https://doi.org/10.1590/jatm.v18.1410 ORIGINAL PAPER

Analysis of Bayesian Hyperparameter Optimization Results in Deep Neural Networks for Wide-Angle Camera Geometric Characterization

Fabiano da Cruz Nogueira[1*], Luan Orion de Oliveira Baráuna Ferreira[1], Ruy Morgado de Castro[2], Elcio Hideiti Shiguemori[2]

1. Instituto Nacional de Pesquisas Espaciais – Programa de Pós-graduação – Curso de Computação Aplicada – São José dos Campos/SP – Brazil.
2. Departamento de Ciência e Tecnologia Aeroespacial – Instituto de Estudos Avançados – São José dos Campos/SP – Brazil.

*Corresponding author: nogueirafcn@fab.mil.br

ABSTRACT

This paper addresses the challenge of geometric camera characterization, a crucial process in various applications such as computer vision, aerial photogrammetry, remote sensing, and robotics. Traditional calibration methods rely on calibration targets or manually defined geometric structures, which limit automation and adaptability, especially in uncontrolled environments. To overcome these limitations, we propose an innovative approach based on deep learning, capable of estimating the camera's intrinsic parameters directly from a single image. The developed method integrates the Optuna hyperparameter optimizer, which utilizes Bayesian optimization to enhance model accuracy while reducing computational cost and training time. The application of this approach accelerates the search for optimal configurations for neural networks, ensuring an efficient balance between performance and architectural complexity. Experimental results demonstrate a significant improvement in model accuracy, with mean absolute errors and standard deviations in distortion rates at the hundredth-order magnitude level and focal length determination below 6 mm. Compared to the model without the integrated optimizer, there was an over 95% gain in reducing Mean Squared Error (MSE) and an 82% reduction in the standard deviation. This research significantly contributes to the enhancement of autonomous navigation and image-based positioning systems, providing a scalable and automated alternative to conventional modeling techniques (SDE).

Keywords: Optuna; Deep learning; Computer vision and image-based navigation.

INTRODUCTION

Precise camera characterization is fundamental for extracting accurate geometric information in fields such as computer vision, aerial photogrammetry, remote sensing, and robotics (Duane 1971; Maybank and Faugeras 1992; Weng *et al.* 1992; Zhang 2000). This task involves determining both intrinsic parameters, including sensor properties and lens distortion and extrinsic parameters, such as rotation and translation relative to a reference frame (Fischer *et al.* 2015; Marcato Junior and Tommaselli 2013; Samii *et al.* 2015).

Traditional characterization techniques differ depending on the camera system, including pinhole models, fisheye lenses, stereo setups, light-field sensors, event cameras, and LiDAR devices (Li *et al.* 2008). Many of these methods rely on manually extracted

Received: Apr. 5, 2025 | **Accepted:** Oct. 17, 2025
Peer Review History: Single Blind Peer Review.
Section editor: Alvaro Damião

CC BY

features or calibration targets, like checkerboards, which require human intervention and hinder automation (Bukhari and Dailey 2013; Gasparini *et al.* 2009; Shah and Aggarwal 1994; Zhang Z 1999). Geometry-based approaches use scene elements, such as lines or vanishing points, to establish 3D-2D correspondences, but their effectiveness is limited to carefully structured environments (Alemán-Flores *et al.* 2014; Barreto and Araujo 2005; Bukhari and Dailey 2013; Carroll *et al.* 2009). Self-calibration methods estimate parameters from sequences of images using multiview geometry, though their accuracy is constrained by the performance of feature detectors, which are sensitive to lighting and surface textures (Faugeras *et al.* 1992; Fraser 1997; Hartley 1994).

Despite advances in industrial and laboratory implementations (MathWorks 2023; OpenCV 2023), achieving automatic real-time camera calibration remains challenging, particularly in uncontrolled environments. Conventional calibration involves multiple stages, including data acquisition, intrinsic parameter computation, and final parameter adjustment (Zhang Z 1999). Detecting control points in aerial images is a major bottleneck that can be automated using computer vision, machine learning, and Convolutional Neural Networks (CNNs) (Habib *et al.* 2013). In applications involving aerial navigation based on imagery, the geometric calibration of sensors is crucial, as navigation accuracy depends on the geometric fidelity of the captured images (Badrloo *et al.* 2022). While conventional monocular and stereo approaches face limitations, deep learning offers the capability for faster and more robust perception, obstacle detection, and feature extraction, even in complex and unknown environments.

This work proposes a deep neural network framework for automatic camera characterization, estimating intrinsic and extrinsic parameters directly from single images without relying on calibration targets or manually defined structures. Neural networks are capable of modeling nonlinear relationships and automatically extracting relevant features from image data, which is particularly important when fixed patterns or calibration objects are unavailable (Yang J *et al.* 2021).

A critical factor for model performance is hyperparameter optimization, including layer size, neuron count, and activation functions, which directly impact accuracy, training time, and computational cost (Marcu and Grava 2021). We employ Optuna, a Bayesian optimization-based hyperparameter search framework, to systematically evaluate multiple configurations, select optimal parameters, and validate their performance on a separate dataset (Akiba *et al.* 2019). This iterative approach ensures robust hyperparameter selection and improves training efficiency while maintaining high characterization accuracy.

By integrating deep learning with optimized hyperparameters, the proposed framework enhances both the precision and efficiency of camera geometric characterization, facilitating real-time applications in autonomous aerial navigation. The study advances current methodologies and offers guidance for future research in deep learning-based camera calibration for aerial imaging and robotics.

RELATED WORK

Types of Camera Models

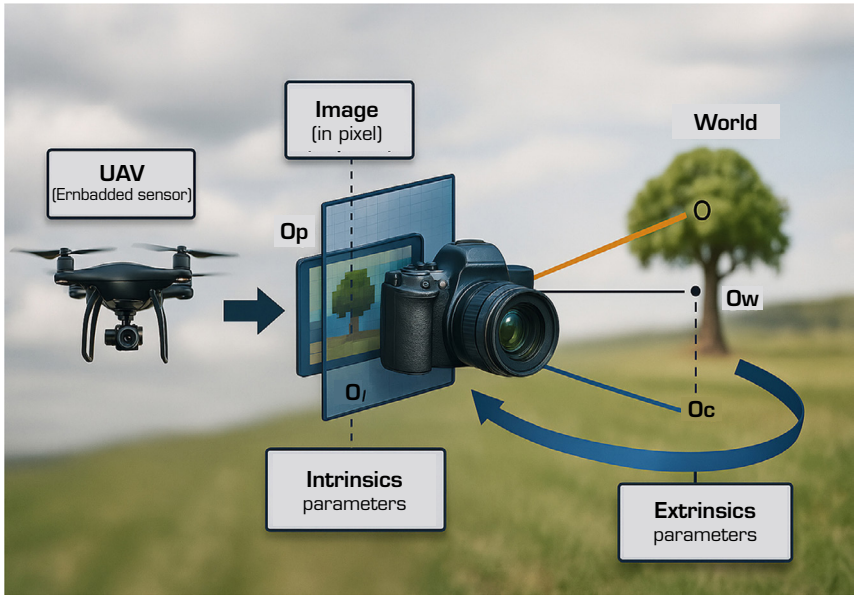
Camera models and their geometric characterizations are fundamental in computer vision and other related fields discussed in Introduction, as they provide a mathematical description of the image formation process and support accurate calibration procedures. These models aim to establish the relationship between 3D world coordinates and their projection onto a 2D image plane, requiring the estimation of intrinsic and extrinsic parameters.

The most commonly adopted configuration is the pinhole camera model, which represents the camera as a single projection center, where light rays intersect and are mapped onto the image plane. Due to its mathematical simplicity and wide applicability, it has been extensively used in theoretical and practical studies (Mahmood *et al.* 2024). However, real optical systems often deviate from this idealized model because of lens distortions, especially in wide-angle or fisheye configurations. To address these deviations, the distortion camera model incorporates additional parameters to account for radial and tangential distortions, allowing more accurate geometric representation after calibration (Huai *et al.* 2024).

Beyond these approaches, advanced imaging systems employ the omnidirectional camera model, designed to capture a 360-degree field of view using specific lenses or mirror-based assemblies. Calibration of these systems requires more complex geometric formulations and often specialized calibration patterns (Caron 2023).

Figure 1 illustrates the general framework of the geometric characterization process, highlighting the coordinate systems involved, the object space coordinate system and the camera coordinate system as well as the intrinsic and extrinsic parameters to be estimated.

This formulation underpins the methodological basis for wide-angle camera characterization, which constitutes the primary focus of this study.



Source: Elaborated by the authors.

Figure 1. Camera model and systems involved in geometric characterization.

Deep Learning Paradigms for Camera Geometric Characterization

Recent advances in deep learning have enabled data-driven alternatives to traditional camera calibration techniques. In intrinsic calibration, methods such as DeepFocal (Workman *et al.* 2015) estimate focal length directly from images using CNNs. Pak *et al.* (2022) integrated active targets and machine learning for high-precision geometric calibration, while Cramariuc *et al.* (2020) introduced APPD, a metric for detecting intrinsic miscalibration over time.

For extrinsic calibration, PoseNet (Kendall *et al.* 2015) pioneered camera pose estimation via CNNs, later extended by DeepFEPE (Jau *et al.* 2020) with learnable modules for feature matching and pose refinement (Tan et al. 2024). Further works explored intermediate representations like surface geometry and depth (Xian *et al.* 2019), as well as sparse 3D landmarks for lightweight calibration (Do *et al.* 2022).

Joint calibration methods aim to simultaneously recover intrinsic and extrinsic parameters. Hold-Geoffroy *et al.* (2018) trained models on panoramic data, proposing a perceptual metric for calibration accuracy. Song *et al.* (2024) combined semantic and geometric features to improve perspective understanding.

Learning-based calibration has also expanded to distortion correction. For radial distortion, DeepCalib (Bogdan *et al.* 2018) and SimFIR (Feng *et al.* 2023) used CNNs and Vision Transformers, while PSEGAN (Shi *et al.* 2018) and OrdinalDistortion (Liao *et al.* 2021) improved interpretability. Reconstruction-based methods, such as DR-GAN (Liao *et al.* 2019b) and DDM (Liao *et al.* 2020), model pixel-wise mappings to rectify distortions without explicit camera parameters. Architectural improvements include FE-GAN (Chao *et al.* 2020), Progressively Complementary Network-PCN (Yang S *et al.* 2021), and PolarRecNet (Zhao *et al.* 2021), which address symmetry and skip-connection artifacts in wide-angle rectification. Additionally, single-frame rolling shutter distortion correction has gained attention in mobile and Unmanned Aerial Vehicle-UAV applications (Zhuang *et al.* 2019), though multi-frame methods (Fan *et al.* 2021) remain outside the scope of this review.

CC BY

Hyperparameters and Tuning Strategies in Deep Learning

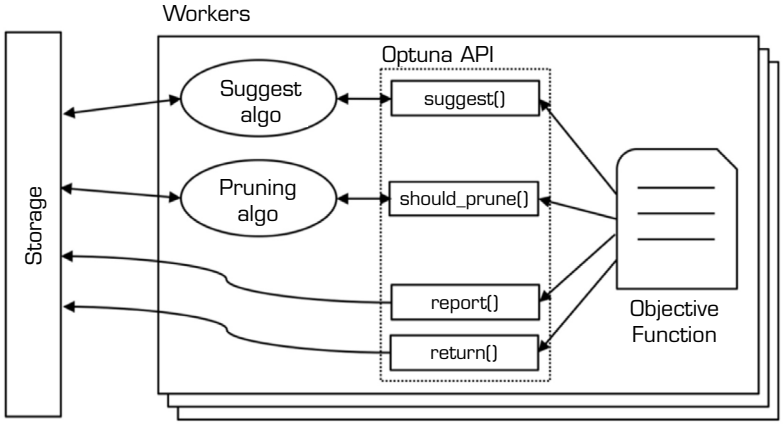
In deep learning, hyperparameters are predefined configurations set before training, directly influencing model performance (Wang *et al.* 2024). Unlike model parameters, which are adjusted during training, hyperparameters control structural and procedural aspects such as complexity, learning rate, and generalization (Balcan *et al.* 2025). Their proper selection, achieved through experimentation and fine-tuning, is essential for optimizing model performance.

Among the key hyperparameters are learning rate, batch size, number of epochs, number of layers and units per layer, dropout rate, weight initialization methods, optimizer types (SGD, Adam, RMSprop), activation functions (ReLU, Sigmoid, Tanh), regularization parameters (L1, L2), kernel and pooling size (for convolutional layers), stride and dilation rate (for convolution and pooling), among others.

Hyperparameter tuning is crucial for developing effective deep learning models (Franceschi *et al.* 2024; Lai 2024). Techniques such as grid search, random search, Bayesian optimization, and evolutionary algorithms provide distinct approaches to exploring the hyperparameter space.

Optuna exhibits superior performance in the combined optimization of algorithm selection and hyperparameters (CASH) compared to frameworks such as HyperOpt and SMAC (Shekhar *et al.* 2021). It provides a dynamic search space through the define-by-run principle, enabling adaptive adjustments not available in traditional methods (Akiba *et al.* 2019), and has demonstrated robustness and adaptability in extensive experiments with real-world datasets (Shekhar *et al.* 2021).

During the trial execution, the objective function leverages the Optuna APIs (Fig. 2).



Source: Akiba *et al.* (2019).

Figure 2. Overview of Optuna's system design.

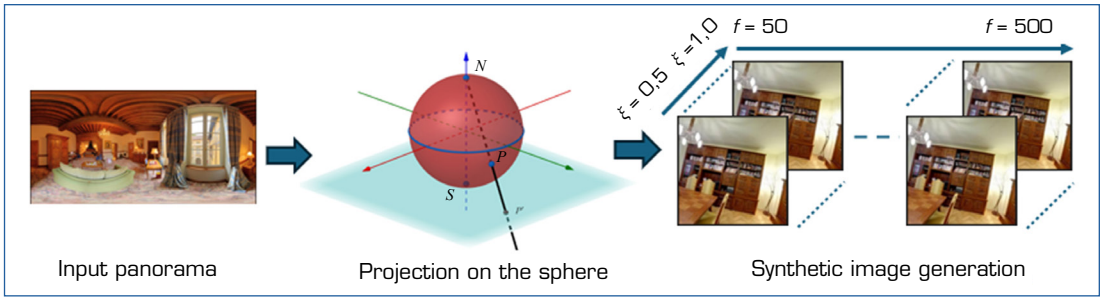
Each process (worker) executes an instance of the objective function for each study. During the trial execution, the objective function leverages the Optuna APIs. Whenever the API is invoked, the objective function accesses the shared storage and retrieves, if necessary, information from previous studies. Each worker operates the objective function independently while simultaneously updating the progress of the current study in the storage.

METHODS

Dataset preparation

This work proposes a deep learning-based method for estimating intrinsic camera parameters, focal length, and distortion coefficient from single images, focusing on wide field-of-view configurations. Due to the absence of large-scale datasets with ground-truth calibration for such conditions, a synthetic dataset was generated from high-resolution 360° panoramas (Chang *et al.* 2017), allowing full control over field-of-view and distortion settings without introducing occlusion artifacts.

The dataset construction was carried out in two main steps. Initially, panoramic images were projected onto a unit sphere through azimuth and elevation transformations (Barreto and Araujo 2005), enabling geometric consistency across diverse viewpoints. Subsequently, a virtual pinhole camera was applied to generate synthetic images by reprojecting selected regions of the sphere with predefined intrinsic parameters specifically varying focal lengths (f) and distortion coefficients (ξ). This process, illustrated in Figs. 3 and 4, enabled the automatic generation of 17,000 training samples with wide coverage of optical distortions, spatial perspectives, and lighting scenarios, without introducing occlusion artifacts.



Source: Elaborated by the authors using data from Barreto and Araujo (2005), Mei and Rives (2007)and Nogueira *et al.* (2025).

Figure 3. Input panorama, automatic generation of images with varying focal lengths f and distortion values ξ, using a unified spherical model.

Source: Elaborated by the authors using data from Nogueira *et al.* (2025).

Figure 4. Dataset of images with varying focal lengths f and distortion values ξ.

The projection and reprojection processes follow Eqs. 1–5:

$$p = (x, y) = \left(\frac{X \cdot f}{\xi \cdot \sqrt{X^2 + Y^2 + Z^2} + Z} + v_0, \frac{X \cdot f}{\xi \cdot \sqrt{X^2 + Y^2 + Z^2} + Z} + u_0 \right) \tag{1}$$

$$P_{s=(\omega \hat{} x, \omega \hat{} y, \omega - \xi))} \tag{2}$$

where ω is defined as:

$$\omega = \frac{\xi \cdot \sqrt{1 + (1 + \xi^2) \cdot (\hat{x}^2 + \hat{y}^2)}}{\hat{x}^2 + \hat{y}^2 + 1} \tag{3}$$

and

CC BY

$$[\omega \cdot \hat{x}\omega \cdot \hat{y}\ 1]^T \cong K^{-1} \cdot p \tag{4}$$

where K is the camera intrinsic matrix:

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

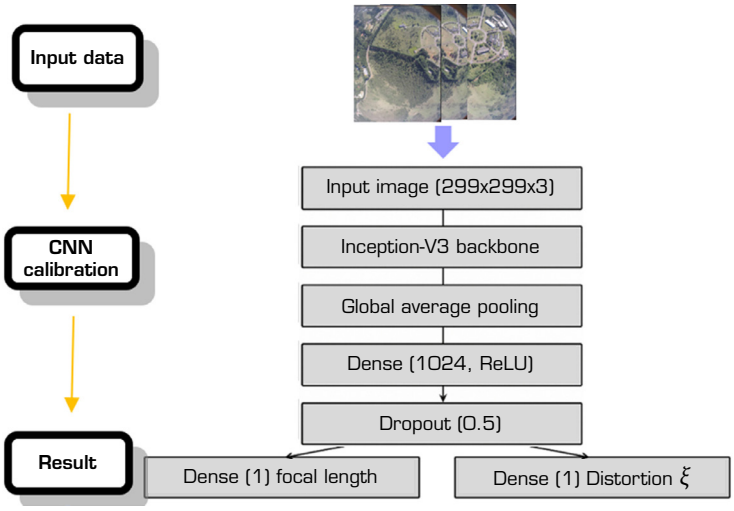
Equations 1–5 describe the geometric modeling based on the unified spherical camera model (Barreto and Araujo 2005), with the following variable definitions:

- $p = (x, y)$: image coordinates of the projected point.
- (X, Y, Z): 3D coordinates of the point in space.
- f: camera focal length.
- ξ: distortion parameter that defines the curvature in the unified projection model.
- u_0, v_0: image coordinates of the optical center (principal point).
- P_s: projected point on the unit sphere.
- $\hat{x}, \hat{y}$: normalized image coordinates.
- ω: radial scaling factor derived from the spherical model.
- K: intrinsic camera matrix comprising the parameters f, u_0, and v_0.

Optimization of Neural Network Architecture Based on a Standard Model: Evaluation and Analysis

Standard Architecture of the Applied Neural Network

Given a single image as input, we adopted a regression-based CNN architecture referred to as SingleNet (Bogdan *et al.* 2018), featuring two output branches for estimating f (focal length) and ξ (distortion). For feature extraction, three widely used CNN backbones were evaluated: Inception-V3 (Szegedy *et al.* 2015), ResNet50 (He *et al.* 2016), and MobileNet (Howard *et al.* 2017). These architectures were selected due to their distinct trade-offs between depth, computational cost, and representational power. Inception-V3 offers robust multi-scale feature aggregation, ResNet50 enables deep residual learning, and MobileNet is optimized for lightweight inference, making it suitable for embedded scenarios. Figure 5 illustrates a general example of the architectural configuration employed.



Source: Elaborated by the authors.

Figure 5. Dataset of images with varying Focal Lengths (f) and Distortion Values (ξ).

SingleNet is composed of a single network, which makes both the training process and the feedforward execution approximately twice as fast when compared to the other architectures considered, such as DualNet and SeqNet, both consisting of two independent networks (Bogdan *et al.* 2018). Considering the potential for embedding the solution in future systems for real-time processing, this neural network configuration was selected.

Based on the previously presented architecture and the results obtained by Nogueira *et al.* (2025), an optimization study was conducted using the Optuna framework to identify a neural network architecture and a set of hyperparameters with improved performance and accuracy. Following this optimization, a comparative analysis was performed between the results of both studies, evaluating the performance gain of the optimized model and its accuracy relative to the reference and the results reported in Nogueira *et al.* (2025).

Optuna: Implementation for Hyperparameter Tuning

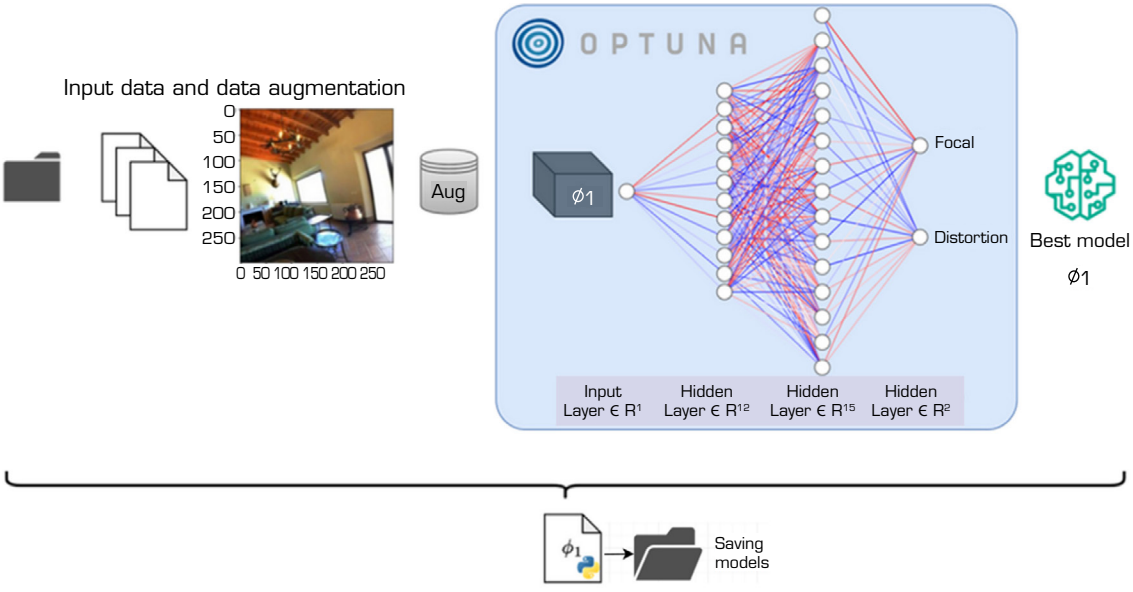
The implementation of Optuna involved defining an objective function to train and evaluate the model, configuring the hyperparameter search space, and executing the optimization process. Additionally, the Optuna library enabled interactive visualization of the hyperparameter exploration through a dedicated dashboard.

Optuna is a modern framework for hyperparameter optimization that leverages advanced sampling and pruning strategies to efficiently explore the hyperparameter space. It employs the Tree-structured Parzen Estimator (Watanabe 2023) for Bayesian optimization and integrates a pruning mechanism based on intermediate results, enabling the early termination of less promising experiments (Akiba *et al.* 2019).

Optimization of Neural Networks and Hyperparameters using Optuna Study

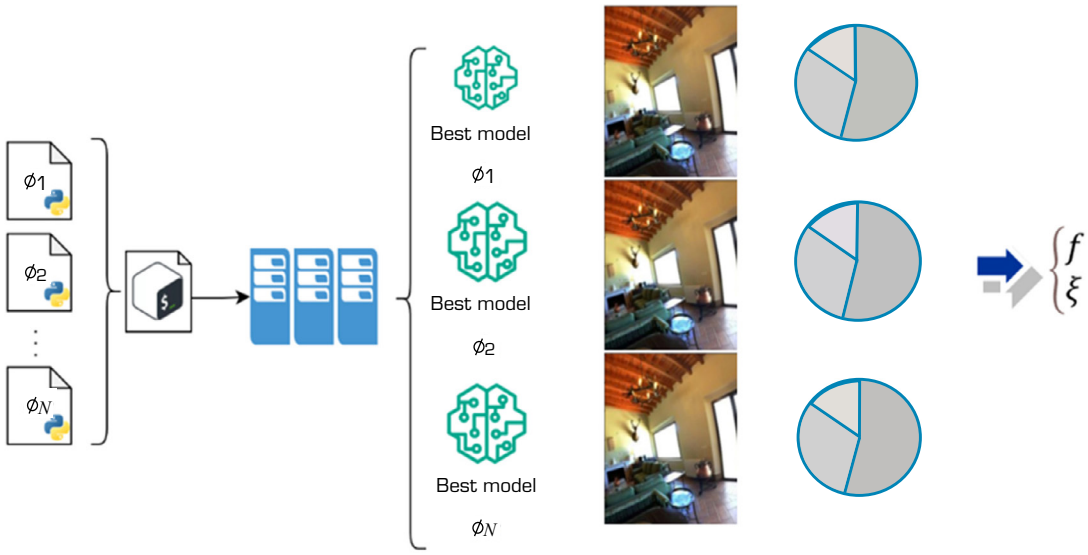
Hyperparameter optimization, as discussed in Section "Hyperparameters and Tuning Strategies in Deep Learning," is crucial in neural network development, directly impacting performance (Yang L and Shami 2020). The number of neurons per layer and the choice of activation function influence the model's capacity, but larger networks increase computational costs, training time, and overfitting. To identify the optimal architecture, the Optuna framework (Akiba *et al.* 2019) was employed, systematically evaluating hyperparameter combinations by training the model and analyzing metrics such as accuracy and loss function on the validation set.

The experimental workflow involves inputting the data x into the network, consisting of a single image, which generates vectors with two model outputs: one for distortion estimation (ξ) and another for focal length estimation (*f*), as illustrated in Figs. 6 and 7.



Source: Elaborated by the authors.

Figure 6. Overview of neural network and hyperparameter optimization – training flowchart.



Source: Elaborated by the authors.

Figure 7. Overview of neural network and hyperparameter optimization: activation flowchart.

After training, the network is evaluated using new data. Following the same procedure, the model's output is processed and compared with the validation target data. The test data values serve as metrics for exploring the hyperparameter space or conducting experiments within the Optuna study.

All models were trained for 1,000 epochs and 10 trials. This limit was established based on empirical evaluations and computational constraints at the time of the experiments. The findings indicated that the networks achieved satisfactory performance across various architectures within this training period. This timeframe also allowed for the observation of stability between training and validation phases.

The Optuna hyperparameter optimizer employs a search process with dependencies on random values, as suggested in its original documentation. Although it is possible to define a seed for hyperparameter search, doing so may compromise the performance of data mining for knowledge discovery.

It is important to highlight that the experiments remain reproducible under the Optuna hyperparameter optimizer, which, as indicated in its original documentation, utilizes a search approach based on randomly seeded values.

Experimental Methodology and Model Evaluation under Real Conditions

The methodology proposed in this study was developed following a structured approach, aiming to ensure high accuracy in position estimation through monocular visual odometry. The initial phase consisted of conducting a photogrammetric flight supported by Global Navigation Satellite System/Global Positioning System (GPS) data, with aerial image acquisition performed using a GoPro camera over a controlled and orthorectified geographic area.

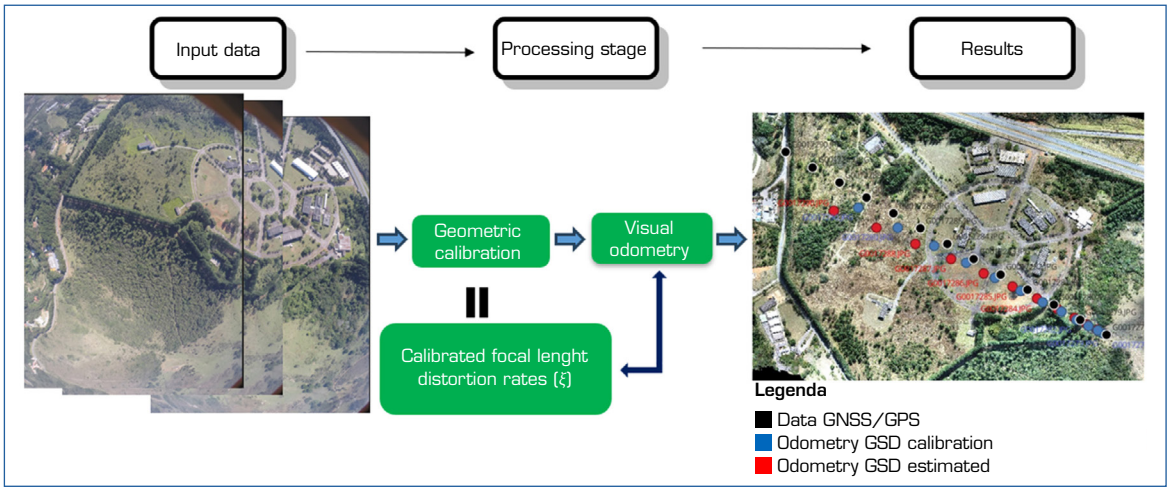
The methodology incorporated a real-time dynamic calibration module to continuously adjust the camera's intrinsic parameters during flight. The impact on processing performance was evaluated on a per-frame basis; however, the assessment of onboard hardware resource utilization has not yet been conducted, as it is currently under implementation and will be reported in future work.

The intrinsic parameters were preadjusted through supervised training using both synthetic and real datasets, generated and processed as described in Section "Dataset Preparation," employing metrics such as accuracy, root mean square error, and loss function to ensure robustness and generalization under real flight conditions.

Figure 8 presents examples of aerial images used for training, testing, and validation, aiming to maximize the solution's generalization and ensure optimized performance in image-based navigation. The adopted methodology was structured as illustrated in Figs. 5 and 9, while the corresponding results are presented, analyzed, and discussed in Results and Discussion, Subsection "Analysis and Testing Using Aerial Imagery Data and Navigation".

Source: Elaborated by the authors.

Figure 8. Image data for experiments under real-world conditions.



Source: Elaborated by the authors.

Figure 9. Diagram illustrating the sequence of steps in the dynamic geometric calibration process.

RESULTS AND DISCUSSION

Parameters of the Network

For training the networks presented in Section "Optimization of Neural Network Architecture Based on a Standard Model: Evaluation and Analysis," two large-scale datasets were generated, each containing thousands of images with resolutions of 299×299 px and 224×224 px, extracted from all panoramic images in the dataset and processed as described in Section "Dataset preparation". This approach was adopted to enable a comparative analysis of three neural network configurations: InceptionV3, MobileNet, and ResNet50, following the same experimental setup as presented in Nogueira *et al.* (2025). The objective is to analyze and demonstrate the improvement in training and prediction performance of intrinsic parameters when compared to the results obtained using conventional neural network training methods.

For training, focal lengths were used within the range of 50 to 500 px, with a step size of 10, and distortion values varying from 0 to 1.2, with a step size of 0.02 (randomly sampled).

The dataset was divided into three subsets: 80% for training, 10% for testing, and 10% for validation. Each original panorama was exclusively assigned to a single stage (training, testing, or validation), ensuring that no original panorama was present in more than one subset.

Standard data augmentation techniques were applied, including the addition of random Gaussian noise, modifications to brightness and contrast, and image mirroring.

The three networks used were pre-trained on the ImageNet dataset and subsequently fine-tuned on the generated dataset, employing an early stopping strategy to prevent overfitting.

CC BY

The training and activation process followed the pipeline illustrated in Figures 6 and 7. Unlike the study conducted in (Nogueira *et al.* 2025), this experiment adopted a distinct approach aimed at determining the optimal neural network architecture and the most suitable hyperparameters. To achieve this, optimization was performed based on studies conducted using the Optuna framework.

The following sections present and analyze some of the results obtained from these experiments, highlighting key improvements observed during both the training and validation stages.

The results obtained in this study suggest that hyperparameter optimization can significantly contribute to enhancing the performance of deep neural networks applied to the geometric characterization of cameras. The conducted experiments indicated that the use of the Optuna framework proved to be an effective strategy for performing fine-grained adjustments, which led to improved prediction of the camera's intrinsic parameters.

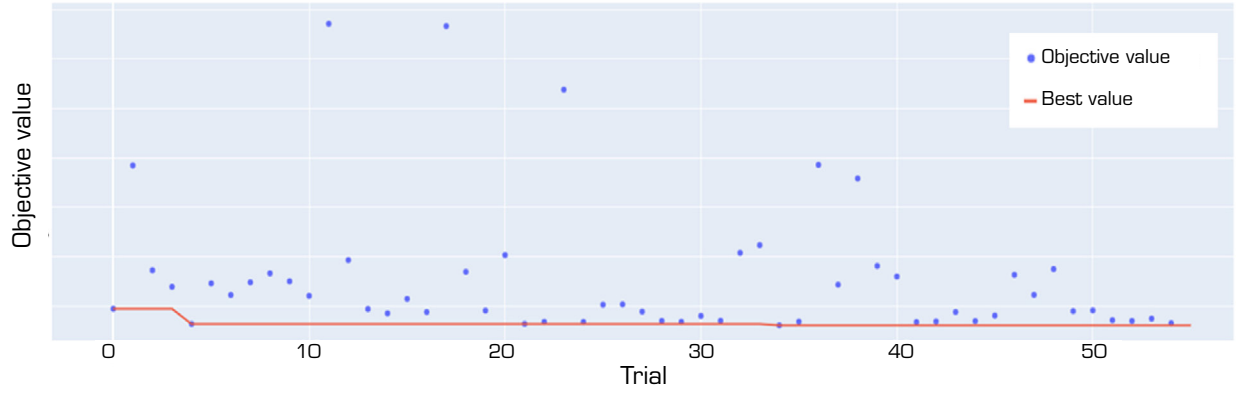
There is a clear interdependence between the model's hyperparameters and its evaluation metrics. Empirical results demonstrate that specific configurations such as variations in learning rate, the number of neurons per layer, and the choice of activation functions have a significant impact on network performance. These findings emphasize the critical need for robust and systematic hyperparameter optimization strategies to enhance model accuracy, as presented in Table 1.

Table 1. Optimized hyperparameters.

Range of hyperparameters	Search Space	Type	Best Value
Hidden layers (n_layers_Hidden)	2 – 12	int	6
Number of neurons per layer (n_neurons_layer_i)	32 – 256	int	128
Activation function (activation_layer_i)	[relu, sigmoid, tanh]	categorical	relu
Dropout rate (dropout_rate_layer_i)	0.0 – 0.5	float	0.2
Network optimizer (optimizer)	[adam, sgd]	categorical	adam
Learning rate (learning_rate)	1e-5 – 1e-2 (log)	float	1e-3
Momentum (SGD)	0.0 – 0.9	float	0.8

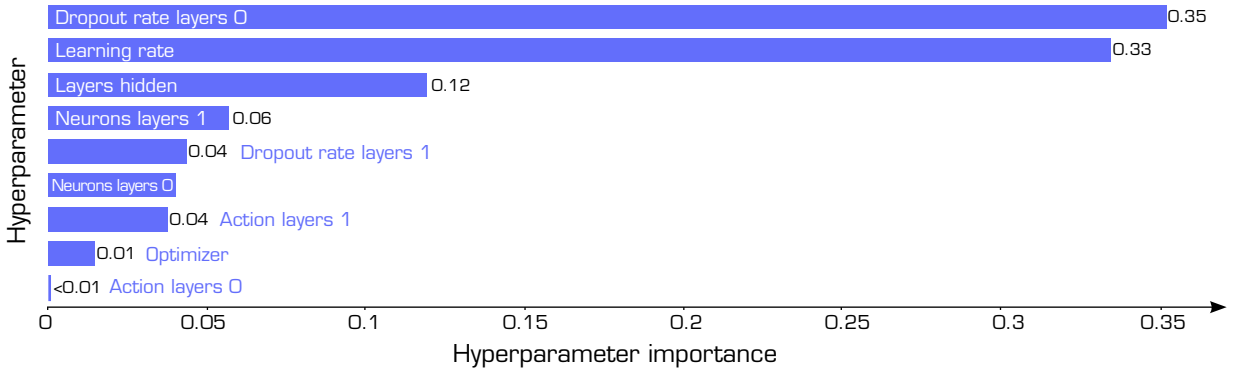
Source: Elaborated by the authors.

The evolution of hyperparameter optimization is depicted in the optimization history (Fig. 10). Initially, the objective metric values were high, indicating that the early tested configurations were suboptimal. The importance of hyperparameters (Fig. 11) provides valuable insights into which parameters exert the greatest influence on the performance of the trained neural network.



Source: Elaborated by the authors.

Figure 10. Optimization history plot.



Source: Elaborated by the authors.

Figure 11. Relative Importance of tuned parameters.

The number of neurons in the second hidden layer (*n* neurons layer 1) was identified as the most critical factor for model performance, suggesting that adjustments in this layer can significantly impact the results.

Similarly, the activation function of the first layer (activation layer 0) and the dropout rate of the second layer (dropout rate layer 1) demonstrated consider able relevance, highlighting the importance of regularization strategies to prevent overfitting.

Other parameters, such as the learning rate and optimizer, also showed some impact, albeit to a lesser extent. Conversely, the number of hidden layers (n layers hidden) had an insignificant influence, indicating that adding more layers did not provide substantial benefits for this specific model.

However, as the trials progressed, a gradual reduction in error was observed, demonstrating that Optuna successfully identified more efficient hyperparameter combinations.

The stabilization of the red line suggests that convergence was achieved, indicating that the model reached an optimization point where further adjustments would yield only marginal improvements.

The optimization results demonstrated that:

- A balanced network depth and width were determinant for overall performance.
- The ReLU activation function outperformed the alternative functions tested.
- Moderate dropout regularization was critical to preventing overfitting.
- The Adam optimizer, with a learning rate close to 10^{-3} (*i.e.*, 0.001), provided the best trade-off between stability and convergence speed.

These findings reinforce that the effectiveness of the model does not rely on a single hyperparameter, but rather on the synergy among architecture, regularization, and optimization strategy.

Evaluation

The results presented in Tables 2 and 3 show the distortion rate and focal length predictions for different deep neural network architectures optimized using Optuna. The statistical error analysis provides valuable insights for identifying the most suitable architecture for this task.

Table 2. Analysis of absolute errors between predictions and actual distortion rate values.

Model	MAE	SD	Max E	Min E
Inception-V3	0.059	0.051	0.110	0.01
Mobilenet	0.100	0.190	0.290	0.01
ResNet50	0.070	0.002	0.005	0.00

The distortion rate is a dimensionless measure. Standard Deviation (SD), Maximum Error (Max E) and Minimum Error (Min E).
Source: Elaborated by the authors.

CC BY

Table 3. Analysis of absolute errors between predictions and actual focal length values.

Model	MAE (mm)	SD (mm)	Max E (mm)	Min E (mm)
Inception-V3	6.85	5.48	10.86	3.30
Mobilenet	42.58	44.01	67.53	7.63
ResNet50	125.64	0.31	125.88	125.20

The values are expressed in millimeters (mm). Standard Deviation-SD, Maximum Error-Max E and Minimum Error-Min E.
Source: Elaborated by the authors.

To assess the accuracy of the models, the following statistical metrics were calculated, based on Vujović (2021) and Chicco *et al.* (2021):

- Mean Absolute Error (MAE): Measures the average of the absolute differences between the predicted and actual values.
- Maximum Error (Max E): Represents the largest absolute difference observed between the actual value (Label) and the model's prediction. This value indicates the worst-case error committed by the model, i.e., the greatest discrepancy between the prediction and the expected value.
- Minimum Error (Min E): Represents the smallest absolute difference observed between the actual value and the prediction. This value indicates the best-case error, showing how close the model was able to approximate the actual value in at least one test instance.
- Standard Deviation (SD) of Errors: Evaluates the dispersion of the predictions relative to the actual value, indicating the consistency of the model's performance.

The results in Table 2 indicate that the model using Inception-V3 achieved the lowest Mean Absolute Error (MAE) of 0.059, making it the most precise option. The ResNet50-based model, although yielding a slightly higher MAE (0.070), achieved a low standard deviation (0.002), indicating consistent predictive performance with minor deviation from the actual values. In contrast, the model using MobileNet demonstrated the worst performance, with an MAE of 0.100 and a high standard deviation (0.19), indicating significant variability in its predictions.

The stability of the ResNet50 model can be attributed to its ability to maintain consistent predictions, whereas the superior accuracy of the Inception-V3 model suggests a better adaptation to data patterns. On the other hand, the MobileNet model may have been affected by overfitting or difficulties in generalizing the data structure, resulting in greater variability in predictions.

In conclusion, in the characterization of camera distortion rates, the model with Inception-V3 demonstrated the best balance between accuracy and stability. Conversely, the model with ResNet50 exhibited limitations during the learning phase, displaying uniformity in predictions that does not adequately reflect the diversity of scenarios and parameters present in the test data. The use of Optuna for optimization was effective in detecting these differences and improving model performance.

The results in Table 3 indicate that the Inception-V3 model exhibited the lowest MAE of 6.85 mm, making it the most precise option for estimating focal length. The ResNet50 model, despite having an extremely low standard deviation (0.31 mm), showed a significant offset from the actual values, resulting in a MAE of 125.64 mm, rendering its predictions inaccurate for this task. The MobileNet model demonstrated an intermediate performance but still exhibited a relatively high MAE of 42.58 mm.

The low standard deviation observed in ResNet50 suggests possible limitations during the learning phase, indicating that its predictions exhibit notable inconsistency and appreciable deviation from the actual values. This may imply a failure of the model to adapt to the characterization data pattern. In contrast, the MobileNet model exhibited greater variability in predictions, resulting in lower overall accuracy.

The Inception-V3 model, with a moderate standard deviation, managed to balance precision and stability, making it the most suitable choice for Focal Length estimation.

These results reinforce that the optimization performed with Optuna enabled the identification of Inception-V3 as the most efficient solution for focal length estimation, ensuring lower errors and predictions closer to the actual values. Selecting the appropriate model is crucial for applications in camera characterization and computer vision, where focal length accuracy directly impacts image reconstruction quality, spatial perception, and navigation outcomes, among other aspects.

Comparison with Recent Training Results of Deep Neural Networks for Camera Characterization

The results of this study, presented in Tables 4 to 7, indicates a significant improvement in the performance of the neural network specialized in camera geometric characterization after hyperparameter optimization. This optimization was performed using the Optuna framework on the AeroCalib network, previously developed and documented in (Nogueira *et al.* 2025).

Table 5 highlights the significant reduction in errors following the optimization process, further supported by the comparative statistical metric graphs. Three key aspects illustrate this improvement: MAE: Reduced from 0.187 to 0.046, corresponding to a 75.4% decrease, reflecting more precise predictions. MSE: Dropped from 0.0613 to 0.0027, representing a 95.6% reduction, highlighting a greater correction of extreme deviations.

Table 4. Summary example of results for distortion rate comparison using the best-performing model with and without optimization by Optuna.

Images	Label (Real)	Prediction (Without Optuna)	Prediction (With Optuna)
1	0.39	0.20	0.43
4	0.58	0.74	0.52
5	0.69	0.70	0.60
9	0.52	0.68	0.46
12	0.73	0.14	0.66
14	0.01	0.08	0.04
16	0.23	0.48	0.21
17	0.78	0.46	0.79
20	0.74	0.78	0.68
28	0.89	0.97	0.87

Source: Elaborated by the authors.

Table 5. Statistical error metrics for distortion rate using inception-v3, with and without optimization by Optuna.

Metrics	Without Optuna (Nogueira et al. 2025)	With Optuna	Reduction (%)
Mean Absolute Error (MAE)	0.187	0.046	75.40
Mean Squared Error (MSE)	0.061	0.003	95.56
Mean Error	0.033	0.030	09.09
Standard Deviation of Errors (SDE)	0.245	0.043	82.62

Source: Elaborated by the authors.

Table 6. Summary example of results for comparing focal length (mm) estimation using the best-performing model with and without optimization by Optuna.

Images	Label	Prediction (without Optuna)	Prediction (with Optuna)
5	183.00	160.83	174.10
7	231.00	255.83	225.04
23	140.00	79.61	135.15
24	146.00	138.63	139.12
29	248.00	252.63	243.70
30	255.00	146.64	262.89
32	51.00	96.17	47.77
37	154.00	88.74	146.14
41	76.00	69.12	76.73
48	412.00	368.34	405.96

Source: Elaborated by the authors.

CC BY

Table 7. Statistical error metrics for models with and without Optuna (focal length).

Metrics	without Optuna (Nogueira *et al.* 2025)	With Optuna	Reduction (%)
Mean Absolute Error (MAE)	38.8700	5.66	85.43
Mean Squared Error (MSE)	2482.54	37.56	98.49
Mean Error	23.9500	3.94	83.55
Standard Deviation of Errors (SDE)	43.6900	4.69	89.26

Source: Elaborated by the authors.

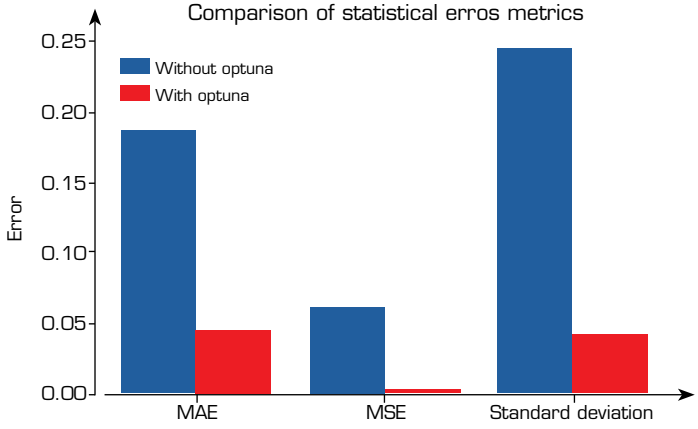
Standard deviation of errors: Decreased from 0.2454 to 0.0427, an 82.6% reduction, indicating greater stability in predictions. Figures 10 and 11 visually illustrate these improvements, confirming the effectiveness of hyperparameter optimization in reducing model errors.

In the analysis of focal length estimation (Table 6), the improvements achieved are of greater magnitude.

- MAE: Reduced by 85.4%, from 38.87 mm to 5.66 mm.
- MSE: Decreased by 98.5%, from 2482.54 mm^2 to 37.56 mm^2.

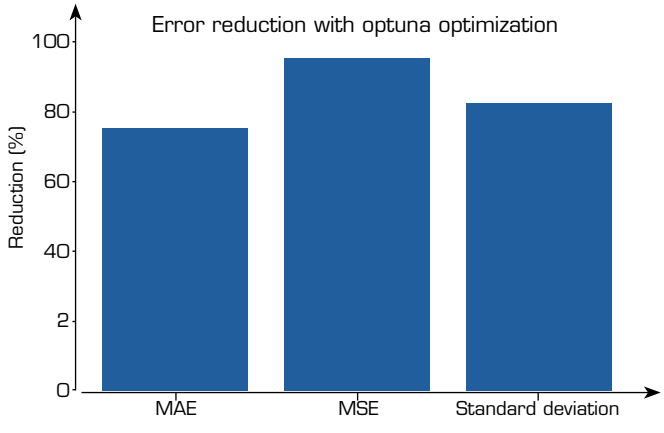
Standard Deviation: Reduced by 89.3%, indicating greater consistency in estimates.

Figures 12 to 15 visually reinforce these findings, highlighting the positive impact of optimization on focal length estimation.



Source: Elaborated by the authors.

Figure 12. Comparison of Statistical Error Metrics Before and After Optuna Optimization.



Source: Elaborated by the authors.

Figure 13. Error reduction achieved through Optuna optimization.

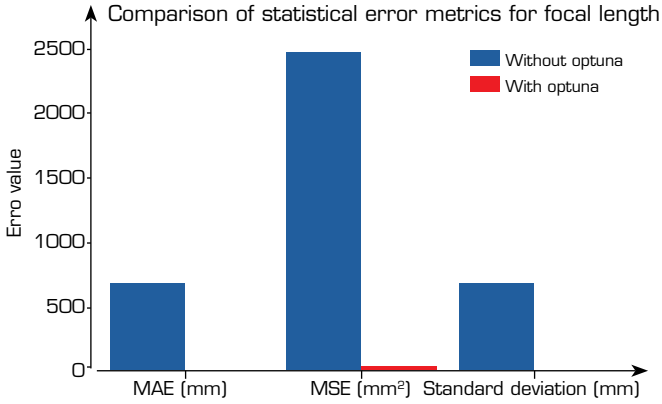


Figure. 14. Comparison of statistical error metrics before and after Optuna optimization for focal length.

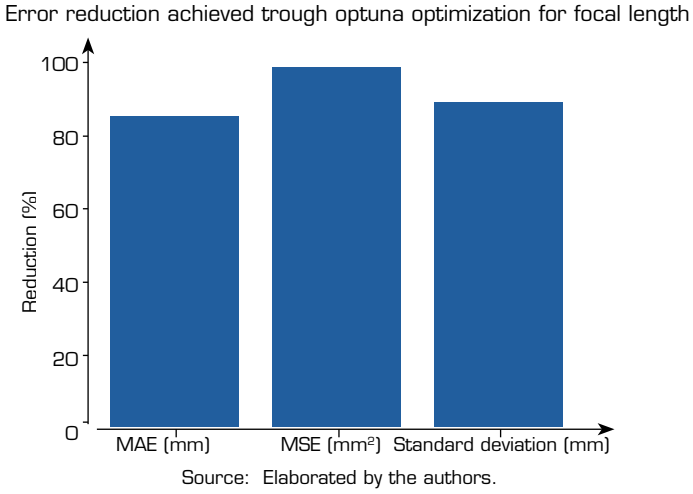


Figure. 15. Error reduction achieved through Optuna optimization for focal length.

These experimental studies demonstrate that hyperparameter optimization using Optuna played a crucial role in enhancing the model's performance, leading to more precise and reliable predictions. The substantial reduction in both absolute and quadratic mean errors for distortion rate and focal length further confirms the effectiveness of the adopted approach.

The application of Optuna enabled an efficient exploration of the neural network's parameter space, even with a limited number of 10 trials. These results suggest that implementing this process in a high-performance computing infrastructure, with a greater number of iterations and a more refined search space, could yield even more significant improvements. The ability to test thousands of hyperparameter combinations would enhance the model's accuracy, bringing it closer to optimal performance.

Thus, the obtained results emphasize the importance of hyperparameter optimization in deep neural networks, particularly for computer vision applications focused on camera calibration and characterization.

The improvement in performance can be explained by the following factors intelligent hyperparameter search:

- Optuna employs advanced techniques, such as the Tree-structured Parzen Estimator, to identify optimal hyperparameter combinations, avoiding less efficient approaches like grid search.
- Reduction of Overfitting and Improved Generalization: Adjustments in learning rate, regularization, and neural network architecture help minimize overfitting, allowing the model to generalize better to new data.
- Acceleration of Model Convergence: Optimization refined the network's learning process, ensuring that adjusted weights minimized prediction errors more efficiently.

CC BY

• Optimization of Cost Function and Regularization: The appropriate selection of hyperparameters contributed to a balance between accuracy and stability, avoiding excessive penalties and ensuring a more robust model.

Efficient hyperparameter optimization is one of the key advantages of Optuna, especially in the context of deep neural networks, where overfitting can be a significant challenge. Optuna allows the adjustment of regularization-related parameters, such as dropout rate and L2 regularization, among others. During the optimization process, Optuna seeks hyperparameter combinations that minimize excessive model complexity, preventing overfitting by avoiding overfitting to the training data. This approach ensures that the model does not become overly complex or specific to the training dataset, thus promoting its generalization to new data.

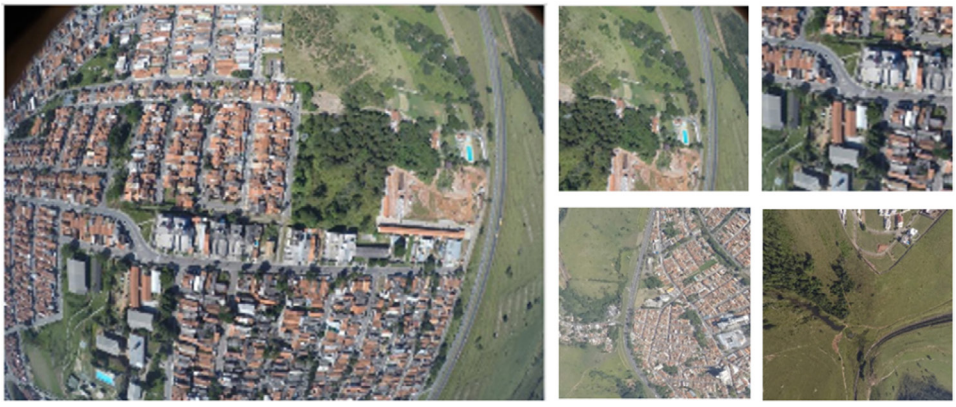
Another important strategy for mitigating overfitting is the use of early stopping during training. Optuna enables the integration of this technique into the optimization process. Early stopping halts training once the model's performance on the validation data starts to deteriorate, indicating that the model is beginning to memorize the training data rather than learning general patterns. This approach prevents the model from continuing to learn spurious patterns from synthetic data, which may not be representative of real-world data and therefore not generalizable.

Although Optuna does not directly implement data augmentation techniques, in this work, it has been configured to optimize hyperparameter combinations that incorporate these techniques as part of the training process. Data augmentation exposes the model to a greater diversity of data variations, such as changes in lighting, zoom, rotation, and distortions, which enhances the robustness of the model and improves its generalization capability. By adjusting hyperparameters that control the application of these techniques, Optuna helps reduce overfitting and improves the model's ability to handle unexpected variabilities.

Furthermore, training with real-world data plays a crucial role in mitigating overfitting. When the model is exposed to data that represents real-world conditions, such as noise, lighting variations, and other imperfections, it is forced to learn how to generalize from more complex and variable information. Optuna facilitates model validation under real-world conditions, helping evaluate the robustness and effectiveness of the model in uncontrolled environments, thereby reducing the likelihood of overfitting to the synthetic dataset.

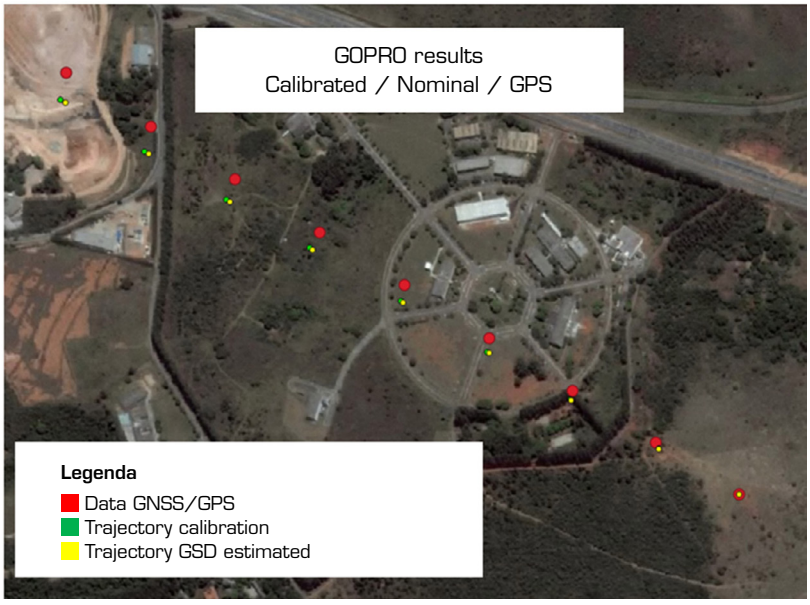
Analysis and Testing Using Aerial Imagery Data and Navigation

To comprehensively evaluate the efficiency and robustness of the proposed methodology, experiments were conducted using aerial trajectories over a previously mapped and orthorectified area. Data were collected with a GoPro Hero4 camera operating over the same geographic segment (east–west orientation) (Fig. 16). The sensor was mounted on an aircraft equipped with a GPS-assisted navigation system, with real-time corrections provided by Real-Time Kinematic (RTK) positioning. Monocular Visual Odometry (VO) was employed to estimate the trajectories, using both calibrated focal length parameters and the nominal values specified in the respective camera technical manuals. Additionally, geometric distortion rates were estimated through the calibration solution, as illustrated in Figs. 5 and 9. The estimated trajectories were compared against reference data obtained via GPS, as shown in Fig. 17 and Table 8.



Source: Elaborated by the authors.

Figure 16. Aerial image datasets with UAV-mounted sensors.



Source: Elaborated by the authors.

Figure 17. Analysis of the estimated trajectories through direct visual comparison.

Table 8. Performance of tested configurations.

Configuration	GPS (m)	VO Estimate (m)	Final Error (m)
GoPro Uncorrected	964.32	823.13	–135.19
GoPro Corrected	964.32	962.22	2.10

VO: visual odometry. Source: Elaborated by the authors.

The experiment was conducted using the previously described dataset, consisting of flight strips containing aerial images acquired in a nadir configuration. This setup ensured sufficient overlap between frames, enhancing the robustness of visual correspondences. The estimated trajectories were qualitatively evaluated through visual comparison, as illustrated in Fig. 10. The results indicated that the trajectory obtained using the calibrated focal length exhibited higher consistency, with more stable and homogeneous rotations between frames compared to the trajectory generated with the nominal focal length. Furthermore, the calibrated trajectory more accurately followed the path recorded by the GPS system dedicated to aerial coverage, particularly in areas with lower vegetation density and greater topographic detail.

Complementing the qualitative analysis, a quantitative evaluation was performed for the estimated trajectories corresponding to each focal length value. Table 8 presents the statistical metrics, including the displacement measured by the GPS, the displacement estimated by visual odometry, and the final error for each flight strip.

Among all configurations, the corrected GoPro setup was the only one to slightly underestimate the actual displacement, demonstrating the highest fidelity to the reference trajectory. The uncorrected GoPro configuration exhibited significant overestimation, whereas the corrected GoPro setup achieved near-ideal performance, with a minimal underestimation of only –0.58 m.

Among all configurations, only the corrected GoPro setup achieved an error below 1 m, while all others remained above 25 m. These results highlight the effectiveness of geometric and focal calibration in enhancing positional accuracy.

The results indicate that the proposed dynamic geometric characterization model, trained and optimized via Optuna hyperparameter tuning, enhances model robustness and generalization. Additional experiments with diverse sensors and image characteristics are underway to validate these findings.

FINAL CONSIDERATIONS

These results indicate that hyperparameter optimization using Optuna has significantly enhanced the automatic characterization of cameras with deep neural networks. In addition to providing more accurate predictions, this approach reduced the variability of estimates, making the model more reliable for practical applications. The results also demonstrate that the proposed methodology, incorporating both geometric and focal calibration, significantly enhances the accuracy and robustness of position estimation through visual odometry. Further experiments with diverse sensors and image characteristics will be conducted to validate the generalizability and effectiveness of the proposed model in real-world applications.

Given the success of the methodology with only 10 trials, it is reasonable to infer that a higher number of iterations and the implementation of the process in a high-performance computing infrastructure could lead to even more significant advancements. To further enhance the model, several additional strategies should be explored and implemented in future developments, such as:

- Deepening the hyperparameter search: Focusing on parameters that have shown the greatest impact on performance, such as the number of neurons and dropout rate.
- Refinement of network architecture: Exploring deeper models with regularization techniques, such as batch normalization and advanced dropout.
- Implementation of ensemble learning techniques: Improving prediction robustness by combining different optimized models.
- Utilization of pruning techniques in Optuna: Accelerating convergence and reducing the need to test ineffective configurations.

In conclusion, the conducted experiments provide evidence that the automatic characterization of cameras using deep neural networks can be effectively enhanced through efficient hyperparameter optimization. The use of Optuna enabled an automated and precise process, allowing refined adjustments that maximize model performance, reduce errors, and ensure greater stability in predictions. This paves the way for applying this approach to more complex scenarios, such as real-time dynamic characterization for advanced computer vision systems. This will be the main subject of study to be tested in future research and experiments.

CONFLICT OF INTEREST

There is no conflict of interest from the authors.

AUTHORS' CONTRIBUTION

Conceptualization: Nogueira FC; **Methodology:** Nogueira FC, Ferreira LOOB; **Software:** Nogueira FC, Ferreira LOOB; **Validation:** Nogueira FC, Ferreira LOOB; **Formal Analysis:** Nogueira FC, Ferreira LOOB; **Investigation:** Nogueira FC; **Writing – Original Draft:** Nogueira FC; **Writing – Review & Editing:** Nogueira FC, Ferreira LOOB, Shiguemori EH; **Supervision:** Shiguemori EH; **Project Administration:** Castro RM, Shiguemori EH; **Project Funding Acquisition:** Nogueira FC; **Final approval:** Nogueira FC.

DATA AVAILABILITY STATEMENT

All dataset were generated or analyzed in the current study.

FUNDING

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

Conselho Nacional de Desenvolvimento Científico e Tecnológico
Grant No: 316948/2023-3.

Financiadora de Estudos e Projetos
Grants No: FINEP 01.20.0195.00 and 01.22.0485.00.

Instituto Nacional de Pesquisas Espaciais

DECLARATION OF USE OF ARTIFICIAL INTELLIGENCE TOOLS

We inform that ChatGPT was used for grammatical correction and for summarizing the work.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Moises José dos Santos Freita for his essential contributions to the development of the first version of the calibration algorithm. We also acknowledge CL Marco Antonio Pizani Domiciano and CP Douglas Damião de Carvalho Honório for their technical and administrative support throughout the work. Finally, we thank all other colleagues from the PITER-N project who provided occasional assistance whenever needed.

REFERENCES

Akiba T, Sano S, Yanase T, Ohta T, Koyama M (2019) Optuna: A next-generation hyperparameter optimization framework. Paper presented Proceedings of the 25th ACM Sigkdd International Conference on Knowledge Discovery & Data Mining, 2623–2631.

Alemán-Flores M, Alvarez L, Gomez L, Santana-Cedrés D (2014) Automatic lens distortion correction using one-parameter division models. Image Process On Lin 4:327-343. https://doi.org/10.5201/ipol.2014.106

Badrloo S, Varshosaz M, Pirasteh S, Li J (2022) Image-based obstacle detection methods for the safe navigation of unmanned vehicles: A review. Remote Sens 14(15):3824. https://doi.org/10.3390/rs14153824

Balcan MF, Nguyen AT, Sharma D (2025) Sample complexity of data-driven tuning of model hyperparameters in neural networks with structured parameter dependent dual function. arXiv preprint arXiv:2501.13734. https://doi.org/10.48550/arXiv.1704.04861

Barreto JP, Araujo H (2005) Geometric properties of central catadioptric line images and their application in calibration. IEEE Trans Pattern Anal Mach Intell 27(8):1327-1333. https://doi.org/10.1109/TPAMI.2005.163

Bogdan O, Eckstein V, Rameau F, Bazin JC (2018) Deepcalib: A deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. Paper presented Proceedings of the 15th ACM Siggraph European Conference on Visual Media Production, 1-10.

Bukhari F, Dailey MN (2013) Automatic radial distortion estimation from a single image. J Math Imaging Vis 45:31-45. https://doi.org/10.1007/s10851-012-0342-2

Caron G (2023) Models and calibration methods. In: Vasseur P, Morbidi F, editors. Omnidirectional Vision: From Theory to Applications. London: Wiley. http://doi.org/10.1002/9781394256440.ch2

Carroll R, Agrawala M, Agarwala A (2009) Optimizing content-preserving projections for wide-angle images. ACM Transactions on Graphics 28(3):43. https://doi.org/10.1145/1576246.1531349

Chang A, Dai A, Funkhouser T, Halber M, Niebner M, Savva M, Song S, Zeng A, Zhang Y (2017) Matterport3D: Learning from RGB-D Data in Indoor Environments. Paper presented 2017 International Conference on 3D Vision (3DV), Qingdao, China, 667-676. https://doi.org/10.1109/3DV.2017.00081

Chao CH, Hsu PL, Lee HY, Wang YCF (2020) Self-supervised deep learning for fisheye image rectification. Paper presented ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2248-2252. https://doi.org/10.1109/ICASSP40776.2020.9054191

Chicco D, Warrens MJ, Jurman G (2021) The coefficient of determination r-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput Sci 7:623. https://doi.org/10.7717/peerj-cs.623

Cramariuc A, Petrov A, Suri R, Mittal M, Siegwart R, Cadena C (2020) Learning camera miscalibration detection. Paper presented 2020 IEEE International Conference on Robotics and Automation (ICRA), 4997-5003.

Do T, Miksik O, Degol J, Park HS, Sinha SN (2022) Learning to detect scene landmarks for camera localization. Paper presented Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 11132-11142. https://doi.org/10.1109/CVPR52688.2022.01085

Duane CB (1971) Close-range camera calibration. Photogramm Eng 37(8):855-866.

Fan B, Dai Y, He M (2021) SUNet: symmetric undistortion network for rolling shutter correction. Paper presented Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 4541-4550.

Faugeras OD, Luong QT, Maybank SJ (1992) Camera self-calibration: Theory and experiments. In: Computer Vision—ECCV'92: Second European Conference on Computer Vision. Santa Margherita Ligure, Italy, 321-334. https://doi.org/10.1007/3-540-55426-2_37

Feng H, Wang W, Deng J, Zhou W, Li L, Li H (2023) SimFIR: A simple framework for fisheye image rectification with self-supervised representation learning. Paper presented Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 12418-12427. https://doi.org/10.1109/ICCV51070.2023.01141

Fischer P, Dosovitskiy A, Brox T (2015) Image orientation estimation with convolutional networks. Paper presented German Conference on Pattern Recognition, 368-378. https://doi.org/10.1007/978-3-319-24947-6_30

Franceschi L, Donini M, Perrone V, Klein A, Archambeau C, Seeger M, Pontil M, Frasconi P (2024) Hyperparameter optimization in machine learning. arXiv preprint arXiv:2410.22854. https://doi.org/10.48550/arXiv.2410.22854

Fraser CS (1997) Digital camera self-calibration. ISPRS J Photogramm Remote Sens 52(4):149-159. https://doi.org/10.1016/S0924-2716(97)00005-1

Gasparini S, Sturm P, Barreto J (2009) Plane-based calibration of central catadioptric cameras. Paper presented 2009 IEEE 12th International Conference on Computer Vision, 1195-1202. https://doi.org/10.1109/ICCV.2009.5459336

Habib A, Lari Z, Kwak E, Al-Durgham K (2013) Automated detection, localization, and identification of signalized targets and their impact on digital camera calibration. Rev Bras Cartogr 65(4):785-803. https://doi.org/10.14393/rbcv65n4-43860

Hartley RI (1994) Self-calibration from multiple views with a rotating camera. Paper presented Computer Vision—ECCV'94: Third European Conference on Computer Vision. Stockholm, Sweden, 471-478.

He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385 https://doi.org/10.48550/arXiv.1512.03385

Hold-Geoffroy Y, Sunkavalli K, Eisenmann J, Fisher M, Gambaretto E, Hadap S, Lalonde JF (2018) A perceptual measure for deep single image câmera calibration. Paper presented Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2354-2363. https://doi.org/10.1109/tpami.2023.3269641

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861. https://doi.org/10.48550/arXiv.1704.04861

Huai J, Shao Y, Jozkow G, Wang B, Chen D, He Y, Yilmaz A (2024) Geometric wide-angle camera calibration: A review and comparative study. Sensors 24(20):6595. https://doi.org/10.3390/s24206595

Jau YY, Zhu R, Su H, Chandraker M (2020) Deep keypoint-based camera pose estimation with geometric constraints. Paper presented Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4950-4957. https://doi.org/10.1109/IROS45743.2020.9341229

Kendall A, Grimes M, Cipolla R (2015) Posenet: A convolutional network for real-time 6-DOF camera relocalization. Paper presented Proceedings of the IEEE International Conference on Computer Vision, 2938-2946. https://doi.org/10.1109/ICCV.2015.336

Lai Y (2024) Application and effectiveness evaluation of Bayesian optimization algorithm in hyperparameter tuning of machine learning models. Paper presented 2024 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEC), 351-355. https://doi.org/10.1109/PEEEC63877.2024.00070

Li Z, Chen J, Baltsavias E (2008) Advances in photogrammetry, remote sensing and spatial information sciences: 2008 ISPRS Congress Book. London: CRC Press.

Liao K, Lin C, Zhao Y (2021) A deep ordinal distortion estimation approach for distortion rectification. IEEE Trans Image Process 30:3362-3375. https://doi.org/10.1109/tip.2021.3061283

Liao K, Lin C, Zhao Y, Gabbouj M (2019b) DR-GAN: Automatic radial distortion rectification using conditional gan in real-time. IEEE Trans Circ Syst Video Technol 30(3):725-733. https://doi.org/10.1109/TCSVT.2019.2897984

Liao K, Lin C, Zhao Y, Xu M (2020) Model-free distortion rectification framework bridged by distortion distribution map. IEEE Trans Image Process 29:3707-3718. https://doi.org/10.1109/TIP.2020.2964523

Mahmood AS, Al-Nuaimi BT, Abdul-Wahab A (2024) Multi-cameras calibration system based deep learning approach and beyond: A survey. Bilad Alrafidain J Eng Sci Technol 3(2):93-126. https://doi.org/10.56990/bajest/2024.030209

Marcato Junior J, Tommaselli AMG (2013) Exterior orientation of CBERS-2B imagery using multi-feature control and orbital data. ISPRS J Photogramm Remote Sens 79:219-225. https://doi.org/10.1016/j.isprsjprs.2013.02.018

Marcu DC, Grava C (2021) The impact of activation functions on training and performance of a deep neural network. Paper presented 2021 16th International Conference on Engineering of Modern Electric Systems (EMES), 1-4. https://doi.org/10.1109/EMES52337.2021.9484108

Mathworks (2023) Camera Calibration Tutorial. https://www.mathworks.com/help/vision/camera-calibration.html

Maybank SJ, Faugeras OD (1992) A theory of self-calibration of a moving camera. Int J Comput Vision 8(2):123-151. https://doi.org/10.1007/BF00127171

Mei C, Rives P (2007) Single view point omnidirectional camera calibration from planar grids. Paper presented Proceedings 2007 IEEE International Conference on Robotics and Automation, 3945-3950. https://doi.org/10.1109/ROBOT.2007.364084

CC BY

Nogueira FC, Freitas MJS, Castro RM, Shiguemori EH (2025) Evaluation of deep neural networks for camera geometric characterization in image-based navigation. Rev Foco 18(4):e8197. https://doi.org/10.54751/revistafoco.v18n4-036

OpenCV (2023) Camera Calibration Tutorial. https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

Pak A, Reichel S, Burke J (2022) Machine-learning-inspired workflow for camera calibration. Sensors 22(18):6804. https://doi.org/10.3390/s22186804

Samii A, Měch R, Lin Z (2015) Data-driven automatic cropping using semantic composition search. Comput Graph Forum 34:141-151. https://doi.org/10.1111/cgf.12465

Shah S, Aggarwal J (1994) A simple calibration procedure for fish-eye (high distortion) lens camera. Paper presented Proceedings of the 1994 IEEE International Conference on Robotics and Automation, 3422-3427.

Shekhar S, Bansode A, Salim A (2021) A comparative study of hyper-parameter optimization tools. Paper presented 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 1-6.

Shi Y, Tong X, Wen J, Zhao H, Ying X, Zha H (2020) Position-aware and symmetry enhanced gan for radial distortion correction. Paper presented 2020 25th International Conference on Pattern Recognition (ICPR), 1701-1708.

Song Z, Peng L, Hu J, Yao D, Zhang Y (2024) A re-calibration method for object detection with multi-modal alignment bias in autonomous driving. arXiv preprint arXiv:2405.16848. https://arxiv.org/html/2405.16848v1

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2015) Rethinking the Inception Architecture for Computer Vision. arXiv preprint arXiv:1512.00567. https://doi.org/10.48550/arXiv.1512.00567

Tan Z, Zhang X, Teng S, Wang L, Gao F (2024) A review of deep learning-based LiDAR and camera extrinsic calibration. Sensors 24(12):3878. https://doi.org/10.3390/s24123878

Vujović ŽD (2021) Classification model evaluation metrics. Int J Adv Comput Sci Appl 12(6):599-606. https://doi.org/10.14569/IJACSA.2021.0120670

Wang R, Nabney I, Golbabaee M (2024) Efficient hyperparameter importance assessment for CNNs. arXiv preprint arXiv:2410.08920. https://doi.org/10.48550/arXiv.2410.08920

Watanabe S (2023) Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. arXiv preprint arXiv:2304.11127, 2023. https://doi.org/10.48550/arXiv.2304.11127

Weng J, Cohen P, Herniou M, *et al.* (1992) Camera calibration with distortion models and accuracy evaluation. IEEE Trans Pattern Anal Mach Intell 14(10):965-980. https://doi.org/10.1109/34.159901

Workman S, Greenwell C, Zhai M, Baltenberger R, Jacobs N (2015) Deepfocal: A method for direct focal length estimation. Paper presented 2015 IEEE International Conference on Image Processing (ICIP), 1369-1373.

Xian W, Li Z, Fisher M, Eisenmann J, Shechtman E, Snavely N (2019) UprightNet: Geometry-aware camera orientation estimation from single images. Paper presented Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 9974-9983.

Yang J, An W, Yan C, Zhao P, Huang J (2021) Context-aware domain adaptation in semantic segmentation. Paper presented Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 514-524.

Yang L, Shami, A (2020) On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing 415:295-316. https://doi.org/10.1016/j.neucom.2020.07.061

Yang S, Lin C, Liao K, Zhang C, Zhao Y (2021) Progressively complementary network for fisheye image rectification using appearance flow. Paper presented Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6348-6357.

Zhang C, Rameau F, Kim J, Argaw DM, Bazin JC, Kweon IS (2020) DeepPTZ: Deep self-calibration for PTZ cameras. Paper presented Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 1041-1049.

Zhang Z (1999) Flexible camera calibration by viewing a plane from unknown orientations. Paper presented Proceedings of the 7th IEEE International Conference on Computer Vision 1:666-673. https://doi.org/10.1109/ICCV.1999.791289

Zhang Z (2000) A flexible new technique for camera calibration. IEEE Trans Pattern Anal Mach Intell 22(11):1330-1334. https://doi.org/10.1109/34.888718

Zhao K, Lin C, Liao K, Yang S, Zhao Y (2021) Revisiting radial distortion rectification in polar-coordinates: A new and efficient learning perspective. IEEE Trans Circ Syst Video Tech 32(6):3552-3560. https://doi.org/10.1109/TCSVT.2021.3119293

Zhuang B, Tran QH, Ji P, Cheong LF, Chandraker M (2019) Learning structure-and-motion-aware rolling shutter correction. Paper presented Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4551-4560.

CC BY