Journal*, 57(4), 4081-4090. https://doi.org/10.1016/j.aej.2018.10.010

Macfarlan, S. (2016). Social Evolution: The Force of the Market. *Current Biology*, 26(16), R756-R758. https://doi.org/10.1016/j.cub.2016.07.041

Marszk, A., & Lechman, E. (2019). Determinants of the European Exchange-Traded Funds Markets Development. *Exchange-Traded Funds In Europe*, 121-163. https://doi.org/10.1016/b978-0-12-813639-3.00004-3

Griffin, P., Jaffe, A., Lont, D., & Dominguez-Faus, R. (2015). Science and the stock market: Investors' recognition of unburnable carbon. *Energy Economics*, 52, 1-12. https://doi.org/10.1016/j.eneco.2015.08.028

Choudhry, T., Papadimitriou, F., & Shabi, S. (2016). Stock market volatility and business cycle: Evidence from linear and nonlinear causality tests. *Journal Of Banking & Finance*, 66, 89-101. https://doi.org/10.1016/j.jbankfin.2016.02.005

Gandhmal, D., & Kumar, K. (2019). Systematic analysis and review of stock market prediction techniques. *Computer Science Review*, 34, 100190. https://doi.org/10.1016/j.cosrev.2019.08.001

M, H., E.A., G., Menon, V., & K.P., S. (2018). NSE Stock Market Prediction Using Deep-Learning Models. *Procedia Computer Science*, 132, 1351-1362. https://doi.org/10.1016/j.procs.2018.05.050

Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems With Applications*, 129, 273-285. https://doi.org/10.1016/j.eswa.2019.03.029

Anbalagan, T., & Maheswari, S. (2015). Classification and Prediction of Stock Market Index Based on Fuzzy Metagraph. *Procedia Computer Science*, 47, 214-221. https://doi.org/10.1016/j.procs.2015.03.200