

Article

Forecasting Agriculture Commodity Futures Prices with Convolutional Neural Networks with Application to Wheat Futures

Avi Thaker¹, Leo H. Chan^{2,*}  and Daniel Sonner¹

¹ Co-Founder, Tauroi Technologies, Pacifica, CA 94044, USA; avi@tauroi.com (A.T.); daniel@tauroi.com (D.S.)

² Department of Finance and Economics, Woodbury School of Business, Utah Valley University, Orem, UT 84058, USA

* Correspondence: lchan@uvu.edu

Abstract: In this paper, we utilize a machine learning model (the convolutional neural network) to analyze aerial images of winter hard red wheat planted areas and cloud coverage over the planted areas as a proxy for future yield forecasts. We trained our model to forecast the futures price 20 days ahead and provide recommendations for either a long or short position on wheat futures. Our method shows that achieving positive alpha within a short time window is possible if the algorithm and data choice are unique. However, the model's performance can deteriorate quickly if the input data become more easily available and/or the trading strategy becomes crowded, as was the case with the aerial imagery we utilized in this paper.

Keywords: machine learning; convolutional neural network; futures price forecasting; commodity futures

JEL Classification: G11; G13; G17



Citation: Thaker, Avi, Leo H. Chan, and Daniel Sonner. 2024. Forecasting Agriculture Commodity Futures Prices with Convolutional Neural Networks with Application to Wheat Futures. *Journal of Risk and Financial Management* 17: 143. <https://doi.org/10.3390/jrfm17040143>

Academic Editor: Thanasis Stengos

Received: 23 February 2024

Revised: 28 March 2024

Accepted: 29 March 2024

Published: 2 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Wheat is one of the most important cereal crops grown in many parts of the world. The price of wheat is influenced by two major factors: demand and supply. On the demand side, economic cycles and population growth both play major roles. On the supply side, total areas of crops planted, conditions, and geopolitical tensions all have major impacts on crop yields. Most agriculture futures price forecasts are based on econometric models using World Agriculture Supply and Demand Estimates (WASDE) projections (see Meyer 1998; Hoffman and Balagtas 1999; Goodwin and Schnepf 2000; Hoffman 2005; Hoffman et al. 2007; Adjemian 2012; Adjemian and Smith 2012; Hoffman et al. 2015; Hoffman and Meyer 2018; Isengildina-Massa and MacDonald 2009). Because the WASDE projections are widely used in the industry, the efficient market hypothesis (Fama 1970) suggests that it would be impossible to generate excess returns with WASDE data. In this paper, we explore an alternative method by using aerial imagery of the planted areas of a specific crop to forecast its future yield (and thus prices). We found that the use of an alternative method could yield positive returns in a short time window. However, the profitable trading algorithm could lose its advantage quickly if the data become more widely accessible.

With the advancements in deep learning architectures and computing powers, it is possible to train machine learning models with a large number of images efficiently and effectively so that they can capture hidden features that are difficult to detect by humans. In this paper, we utilize satellite imagery of wheat fields combined with weather imagery to generate a trading rule for wheat futures. The trading rule is a simple long or short position for 20 days at the beginning of the month (often correlated with rolls). If the model is correct, the trading rule will yield excess returns. Using data from 1984 to 2023, our model

produces excellent in-sample results (the training set), with an annual return of 24.12% unleveraged, and relatively modest results for out-of-sample results (the holdout set) until 2018. The performance of the model deteriorated rapidly after 2018, which coincides with the start of Airbus and other satellite imagery companies providing the real-time images used in our model to hedge funds.

Our paper makes two important contributions to the existing literature. First, we show that it is possible to use image data as a proxy for fundamentals in forecasting the prices of agricultural commodities. In order to use image data in machine learning, a convolutional neural network (CNN) must be utilized. The use of a CNN to analyze fundamentals rather than using it as an augmentation tool is also a new approach. Aerial images are less costly to collect than WASDE data. Allowing machine learning models to interpret data can improve the accuracy of forecasts compared to using data summarized by humans such as the WASDE. [Liu et al. \(2021\)](#) also show that combining time series data and other fundamental data can improve the forecasting performance of ML models. Second, we utilize live trading as part of the out-of-sample dataset for our paper. While the use of live data to trade makes it impossible to replicate the results from our out-of-sample results, the approach is what would happen in real life if certain algorithms or models were used to trade with live data. The existing literature on the application of machine learning models to estimate financial asset prices only reports the model's training accuracy and one-period ahead out-of-sample predictions. When real trades are involved, a model with 90% accuracy in the training and testing set could still face significant losses if the out-of-sample data have a different distribution than the training and testing data.

Our approach shows that it is possible to achieve excess returns by utilizing alternative data and applying image processing techniques. However, the rapid decline in model performance as more participants access similar data highlights the risk of using machine learning for trading strategies: the saturation of similar data and models can lead to crowding of good trades and eroding returns.

2. Literature Review

Due to the potential for capturing non-linear relationships, the application of neural networks in finance began as soon as the foundation models were developed. [Zhang et al. \(1998\)](#) summarize the application of artificial neural networks (ANNs) in forecasting. Due to the primitive architecture of ANNs in the 1990s and the complexity of financial data, ANNs failed to outperform the forecasting ability of more traditional forecasting methods such as ARIMA and GARCH-type models. Moreover, with the exception of a few papers surveyed (such as [Grudnitski and Osburn 1993](#); [Kaastra and Boyd 1996](#); [Kuan and Liu 1995](#); [Kryzanowski et al. 1993](#); [Wong et al. 1992](#); [Wong and Long 1995](#); [Yu and Yan 2020](#)), most applications of ANNs remained in non-finance/economics areas.

Since 2014, the number of papers published that are related to the use of long short-term memory (LSTM) architecture in financial forecasting has exploded. LSTM architecture has been used to forecast stock market returns from around the world ([Chopra and Sharma 2021](#); [Jafar et al. 2023](#); [Liu et al. 2021](#); [Oukhouya and Himdi 2023](#); [Qiu et al. 2020](#); [Zaheer et al. 2023](#)), the prices of cryptocurrencies ([Seabe et al. 2023](#)), the futures market ([Ly et al. 2021](#)), and foreign exchange markets ([Yıldırım et al. 2021](#)). However, [Chudziak \(2023\)](#) shows that the long-term performance of neural net-based models is generally relatively poor. In addition to the issues of exploding or diminishing gradients during the training process, financial time series typically have very limited data points and, as a result, are prone to overfitting in the training process, which leads to poor out-of-sample performance ([Liu and Long 2020](#)). The apparent good results in some of the papers mentioned above may simply be an illusion of the overfitting issue.

Alternative models such as the hybrid fuzzy neural network ([García et al. 2018](#)), deep ensemble-CNN-SAE ([Parida et al. 2021](#)), and CNN-BiLSTM-AM ([Lu et al. 2021](#)) seem to have produced better results. However, these models have only been applied to forecast the next day's stock market prices using in-sample data. Their models will not work in

the context of commodity markets where hedgers tend to hold positions longer than a day. [Endri et al. \(2020\)](#) utilized a support vector machine (SVM) to predict the probability of a certain company being delisted from the Malaysia Stock Exchange. While the SVM is useful for classification purposes, its ability to predict financial asset prices is not conclusive.

Due to the poor performance and limitations of using price data alone, researchers have investigated models focused on incorporating factors of supply and demand into the models. [Haile et al. \(2016\)](#) developed an acreage response model to forecast the likely acreages to be planted for four major crops and found that the current price can predict future acreages planted. Their study established the link between planted areas and futures prices. The total acreages planted is only one of the two major factors that affect future yields, however. Weather also plays a major role. Bad weather conditions (too wet, too cold, or too hot could all have major impacts) during planting season can have devastating effects on future yields. [Hammer et al. \(2001\)](#) and [Bekkerman et al. \(2016\)](#) show that even with all the advances in technology, it is no easy task to forecast the weather conditions in a specific area when the planting season for certain crops starts.

Once the crops are planted, weather conditions until harvest play the most critical role in determining the yield. There are a number of studies that utilize the normalized difference vegetation index (NDVI) derived from NOAA's advanced very high resolution radiometer (AVHRR) and found a positive correlation between NDVI data and future crop yields ([Mkhabela and Mashinini 2005](#); [Funk and Budde 2009](#)). [Wall et al. \(2008\)](#) found that the AVHRR-NDVI data have better explanatory power four weeks earlier than using the cumulative moisture index (CMI) alone when forecasting the crop yields for wheat. [Mkhabela et al. \(2011\)](#) show that using Moderate Resolution Imaging Spectroradiometer (MODIS) data yield better forecasting results (up to two months ahead) than AVHRR-NDVI data. These studies show that acreage and weather data are effective predictors of future yields.

In this study, we utilize aerial imagery of selected wheat farms and weather conditions (proxied by cloud coverage over the planted area) within a machine learning framework to predict future prices for wheat. This paper explores utilizing convolutional neural networks (CNNs) to predict the future yields and prices of wheat. There were a few studies utilizing CNNs to predict financial asset prices prior to [Nair and Hinton \(2010\)](#), where the rectified linear unit (ReLU) activation function was proposed. Before the ReLU activation function was invented, most papers related to the use of CNNs utilized the hyperbolic tangent (tanh) or the Sigmoid activation function ([Laxmi and Kumar 2011](#)). Both tanh and Sigmoid functions are more computationally complex than the ReLU function. By utilizing ReLU activations instead of Sigmoid or tanh, we can reduce training time by improving the sparsity of the model, helping to avoid the vanishing/exploding gradient problem during the model training process.

3. Data and Methodology

While there are many varieties of wheat grown across the world, this paper focuses on the hard red winter (HRW) since it has the widest distribution across the USA. HRW is used in bread, hard rolls, flatbreads, all-purpose flour, and Asian noodles.

Aerial imagery that shows cloud cover, sun elevation, and azimuth over the planted areas for HWR was pulled from the [Landsat Collection 2](#) database through the [Google Earth Engine](#). More specifically, we used the following datasets:

- Landsat 1–5 MSS
- Landsat 4 TM
- Landsat 5 TM
- Landsat 7 ETM+
- Landsat 8 OLI/TIRS
- Landsat 9 OLI-2/TIRS-2

Much of the historical data are trained on Landsat 5 and 8 starting from 1984 and continuing to the end of the study period. The dataset containing Landsat 9 images (which

is available after late 2021) was used in the test set to check whether the models are robust (if they have high variance issues).

Figure 1 shows the general areas where HRW is planted. We used the Landsat 8 and Landsat 5 satellites to show cloud covers in this area.

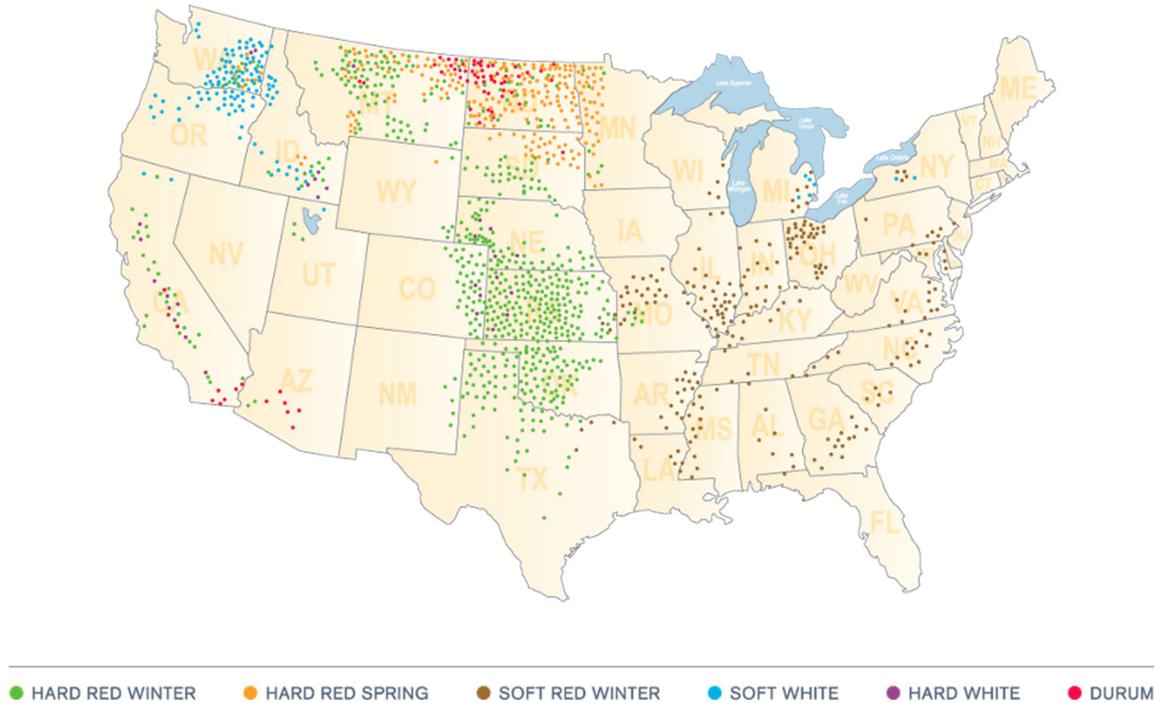


Figure 1. The distribution of various wheat varieties across the United States. <https://www.uswheat.org/working-with-buyers/wheat-classes/>, accessed on 2 April 2019.

Figure 2 shows the Google Earth Image without cloud cover and the same location with varying levels of cloud cover.



Figure 2. Example of a selected area with and without cloud cover.

In addition to the planting data, we also collected special weather features. These features are:

- Cloud cover: Percentage of the sample that is obscured by clouds.

- Sun elevation: A single value for the Sun's elevation (in degrees) above the horizon at the time of image acquisition. This is also normalized between 0–1.
- Sun azimuth: The horizontal angle between the sun and a reference direction, usually north. This defines the Sun's relative direction along the local horizon. The azimuth angle ranges from -180° (south) to -90° (west) to 0° (north) to 90° (east) and back to 180° (south). This is normalized to a value between 0 to 1.

Wheat futures data were collected from the Chicago Mercantile Exchange. We collected the price data from 1984 onward to match our imagery data time frame. When simulating trades, we utilized the front-month contract price at the close of each trading day.

We combined the image data with the front future and predicted the future price 20 trading days ahead. The model predicts the direction of the future price 20 days into the future and takes a long position if the model predicts that the price will go up and takes a short position if it predicts that prices will go down from the current futures price. We chose the 20-day time window to avoid the price effects of futures contract rolling practices, which occur on the third Friday of each month, before the 20-day window.

3.1. Data Preparation and Model Selection

To improve the model generalizability of the data, we segmented the images in our dataset with image transformations. Many images can appear to be strikingly similar due to inconsistent capturing conditions and vast homogenous landscapes. Image augmentations prove instrumental when working with imperfect or small datasets. The process can amplify the dataset size and strengthen model robustness by providing variety in the images. This provides robustness against frequently encountered real-world imperfections of aerial imagery captures such as inconsistent lighting or orientation distortions. Moreover, augmentations provide a cost-effective alternative to extensive aerial recaptures. The following augmentations were performed on all the images in our dataset.

Applying Transformations to Improve Model Generalizability

Satellite imagery often suffers from distortions and artifacts due to atmospheric conditions, sensor limitations, and processing anomalies. To counter these challenges and enhance the generalization capability of our model to new, unseen data, we implemented a series of programmatic image transformations. These transformations equip the model to handle a diverse array of real-world scenarios more effectively. Since manually curating and adjusting each image is impractical, automated transformations offer a scalable solution to improve our model's performance.

More specifically, we applied the following transformations:

- Translation: To mimic positional variances.
- Rotation: To account for changes in orientation.
- Scaling: To simulate different zoom levels.
- Noise injection: To represent sensor and environmental noise.
- Random cropping: To generate variability in scene composition.

Each transformation addresses a specific type of variability or distortion that can occur in satellite imagery, thus preparing our model to generalize well across different conditions and geographical areas. For example, noise injection helps the model remain robust against grainy or pixelated images while scaling ensures that it can interpret features of varying sizes effectively.

By automating the application of these transformations, we significantly enhance the model's ability to generalize from the training data to real-world scenarios, a crucial step for achieving reliable and accurate predictions from satellite imagery.

3.2. Selection of Model Architecture

Our model was built by feeding multiple picture inputs before a specified time to predict the direction (up or down) of a wheat futures price. Images are randomly selected nearest and prior to the time of the future for a given region. For example, if an image of

an area has a timestamp at 6:45 p.m. (when the data are published), we obtain the time series inputs (in images) of the same area before that time. This model utilizes images from multiple regions throughout North America. Figure 3 shows the different regions for the images (up to a maximum of 50 regions used).

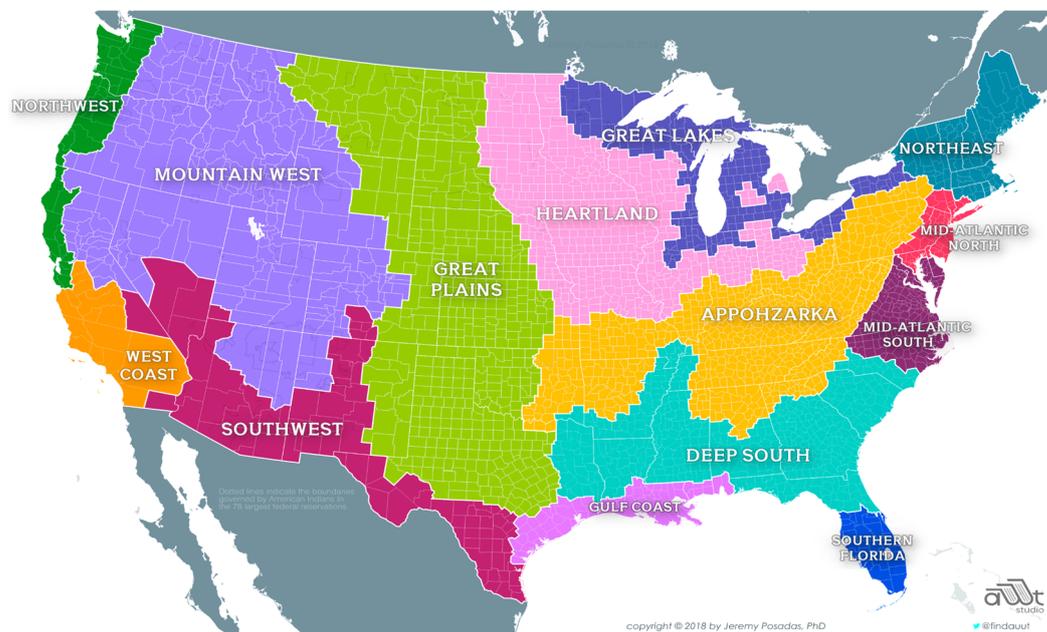


Figure 3. Regional, map of the US where the aerial images were obtained.

To keep the model simple to visualize, let us use an example of this model using 2 picture inputs (see Figure 4). These inputs use a standard convolution with a ReLU activation function and maxpooling (Conv + pool). To mimic a mobile net architecture, we experimented with different models' multiple layers of convolution and pooling. In practice, multiple picture inputs are utilized. Each image has two convolutional layers (kernel 3, stride 1, and padding 1), with two max-pooling layers, and ReLU activations.

In parallel, we fed alternative data (cloud cover, sun elevation, previous wheat prices, etc.) into a tabular format. This was fed into a dense net. All these models were put into a single vector combined for a single prediction. There is a linear dense layer for this vector.

A dense network was finally used to combine the final vectors. This was carried out so that each part could be learned and to give region-specific performance, and alternative data could be supplemented and learned to predict a final feature. This is similar to how multi-modal models are built in other stacks so each one can learn a feature and that feature can be used to explain and provide weights. Since this prediction layer is only a single layer deep, you can obtain the feature contributions.

We used the ReLU activation function instead of the sigmoid or tanh activation functions due to its much better training speed. ReLU is computationally more efficient than sigmoid and tanh because it only requires simple thresholding at zero, allowing models to train faster with fewer computational resources. ReLU activation can also lead to sparsity. When the output is zero, it is essentially ignoring that particular neuron, leading to sparse representation. Sparsity is beneficial because it makes the network easier to optimize and can lead to a more efficient model.

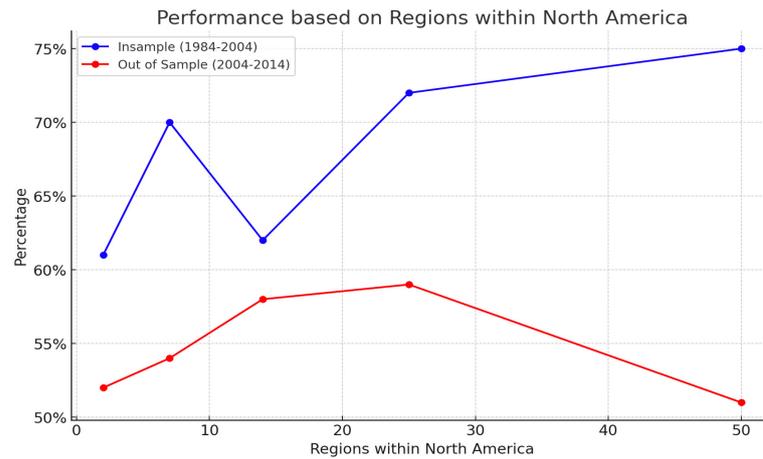


Figure 5. Model performance for the accuracy measure when predicting the direction of a wheat futures price 20 days in the future based upon the number of regions of data. Picture inputs will never be more than the number of regions; one picture is randomly selected for each region. While in-sample performance continues to rise by adding more regions, out-of-sample performance rises for a while and then degrades, showing no benefit of adding more regions after a certain point.

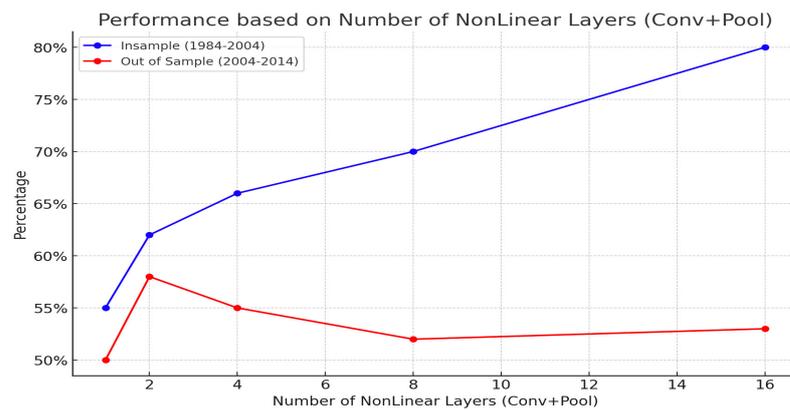


Figure 6. When holding the number of regions constant at 25, we can see what happens when we adjust the number of non-linear layers (a conv + pool is counted as a single layer). We can see the model stabilizes but tends to overfit quickly.

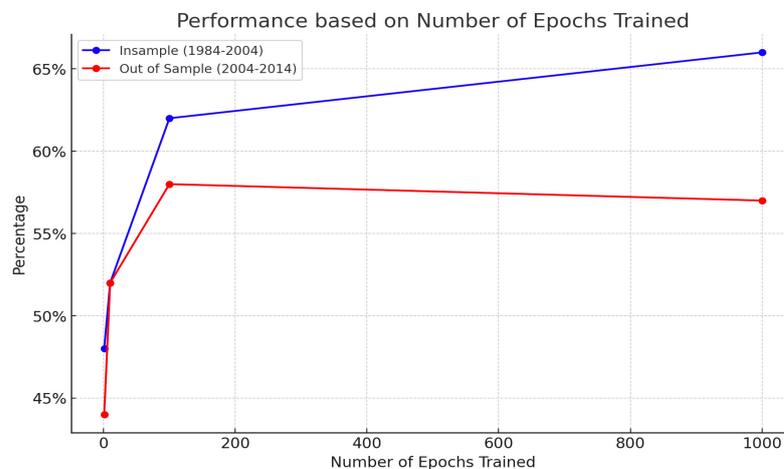


Figure 7. Training for longer usually resulted in increased performance with minimal degradation in out-of-sample performance. Again, the regions are held constant here and two non-linear layers are selected.

From the hyperparameter tests we performed, we selected a model with two layers of non-linearity, 100 epochs trained, and 25 regions within North America (codes provided in the Appendix A). We split the data into two sets: the training set (in-sample data) contains a data sample covering 1984–2004, and the testing set (out-of-sample data) contains data covering 2004–2014. The 67%/33% split of the data could help reduce the potential of overfitting when training the model. We will use an accuracy metric (number of correct predictions/total number of predictions) to measure the performance of the model first. We then evaluated the performance of the training set and testing set using a common measure used in investment management, the profits and losses (PNL) measure. The PNL measure is a better performance measure for models involving trading with live data. The reason is that a model with 90% accuracy could continue to roll the profit into trade and lose everything in one bad trade.

4. Results

Traditionally, machine learning evaluation methods consist of dividing the data into three subsets: the training set, the testing set, and the holdout set. In this paper, we took the historical data from 1984 to 2014 as the training and testing set. The holdout set is derived from trading with live data using the model we selected. The holdout set is from 2014 onward and is considered the out-of-sample set. Table 1 summarizes the profit and losses (PNL) results of the model.

Table 1. PNL values for the training set and the testing set.

Metric	Value
Average Train PNL	2.01% monthly
Average Test PNL	1.01% monthly
Standard Deviation Train PNL	6.2% monthly
Standard Deviation Test PNL	6.6% monthly
Sharpe Ratio Train PNL	1.12
Sharpe Ratio Test PNL	0.54

The result shows that the model performs well for the in-sample sets. Both the training set and the testing set produce positive average monthly returns. The decline in PNL for the testing set coincides with the increased financialization of the commodity markets (Chan et al. 2015). We observed a similar negative impact from increases in participations on returns when we used the model to trade with live data.

4.1. Robustness Test of the Model with Live Data

One of the major issues found in most other studies related to financial data forecasting is that those models might perform well in sample but perform poorly out of sample because the distribution of the out-of-sample data might be completely different from the in-sample data distribution. To test the robustness of our model, we utilized an approach not common in the finance literature: we tested the model's performance by trading with real-time data and real funds. We trained our model with data from 1984 to the end of 2003 as the training set and data from 2004 to the end of 2013 as the testing set. We also had an initial holdout set (out-of-sample set) from 2014 to the end of 2015. We then instructed the model to start trading live in 2016. The results are shown in Figure 8.

As the results indicate, the performance of the testing set is not as good as the training set. The average monthly PNL is only half of the results for the training set (Table 1). This relates to the drastic deterioration in the model's performance after 2018. The model performs well in the sample and partially out of the sample until about 2018. This is about the same time when Airbus and other companies started selling their data in real time to hedge funds and other traders in the industry. Most likely, more traders have utilized these alternative datasets to make trades, resulting in lower profitability for all trades.

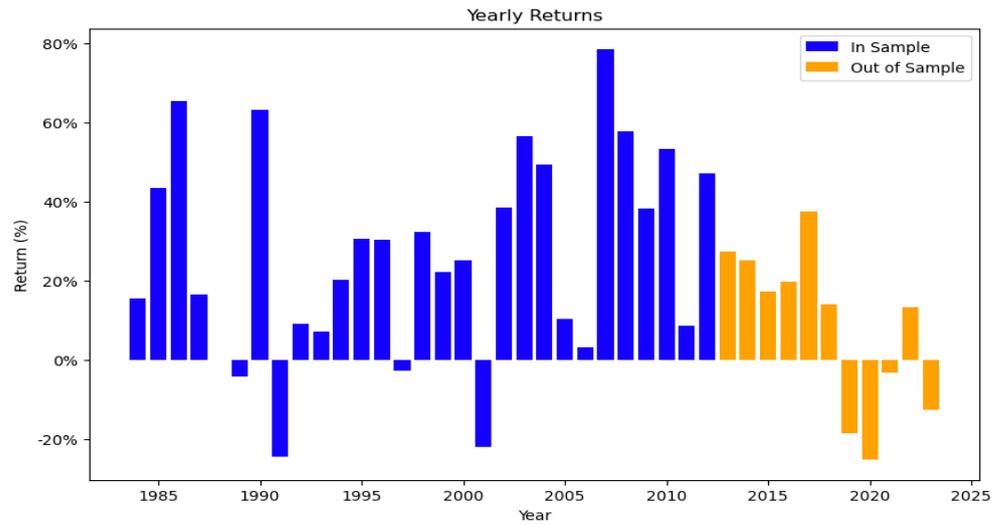


Figure 8. Trained model’s annual trading performance.

When we performed live trades with the model, we experienced decays in performance around 2018. The availability of real-time image data increases the availability of observations in the data and could result in better model performance for those who use the real-time data. We expect that this is likely the case. Since we observed that the model’s performance declined (which could be due to better model performance by other traders), we needed to run a real-time analysis looking at the distribution of returns to see when the value of the last 12 months of returns no longer looked profitable. This was achieved by looking at a distribution pairwise test and testing if the distribution of returns has shifted (measured by average monthly returns). Linear returns instead of compounding returns help highlight the model performance and drawdowns. Knowing that the model’s performance would decay, we instructed it not to reinvest profits but to instead trade with a fixed amount of capital. The results are shown in Figures 9 and 10. The model’s performance in 2019 and 2020 was particularly poor. The decline in performance might be partly due to the global economic conditions, i.e., the COVID-19 pandemic. However, the model failed to capture the benefits of the sharp increases in wheat prices during late 2021 and 2022. The decision to abandon live trade with cumulative funds was correct.

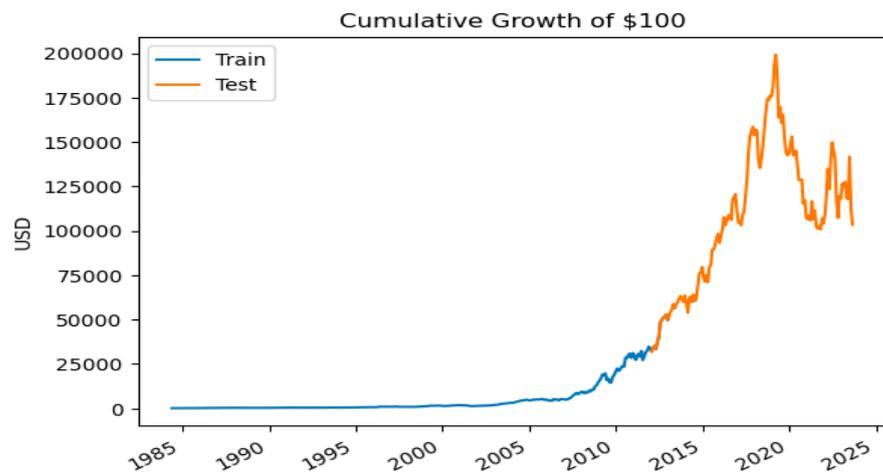


Figure 9. Cumulative growth of USD 100 invested, with gains and initial capital reinvested.

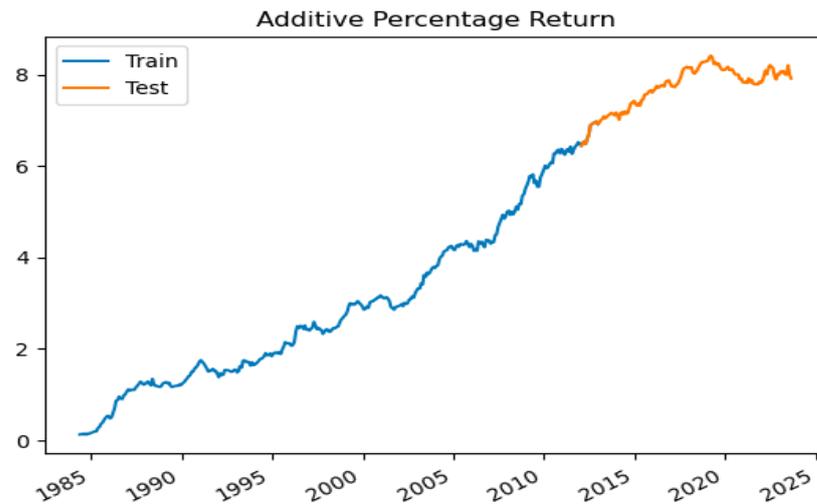


Figure 10. Cumulative growth of USD 100 invested without any reinvestment.

4.2. Robustness Test with Traditional Machine Learning Models

To test the robustness of our approach, we compared the model’s PNL performance using more traditional machine learning methods such as support vector machine (SVM) and multilayer perceptron (MLP). The results are shown in Table 2. The results show that the convolutional neural network (CNN) architecture outperformed both SVM and MLP.

Table 2. PNL values for different machine learning architectures.

Metric	CNN	SVM	MLP
Average Trained PNL	2.1%	0.52%	0.45%
Average Test PNL	1.01%	−0.21%	0.25%

In summary, our results show that the use of alternative data (such as aerial images) as proxy for fundamentals can yield excess returns for wheat futures. The ability to keep excess return depends on the novelty of the data and machine model architecture. While our model performed well against more traditional machine learning architectures (such as the SVM and MLP), its advantage can deteriorate quickly in live trades if other traders have access to better data and/or superior algorithms. Therefore, it is important to observe the out-of-sample performance and adjust the strategy and/or trading algorithm to avoid large losses. This is particularly important in markets with few participants and low trading volumes such as the wheat futures market.

5. Conclusions

In this paper, we utilize a convolutional neural network to analyze aerial images of areas of winter hard red wheat planted and cloud coverages over the planted areas as a proxy for future yield forecasts. Because future crop yields can affect futures prices directly, the approach we took in this paper was driven by fundamentals. We trained the model to forecast the futures price 20 days ahead and give recommendations to either to take long or short positions on wheat futures. Our method shows that it is possible to achieve positive alpha within a short time window if the algorithm and data choice are unique. We also took a unique approach to handle out-of-sample distribution by trading with real-time (live) data. When trading with live data, the model’s performance can deteriorate quickly if the input data become more easily available, as is the case with the aerial imagery we utilized in this paper. Another potential contributor to the decline in performance, since we traded with live data, is that other algorithm trading systems might be able to pick up our approach and create additional competitions that drive prices further away from fundamentals, leading to poor performance out of sample.

The initial successes of our model point to the potential that this approach can be generalized for other crops. However, the rapid decline in performance in our model due to the improved access to the data shows the limits of algorithmic trading. If an investor wants to maintain an excess return position, he/she must constantly innovate the algorithm or data sources. Contrary to other studies using machine learning tools to forecast financial time series with historical data and model accuracy above 50%, our findings suggest that those models will likely not be able to achieve above 50% consistently when dealing with out-of-sample data (trading with live data). In short, if the algorithm is easy to implement and data are widely available, the efficient market hypothesis should hold.

Author Contributions: Conceptualization, A.T.; data curation, A.T.; investigation, A.T.; methodology, A.T.; project administration, A.T. and D.S.; software, A.T. and D.S.; supervision, L.H.C.; validation, A.T. and L.H.C.; visualization, A.T. and L.H.C.; writing—original draft, A.T. and L.H.C.; writing—review and editing, A.T., L.H.C. and D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Available upon request.

Conflicts of Interest: The Authors declare no conflict of interest.

Appendix A

Codes for the custom model with two image inputs.

```
class CustomModel(nn.Module):
    def __init__(self):
        super(CustomModel, self).__init__()

        # For image 1
        self.conv1_1 = nn.Conv2d(3, 16, kernel_size=3, stride=1, padding=1)
        self.conv2_1 = nn.Conv2d(16, 32, kernel_size=3, stride=1, padding=1)
        self.fc1_1 = nn.Linear(
            32 * 64 * 64, 1
        ) # Size after pooling layers: 2 * [256/2/2] = 64

        # For image 2
        self.conv1_2 = nn.Conv2d(3, 16, kernel_size=3, stride=1, padding=1)
        self.conv2_2 = nn.Conv2d(16, 32, kernel_size=3, stride=1, padding=1)
        self.fc1_2 = nn.Linear(
            32 * 64 * 64, 1
        ) # Size after pooling layers: 2 * [256/2/2] = 64

        # For vector of length 26
        self.fc1_vec = nn.Linear(26, 1)

        # Final dense layers
        self.fc_final = nn.Linear(3, 3) # Increasing output features
        self.fc2 = nn.Linear(3, 3) # New layer
        self.fc3 = nn.Linear(3, 3) # New layer
        self.fc4 = nn.Linear(3, 1) # New layer

    def forward(self, x_img1, x_img2, x_vec):
        # Image 1
        x1 = F.max_pool2d(F.relu(self.conv1_1(x_img1)), 2)
        x1 = F.max_pool2d(F.relu(self.conv2_1(x1)), 2)
        x1 = x1.view(-1, 32 * 64 * 64)
```

```

x1 = self.fc1_1(x1)
# Image 2
x2 = F.max_pool2d(F.relu(self.conv1_2(x_img2)), 2)
x2 = F.max_pool2d(F.relu(self.conv2_2(x2)), 2)
x2 = x2.view(-1, 32 * 64 * 64)
x2 = self.fc1_2(x2)
# Vector of length 26
x3 = self.fc1_vec(x_vec)
# Concatenating
x = torch.cat((x1, x2, x3), dim=1)
# Passing through final dense layers
x = F.relu(self.fc_final(x))
x = F.relu(self.fc2(x))
x = F.relu(self.fc3(x))
x = self.fc4(x)

return x

```

References

- Adjemian, Michael K. 2012. Quantifying the WASDE announcement effect. *American Journal of Agricultural Economics* 94: 238–56. [\[CrossRef\]](#)
- Adjemian, Michael K., and Aaron Smith. 2012. Using USDA forecasts to estimate the price flexibility of demand for agricultural commodities. *American Journal of Agricultural Economics* 94: 978–95. [\[CrossRef\]](#)
- Bekkerman, Anton, Gary W. Brester, and Mykel Taylor. 2016. Forecasting a Moving Target: The Roles of Quality and Timing for Determining Northern U.S. Wheat Basis. *Journal of Agricultural and Resource Economics* 41: 25–41.
- Chan, Leo H., Chi M. Nguyen, and Kam C. Chan. 2015. A new approach to measure speculation in the oil futures market and some policy implications. *Energy Policy* 86: 133–41. [\[CrossRef\]](#)
- Chopra, Ritika, and Gagan Sharma. 2021. Application of Artificial Intelligence in Stock Market Forecasting: A Critique, Review, and Research Agenda. *Journal of Risk and Financial Management* 14: 526. [\[CrossRef\]](#)
- Chudziak, Adam. 2023. Predictability of stock returns using neural networks: Elusive in the long term. *Expert Systems with Applications* 213: 119203. [\[CrossRef\]](#)
- Endri, Endri, Kasmir Kasmir, and Andam Syarif. 2020. Delisting Sharia stock prediction model based on financial information: Support Vector Machine. *Decision Science Letters* 9: 207–14. [\[CrossRef\]](#)
- Fama, Eugene F. 1970. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance* 25: 383–417. [\[CrossRef\]](#)
- Funk, Chris, and Michael E. Budde. 2009. Phenologically-tuned MODIS NDVI-based production anomaly estimates for Zimbabwe. *Remote Sensing of Environment* 113: 115–25. [\[CrossRef\]](#)
- García, Fernando, Francisco Guijarro, Javier Oliver, and Rima Tamošiūnienė. 2018. Hybrid fuzzy neural network to predict price direction in the German DAX-30 index. *Technological and Economic Development of Economy* 24: 2161–78. [\[CrossRef\]](#)
- Goodwin, Barry K., and Randy Schnepf. 2000. Determinants of endogenous price risk in corn and wheat futures markets. *The Journal of Futures Markets* 20: 753–74. [\[CrossRef\]](#)
- Grudnitski, Gary, and Larry Osburn. 1993. Forecasting S and P and gold futures prices: An application of neural networks. *The Journal of Futures Markets* 13: 631–43. [\[CrossRef\]](#)
- Haile, Mekbib G., Jan Brockhaus, and Matthias Kalkuhl. 2016. Short-term acreage forecasting and supply elasticities for staple food commodities in major producer countries. *Agricultural and Food Economics* 4: 17. [\[CrossRef\]](#)
- Hammer, G. L., J. W. Hansen, J. G. Phillips, J. W. Mjelde, H. Hill, A. Love, and A. Potgieter. 2001. Advances in application of climate prediction in agriculture. *Agriculture Systems* 70: 2–3. [\[CrossRef\]](#)
- Hoffman, Linwood A. 2005. *Season-Average Price Forecasts*; Data Product. Washington, DC: U.S. Department of Agriculture, Economic Research Service.
- Hoffman, Linwood A., and J. Balagtas. 1999. Providing Timely Farm Price Forecasts: Using Wheat Futures Prices to Forecast U.S. Wheat Prices at the Farm Level. Paper presented at the 10th Federal Forecasters Conference, Washington, DC, USA, June 24; p. 13.
- Hoffman, Linwood, and Leslie A. Meyer. 2018. *Forecasting the U.S. Season-Average Farm Price of Upland Cotton: Derivation of a Futures Price Forecasting Model*; CWS-181-01; Washington, DC: U.S. Department of Agriculture, Economic Research Service, September.
- Hoffman, Linwood A., Scott H. Irwin, and Jose I. Toasa. 2007. Forecasting performance of futures price models for corn, soybeans, and wheat. Paper presented at the Annual Meeting of the American Agricultural Economics Association, Portland, OR, USA, July 29–August 1.
- Hoffman, Linwood A., Xiaoli L. Etienne, Scott H. Irwin, Evelyn V. Colino, and Jose I. Toasa. 2015. Forecast Performance of WASDE Price Projections for U.S. Corn. *Agricultural Economics* 4622: 157–71. [\[CrossRef\]](#)
- Isengildina-Massa, Olga, and Stephen MacDonald. 2009. *U.S. Cotton Prices and the World Cotton Market: Forecasting and Structural Change*; ERR-80; Washington, DC: U.S. Department of Agriculture, Economic Research Service, September.

- Jafar, Syed Hasan, Shakeb Akhtar, Hani El-Chaarani, Parvez Alam Khan, and Ruaa Binsaddig. 2023. Forecasting of NIFTY 50 Index Price by Using Backward Elimination with an LSTM Model. *Journal of Risk and Financial Management* 16: 423. [\[CrossRef\]](#)
- Kaastra, Iebling, and Milton Boyd. 1996. Designing a neural network for forecasting financial and economic time series. *Neurocomputing* 10: 215–36. [\[CrossRef\]](#)
- Kryzanowski, Lawrence, Michael Galler, and David W. Wright. 1993. Using artificial Neural nets: An approach to the forecasting neural networks to pick stocks. *Financial Analysts Journal* 49: 21–27.
- Kuan, Chung-Ming, and Tung Liu. 1995. Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of Applied Econometrics* 10: 347–64. [\[CrossRef\]](#)
- Laxmi, Ratna Raj, and Amrender Kumar. 2011. Weather based forecasting model crops yields using neural network approach. *Statistics and Applications* 9: 55–69.
- Liu, Hui, and Zhihao Long. 2020. An improved deep learning model for predicting stock market price time series. *Digital Signal Processing* 102: 102741. [\[CrossRef\]](#)
- Liu, Keyan, Jianan Zhou, and Dayong Dong. 2021. Improving stock price prediction using the long short-term memory model combined with online social networks. *Journal of Behavioral and Experimental Finance* 30: 100507. [\[CrossRef\]](#)
- Lu, Wenjie, Jiazheng Li, Jingyang Wang, and Lele Qin. 2021. A CNN-BiLSTM-AM method for stock price prediction. *Neural Comput & Applic* 33: 4741–53. [\[CrossRef\]](#)
- Ly, Racine, Fousseini Traore, and Khadim Dia. 2021. *Forecasting Commodity Prices Using Long-Short-Term Memory Neural Networks*. IFPRI Discussion Paper 2000. Washington, DC: International Food Policy Research Institute. [\[CrossRef\]](#)
- Meyer, Leslie A. 1998. Factors Affecting the U.S. Farm Price of Upland Cotton. In *Cotton and Wool Situation and Outlook; CWS-1998*; Washington, DC: U.S. Department of Agriculture, Economic Research Service.
- Mkhabela, Manasah S., and Nkosazana N. Mashinini. 2005. Early maize yield forecasting in the four agro-ecological regions of Swaziland using NDVI data derived from NOAA's-AVHRR. *Agricultural and Forest Meteorology* 129: 1–9. [\[CrossRef\]](#)
- Mkhabela, M. S., P. Bullock, S. Raj, S. Wang, and Y. Yang. 2011. Crop yield forecasting on the Canadian Prairies using MODIS NDVI data. *Agricultural and Forest Meteorology* 151: 385–93. [\[CrossRef\]](#)
- Nair, Vinod, and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. Paper presented at the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, June 22–24; pp. 807–14.
- Oukhouya, Hassan, and Khalid El Himdi. 2023. Comparing Machine Learning Methods—SVR, XGBoost, LSTM, and MLP—For Forecasting the Moroccan Stock Market. *Computer Sciences and Mathematics Forum* 1: 39.
- Parida, Nirjharinee, Debahuti Mishra, Kaberi Das, Narendra Kumar Rout, and Ganapati Panda. 2021. On deep ensemble CNN-SAE based novel agro-market price forecasting. *Evolutionary Intelligence* 14: 851–62. [\[CrossRef\]](#)
- Qiu, Jiayu, Bin Wang, and Changjun Zhou. 2020. Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS ONE* 15: e0227222. [\[CrossRef\]](#)
- Seabe, Phumudzo Lloyd, Claude Rodrigue Bambe Moutsinga, and Edson Pindza. 2023. Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach. *Fractal and Fractional* 7: 203. [\[CrossRef\]](#)
- Wall, Lenny, Denis Larocque, and Pierre-Majorique Léger. 2008. The early explanatory power of NDVI in crop yield modelling. *International Journal of Remote Sensing* 29: 2211–25. [\[CrossRef\]](#)
- Wong, Shee Q., and J. Allen Long. 1995. A neural network approach to stock market holding period returns. *American Business Review* 13: 61–64.
- Wong, F. S., P. Z. Wang, T. H. Goh, and B. K. Quek. 1992. Fuzzy neural systems for stock selection. *Financial Analysis Journal* 48: 47–52. [\[CrossRef\]](#)
- Yıldırım, Deniz Can, Ismail Hakkı Toroslu, and Ugo Fiore. 2021. Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators. *Financial Innovation* 7: 1. [\[CrossRef\]](#)
- Yu, Pengfei, and Xuesong Yan. 2020. Stock price prediction based on deep neural networks. *Neural Computing and Applications* 32: 1609–28. [\[CrossRef\]](#)

Zaheer, Shahzad, Nadeem Anjum, Saddam Hussain, Abeer D. Algarni, Jawaid Iqbal, Sami Bourouis, and Syed Sajid Ullah. 2023. A Multi Parameter Forecasting for Stock Time Series Data Using LSTM and Deep Learning Model. *Mathematics* 11: 590. [[CrossRef](#)]

Zhang, Guoqiang, B. Eddy Patuwo, and Michael Y. Hu. 1998. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting* 14: 35–62. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.