Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Optimizing Hyperparameters for Enhanced Satsuma Fruit Disease Detection

Jann Vincent Paul C. Lagmay [1]
[1] *Graduate Student, Department of Information Technology, College of Information Technology and Computer Science, University of the Cordilleras, Baguio City, Philippines. Email: jcl9134@students.uc-bcf.edu.ph, Orcid Id: https://orcid.org/0009-0004-7811-3934*

ARTICLE INFO

Received: 30 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

Hyperparameter tuning is an important process for optimizing the performance of machine learning models by fine-tuning parameters such as learning rate, batch size, and the number of epochs. This study systematically explored these parameters using a grid search optimization approach, conducting 120 experiments to enhance model accuracy and minimize loss. Key performance metrics, such as accuracy and loss, were used to evaluate the system's performance. Visualizations like line graph, heatmap, and pair plot gained insights into parameter interactions. The optimal configuration identified consisted of a learning rate of 0.001, a batch size of 32, and 50 epochs, achieving a test accuracy of 100.0% and a test loss 0.0027. These results represented a significant improvement over the approximated baseline configuration, which yielded a test accuracy of 90.4% and a test loss of 0.4164. The findings underscore the importance of moderate parameter values to ensure stable convergence, efficient training, and prevention of overfitting. By achieving substantial gains in accuracy and reductions in loss, the study demonstrates the transformative impact of hyperparameter tuning on model performance.

Keywords: Hyperparameter tuning, learning rate, batch size, epochs, model optimization.

INTRODUCTION

The citrus industry is a cornerstone of global agriculture, contributing significantly to food production, economic growth, and international trade. Within this diverse industry, Satsuma mandarins hold a prominent position, particularly in Asia and the Philippines. These mandarins, known for their distinct flavor and adaptability to different climates, enhance citrus diversity and sustain agricultural livelihoods. The global demand for citrus fruits, including Satsuma mandarins, underscores their economic and nutritional significance, making them an indispensable crop in the fruit trade [1].

Citrus fruits are cultivated extensively in tropical and subtropical regions, with leading contributions from the European Union, the Middle East, and Africa. These regions have emerged as global exporters, fostering economic interdependence through citrus trade. Notably, the citrus sector has catalyzed local economic development, especially in citrus-producing nations like the Philippines, where it has become a linchpin of rural development and farmer empowerment. In 2021, these regions dominated the global trade in citrus fruits, reflecting the widespread importance of this industry [2][4].

Satsuma mandarins are especially significant in the agricultural landscapes of certain regions, such as the Gulf Coast states of the United States and Nueva Vizcaya in the Philippines. These mandarins serve as a primary source of income for many citrus growers and enhance regional agricultural identities. For instance, Nueva Vizcaya contributes up to 80% of the annual citrus output in the Cagayan Valley Region of the Philippines, demonstrating the critical role of crops in local economies. A recent study in Malabing Valley, Municipality of Kasibu, Nueva Vizcaya, Philippines, surveyed 196 Satsuma growers to assess the prevalent fruit diseases, current practices, and technology adoption approaches. The study identified Scab, Huanglongbing (Citrus Greening Disease), and Canker as the most prevalent diseases affecting Satsuma production in the area, highlighting local growers' challenges. Their staggered harvesting

seasons and early bearing characteristics also provide growers with extended market opportunities, ensuring sustained incomes [3]. Furthermore, their staggered harvesting seasons and early bearing characteristics provide growers extended market opportunities, ensuring sustained incomes [4][5].

Recent advancements in fruit disease detection techniques have revolutionized agricultural practices by leveraging computer vision and machine learning. Convolutional Neural Networks (CNNs) have been pivotal, showcasing high accuracy in various applications. For instance, a study on Apple defect detection using a CNN-based model achieved 96.5% accuracy and outperformed traditional methods, showing its effectiveness for real-time use in commercial packing lines [7]. Expanding the scope, researchers utilized a hybrid approach combining Convolutional Neural Networks (CNNs) and machine learning techniques to classify various fruit diseases across a heterogeneous dataset, achieving an impressive accuracy of 97.10% [8]. This approach demonstrates the potential of CNN models in addressing diverse fruit disease detection challenges while enhancing performance compared to traditional methods. However, specific fruit types and conditions have often been the focus.

Research in this area aims to explore how hyperparameter optimization can enhance the accuracy and reliability of Satsuma fruit disease detection models. This study seeks to address gaps in the existing literature and contribute to advancements in precision agriculture by leveraging machine learning models, optimizing critical parameters, and evaluating performance improvements.

The accuracy and efficiency of identifying and managing plant diseases have significantly improved. Integrating advanced deep learning techniques, such as transfer learning and hybrid models, has enhanced model robustness and efficiency [9]. For instance, transfer learning has been effectively utilized in banana disease detection with ResNet and InceptionV2 architectures, demonstrating its ability to handle complex features with minimal data [10].

Furthermore, applying multi-spectral imaging technologies has expanded the capabilities of automated fruit disease detection systems. Researchers have developed systems capable of identifying external fruit defects by integrating near-infrared (NIR) and RGB imaging, employing adaptive preprocessing and threshold-based analysis [11]. These advancements enhance detection accuracy and address challenges posed by limited training data and diverse environmental conditions.

While these methods show great promise, they also highlight the need for further research to improve model generalization across various fruit types and environments. Future efforts should focus on integrating these technologies with real-time monitoring systems to enable widespread adoption in precision agriculture. The agricultural industry can reduce economic losses due to fruit diseases and improve overall crop management practices.

The reviewed studies highlight significant advancements in machine learning and hyperparameter optimization for fruit disease detection. While existing studies have explored CNN-based models for various fruits and achieved high accuracy rates, a clear gap still needs to be bridged in systematically addressing hyperparameter tuning precisely for Satsuma fruit disease detection. For instance, studies have optimized CNNs with techniques such as Dung Beetle Optimization for detecting tomato disease [12] and transfer learning for identifying grape leaves [13]. The benefits of these methods for Satsuma fruit detection still need to be explored. A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks further emphasizes the importance of this approach in improving model performance [14]. However, applying these advanced techniques to Satsuma-specific datasets and disease categories still needs to be improved, presenting an opportunity for future research.

Recent studies have explored grid search optimization in fruit disease detection, leveraging deep learning techniques to enhance accuracy and efficiency. This approach has been applied to various aspects of fruit disease detection models, including hyperparameter tuning, model architecture optimization, and feature selection. Moreover, the impact of hyperparameter tuning strategies such as the grid search approach has been well-documented across diverse machine learning domains.

Deep learning models for fruit disease detection have their hyperparameters optimized through grid search. For instance, in a study on strawberry disease and quality detection, researchers used Grid Search (GS) to systematically evaluate hyperparameter combinations by discretizing their ranges [15]. This method helped fine-tune numeric and

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

integer hyperparameters for vision transformers and attention-based convolutional neural networks. Additionally, studies on optimization techniques like Dung Beetle Optimization and multi-objective frameworks provide valuable insights, although these strategies are minimally applied to Satsuma fruit disease detection.

Furthermore, EfficientNet, a popular deep learning architecture, initially performs a grid search for the base network to determine the relationships between unique scaling dimensions [16]. This approach has been adapted in fruit detection and classification studies to optimize model architectures for specific fruit types and diseases. For example, the YOLO-v8 framework has been utilized for fruit detection and recognition, demonstrating the application of optimized architectures in this field [16]. As tuning hyperparameters can enhance model performance significantly, further research is needed to adapt and evaluate these strategies specifically for the Satsuma dataset.

Moreover, several studies have used grid search to compare the performance of different deep learning models for fruit disease detection. CNN-based methods have shown promising results in single-stage fruit detection and recognition [17]. These models have been optimized using grid search to improve their performance detecting various fruit diseases [17] [18]. Similarly, two-stage methods based on R-CNN variants, including Faster R-CNN and Mask R-CNN, have also been optimized using grid search for improved accuracy in fruit disease detection [18]. For instance, a study on citrus fruit disease detection used a Faster-CNN model with context data fusion, achieving high accuracy in various diseases like citrus canker, scab, melanosis, greening, blackspot, and healthy with accuracy, 97%, 95%, 99%, 97%, 97%, and 97%, respectively [19].

Finally, while comprehensive reviews on the broader effects of hyperparameter optimization on deep learning models exist, including their influence on CNN accuracy and multi-objective applications, these insights have yet to be systematically extended to address the unique challenges of Satsuma fruit disease detection [20]. Future work could bridge this gap by developing optimization frameworks tailored to Satsuma datasets, potentially enhancing accuracy and applicability. Addressing these gaps, researchers can contribute to the broader goal of improving agricultural productivity and disease management for this economically significant fruit, the Satsuma mandarin.

METHODS AND METHODOLOGY

This study followed a structured approach to develop and optimize a Convolutional Neural Network (CNN) for classifying Satsuma fruit diseases, focusing on canker and scab. The process was divided into key stages: dataset preparation, data preprocessing, model selection, initial model development, hyperparameter tuning, final selection and testing, and analysis and reporting.

A. Dataset Preparation

Data collection and preparation are vital for any image-based disease detection system. In the case of the Satsuma mandarin fruit, the researcher meticulously acquired high-quality images of disease conditions such as scab and canker. The study focused on citrus orchards in the Malabing Valley of the Municipality of Kasibu, Nueva Vizcaya, Philippines.

Images were captured from various angles to ensure comprehensive representation, providing a holistic view of the disease manifestations on the fruit. A key aspect of the data collection process was using a photo studio box, which provided consistent lighting conditions across all images. This controlled environment is essential for minimizing variations that could affect subsequent analysis's accuracy.

The imaging equipment played a significant role in maintaining data quality. A high definition camera was employed for its advanced capabilities, while the researcher utilized a tripod and shutter to ensure stability during image capture. These measures contribute to the overall clarity and consistency of the dataset.

The dataset comprised 1,000 images, equally distributed between the two disease conditions under study. Specifically, 500 images were collected for canker-affected Satsuma mandarins and another 500 for those exhibiting scab. This balanced dataset is crucial for training machine learning models, as it helps prevent bias towards one condition.

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

B. Data Preprocessing

Data preprocessing began with meticulous image enhancement using the Canva's product photos feature to eliminate subtle shadows caused by varying capture angles. All images were standardized with consistent margins and dimensions of 1440x1440 pixels to ensure uniformity and optimal focus on the fruit samples.

The annotation process was conducted through the Roboflow platform, leveraging the expertise of five agricultural experts. The annotation team consisted of three academic experts specializing in citrus research and two professionals from the agriculture sector - one municipal agriculture officer and one agricultural technician.

Following standard machine learning practices, the researcher strategically split the dataset into three sets: a training set comprising 80% of the data, the validation, and test sets, each receiving 10% of the total dataset. This distribution ensures robust model training while maintaining sufficient data for validation and testing purposes.

C. Initial Model Development with Google Teachable Machine

The initial model development for the Satsuma fruit disease classification project began with using Google Teachable Machine (GTM), a user-friendly platform for creating machine learning models. The process commenced by accessing the GTM application and initiating a new image project. The standard image model was selected as the most suitable option for this task, balancing performance and versatility.

The standard image model operates with 224x224 pixel color images and provides the flexibility to export the trained model to various formats, including TensorFlow, TFLite, and TF.js. This versatility ensures compatibility with different deployment environments and platforms. The model's compact size of approximately 5MB makes it ideal for applications with limited storage or bandwidth constraints.

The researcher built a robust dataset for training by downloading images from Roboflow, a popular platform for managing computer vision data. To maintain a balance between disease classes, the researcher selected 500 images for scab and canker diseases. These preprocessed Satsuma fruit images were then uploaded to the GTM platform, as shown in Figure 1. Careful labeling was performed to categorize the disease images into their respective classes, such as canker and scab.

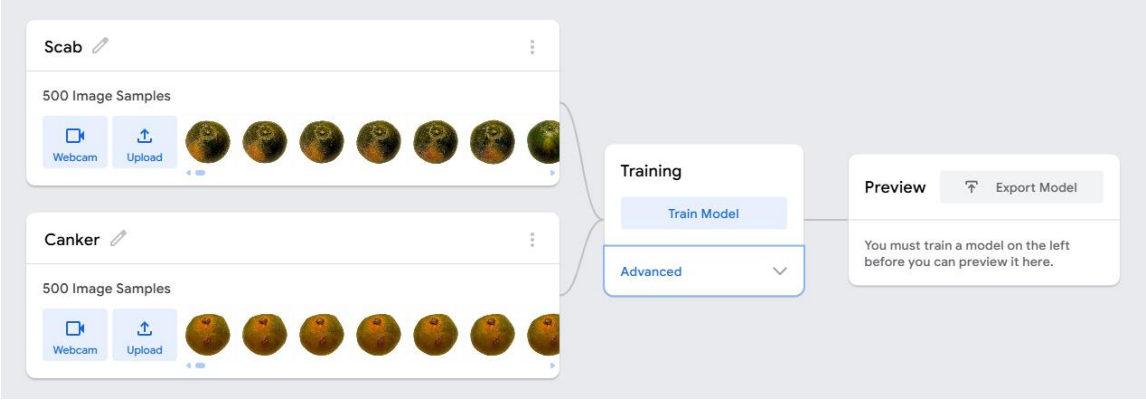

Figure 1. Uploaded Satsuma fruit images in Google Teachable Machine.

The initial model training utilized the MobileNet architecture, a state-of-the-art convolutional neural network for mobile and edge devices. This choice of architecture aligns well with the model's goals, as it offers a good trade-off between accuracy and computational efficiency. Using the GTM and the MobileNet architecture, the study laid a solid foundation for developing an effective Satsuma fruit disease detection model.

D. Hyperparameter Tuning using Grid Search

The researcher conducted hyperparameter tuning using the grid search method to optimize the model's performance. This approach systematically evaluated combinations of key hyperparameters, including the number of epochs, learning rate, and batch size. The results of these experiments are summarized in Tables 1(a) to 1(d), which detail the performance metrics (accuracy and loss) for both training and testing datasets under varying configurations.

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 1(a) presents the results for experiments conducted over 10 epochs, while Table 1(b), Table 1(c), and Table 1(d) outline results for 50 epochs, 100 epochs, and 200 epochs, respectively. Each table highlights the influence of different hyperparameter combinations on model performance.

These tables collectively illustrate the systematic evaluation of hyperparameter combinations, providing insights into their effects on both accuracy and loss. The analysis reveals the importance of achieving a balanced configuration to ensure optimal model performance.

E. Final Model Selection and Testing

This study began with identifying the best-performing model based on the results of hyperparameter tuning. The model with the optimal combination of hyperparameters was selected for further refinement, as determined by the highest test accuracy and lowest test loss. The researcher retrained this model using the combined training and validation datasets to maximize its learning capacity and ensure robust generalization. Following retraining, the researcher evaluated the selected model on a held-out test dataset to assess its performance on unseen data. Key performance metrics, including accuracy and loss, were calculated and analyzed to validate the effectiveness of the fruit disease detection model. Finally, the optimized model was exported and prepared for deployment, ensuring its integration into the target application or system for practical use.

Table 1(a). Experiment results showcasing the impact of varying hyperparameter combinations (learning rate, batch size, accuracy, and loss) for 10 epochs

Experiment No.	Learning Rate	Batch Size	Accuracy		Loss	
			Train	Test	Train	Test
1	0.1	16	0.500235295	0.500000000	4.186222887	4.605220795
2	0.1	32	0.499882358	0.500000000	4.584222651	4.605220366
3	0.1	64	0.727764738	0.734666681	2.362277704	2.439478219
4	0.1	128	0.503411770	0.500000000	4.554040575	4.605220795
5	0.1	256	0.499882355	0.500000000	4.488973260	4.605220318
6	0.1	512	0.501529413	0.500000000	4.370074511	4.605221272
7	0.01	16	0.996000010	1.000000000	0.013361842	0.000203080
8	0.01	32	0.990235317	0.998000002	0.130621797	0.010382780
9	0.01	64	0.989176482	0.992000008	0.071627125	0.014184124
10	0.01	128	0.965294147	0.980000007	0.299268581	0.150237464
11	0.01	256	0.945058852	0.991999966	0.232302721	0.023955843
12	0.01	512	0.889411813	0.941999990	0.627381901	0.193951146
13	0.001	16	0.996470594	1.000000000	0.222619582	0.203781826
14	0.001	32	0.993411773	1.000000000	0.040126299	0.009117263
15	0.001	64	0.990588242	1.000000000	0.071060724	0.007231903
16	0.001	128	0.981764740	0.994000006	0.116686334	0.054659977
17	0.001	256	0.984470618	0.997333282	0.075211652	0.034798627
18	0.001	512	0.961764753	0.982000017	0.142378217	0.107636555
19	0.0001	16	0.891635124	0.992666674	0.077844089	0.050116538
20	0.0001	32	0.973176521	0.986000013	0.150369481	0.119448031
21	0.0001	64	0.970117694	0.986666673	0.182678549	0.147170970

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Experiment No.	Learning Rate	Batch Size	Accuracy		Loss	
			Train	Test	Train	Test
22	0.0001	128	0.894705939	0.954000014	0.408366266	0.369308048
23	0.0001	256	0.909418476	0.904000017	0.457296905	0.416415323
24	0.0001	512	0.726823574	0.770666665	0.635090500	0.605100194
25	0.00001	16	0.928117704	0.367142709	0.932666689	0.352248397
26	0.00001	32	0.790117681	0.823333359	0.599628720	0.523257282
27	0.00001	64	0.919294167	0.924000037	0.505561969	0.495822170
28	0.00001	128	0.662352979	0.653196579	0.652805072	0.654437643
29	0.00001	256	0.576000023	0.596666682	0.649600452	0.634260952
30	0.00001	512	0.497764736	0.513745189	0.755617070	0.758092856

Table 1(b). Experiment results showcasing the impact of varying hyperparameter combinations (learning rate, batch size, accuracy, and loss) for 50 epochs

Experiment No.	Learning Rate	Batch Size	Accuracy		Loss	
			Train	Test	Train	Test
31	0.1	16	0.499647063	0.500000000	4.600047588	4.605220318
32	0.1	32	0.501529413	0.500000000	4.579988575	4.605220795
33	0.1	64	0.502352947	0.500000000	4.554415846	4.605220318
34	0.1	128	0.497176474	0.500000000	4.560686111	4.605219841
35	0.1	256	0.912117696	0.920666671	0.687031870	0.713608091
36	0.1	512	0.525058830	0.500000000	4.244945359	4.605220318
37	0.01	16	0.992352957	0.992666674	0.023971816	0.020930748
38	0.01	32	0.992235303	0.996666670	0.026078442	0.008257028
39	0.01	64	0.975764734	0.995333338	0.066179253	0.012828836
40	0.01	128	0.971058851	0.998000002	0.077336651	0.007497785
41	0.01	256	0.940235323	0.989333302	0.258844613	0.026697355
42	0.01	512	0.915882373	0.968666637	0.197101715	0.072640024
43	0.001	16	0.997294128	1.000000000	0.010958936	0.003400818
44	0.001	32	0.994470596	1.000000000	0.017323309	0.002661103
45	0.001	64	0.984000027	1.000000000	0.042381167	0.012898028
46	0.001	128	0.973647088	0.996000004	0.058958527	0.022378182
47	0.001	256	0.970470607	0.997333276	0.102879866	0.051043681
48	0.001	512	0.873069602	0.991333282	0.133973394	0.089443948
49	0.0001	16	0.910493729	1.000000000	0.075445634	0.045487585
50	0.0001	32	0.988000035	0.992666674	0.098567154	0.071142640
51	0.0001	64	0.976941204	0.990666676	0.151083306	0.123757328
52	0.0001	128	0.902567941	0.910544086	0.293518953	0.229656052

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Experiment No.	Learning Rate	Batch Size	Accuracy		Loss	
			Train	Test	Train	Test
53	0.0001	256	0.900941199	0.944666636	0.291208424	0.330217773
54	0.0001	512	0.847111708	0.406190001	0.898666686	0.347362870
55	0.00001	16	0.908705926	0.936000013	0.304948511	0.288376336
56	0.00001	32	0.957411808	0.974666685	0.292893664	0.290249754
57	0.00001	64	0.784705919	0.794000012	0.421684742	0.418428443
58	0.00001	128	0.799176532	0.822000009	0.442271334	0.437186429
59	0.00001	256	0.854117697	0.466670147	0.850666690	0.483631414
60	0.00001	512	0.706117678	0.728666687	0.607600242	0.587479722

Table 1(c). Experiment results showcasing the impact of varying hyperparameter combinations (learning rate, batch size, accuracy, and loss) for 100 epochs

Experiment No.	Learning Rate	Batch Size	Accuracy		Loss	
			Train	Test	Train	Test
61	0.1	16	0.500000000	0.500000000	4.601576400	4.605220318
62	0.1	32	0.500705883	0.500000000	4.592268491	4.605220318
63	0.1	64	0.509411767	0.500000000	4.506344438	4.605220318
64	0.1	128	0.499764708	0.500000000	4.577202201	4.605220318
65	0.1	256	0.499588236	0.500000000	4.550764680	4.605220938
66	0.1	512	0.920411813	0.901432388	0.615778235	0.596673449
67	0.01	16	0.941117671	0.963666663	0.154696742	0.090282939
68	0.01	32	0.995705888	1.000000000	0.014666312	0.000440174
69	0.01	64	0.992313731	0.997000003	0.067000630	0.105332136
70	0.01	128	0.983588248	0.996000001	0.038336037	0.011435778
71	0.01	256	0.966058844	0.980999961	0.109200791	0.056469477
72	0.01	512	0.941117671	0.963666633	0.154741635	0.140565804
73	0.001	16	0.997117653	1.000000000	0.007944104	0.002122768
74	0.001	32	0.996352947	0.999666667	0.010343391	0.002787920
75	0.001	64	0.996235305	0.999333334	0.016093008	0.004415038
76	0.001	128	0.988470611	0.996666670	0.030996326	0.019935395
77	0.001	256	0.989000025	0.949496365	0.039323474	0.019600416
78	0.001	512	0.969470608	0.963556138	0.092485594	0.059976876
79	0.0001	16	0.992117661	0.999333334	0.045187414	0.026718322
80	0.0001	32	0.992294142	0.996000004	0.063690395	0.044323306
81	0.0001	64	0.984058851	0.945562404	0.094350549	0.124031009
82	0.0001	128	0.969411799	0.981666678	0.159569135	0.179218113
83	0.0001	256	0.955647090	0.978999981	0.194962953	0.225335468
84	0.0001	512	0.925143239	0.945682451	0.312952035	0.279205376

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Experiment No.	Learning Rate	Batch Size	Accuracy		Loss	
			Train	Test	Train	Test
85	0.00001	16	0.973823556	0.993000004	0.188411326	0.172941799
86	0.00001	32	0.901294143	0.912333351	0.285837300	0.281518676
87	0.00001	64	0.903588280	0.910000008	0.322407996	0.329799598
88	0.00001	128	0.870117709	0.893333346	0.362400334	0.397233804
89	0.00001	256	0.873117703	0.899666634	0.380456151	0.418019665
90	0.00001	512	0.752117693	0.523881967	0.746000010	0.565707110

Table 1(d). Experiment results showcasing the impact of varying hyperparameter combinations (learning rate, batch size, accuracy, and loss) for 200 epochs

Experiment No.	Learning Rate	Batch Size	Accuracy		Loss	
			Train	Test	Train	Test
91	0.1	16	0.500000000	0.500000000	4.603796371	4.603795910
92	0.1	32	0.501862746	0.500000000	4.600944455	4.605220795
93	0.1	64	0.501647059	0.500000000	4.586048142	4.605220318
94	0.1	128	0.498666668	0.500000000	4.594326576	4.605219841
95	0.1	256	0.498941177	0.500000000	4.569311420	4.605221272
96	0.1	512	0.500000000	0.500000000	4.527073089	4.605221272
97	0.01	16	0.997882356	1.000000000	0.006205114	0.000234933
98	0.01	32	0.997647063	0.999333334	0.006566757	0.002770884
99	0.01	64	0.993098046	0.999333334	0.025767571	0.001409600
100	0.01	128	0.988862753	0.992888894	0.027347423	0.015803373
101	0.01	256	0.970627477	0.991777730	0.092543424	0.018577503
102	0.01	512	0.956509826	0.968888863	0.113300156	0.097804258
103	0.001	16	0.999019611	1.000000000	0.003776477	0.000512920
104	0.001	32	0.996745102	0.998888890	0.009550057	0.004031184
105	0.001	64	0.994823533	0.997777780	0.015208454	0.007452758
106	0.001	128	0.995058835	0.993333340	0.020700078	0.019739335
107	0.001	256	0.991450995	0.996444398	0.029903942	0.054956274
108	0.001	512	0.973764727	0.993555508	0.068778149	0.047225130
109	0.0001	16	0.993921586	0.998222224	0.035561693	0.024697517
110	0.0001	32	0.992078455	0.999333334	0.046690202	0.034508211
111	0.0001	64	0.986039239	0.990444452	0.079205729	0.071272759
112	0.0001	128	0.977960811	0.987333337	0.123259639	0.123206185
113	0.0001	256	0.968509833	0.973999959	0.162231565	0.154155389
114	0.0001	512	0.955568647	0.967777753	0.221043634	0.212814956
115	0.00001	16	0.974196108	0.981333339	0.168757119	0.161574151
116	0.00001	32	0.938196106	0.925111120	0.224227631	0.235501641

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Experiment No.	Learning Rate	Batch Size	Accuracy		Loss	
			Train	Test	Train	Test
117	0.00001	64	0.917647084	0.925333343	0.273670566	0.267966468
118	0.00001	128	0.881460782	0.895555570	0.319762941	0.313311989
119	0.00001	256	0.873117703	0.899666634	0.380456151	0.418019665
120	0.00001	512	0.745217693	0.756000010	0.533484967	0.565707110

F. Analysis and Reporting

This study examined how hyperparameters affect model performance. The researcher analyzed various combinations of learning rate, batch size, and epochs to find the optimal configuration. Through detailed analysis and visualization techniques like heatmaps and line plots, patterns emerged showing how different hyperparameters influenced accuracy and loss.

The optimal configuration with a learning rate of 0.001, batch size of 32, and 50 epochs significantly improved model performance compared to the baseline. This research highlights the importance of systematic hyperparameter tuning in enhancing the accuracy and efficiency of machine learning models. The findings provide valuable insights for future projects and practical applications.

RESULTS AND DISCUSSION

This study optimized hyperparameters to improve test accuracy and minimize test loss by tuning the learning rate, batch size, and number of epochs. Key findings are discussed below, with supporting evidence from tables and visualizations.

A. Performance Metrics

The optimal hyperparameters identified were a learning rate of 0.001, a batch size 32, and 50 epochs. This configuration achieved a training accuracy of 99.45%, a test accuracy of 100.0%, and a test loss of 0.0027, as shown in Table 2. Compared to the approximated baseline configuration, which achieved a test accuracy of 90.4% and a test loss 0.4164, the tuned model demonstrated a significant improvement. The baseline metrics correspond to Experiment 23 as shown in Table 1(c), which used 10 epochs, a learning rate 0.0001, and a batch size 256. This improvement highlights the impact of hyperparameter tuning on optimizing model performance.

Table 2. Summary of optimal hyperparameters and their corresponding performance metrics

Hyperparameter		Optimal Value	
Learning Rate		0.001	
Batch Size		32	
Number of Epoch		50	
Metric	**Training**	**Validation**	**Test**
Accuracy (%)	99.45	99.00	100.00
Loss	0.0173	0.0200	0.0027

B. Impact of Epochs

Figure 2 illustrates the relationship between epochs and performance. Accuracy improves steadily as the number of epochs increases, stabilizing at 50 epochs. At 50 epochs, the researcher does not observe significant performance gains, suggesting an ideal balance between learning and computational efficiency.

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

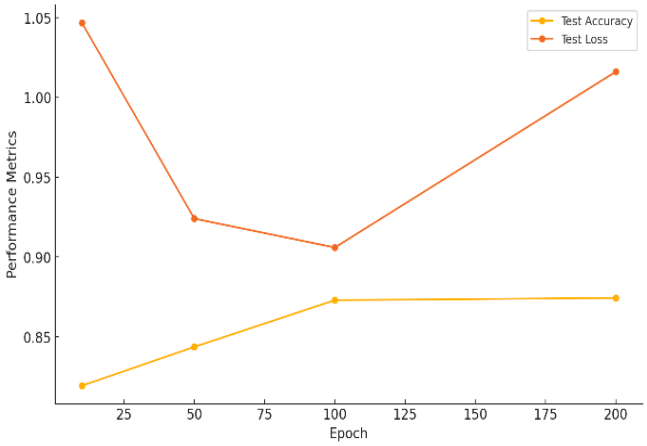

Figure 2 Impact of epochs on test accuracy and test loss

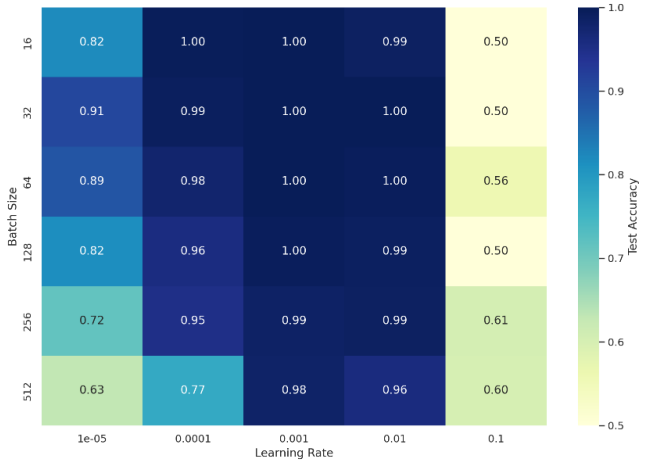

Figure 3 Test accuracy heatmap for batch size and learning rate

C. Learning Rate and Batch Size

Figure 3 shows that the optimal combination of a learning rate of 0.001 and batch size of 32 yielded the highest test accuracy and lowest loss. Larger batch sizes of 128 or more and extreme learning rates, such as 0.1, resulted in degraded performance. This finding highlights the importance of selecting moderate parameter values to ensure stable convergence and improved outcomes.

D. Relationships Between Hyperparameters

The pair plot in Figure 4 reveals crucial insights into the relationships between hyperparameters and performance metrics. A learning rate of 0.001 emerges as optimal, consistently delivering the highest test accuracy and lowest test loss, while higher rates lead to instability, and lower rates result in suboptimal performance. Batch size also significantly impacts model performance, with a size of 32 producing the highest test accuracy by balancing frequent weight updates and gradient noise. As for epochs, test accuracy improves steadily up to 50 epochs before plateauing, with excessive epochs risking overfitting.

The interactions between hyperparameters underscore the need for a balanced approach. A learning rate of 0.001 combined with a batch size of 32 and 50 epochs consistently achieves the best trade-off between accuracy and loss. Larger batch sizes paired with higher learning rates produce erratic results, while smaller batch sizes and moderate learning rates ensure stable learning and robust generalization. These insights emphasize the importance of tuning hyperparameters in combination, as their interactions significantly affect model performance and stability. Figure 4 validates the selected optimal configuration and highlights the necessity of systematic exploration for achieving superior outcomes.

Journal of Information Systems Engineering and Management
2025, 10(33s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

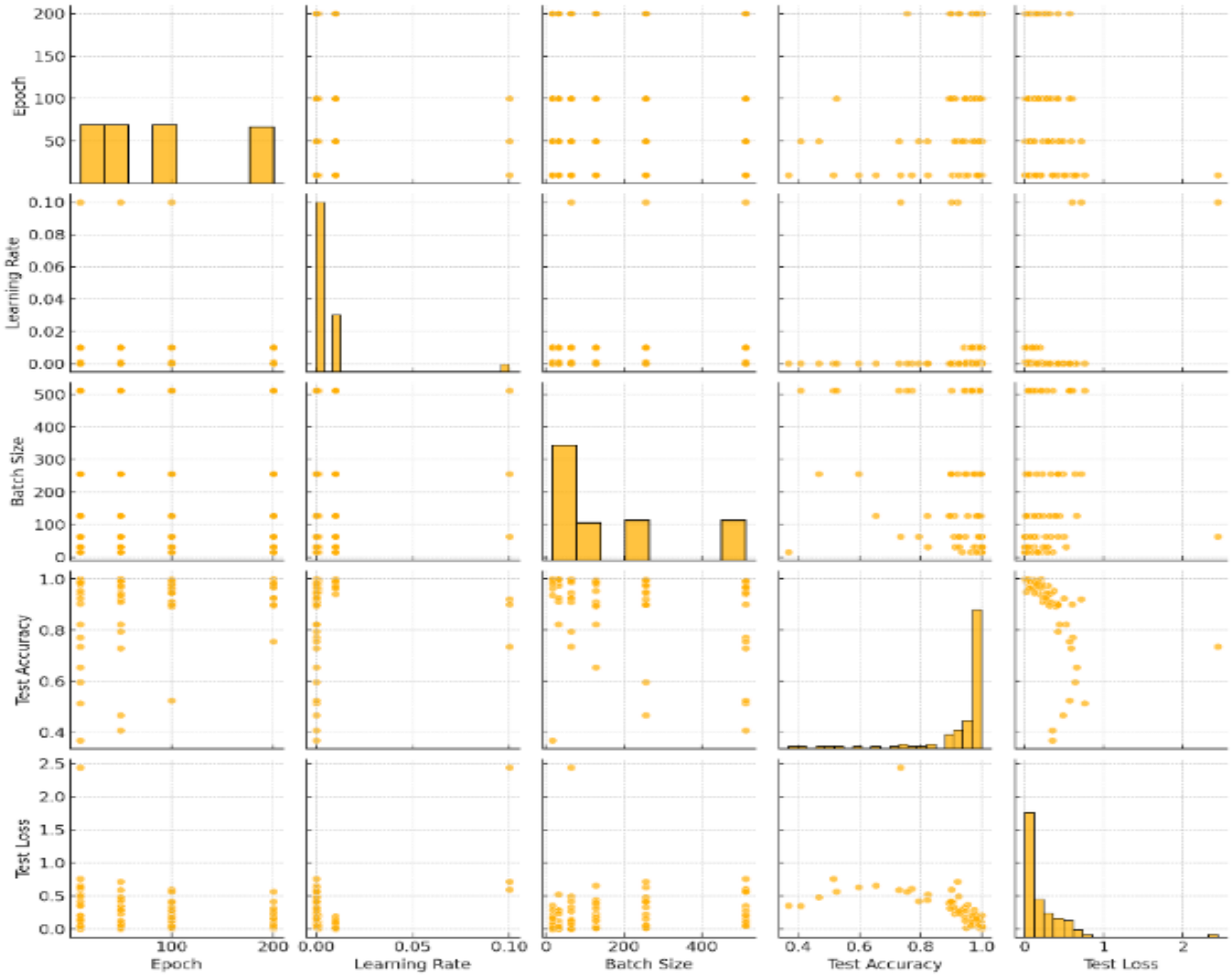

Figure 4 Pair plot visualizing hyperparameter relationships with performance metrics

E. Interpretation of Results

The optimal configuration demonstrates the effectiveness of systematic hyperparameter tuning in enhancing performance. By balancing convergence stability, generalization, and computational cost, the model achieves 100.0% accuracy with a test loss of 0.0027, far outperforming the baseline configuration.

F. Comparison to Baseline

Compared to the approximated baseline, represented by Experiment 23, the tuned configuration improved test accuracy by 9.6%, increasing from 90.4% to 100.0%, and reduced test loss by 0.4137, decreasing from 0.4164 to 0.0027. These substantial improvements underline the importance of optimizing hyperparameters to realize a model's potential fully.

CONCLUSION

This study highlights the critical role of hyperparameter tuning in maximizing machine learning model performance. By systematically exploring the effects of learning rate, batch size, and number of epochs, the researcher identified the optimal configuration: a learning rate of 0.001, 32, and 50 epochs. This setup achieved exceptional results, with a test accuracy of 100.0% and a test loss 0.0027. These results represent a significant improvement over the approximated baseline configuration, which achieved a test accuracy of 90.4% and a test loss 0.4164. The tuned

model demonstrated a 9.6% increase in accuracy and a substantial reduction in loss, underscoring the transformative impact of targeted hyperparameter optimization.

The findings emphasize the importance of selecting moderate learning rates and batch sizes to ensure stable convergence and practical training. Limiting the number of epochs also prevents overfitting and avoids unnecessary computational costs. These adjustments allow for a balanced approach that optimizes both accuracy and efficiency. The visualizations provided, including line graphs, heatmaps, and pair plots, offer valuable insights into parameter interactions, helping to guide the tuning process and ensuring reproducible results.

This study demonstrates the potential of hyperparameter tuning to unlock the full potential of machine learning models, making them more accurate and resource efficient.

Acknowledgement:

We thank Jann Vincent Paul Cadiz Lagmay for his contributions to this work. Special thanks to the University of the Cordilleras, Baguio City, for their invaluable suggestions and guidance and the Commission on Higher Education - Cordillera Administrative Region for their unwavering financial support.

REFERENCES

[1] Antonio LRF, Cruz MB, Madamba JaB, Williams JB. Assessing the performance of the citrus industry in Kasibu, Nueva Vizcaya, Philippines: the case of farmers and traders of the Malabing Valley Agri-Trading Center. CORE. 2015. https://core.ac.uk/outputs/229567513/?source=oai.

[2] Beshara S, Kassem A, Fors H, Harraz N. A Comprehensive Review and Mapping Citrus Supply Chains from a Sustainability Perspective across the European Union, Middle East, and Africa. Sustainability. 2024; 16(19): 8582. https://doi.org/10.3390/su16198582.

[3] Lagmay JVPC. Integrating Machine Learning-based Fruit Disease Detection Algorithm for Satsuma Growers' Knowledge, Practices, and Technology Adoption. Nanotechnology Perceptions. 2024; 20(S4): 27–38. https://doi.org/10.62441/nano-ntp.vi.624.

[4] Idquival JLA, Layog PMP, Victoria CGC, Rosete MAL. An Analysis of the Citrus Industry in Nueva Vizcaya, Philippines. International Journal of Research in Engineering, Science and Management. 2023; 6(12): 144–145. https://www.ijresm.com.

[5] Sarmiento GP. Enhancing the Mandarin Industry of Malabing Valley, Nueva Vizcaya, Philippines. Proceedings of 78th the IIER International Conference. 2016.

[6] Andersen PC, Ferguson JJ, Shahid MA. Satsuma Mandarin. EDIS. 2023; 2023(5). https://doi.org/10.32473/edis-ch116-2023.

[7] Fan S, Li J, Zhang Y, et al. Online detection of defective apples using a computer vision system combined with deep learning methods. Journal of Food Engineering. 2020; 286: 110102. https://doi.org/10.1016/j.jfoodeng.2020.110102.

[8] Pawar S, Surana A, Sharma P, Pujari R. Fruit Disease Detection and Classification using Machine Learning and Deep Learning Techniques. International Journal of Intelligent Systems and Applications in Engineering. 2023: 440–453. https://ijisae.org/index.php/IJISAE/article/view/3805.

[9] Chaudhari T. Fruit Scan - Disease Identification in Fruits Using Image Processing. International Journal for Research in Applied Science and Engineering Technology. 2024; 12(12): 3061–3065. https://doi.org/10.22214/ijraset.2024.59572.

[10] Shoaib M, Shah B, Ei-Sappagh S, et al. An advanced deep learning models-based plant disease detection: A review of recent research. Frontiers in Plant Science. 2023; 14. https://doi.org/10.3389/fpls.2023.1158933.

[11] Chaudhari T, Raut A, Jadhavrao D. Fruit Inspect - Disease Identification in Fruits using Image Processing. International Research Journal of Modernization in Engineering Technology and Science. 2023. https://doi.org/10.56726/irjmets45799.

[12] K S, Kamarasan M. Enhancing Tomato Fruit Disease Detection using Dung Beetle Optimization with Deep Transfer Learning based Feature Fusion Model. International Journal of Engineering Trends and Technology. 2024; 72(9): 127–138. https://doi.org/10.14445/22315381/ijett-v72i9p111.

Research Article

[13] Vo H, Mui KC, Thien NN, et al. Optimizing Grape Leaf Disease Identification Through Transfer Learning and Hyperparameter Tuning. International Journal of Advanced Computer Science and Applications. 2024; 15(2). https://doi.org/10.14569/ijacsa.2024.0150293.
[14] Raiaan MaK, Sakib S, Fahad NM, et al. A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks. Decision Analytics Journal. 2024; 11: 100470. https://doi.org/10.1016/j.dajour.2024.100470.
[15] Shankar K, Kumar S, Dutta AK, et al. An Automated Hyperparameter Tuning Recurrent Neural Network Model for Fruit Classification. Mathematics. 2022; 10(13): 2358. https://doi.org/10.3390/math10132358.
[16] Ukwuoma CC, Zhiguang Q, Heyat MBB, et al. Recent Advancements in Fruit Detection and Classification Using Deep Learning Techniques. Mathematical Problems in Engineering. 2022; 2022: 1–29. https://doi.org/10.1155/2022/9210947.
[17] Nasir IM, Bibi A, Shah JH, et al. Deep Learning-based Classification of Fruit Diseases: An Application for Precision Agriculture. Computers, Materials & Continua. 2020; 66(2): 1949–1962. https://doi.org/10.32604/cmc.2020.012945.
[18] Wang C, Liu S, Wang Y, et al. Application of Convolutional Neural Network-Based Detection Methods in Fresh Fruit Production: A Comprehensive Review. Frontiers in Plant Science. 2022; 13. https://doi.org/10.3389/fpls.2022.868745.
[19] Dhiman P, Manoharan P, Lilhore UK, et al. PFDI is a precise fruit disease identification model based on context data fusion with faster-CNN in an edge computing environment. EURASIP Journal on Advances in Signal Processing. 2023; 2023(1). https://doi.org/10.1186/s13634-023-01025-y.
[20] Firdaus MH, Utami E, Ariatmanto D. Detection And Classification of Citrus Diseases Based on A Combination of Features Using the Densenet-169 Model. SinkrOn. 2023; 8(4): 2592–2601. https://doi.org/10.33395/sinkron.v8i4.12974.