Deutsches Zentrum für Luft- und Raumfahrt
German Aerospace Center

University of Stuttgart
Institute of Industrial Automation and Software Engineering

Master Thesis

– M.Sc. Autonomous Systems –

Development of a method for brightness calibration of light curves

by Kai Riegler
Student ID: 3730872
Faculty 5: Computer Science, Electrical Engineering and Information Technology

First Reviewer:
Prof. Dr.-Ing Michael Weyrich
Institute of Industrial Automation and Software Engineering (IAS)
University of Stuttgart

Second Reviewer:
Prof. Dr. rer. nat. Thomas Dekorsy
Institute of Aerospace Thermodynamics (ITLR)
University of Stuttgart
and
Institute of Technical Physics
German Aerospace Center (DLR)

Supervisor:
Tristan Meyer
Institute of Technical Physics
German Aerospace Center (DLR)

October 15, 2025

Declaration of Originality

I hereby declare that I have completed this Master's thesis independently with the support of my supervisor and have not used any sources or aids other than those specified. The thesis or essential parts of it have not been submitted to this or any other educational institution for the award of a degree.

I further declare that I have complied with the relevant provisions on copyright protection of third-party contributions in accordance with the rules of good scientific practice when preparing the thesis. Insofar as my thesis contains third-party contributions (e.g., images, drawings, text passages, etc.), I have marked these contributions as such (citation, source reference) and obtained any necessary consent from the authors to use these contributions in my thesis. I am aware that in the event of a culpable breach of these obligations, I must bear the resulting consequences.

Stuttgart, October 15, 2025

Place, Date

Signature of the author

Abstract

This thesis aims to perform a photometric calibration on non-sidereal imagery from the German Aerospace Center (DLR)'s miniSLR® system. The goal is to reduce environmental influences and camera effects. To solve this, a multi-stage pipeline was designed, implemented, and validated, using Deep Learning (DL) for instance segmentation and differential photometry for light curve measurements. A Mask Region-based Convolutional Neural Network (R-CNN) model was trained on a large synthetic dataset and is able to identify star streak and a central target object within noisy images. The validation of this system was in a controlled synthetic environment, where the Artificial Intelligence (AI) model demonstrated a high proficiency in locating objects and achieved a Mean Average Precision (mAP) of 0.814 on the synthetic data. Subsequently, the photometric pipeline, which combines the model's prediction with a Point Spread Function (PSF)-fitting algorithm, proved capable of correcting environmental factors. The pipeline reduces a measurement error of almost 39 %, caused by a passing cloud, to 1.03 %. While the pipeline validation was successful on the synthetic data, applying the model to real-world images indicates a significant limitation: the sim-to-real gap. The fine-tuned model on a small set of real data resulted in poor performance (mAP of 0.088), as the model struggled to generalize from the simulation to reality. This prevented the validation of the photometric transformation on real data within the scope of this thesis. In conclusion, the thesis has successfully established the theoretical and practical groundwork for a frame-by-frame, real-time approach for photometric calibration.

Keywords: miniSLR®, light curve calibration, differential photometry, real-time calibration, instance segmentation, deep learning

Kurzzusammenfassung

Das Ziel dieser Arbeit ist die Durchführung einer photometrischen Kalibrierung von nicht siderischen Bildern vom miniSLR® System des Deutschen Zentrums für Luft- und Raumfahrt DLR. Das Ziel darin besteht, Umwelteinflüsse und Kameraeffekte zu reduzieren. Um dies zu erreichen, wurde eine mehrstufige Pipeline entworfen, implementiert und validiert, wobei Deep Learning für die Instanzsegmentierung und differentielle Photometrie für die Lichtkurvenmessungen verwendet wurde. Ein Mask Region-based Convolutional Neural Network (R-CNN)-Modell wurde auf einem großen synthetischen Datensatz trainiert und kann Sternspuren und ein zentrales Zielobjekt in Bildern erkennen. Die Validierung dieses Systems erfolgte in einer kontrollierten synthetischen Umgebung, in dem das Modell eine hohe Kompetenz bei der Lokalisierung von Objekten zeigte und eine mittlere durchschnittliche Präzision (mAP) von 0.814 auf den synthetischen Daten erreichte. Anschließende erwies sich die photometrische Pipeline, die die Vorhersage des Modells mit einem Punktspreizfunktionsalgorithmus kombiniert, als fähig, Umweltfaktoren zu korrigieren. Das System reduzierte einen durch eine vorbeiziehende Wolke verursachten Messfehler von fast 39 % auf 1.03 %. Während die Pipeline-Validierung mit den synthetischen Daten erfolgreich war, zeigte die Anwendung des Modells auf reale Bilder eine erhebliche Einschränkung: die Diskrepanz zwischen Simulation und Realität. Das auf einem kleinen Satz realer Daten fein abgestimmte Modell führte zu einer schlechten Leistung (mAP von 0,088), da das Modell Schwierigkeiten hatte, von der Simulation auf die Realität zu verallgemeinern. Diese machte die Validierung der photometrischen Transformation auf realen Daten im Rahmen dieser Arbeit nicht möglich. Zusammenfassend lässt sich sagen, dass die Arbeit erfolgreich die theoretischen und praktischen Grundlagen für eine Frame-für-Frame-Echtzeitkalibrierung geschaffen hat.

Schlüsselwörter: miniSLR®, Lichtkurvenkalibrierung, Differentialphotometrie, Echtzeitkalibrierung, Instanzsegmentierung, Deep Learning

Acknowledgement

I would like to take this opportunity to express my sincere gratitude to those who supported me during the preparation of this master's thesis.

First, I would like to thank my supervisor, **Tristan Meyer**, for his valuable guidance and support throughout this master's thesis.

I am also grateful to the members of my thesis committee, **Prof. Dr. Thomas Dekorsy** and **Prof. Dr. Michael Weyrich**, for the opportunity to undertake this thesis and for agreeing to review this work.

Finally, I sincerely thank my family and friends for their support and encouragement throughout my studies. This achievement would not have been possible without them.

Contents

List of Figures

List of Tables

List of Abbreviations

ADRAS-J	Active Debris Removal by Astroscale-Japan
AI	Artificial Intelligence
ANN	Artificial Neural Network
AGI	Artificial General Intelligence
CNN	Convolutional Neural Network
COCO	Common Objects in Context
DLR	German Aerospace Center
DL	Deep Learning
ESA	European Space Agency
FPN	Feature Pyramid Network
FCN	Fully Convolutional Network
FWHM	Full Width at Half Maximum
FP	False Positive
FN	False Negative
IoU	Intersection over Union
MEV	Mission Extension Vehicle
ML	Machine Learning
mAP	Mean Average Precision
MAPE	Mean Absolute Percentage Error
MdAPE	Median Absolute Percentage Error
NN	Neural Network
PSF	Point Spread Function
PCA	Principal Component Analysis
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
RPN	Region Proposal Network
RoI	Region of Interest
SSA	Space Situational Awareness
SED	Spectral Energy Distribution

SLR	Satellite Laser Ranging
TP	True Positive
TN	True Negative
YOLO	You Only Look Once

1. Introduction

Current models from the European Space Agency (ESA) estimate that around 54,000 space objects larger than 10 cm, along with millions of smaller debris pieces between 1 and 10 cm, are traveling through Earth's orbit. This total, which includes both active controlled satellites and uncontrolled space debris, is rising rapidly [10]. Each one of these objects poses a potential threat to the satellite infrastructure on which our modern communication, navigation, and Earth observation depend. With the increasing density of space objects in Earth's orbit, ensuring the safety and sustainability of space operations presents a significant challenge. Therefore, the discipline of Space Situational Awareness (SSA) has become critical for monitoring, tracking, and managing these objects to prevent catastrophic collisions [11]. The removal of space debris is one of the biggest challenges for the future of spaceflight. Before such an object can be captured and removed, a seemingly simple but crucial question must be answered: How exactly is the object moving?

The need to answer this question is driving a new era in spaceflight. That of space debris removal and on-orbit servicing. To push forward this technology, there are missions to tackle the growing problem of space debris. One is the Active Debris Removal by Astroscale-Japan (ADRAS-J) mission, which launched successfully in 2024. It's the world's first mission to safely approach and inspect a non-cooperative piece of debris to gather data and close-up imagery. This information will be used to plan a future follow-up mission to attempt a capture and removal of space debris [21]. Another example is the ClearSpace-1 mission. Exercised by ESA and the Swiss start-up ClearSpace SA, it's the world's first mission to capture and remove a piece of space debris, the malfunctioning satellite PROBA-1, to actively de-orbit it [2, 41].
On the other hand, the Mission Extension Vehicle (MEV) of Northrop Grumman was the first satellite life extension vehicle, which successfully performed an on-orbit servicing by docking onto the client satellite Intelsat IS-901 in 2020, extending its operational life. Despite the differences in the missions, all of these have the same prerequisite: they require precise knowledge of the target object before a docking maneuver can take place or a robotic arm can grab it [8].

A particularly powerful method for this is the analysis of light curves, which are the time-based brightness variations of an object [29, 45]. Through a technique known as light curve inversion, these variations can be used to determine object characteristics such as rational dynamics, shape, size, and surface properties. Determining the rotational dynamics is of particular importance for robotic missions, as they influence tasks such as trajectory planning or manipulation tasks. However, the interpretation of these measurements can be very complex, because the measured brightness depends on factors that are constantly changing. For example, the position of the sun, which illuminates the object, and the position of an observer [4]. For this reason, light curve simulation is a necessary and powerful tool. A comparison of the simulation with measurements is made possible by generating physically accurate synthetic light curves with known dynamic parameters. This makes estimating object properties easier and improves the interpretation of measured light curves and the overall understanding of the object [29, 34].

Nevertheless, extracting reliable information from these curves is a complex task. This means that not only the dynamic factors of the light curves themselves have to be considered, but one must also take factors that influence the raw photometric measurements into account. Variables such as atmospheric conditions (e.g., air turbulence, humidity, cloud coverage) or varying seeing conditions, and camera-related factors like sensor noise make precise light curve calibration a non-trivial but necessary task [38, 45].

1.1. Research Goal

This study aims to establish a more robust calibration process that minimizes the impact of environmental variables. To achieve this, the thesis details the development and validation of a processing pipeline for brightness calibration of space objects observed with the miniSLR® of the DLR in Stuttgart, Germany [17]. This pipeline combines modern machine learning approaches with innovative calibration techniques for a robust brightness extraction of objects in the miniSLR® imagery. Notably, for tracked objects, this calibration is performed on a per-image basis in real-time. Correcting for atmospheric and instrumental disturbances, the comparison of measured and simulated light curves improves the reliability of light curve inversion. This allows a more accurate analysis of the rotational dynamics of space objects. This improved precision is of critical importance for future in-orbit or debris removal activities, as it provides the reliable data basis for the planning and execution of complex servicing and de-orbiting missions.

1.2. Methodology

To reach the stated research goal, this thesis develops and validates a multi-stage processing pipeline. The first stage is an automated detection of the tracked object and the surrounding star streaks in the raw image data through a deep learning approach. In the second stage, a precise differential calibration method is implemented, utilizing stellar reference sources within the same field of view to correct atmospheric and instrumental effects. Finally, the performance of the complete pipeline is evaluated to quantify the improvement in the final light curve quality. The detailed concept and technical implementation of this approach are presented in chapter 4.

1.3. Thesis Outline

This thesis is divided into the following six chapters: First, chapter 2 begins by providing the necessary theoretical background for this thesis, covering the core principles of photometry, the fundamentals of deep learning, AI-based image segmentation, and a brief description of the miniSLR®. Following this, chapter 3 presents a review of the state-of-the-art, examining traditional and machine learning-based approaches to object detection and photometric calibration. Chapter 4 then details the design and methodology of this project's end-to-end pipeline, explaining the synthetic data generator, the AI model training strategy, the photometric analysis, and the basics for the calibration transformations. Afterwards, chapter 5 presents the evaluation of the developed system, analyzing the AI model before validating the accuracy of the photometric pipeline. Finally, chapter 7 summarizes the key findings of the thesis, discusses the project's limitations, and proposes directions for future research and development.

2. Theoretical Fundamentals

This chapter provides the necessary context for this research and reviews the core methodologies and concepts applied. It details the fundamental principles of astronomical photometry and introduces the key aspects of AI and the implemented model architecture.

2.1. Fundamentals of AI

Machine Learning (ML) and Deep Learning (DL) are terms often used for approaches in the area of AI, such as in autonomous driving, data analysis, speech recognition, and many more. But where exactly do these terms fit in, and what are the differences between them?

The main goal of AI is to display human intelligence and behavior in machines. Within this field, fundamental distinctions can be made [13]. The first one relates to the degree of intelligence: *Narrow-AI* (or weak AI) describes systems that are designed to do specialized tasks inside a defined problem space. This means that it cannot transfer its knowledge into another area. Examples for Narrow-AI are voice assistance like Siri and Alexa, systems of image and face recognition, or even autonomous vehicles. In contrast to that, there is *Artificial General Intelligence (AGI)*, which is a theoretical concept that describes an AI with human cognitive abilities. An AGI would be able to understand, learn, and solve intellectual tasks, like a human. It could use its knowledge flexibly and apply it to new and unknown situations. Currently, such an AGI does not exist [40].

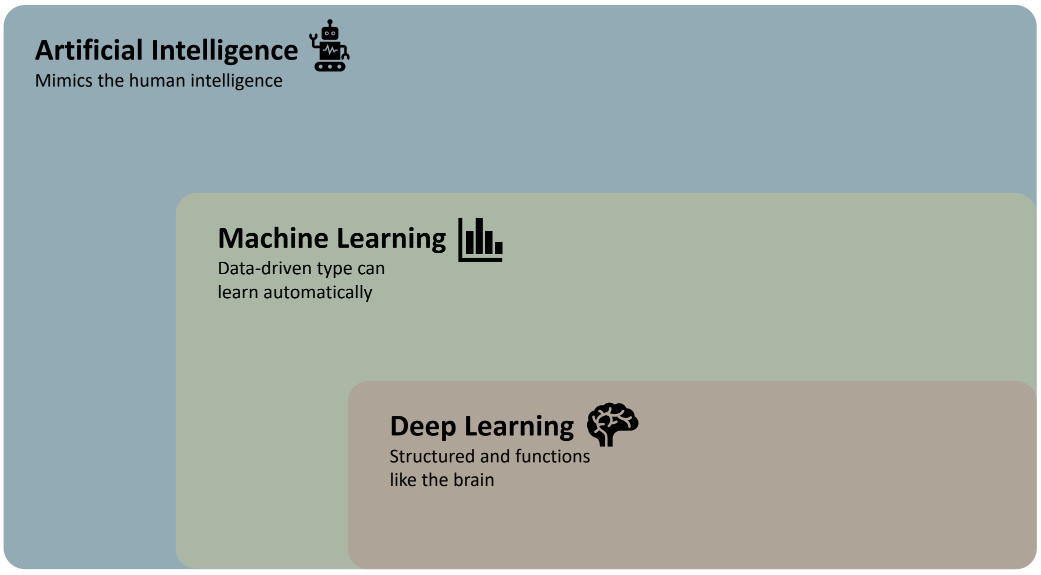

Artificial Intelligence
Mimics the human intelligence
Machine Learning
Data-driven type can learn automatically
Deep Learning
Structured and functions like the brain

Figure 2.1.: The hierarchical relationship between Artificial Intelligence, Machine Learning, and Deep Learning. The diagram shows that DL is a subset of ML, which is an approach to realizing AI (adapted from [13]).

The second important differentiation is the implementation approach: *symbolic AI* (classic AI) represents knowledge that is predefined through rules, on the basis of which they are deterministic. Traditional computer algorithms fall into this category. On the opposite side, there is *non-symbolic AI*, which, unlike its counterpart, does not follow predefined rules. In this case, it creates its own set of rules through a large amount of data to detect patterns. This is called Machine Learning [13]. Especially, Artificial Neural Networks (ANNs) are often used for such complex tasks. Their special structure, which is inspired by the brain (category of DL), enables them to solve complex tasks much more easily and with better accuracy. Figure 2.1 gives a brief overview of the integration of ML and DL in the environment of AI.

2.1.1. Types of ML Approaches

In Machine Learning, there are different approaches for all kinds of tasks. In general, they can be divided into the following four categories:

Supervised Learning Supervised learning is the most common and easiest approach. It is mainly used for classification and regression tasks, the overall principle is shown in Figure 2.2. The available data for this approach has to be labeled [13], which means that the extracted features of the raw data can be associated with a specific class. For example, in an image with a dog and a cat, they are either labeled together as animals or separately as dog and cat. This makes this approach so simple and optimal for ML tasks. However, the downside is that data typically comes unlabeled, and an expert has to label it manually. This is an enormous effort and very time-consuming [16, 28].

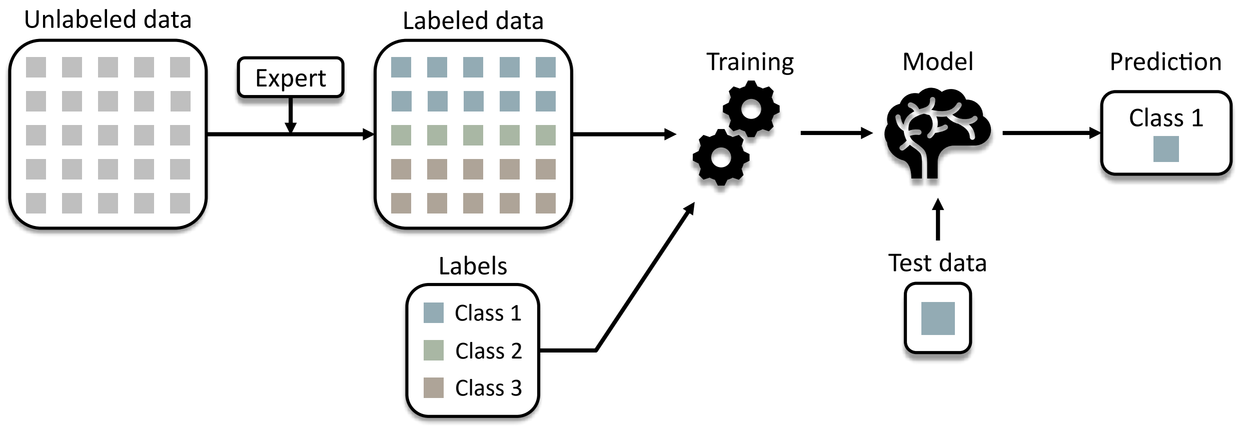


Figure 2.2.: Principle of Supervised Learning. An expert first assigns labels to unlabeled data. The model is then trained with this labeled data to learn the relationship between input data and the correct output labels. After training, it can predict labels for new, unseen data.

Unsupervised Learning The next approach is unsupervised learning, which works entirely without labels [13]. This approach is less time-consuming, however, it introduces other limitations. Unsupervised methods are not able to classify data, they only group them into clusters. An expert has to interpret these clusters manually. Therefore, these are often used to learn the underlying structure or distribution of data to find clusters or used for dimension reduction to compress data without losing important information. Another use case is generative AI models, because they learn the underlying data distribution to generate new data objects [16, 28]. Figure 2.3 visualizes this approach.

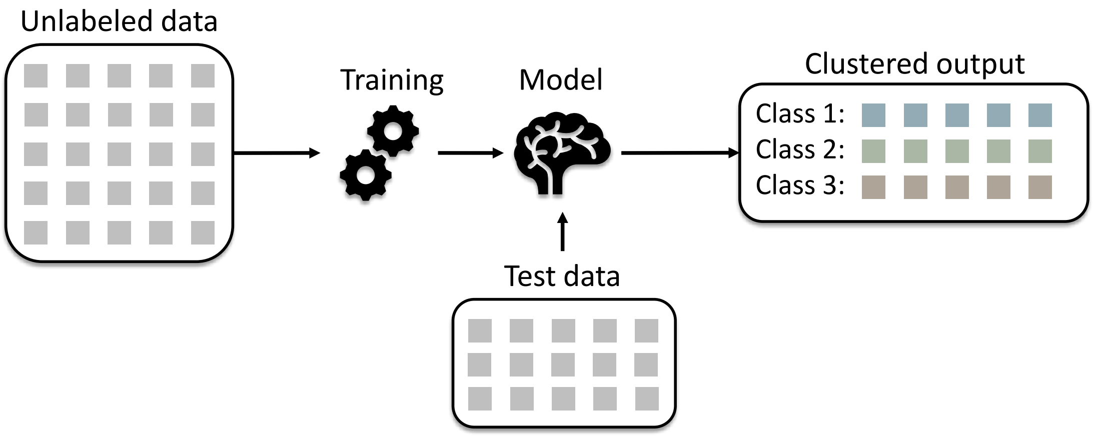


Figure 2.3.: Principle of Unsupervised Learning. The model is trained with only unlabeled data. As a result, the model delivers data in the form of clusters without knowing the meaning of these clusters.

Semi-Supervised Learning Semi-supervised Learning is a middle ground between supervised and unsupervised learning. It often uses a smaller labeled dataset and a large unlabeled dataset [13]. This type is often used to label data or fill in missing data and send it back to the classifier to train it further, which can be seen in Figure 2.4 [16, 28].

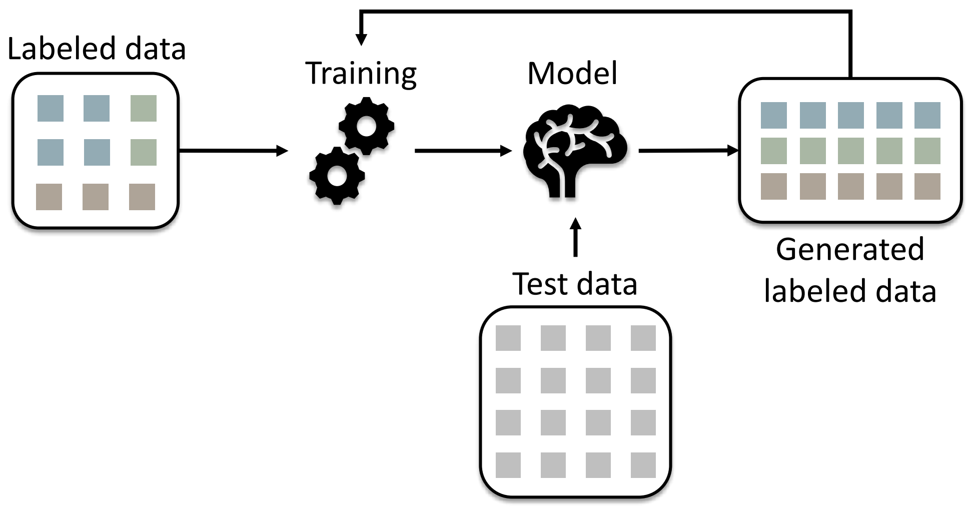


Figure 2.4.: Principle of Semi-Supervised Learning. This approach uses a small amount of labeled data to train an initial model. This model is then used to generate predictions for a larger set of unlabeled data. This newly labeled data is then added to the training data to improve the model.

Reinforcement Learning This approach differs from the others. Reinforcement learning learns through an agent in a dynamic environment, as shown in Figure 2.5. The agent carries out decisions and actions, based on these, it gets a reward or a penalty [13]. Therefore, this constitutes a trial-and-error approach, whereby the agent learns over time to improve its actions. This concept does not require a large amount of data because it generates its own through multiple trials, however, this can be computationally expensive [16, 28].

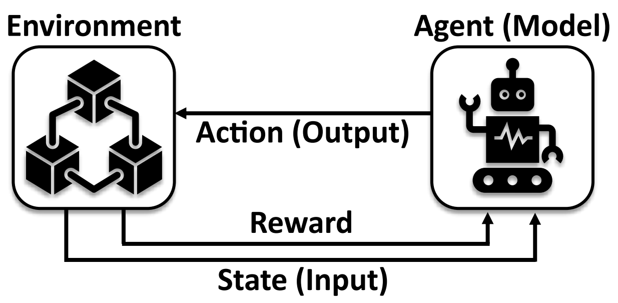


Figure 2.5.: Principle of Reinforcement Learning. An agent interacts with an environment by performing actions. Each action puts the environment into a new state and generates a reward. Through this feedback, the agent learns to optimize its actions.

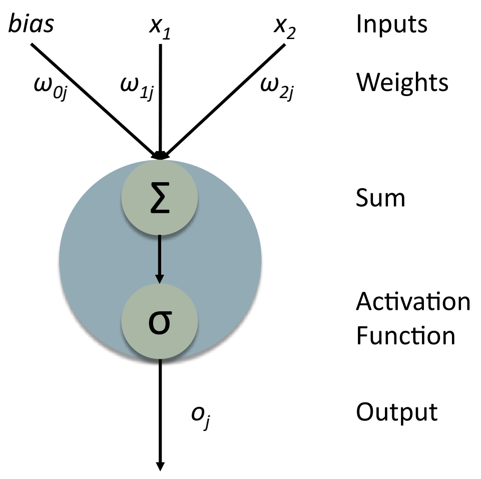
2.1.2. Introduction to Neural Networks

Neural Networks (NNs), or so-called ANNs, are a core component of modern ML, particularly in Deep Learning. Inspired by the structure and functionality of its biological counterparts, these networks are designed to detect complex patterns and connections in data. Instead of being programmed with fixed rules, like traditional algorithms, they learn directly from examples [13, 16].

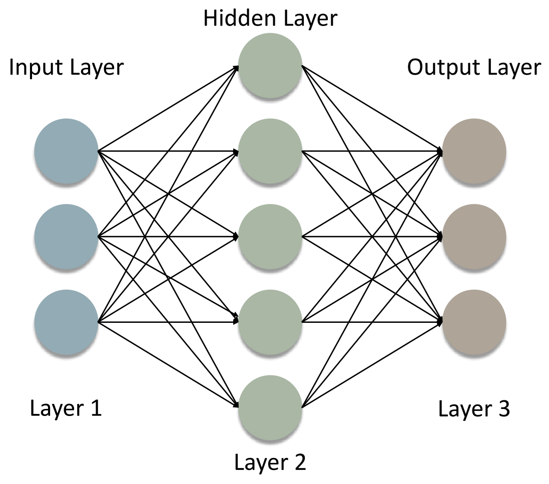
Perceptron (Basic Building Block) Every perceptron in a NN receives one or more input signals, equipped with a weight ω and a bias b, with a processor and one output. This structure is visualized in Figure 2.6a.

$$o_j = a(\omega_{0j} \cdot b + \sum_{i=1}^{n} \omega_{ij} \cdot x_i) \tag{2.1}$$

The inputs x_i and the bias b are multiplied with the weights ω_{ij} and passed to an activation function a, see Equation 2.1. This function represents the processor and determines if the perceptron is active or not. There are different types of activation functions, the most common ones are Rectified Linear Unit (ReLU), SoftMax and Sigmoid. Even a single perceptron is already able to solve simple linear problems [13, 14, 16].



(a) Perceptron



(b) Structure of Neural Network

Figure 2.6.: Basic structure of a Neural Network. (2.6a) A perceptron, the smallest unit, sums weighted inputs and generates an output value using an activation function. (2.6b) A Neural Network consisting of several connected perceptrons arranged in layers.

Multi-Layer Perceptron The real strength of a neural network is created by the connections of many single perceptrons in different layers. Such a network is called a Multi-Layer Perceptron, and consists of three layer types (visualized in Figure 2.6b):

- Input layer: represents the input data (e.g., image pixels)
- Hidden layer (at least one): learns the complex pattern of the data
- Output layer: produces the result like a classification

Through stacked layers, these networks can learn highly complex and non-linear connections [13, 14, 16].

Learning Process Learning is the most important part of the NN, because this process tries to find the best weights of the network. The learning process includes four basic steps:

- Forward Pass: training data is fed trough the network and a prediction is made (at the beginning, the weights are random, therefore the prediction is also random).
- Error Calculation: the prediction error gets calculated using a loss function.
- Backpropagation: the calculated error gets backpropagated trough the network - from output back to the input. Thereby, the proportion of the total error to each individual weights is calculated.
- Gradient Descent: based on the calculations all weights get adjusted with help of an optimizer like Gradient Descent, reducing the error in the next iteration.

These steps are repeated many times. With every iteration the weights get better adjusted and the network learns to make better predictions [13, 14, 16].

2.1.3. Over- and Underfitting

The main goal of training an AI model is to get a good *generalization*. This is the ability to make correct predictions on new and unseen data [13, 16].

The most typical issues of ML algorithms are over- and underfitting. If the model fails to learn the general patterns in the data and instead memorizes the training examples, this phenomenon is referred to as *overfitting* [13]. This means that the model shows an excellent performance on the training data, but fails when it sees new, unseen data. A potential reason could be that the model is overly complex, therefore, the model has too many parameters in proportion to the data, or the amount of data is too small, and the model is not able to detect patterns. In contrast *underfitting* is the exact opposite [16]. The model structure is too simple and is not able to detect any pattern or connection. Its performance is poor on both the training data and unseen data, because of this, the model does not have enough parameters or does not get enough features to make good decisions [13, 16].

To counter overfitting, there are some techniques that are used to improve generalization, which are called *regularization*. These methods punish the model during the training for a too high complexity. They force the model to become simpler and, therefore, more robust [13, 16]. The most common techniques are:

- L1- and L2-Regularization: adds a penalty term to the loss function, limiting the size of the model weights [13].

- Dropout: deactivates random perceptrons during the training. This forces the model to learn redundant paths [16].
- Early Stopping: Stops the training process as soon as the model's performance is decreasing [16].

2.1.4. Transfer Learning

Transfer Learning is a ML method that shifts away from the normal approach of training an AI model from scratch. The main idea is to use the core knowledge, which is based on source tasks, and transfer it to new but similar goal tasks. Rather than initializing model training from scratch, a preexisting model pretrained on a large dataset (e.g., ImageNet) is used as the starting point.

This approach has a significant advantage, especially when the task has limited data. It reduces the need for large amounts of data and decreases both training time and computational costs.

In practice, it works by keeping lower layers of the network, because these detect features like edges, shapes, and patterns. Only the top layers that are task-specific are adjusted or replaced and trained with the specific and smaller dataset. This process is also called *Fine-Tuning* [35].

2.2. Synthetic Data

Synthetic data is artificially created information. This kind of data is not obtained through direct measurements or real situations. Instead, they are generated through computer algorithms, simulations or statistical models, to replicate statistical properties and patterns of real data. In the field of AI, synthetic data has become an fundamental concept due to the lack and need of large amounts of data [33].

2.2.1. Lack of Data

One of the main reasons to use synthetic data is the lack of high-quality and labeled information. Modern DL approaches are data hungry and need thousands or millions of examples to learn properly. In many cases, obtaining such an amount of data is often difficult or even impossible [13, 33].

In areas such as medicine or finance information are strongly protected by privacy policies. Synthetic information can represent statistical patterns of the original data, without leaking personal or sensitive information. This enables training without the need to violate other's privacy [33].

Other factors are cost and time. Gathering data and manually labeling it is extremely costly and time-intensive. This is especially the case for complex tasks like instance segmentation. The automated process of generating labeled data can accelerate this process significantly.

Special cases are important for training an AI model because they need to be prepared for uncommon and critical events. For example, an autonomous car has to learn how to behave when an animal jumps onto the street. Even if these events are represented in real data, they can be generated in simulations in larger numbers. This is often the only feasible method, as collecting this type of data in the real world would require actions that are ethically unacceptable and practically impossible, such as harming thousands of animals. [13, 33].

2.2.2. Test and Validation

The same reasons why synthetic data is useful for model training also apply to validation and testing. The possibility of creating edge cases is very important. It makes the model more robust and reliable. Therefore, it is possible to, e.g., change weather conditions, lighting effects, object textures, or noise. This improves the overall generalization of the model, which is impossible with a limited real dataset [13, 33].

2.2.3. Reality Gap and Domain Gap

One of the greatest challenges of using synthetic data is the so-called Reality Gap. This term describes the mismatch between the simulated and the real world. A model that is only trained on synthetic data will have problems generalizing to real-world scenarios, due to the fact that synthetic data cannot cover the unpredictability and complexity of the real world [13, 33].

Such a gap can appear in all kinds of ways: physically incorrect shadows, unnatural textures, visual artifacts, and missing noise. To close this gap, there are different techniques of Domain Adaptation. One of the most common ones is Fine-Tuning or Transfer learning, see subsection 2.1.4. A model is pretrained on a large dataset of synthetic data and then trained on a smaller real dataset to adjust it to the real-world properties [13, 33].

2.2.4. Types of Synthetic Data

Synthetic data can be created in two different ways:

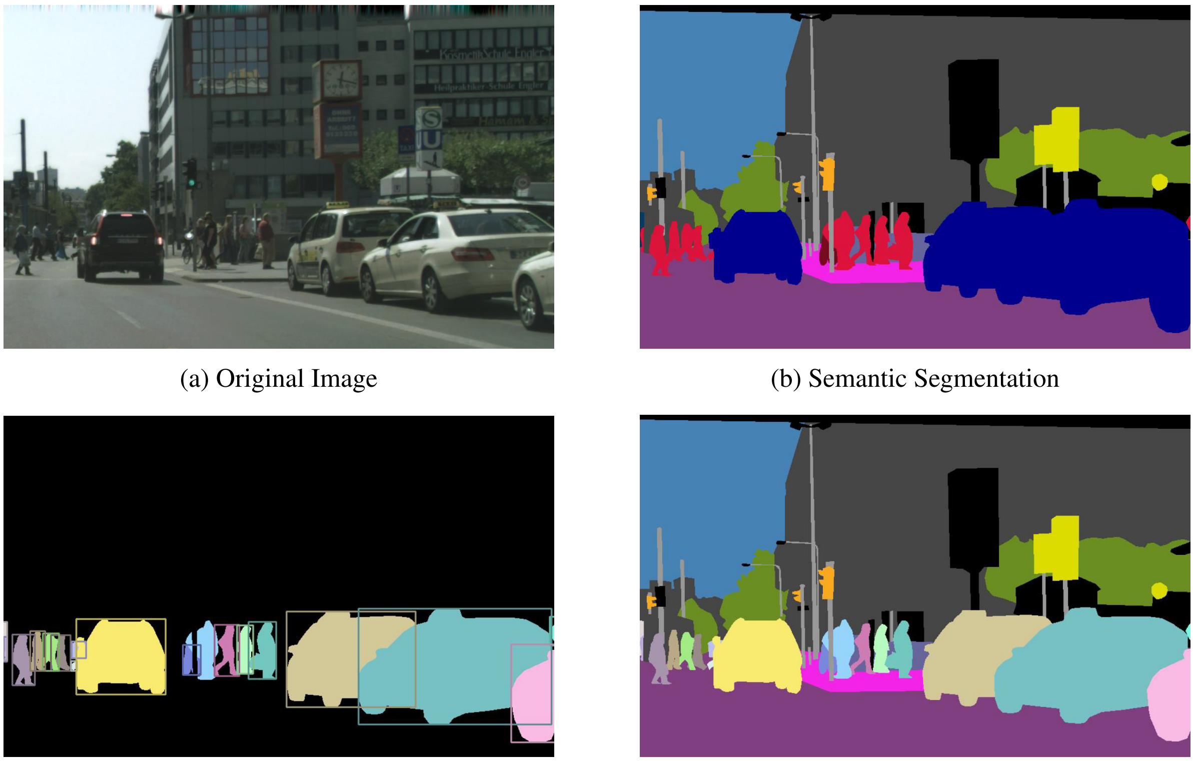
- Procedurally generated data from simulations:
  For this technique the data ins generated in a virtual environment, based on a set of rules. This method allows for maximum control and generates perfect annotation (e.g., bounding boxes, segmentation masks). It is the standard for most applications [33].
- Generative Models:
  Instead of using a simulation, DL models use the underlying data distribution to generate new information. The most common approaches are generative adversarial networks, variational autoencoders, or diffusion models [33].

2.3. AI-based Image Processing and Segmentation

Image segmentation is a fundamental and important process in image processing and ML. While for humans it is easy to distinguish between different objects, like cars, trees or buildings in images, for computers it is a challenging task. This is where image segmentation takes place. It divides a digital image into regions or segments, simplifying it and make it more understandable. Therefore, the main goal is to determine a label to every pixel in an image [23, 46].

2.3.1. Types of Image Segmentation

Modern image segmentation, especially in the area of DL, is divided into three main categories, which build on each other.



(a) Original Image

(b) Semantic Segmentation

(c) Instance Segmentation

(d) Panoptic Segmentation

Figure 2.7.: Comparison of the three main types of image segmentation using the example of a street scene (2.7a). (2.7b) Semantic segmentation classifies each pixel into a category without distinguishing between individual objects (e.g., all cars are blue). (2.7c) Instance segmentation detects and isolates each individual object instance (e.g., each car in a different color). (2.7d) Panoptic segmentation combines both approaches by assigning each pixel both a semantic class and a unique instance ID [23].

Semantic Segmentation This type aims to assign a label to every pixel of the image. This means that each pixel gets an object class, like a car or a tree. However, the disadvantage of semantic segmentation is that it does not distinguish between different instances. For example, all pixels of a tree would be classified as tree, regardless of whether it is tree A or tree B, see Figure 2.7b [23, 46].

Instance Segmentation Instance Segmentation is a more advanced technique. It solves the problem of object-detection and semantic segmentation at the same time. This means it not only identifies the class of an object on a pixel level, but it also differentiates between single object instances, see Figure 2.7c [23, 46].

Panoptic Segmentation Panoptic segmentation combines the tasks of semantic and instance segmentation, see Figure 2.7d. The powerful aspect is that it processes both countable objects (e.g. cars, people) and amorphous background regions (e.g. sky, grass, road) in a single image. Each pixel in the final image has exactly one class label and one instance ID. Nevertheless, a drawback is that panoptic segmentation does not allow overlapping segments. This makes it different from instance segmentation, where overlaps can happen [23].

2.3.2. Mask R-CNN

The Mask Region-based Convolutional Neural Network (R-CNN) is a modern and state-of-the-art DL architecture designed to solve complex tasks of instance segmentation, see subsection 2.3.1, in the field of computer vision. While a classical object detection system only identifies objects in an image and assigns a label to them (e.g., ”this region is a car”), the Mask R-CNN includes two further steps. It identifies every object (e.g., car) as a separate instance and draws a pixel-wise segmentation mask on it [18].

The ability to identify individual instances and accurately outline them is essential for advanced image processing tasks. For example, an autonomous car must not only detect the presence of pedestrians on the road but also distinguish between each pedestrian to predict their individual motion path. Another example is in the field of medicine. An analysis system should not only detect, e.g., overlapping cells in an image, it should also be able to separate them into single units and count or measure them.

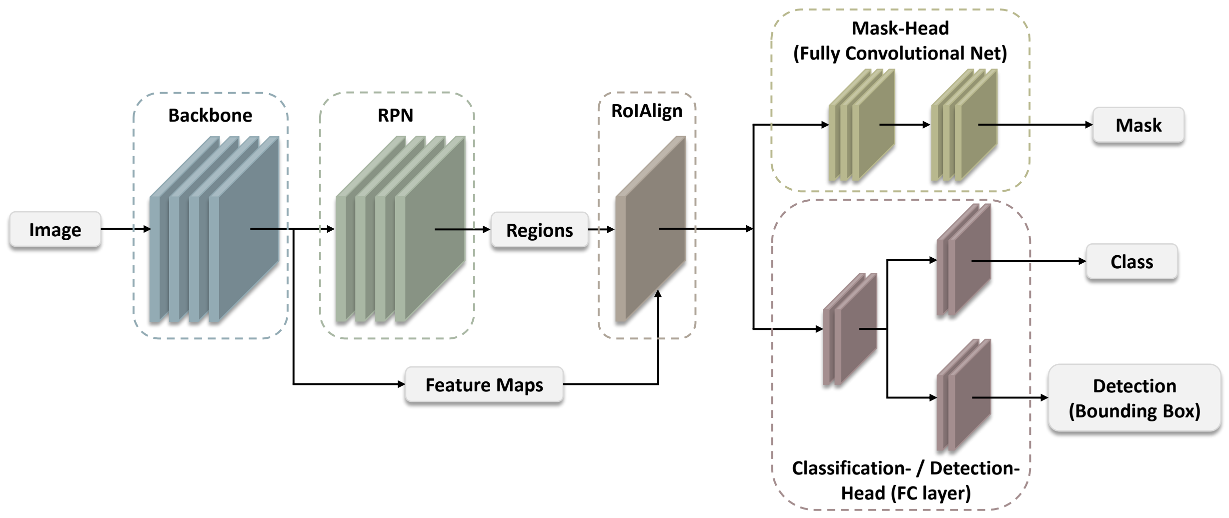


Figure 2.8.: The Mask R-CNN architecture for instance segmentation. A backbone network extracts features from the input. The Region Proposal Network identifies potential object regions from these features. These regions are brought to a uniform size using RoIAlign and then forwarded to two parallel output branches (heads): One classifies the object and predicts its bounding box, while the other generates a pixel-accurate segmentation mask (adapted from [18]).

Architecture The Mask R-CNN is an upgrade of the object detection model Faster R-CNN [18, 37]. Its strength lies in a modular architecture that solves complex tasks and parallelizes them [18]. Figure 2.8 shows a visualization of the architecture.

1. Backbone Network and Feature Pyramid Network:
   The network starts with a Backbone, which is a pretrained Convolutional Neural Network (CNN). Typically, some sort of ResNet is used as a CNN. This NN analyses the input image and extracts different kinds of features on different levels. On the bottom level, it detects features like edges, shapes, and colors. On the upper level, it learns more complex shapes which consist of different

objects and shapes (e.g., an eye or a wheel) [18]. Additionally, the architecture incorporates the Feature Pyramid Network (FPN). This FPN takes different features from the different levels of the backbone, combines them in an intelligent way, and creates a feature pyramid. The pyramid includes high-resolution and highly detailed information from the lower levels and semantic information of the upper levels. This helps the overall network to detect objects in all kinds of sizes, with a consistent accuracy [18, 25].

2. Region Proposal Network:
   The Region Proposal Network (RPN) is a smaller network which iterates over the created feature of the FPN. Its only purpose it to identify image regions which contain an object. These regions are called Region of Interest (RoI). The RPN generates a large amount of RoI's as bounding boxes and labels them with an objectness score. It's a rating, how sure the network is that there is an object at all [18].

3. RoIAlign:
   This is the most important part of the Mask R-CNN. After the RPN proposed some RoIs, the corresponding feature has to be extracted from the overall features. The previous model, Faster R-CNN, uses RoIPool for this, but this leads to a rough quantization of the coordinates [18, 37]. The RoIPool process produces inaccuracies that are tolerable for object detection but critical for pixel-precise segmentation. RoIAlign solves this problem. Instead of rounding the calculations, it uses a bilinear interpolation to calculate the exact coordinates within the RoI. Thereby, the extracted features remain perfectly inside the image, which is crucial for a precise segmentation mask [18].

4. Parallel Heads for Goal Tasks:
   For every RoI that is produced by the RoIAlign, the features are forwarded to three parallel heads. These heads solve the following tasks at the same time:

   - Classification-Head: This head answers the question "What kind of object is it?" and assigns a class label to it.
   - Detection-Head: This head answers the question "Where exactly is the object?" and refines the coordinates of the bounding box.
   - Mask-Head: It's the main addition of the Mask R-CNN. This head is a small Fully Convolutional Network (FCN), that is used on the features to generate a binary pixel mask of the object [18].

2.4. Evaluation Metrics

To measure and compare the performance of AI models, metrics that are clearly defined are required. These metrics are based on the comparison of the model predictions and the ground truth of the dataset.

Fundamental Classification Metrics For every prediction, there are four fundamental states [19, 43]:

- True Positive (TP): The model has correctly identified an object (e.g., a star is classified as a star at the correct position).
- False Positive (FP): The model has identified an object where no objects are or classified it wrong (e.g., an image artifact is identified as a star).

- True Negative (TN): Refers to the case where no object is present and none has been detected.
- False Negative (FN): The model did not recognize an object that actually exists. (e.g., a faint star was overlooked).

Based on these fundamental states, two important rates are derived. The first one is called *Precision*. This asks how reliable the predictions are - Of all the objects detected, how many were correct? This metric is described through Equation 2.2.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.2}$$

The second one is the *Recall*. It asks how complete the detection is - Of all real objects, how many were found? - and is defined through Equation 2.3.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.3}$$

A good model has to find a balance between high precision and recall [13, 19, 43].

Intersection over Union Evaluation is a crucial step following model training, where various metrics are used. In object detection, the most common metric is the Intersection over Union (IoU), which evaluates how well the model can localize objects and mark them with a bounding box. For this, the predicted bounding box B_{pred} is compared with the correct bounding box B_{gt}. It calculates the ratio of the intersection area (overlapping area of boxes) to the union area (total area covered by boxes) [13]:

$$\text{IoU} = \frac{\text{Area}(B_{\text{pred}} \cap B_{\text{gt}})}{\text{Area}(B_{\text{pred}} \cup B_{\text{gt}})} \tag{2.4}$$

IoU provides a value between zero (no overlap) and one (perfect overlap), shown in Figure 2.9. In practice, a threshold (e.g., IoU > 0.5) is set to decide whether a prediction is correct or not [13].

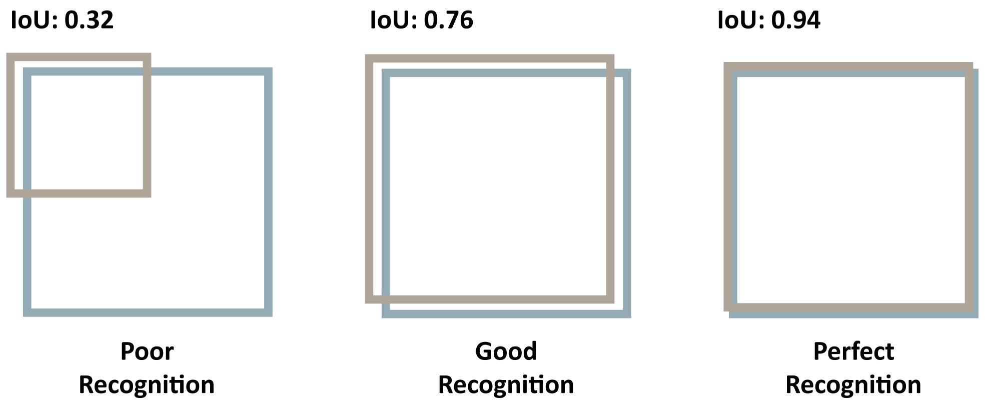


Figure 2.9.: Schematic representation of Intersection over Union. IoU measurement is used to evaluate the matching of two bounding boxes. Three examples are shown for poor (left), good (middle), and perfect (right) detection. The blue frame represents the ground truth, while the brown frame represents the model's prediction (adapted from [13]).

Mean Average Precision The Mean Average Precision (mAP) is the main metric for the performance evaluation of an object detection model. It combines precision, recall, and the IoU metric into one value. At first, the average precision for every object class is determined. Afterwards, the mean values of all average precision values across all classes are calculated [13]:

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \tag{2.5}$$

where N is the number of classes and AP_i is the average precision for the i-th class.

The mAP is often specified with the IoU-threshold [13, 19, 43]:

- mAP@0.5: A prediction counts as TP if the IoU is greater than 0.5, this is a common but soft standard.
- mAP@[.5:.95]: Here, the mAP is calculated for ten different IoU thresholds (from 0.5 to 0.95 in 0.05 steps) and the results are averaged. This version is more rigorous. Higher values mean that the model finds an object and can localize it with high accuracy.

2.5. Photometry and Light Curve Analysis

Photometry is the science of measuring the brightness and color of an object and is a fundamental part of astronomy [45]. It mainly deals with precise measurements of flux (amount of energy, in the form of electromagnetic radiation) or the received light intensity of a celestial object [38]. The measured brightness is a fundamental physical information that helps to derive a variety of physical properties of an object. By analyzing variations in this brightness, we can determine several of its properties, including size, shape, material, and rotational dynamics [29, 45].

2.5.1. Magnitude Scale and Flux

The brightness of a celestial body is a fundamental property in astronomy. To measure the brightness, astronomers used the magnitude system. This system is based on an inverse logarithmic scale. Therefore, bright objects have a small magnitude value, while faint objects have a higher value. The object brightness, which is measured from Earth, is referred to as apparent magnitude. It not only depends on the luminosity of the object but also on factors like distance, light pollution, and atmospheric conditions [45].

The physical value on which brightness is based is the flux. This flux is the amount of energy that hits an unit area per unit of time. The relationship between the apparent magnitude (m_1, m_2) and the flux (f_1, f_2) of two celestial bodies is defined through Equation 2.6. This definition states that a difference in brightness of five magnitudes corresponds to a flux ratio of 100 [38]:

$$f_1/f_2 = 10^{-0.4(m_1-m_2)} \tag{2.6}$$

This equation shows the logarithmic nature of the scale. It can also be rearranged to calculate the magnitude difference directly from the flux ratio. This leads to Equation 2.7:

$$m_1 - m_2 = -2.5 \cdot \log_{10}\left(\frac{f_1}{f_2}\right) \tag{2.7}$$

This equation shows that a star of magnitude one is 2.512 times brighter than a star of magnitude two ($\sqrt[5]{100} = 2.512$) [38, 45].

2.5.2. Atmospheric Extinction

Atmospheric extinction is the reduction in brightness of light passing through Earth's atmosphere. It is caused by scattering from air molecules, as well as by absorption. The extinction is different for all colors. Blue light is scattered more than red light, which is why objects closer to the horizon appear darker and redder. This effect increases the longer the light travels through the atmosphere (air mass). Extinction varies with location and weather and can change overnight. In practice, it needs to be corrected for precise measurements. An observation as close to the zenith as possible is in favor, because in this case the light travels the shortest path through the atmosphere [7, 45].

2.5.3. Light Curves

A light curve is a graphical visualization of the measured object brightness as a function of time. The vertical axis shows the brightness (e.g., in magnitude), while the horizontal axis represents the time. An example curve is shown in Figure 2.10. Such a light curve enables astronomers to research dynamical processes in the universe (e.g., exoplanet transits), and derive physical object properties [45].

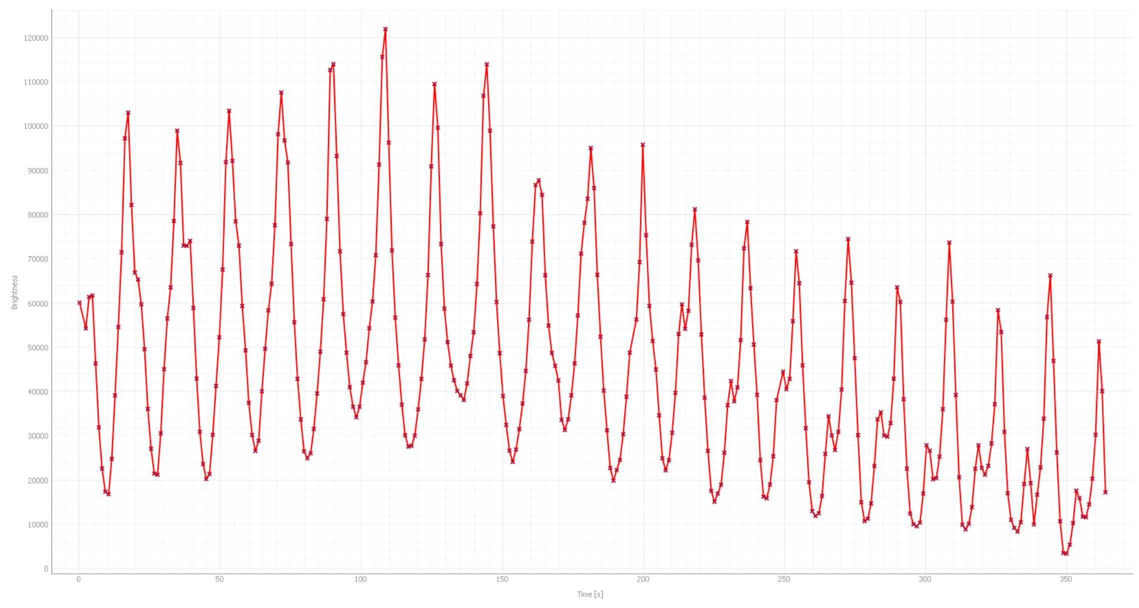


Figure 2.10.: Light curve example of the SL-14 rocket body. This relative brightness is shown as a function of time. The periodic brightness flares are caused by sunlight's reflection on the surfaces of the rotating rocket debris. The data was recorded on 2025-08-25, with the miniSLR®.

2.5.4. The miniSLR®

The miniSLR® is a compact and mobile Satellite Laser Ranging system concept that contains only the necessary components and allows for an affordable laser ranging solution. The system is placed at the DLR facility in Stuttgart, Germany, see Figure 2.11. It consists of two main parts: the first one is the lower cabinet, which contains the controlling electronics, whereas the top part encloses the telescope and the laser head. All parts together have a footprint of 2.3 m x 2.1 m x 2.0 m. The whole system weighs about 600 kg and only needs a stable underground, as well as a power and internet connection [31].

The miniSLR® uses a repeating and pulsing laser source. These pulses are sent to a space object equipped with a retroreflector. Retroreflectors reflect the light back to its source. The receiving telescope captures the returning light, and the exact distance can be calculated. In addition, the system is also equipped with a monochrome Andor Zyla camera, which allows the use of the miniSLR® as an imaging device. Captured images are used, for example, to determine object characteristics via light curve analysis. Due to the minimal construction design, the station has a small field of view of 0.92° x 0.69°, which presents significant requirements for satellite tracking, such as the need for precise orbital data, a mount with high tracking accuracy, and closed-loop tracking, which corrects the system alignment in real-time to keep the tracking object in the center [31, 32].

Despite its small size, the miniSLR® achieves high-precision and can compete with regular systems [31].

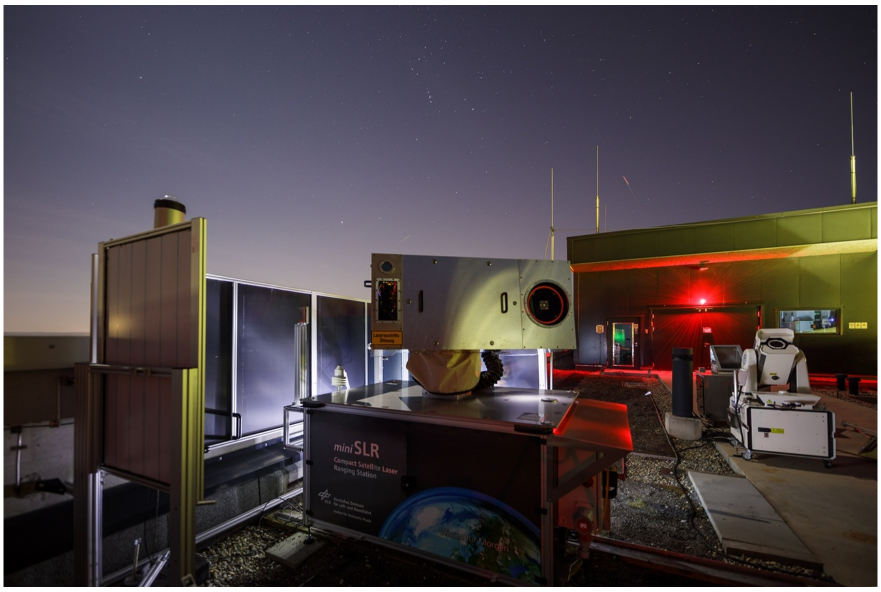


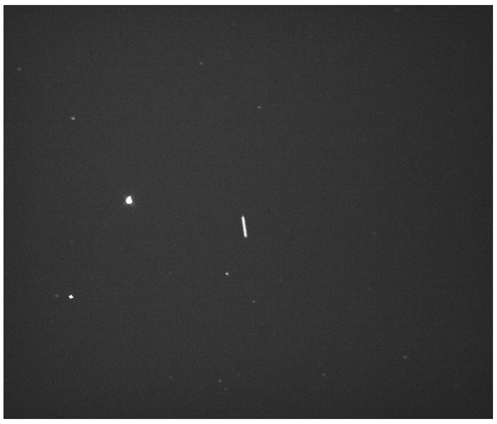
Figure 2.11.: The miniSLR® system at the DLR site in Stuttgart. From this station on the roof, high-precision distance measurements to satellites are performed using laser pulses (Photo: Paul Wagner / DLR).

3. State of the Art

This chapter begins by establishing the foundation methods of astronomical observation, distinguishing between sidereal and non-sidereal tracking. It then addresses the task of object detection in astronomical images, comparing traditional with DL algorithms. Building on this, the next chapter examines state-of-the-art techniques for photometric calibration, detailing differential photometry and absolute referencing. The chapter concludes by discussing the overall role of AI and ML in Astronomy.

3.1. Astronomical Observation of Space Objects

Astronomical Observation is about capturing and interpreting light from celestial bodies. Methods for these observations can be categorized into two types, depending on the motion of the observed object relative to the background stars. The first technique is called sidereal tracking or observation. In this method, the telescope is set up to track the motion of the stars across the sky, which is caused by Earth's rotation. This is the standard method for observing distant and fixed objects such as stars, galaxies, and nebulae. While observing in sidereal mode, objects appear sharp and point-like in the resulting image, see Figure 3.1a. Non-sidereal tracking or observation is used for objects within our Solar System, such as asteroids, comets, or satellites, which move differently against the background stars. The telescope tracks the motion of the observed object across the sky. Figure 3.1b shows that the object appears as a point-like source and the background stars are distorted into streaks [42].


(a) Sidereal Image

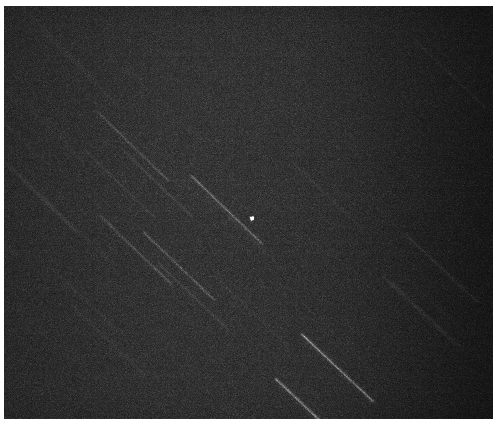

(b) Non-Sidereal Image

Figure 3.1.: Comparison of sidereal and non-sidereal images. (3.1a) A sidereal tracking image in which the stars appear as points, while objects such as satellites would be lines. (3.1b) A non-sidereal image where an observed object (e.g., satellite) would appear as a point, while the stars in the background appear as streaks.

As explained by Sharma et al., the analysis of these star streaks can be used for astrometric and photometric purposes. This technique is crucial for the study of near-Earth and other fast-moving objects [42]. The choice between these observation techniques depends on the scientific goal, but both produce raw image data that require further processing and analysis to extract meaningful information.

3.2. Detection of Space Objects in Astronomical Imagery

Detection of objects in astronomical imagery is one of the first steps. While the human brain can easily and instinctively detect patterns and objects, the sheer amount of data from modern telescopes requires automated and robust algorithms. As described in the following section 3.4, the progress in AI is essential for modern astronomy. These advanced methods replace simple threshold algorithms.

A traditional approach for source detection is the so-called Source Extractor, which is based on a threshold analysis. In this case, the sky background gets modeled and subtracted from the image. Afterwards, coherent pixel groups that exceed a certain threshold above the image noise are identified as a potential object. This approach is fast and effective for the detection of point-like objects [5, 12].

The modern standard shifts to DL models, especially object detection models like You Only Look Once (YOLO) [44] or Mask R-CNN [46]. These models learn not only to identify the presence of an object but also to determine its precise location, shape, and class label within the image. The main advantage lies in the ability to detect complex patterns and shapes, even if these objects overlap or exhibit a low signal-to-noise ratio, where traditional methods fail [5, 46].

One relevant and complex problem is the detection of features like star streak. These occur in two prime scenarios:

- Fast-moving objects: satellites, meteors, or asteroids move very fast in contrast to the fixed starry sky and leave a line mark instead of a point-like shape [42].
- Tracked observation: the telescope gets pointed at a moving object, like a satellite, asteroid, or space debris. In this case, the telescope follows the path of the object, and the object itself is displayed as a point-like shape. The stars in the background, on the other hand, get distorted and form a star streak [42].

Streak detection and analysis in images is important for several reasons. In some contexts, the goal is to identify and remove unwanted satellite streaks to clean scientific data. In other applications, such as tracking asteroids, these streaks are analyzed to extract information about the object's length and orientation. In this case, the star streaks are used for a photometric calibration [5, 46].

In conclusion, it can be said that object detection in astronomy made a jump from procedural algorithms like the Source Extractor [5] or Astreaks [42] to data-driven models such as Mask R-CNN. Table 3.1 summarizes that traditional algorithms are often simpler, faster, and their logic is easier to interpret, making them relevant and ideal for straightforward tasks [13, 28]. In contrast, DL models excel at complex problems but require significant computational resources and a large dataset for training. Nevertheless, one of the biggest issues is that their decision-making process remains a black box, making it unclear why they reach to some conclusions [13, 28]. In cases such as star streak detection, it has demonstrated superior performance, achieving significantly higher sensitivity and accuracy [12, 22, 36]. This is essential for the next generation of detection algorithms.

Table 3.1.: Comparison between traditional and DL algorithms. The table highlights the differences in terms of method, flexibility, development, interpretability, speed, and accuracy.

	Traditional Algorithm	DL Algorithm
Method	Based on pre-defined rules and thresholds	Learns features and rules directly from data
Flexibility	Inflexible - changes in the problem often require redesigning the algorithm	Highly flexible - can be retrained with new data to solve different problems
Development	Requires expert knowledge for feature engineering	Requires large, high-quality training datasets
Interpretability	White box - decisions are easy to understand	Black box - decisions are difficult or unclear to understand
Speed	Generally faster for simple tasks	Can be computationally intensive, especially training
Accuracy	High when input data closely matches the pre-defined rules	Superior for complex or noisy data (assuming a sufficient training dataset)

3.3. Photometric Calibration and Light Curve Analysis

A light curve is a powerful measurement, but only as valuable as the accuracy of the calibration [15]. Photometric calibration is a complex and important process that converts the measured counts of a detector into a standardized physical unit (normally magnitude) [45]. In a time when sky surveys produce thousands or millions of light curves, nightly survey routines have evolved into a highly automated process [22, 45]. The goal is to eliminate instrumental and atmospheric effects to ensure accurate and comparable values of the night sky [45].

Each astronomical image is influenced by the signature of the measuring tool (e.g., camera), atmospheric effects, and other factors such as background brightness, light pollution, or optical aberrations introduced by the telescope. Eliminating these effects is the first and fundamental step for a photometric analysis. Bias-frame, dark-frame, and flatfield correction are the main steps for this preprocessing phase. After that, there is a clean image, whose pixel values are proportional to the captured light. The following steps are the conversion into a standardized physical unit or brightness system [7, 20]. Traditionally, the method for this conversion is based on observations of standard stars. These stars are precisely measured in different color indices. This process includes the correction of atmospheric extinction and the transformation of the instrumental magnitudes into the standard magnitude system [15, 45]. However, modern state-of-the-art methods have replaced this approach [45]. Instead of observing dedicated standard star fields, astronomers use data from massive sky surveys like the Sloan Digital Survey, Pan-STARRS, or the Gaia mission [19, 27]. These catalogs contain precise photometric information for billions of stars over the whole night sky. Rather than a few primary standards, dozens or hundreds of stars directly in the field of view of observation now serve as local reference standards. This increases the accuracy and saves observation time [6].

For the observation of objects, the primary tool is a light curve. Differential photometry is the-state-of-the art approach to creating precise light curves, instead of determining the absolute magnitude of a target object at an exact time. This simple and useful approach measures the brightness of a group of nearby stars in the same image. This offers several advantages, such as the elimination of atmospheric extinction, since the light from the target object and surrounding stars travels through the same path in the atmosphere. Therefore, small atmospheric changes cancel each other out. The other advantage is the compensation of instrumental effects, small changes in sensitivity or the focus of the camera influence all stars and the target object in the same way. This method allows a measurement of brightness variations with high accuracy [20, 45].

A light curve obtained by differential photometry is relative at first. This is sufficient for many scientific analyses (e.g., determining the periods of variable stars). However, absolute calibration is often necessary in order to derive physical parameters of the object [20]. Such a process could look like this and is visualized in Figure 3.2:

- Creation of a precise and relative light curve: the target object is measured based on comparison stars in the same image field.
- Absolute referencing: standard magnitudes of the comparison stars are taken from an all-sky catalog like Gaia.
- Scaling: a calibration factor is calculated from the known brightness of the reference star, which converts the relative light curves of the target into an absolute, calibrated light curve in the standard magnitude system

This hybrid approach ensures that the light curve has both the highly relative precision of the differential method and the absolute accuracy of a calibration against established standards [20, 45].

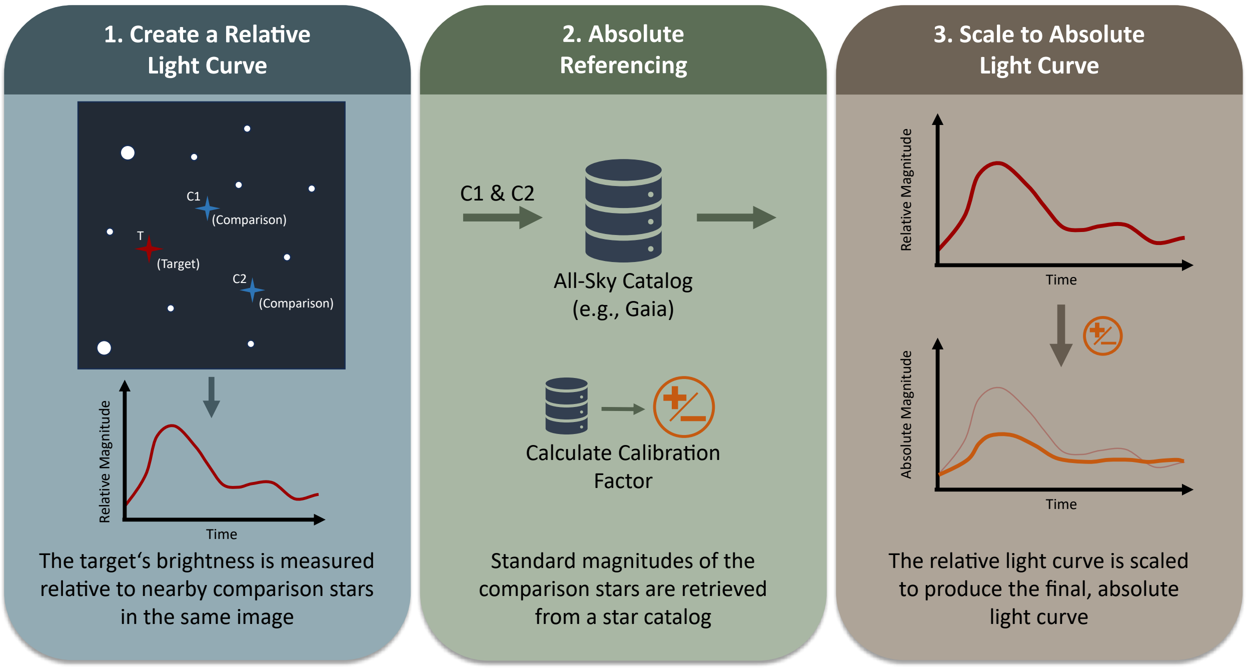

1. Create a Relative Light Curve
C1
(Comparison)
T
(Target)
C2
(Comparison)
Relative Magnitude
Time
The target's brightness is measured relative to nearby comparison stars in the same image
2. Absolute Referencing
C1 & C2
All-Sky Catalog (e.g., Gaia)
Calculate Calibration Factor
Standard magnitudes of the comparison stars are retrieved from a star catalog
3. Scale to Absolute Light Curve
Relative Magnitude
Time
Absolute Magnitude
Time
The relative light curve is scaled to produce the final, absolute light curve

Figure 3.2.: Schematic representation of the three-step process of absolute photometric calibration. First, a relative light curve of the target (T) is created in relation to nearby comparison stars (C1, C2) (1). Next, the standard magnitudes of the comparison stars are retrieved from a star catalog, such as Gaia, to calculate a calibration factor (2). Finally, this factor is applied to the relative light curve to scale it to an absolute light curve (3).

In conclusion, photometric calibration has evolved from a complex, procedural process to a data-driven, highly automated field. The availability of precise all-sky catalogs has reduced the need for observing traditional standard stars and increased the overall accuracy. In the field of time series analysis, differential photometry dominates, leading to absolutely calibrated light curves of the highest precision by connecting them to cataloged reference stars.

3.4. AI and ML in Astronomy

Modern sky surveys create enormous amounts of data, which makes it impossible to analyze manually. For example, the newly built Vera C. Rubin Observatory will produce about 20 terabytes per night [39]. Therefore, automatic image recognition, driven by advances in ML — particularly DL — is no longer a convenient tool but a necessity for modern astronomical research. It enables detection, classification, and analysis of all kinds of objects in the sky [28].

Traditional astronomical image analysis is based on feature-based algorithms. Here, physical parameters are extracted from an image, like brightness, size, and color indices. Afterwards, ML algorithms like support vector machines or random forests are used to extract such features to classify objects. This approach is effective, but still needs an expert for the selection and definition of such features [28][46].

The current state of the art is dominated by DL algorithms, especially of CNNs. The primary advantage is that such approaches are able to extract important features directly and automatically from raw data. Therefore, the network identifies the optimal pattern on its own. These can be simple shapes like edges or corners, or more complex structures like galaxies. This process is not only more effective but also more capable, as the model can detect patterns that the human eye cannot see [46].

Despite significant successes, numerous challenges remain. The performance of a DL model depends on the size and quality of the training dataset. Creating such a dataset can be time-consuming. Another point is the black box behavior of these models, which means that it is unclear to determine why a model made this decision [28, 46].

4. Concept Development

This chapter details the developed concept of this thesis, beginning with the methodology for synthetic data generation, followed by the strategy for training the AI model, and concluding with the photometric analysis pipeline.

This project proposes a multi-stage processing pipeline to solve this challenge. The core concept is to leverage the power of DL to create a robust and automated system for star streak detection, thereby enabling precise photometric calibration. Unlike conventional approaches that rely on pre-scanned star fields that quickly become invalid when conditions change, the proposed concept performs a frame-by-frame calibration, therefore, each frame is calibrated in real-time to adjust to environmental changes. The calibration pipeline is structured into six primary stages, as illustrated in Figure 4.1.

4.1. Synthetic Data Generation

Modern DL models, particularly for complex computer vision tasks, demand a vast amount of accurately annotated training data. Manually creating such a dataset from real telescope imagery would be a time-consuming and error-prone task, requiring an expert to outline thousands of star streaks with pixel-perfect precision. To overcome these demanding tasks, this concept adopts a simulation-based approach by programmatically generating a large synthetic dataset, which creates a controlled environment where the ground truth for every object is known. This provides the necessary volume of data for training an AI model and establishes the foundation for the quantitative validation.

4.1.1. Methodology

The generation process is designed to create realistic 16-bit astronomical images, based on the miniSLR® imagery. This process systematically layers various physical and instrumental effects to construct a synthetic image as close as possible to the original one. Figure 4.2 illustrates the whole generating process of the following steps:

1. Black Image:
   The process begins with a black 16-bit image, which serves as a black canvas.
2. Star Streak Simulation:
   To simulate the appearance of reference stars in a non-sidereal tracked image, each streak is modeled individually. A 2D Gaussian kernel is generated to represent the telescope's PSF, which dictates the blurriness of a point source. The brightness of the kernel at any point (x, y) is given by the formula:

$$G(x,y) = A\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \qquad (x,y) \in [-4\sigma, 4\sigma], \;\; \sigma \in [1.5, 3.5]\ \text{px} \tag{4.1}$$

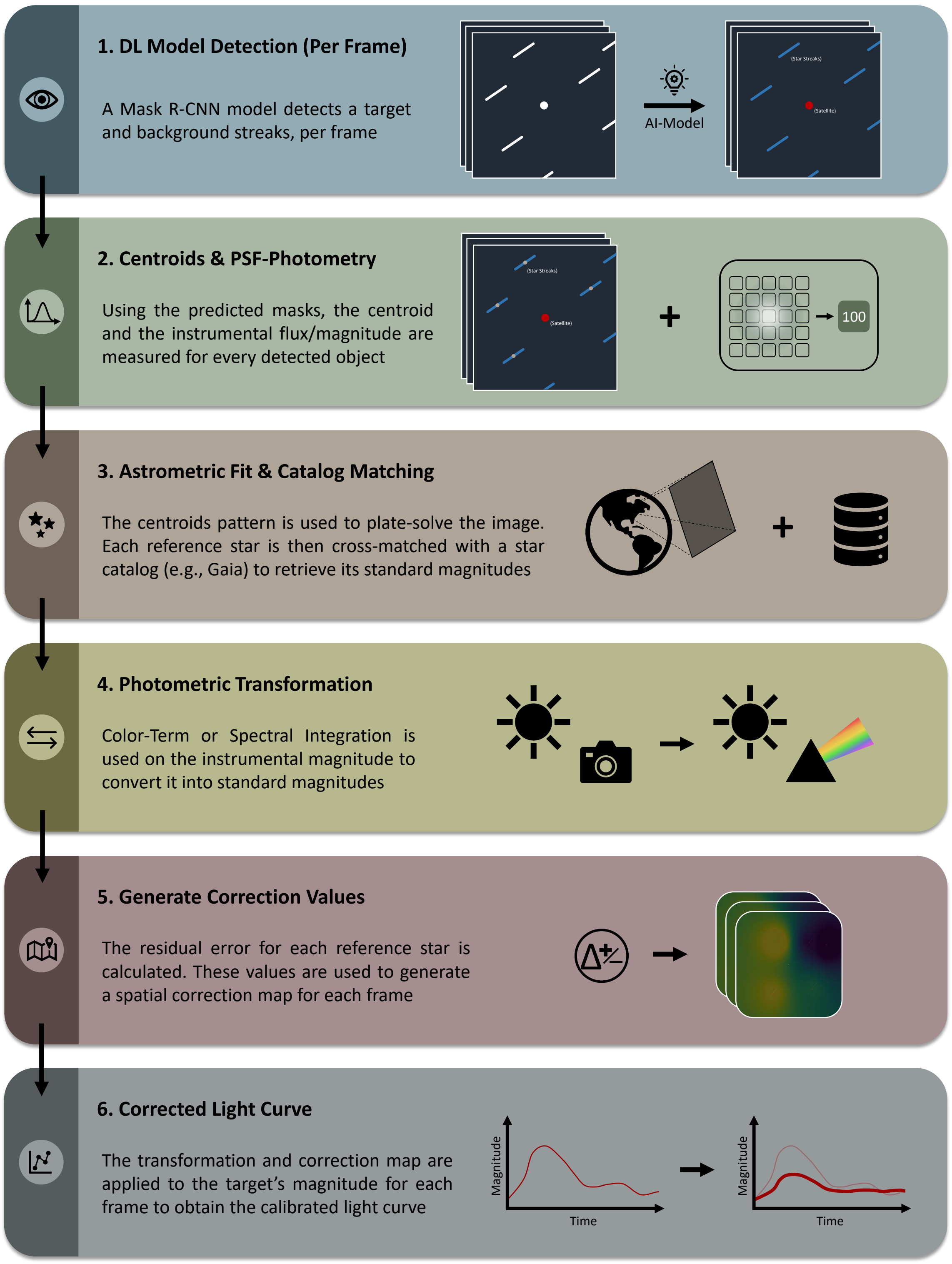

1. DL Model Detection (Per Frame)
A Mask R-CNN model detects a target and background streaks, per frame
AI-Model
(Star Streaks)
(Satellite)
2. Centroids & PSF-Photometry
Using the predicted masks, the centroid and the instrumental flux/magnitude are measured for every detected object
100
3. Astrometric Fit & Catalog Matching
The centroids pattern is used to plate-solve the image. Each reference star is then cross-matched with a star catalog (e.g., Gaia) to retrieve its standard magnitudes
4. Photometric Transformation
Color-Term or Spectral Integration is used on the instrumental magnitude to convert it into standard magnitudes
5. Generate Correction Values
The residual error for each reference star is calculated. These values are used to generate a spatial correction map for each frame
6. Corrected Light Curve
The transformation and correction map are applied to the target's magnitude for each frame to obtain the calibrated light curve
Magnitude
Time

Figure 4.1.: Schematic representation of the calibration pipeline concept. The process begins with the instance segmentation using Mask R-CNN (1). Followed by the astrometric fitting and comparison with star catalogs (2-3), these transform the instrumental magnitude into standard magnitudes (4). A spatial correction map (5) is applied to generate the calibrated light curve of the target object (6).

where (x, y) are the coordinates relative to the center of the kernel, and σ is the standard deviation, which controls the width or "blurriness" of the PSF. This PSF is then *smeared* by being stamped repeatedly along a randomized linear path. Key parameters such as the streak's length, angle, brightness, and PSF width are varied in each image to ensure a diverse dataset.

3. Target Object Simulation:
   A target object is added near the center of the frame. To increase realism, the target is not a simple shape but is generated as either a symmetrical PSF-based object or an irregular shape with a soft PSF falloff, simulating a non-uniform shape.

4. Physical Noise Model:
   A critical step for reducing the *sim-to-real gap* is applying an advanced noise model. The clean image containing the streaks and a target is degraded with the following effects:

   - Read Noise: Signal-independent Gaussian noise (so-called read noise) is added. This term models thermal fluctuations and imperfections in the readout noise electronics. It is typically represented as
   $$\mathrm{n}_{\mathrm{read}} \sim \mathcal{N}(0, \sigma^2_{\mathrm{read}}) \tag{4.2}$$
   where σ^2_{read} is the standard deviation of the read noise distribution.

   - Shot Noise: Signal-dependent Poisson noise (so-called shot noise) is used, which realistically models the quantum nature of light detection. Because the photons arrive randomly, the number of detected photons follows a Poisson distribution:
   $$\mathrm{n}_{\mathrm{shot}} \sim \mathrm{Poisson}(I^2_{\mathrm{signal}}) \tag{4.3}$$
   where I_{signal} is the expected photon count proportional to the input intensity.

   - Dark Noise (Dark Current): Even in complete darkness, thermal energy within the sensor can generate electrons. This dark noise is indistinguishable from electrons generated by photons. The number of dark electrons $\mathrm{n}_{\mathrm{dark}}$ in a pixel follows a Poisson distribution whose mean is determined by the dark current rate and the exposure time:
   $$\mathrm{n}_{\mathrm{dark}} \sim \mathrm{Poisson}(D \cdot t) \tag{4.4}$$
   where D is the dark current rate (in $\mathrm{e}^-/\mathrm{pixel/s}$) and t is the exposure time.

   The final noisy measurement can be expressed as:
   $$\mathrm{I}_{\mathrm{noise}} = \mathrm{n}_{\mathrm{read}} + \mathrm{n}_{\mathrm{shot}} + \mathrm{n}_{\mathrm{dark}} \tag{4.5}$$
   This formulation realistically reproduces both fixed electronic noise and intensity-dependent fluctuations, closely approximating the behavior of real imaging sensors.

5. Sensor Artifacts:
   To simulate common detector defects, artifacts such as random hot/dead pixels and vertical or horizontal band noise are introduced.

6. Background Gradients:

Large-scale, non-uniform variations are added, including vignetting (darkening of corners), and random linear gradients to simulate skyglow or instrumental effects.

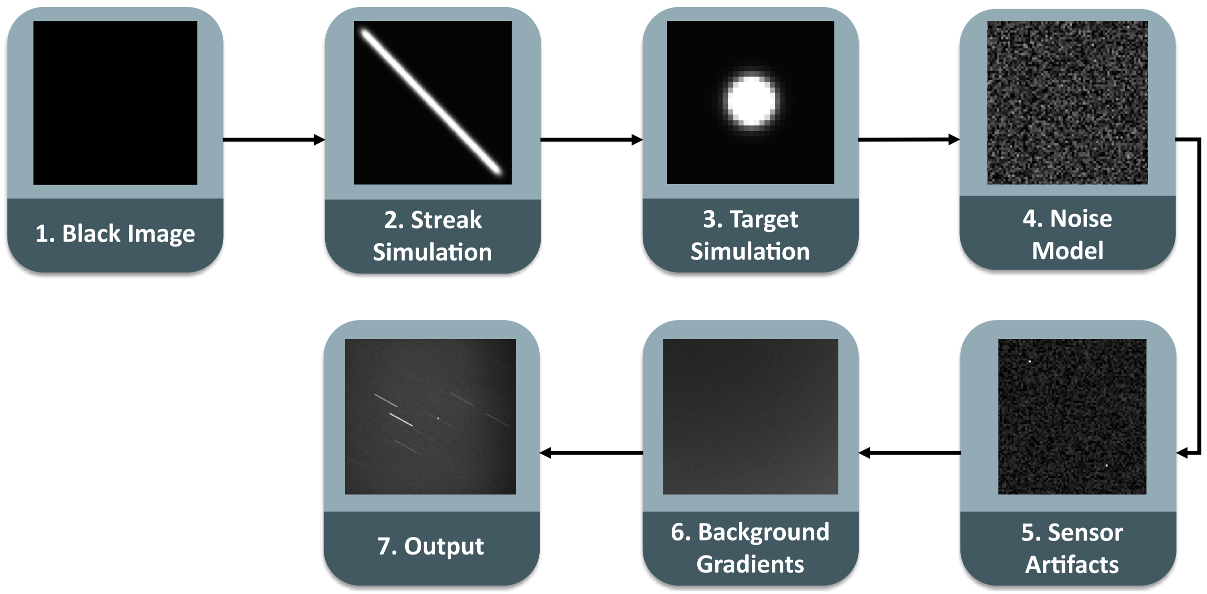


Figure 4.2.: The step-by-step pipeline for generating synthetic astronomical imagery. Based on a blank image (1), a star streak (2), a target object (e.g., satellite)(3), realistic noise (4), sensor artifacts (5), and background gradients (6) are added in sequence to generate the final output image (7).

4.1.2. Ground Truth Generation

A key advantage of this synthetic approach is the ability to generate an accurate ground truth for both model training and validation. This process runs in parallel with image generation and consists of three components:

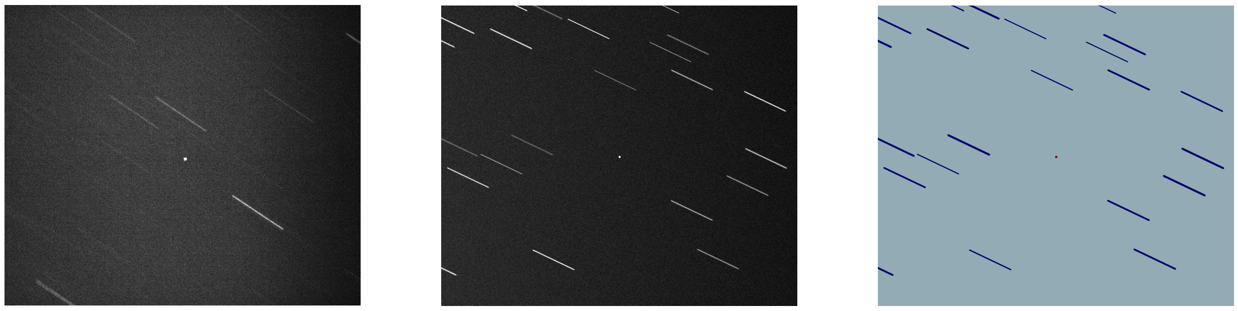
1. Segmentation Mask Generation:
   For every object drawn onto an image canvas, a corresponding, simplified geometric shape is drawn onto a parallel mask canvas. For instance, a smeared PSF streak on the main image corresponds to a simple, solid line with rounded ends on the mask image. This ensures a pixel-perfect correspondence between the visual objects and their ground truth. Furthermore, each streak gets its own shade of blue, the target object its own shade of red, and the background is colored black. This helps to distinguish between each instance and each class in the mask image.

2. COCO-Standard Annotation File:
   Then, the images and masks are created. The scripts generate a JSON annotation file in the industry standard Common Objects in Context (COCO) format [24, 26]. The script analyzes the generated masks to extract the necessary information for each object, such as its category (e.g., star trail or target), its precise bounding box coordinates, and its segmentation data. This structured JSON file is the input for training the Mask R-CNN model. The main advantages of this kind of data style are a better instance-level distinction, more precise geometry of instances, the possibility to add additional metadata that can be used to perform multi-task feature learning, and its tool-friendly and standardized working process with the COCO frameworks.

3. Photometric Analysis Ground Truth:
   To enable the final validation of the system's accuracy, the true brightness and center points of each object are extracted and stored in an external CSV-file. This provides the quantitative ground truth against which the photometric analysis is evaluated.

4.1.3. Final Output and Visual Comparison

The goal of the data generation stage is to produce a dataset that is not only large but also closely mirrors the characteristics of real-world data. Figure 4.3 provides a qualitative comparison between a real and a synthetic image, along with the corresponding ground trough mask of the generated image.



(a) Real Image of the miniSLR® (b) Synthetic Generated Image (c) Synthetic Generated Mask

Figure 4.3.: Side-by-side comparison of a real miniSLR® image (4.3a) and a synthetic image (4.3b), with the corresponding segmentation mask (4.3c) for the synthetic image.

The final output of this generation pipeline is a comprehensive dataset, structured into training, validation, and test sets. Each set contains the generated images and corresponding masks, the annotation file in COCO-style, and the validation CSV-file for the photometric analysis validation.

4.2. Deep Learning-based Streak Detection

With a robust synthetic dataset, the next stage is to develop an AI model capable of accurately and automatically identifying the objects of interest within the images. This section details the selection of the appropriate computer vision task, the state-of-the-art architecture chosen, and the multi-stage strategy used to train the model.

4.2.1. Task Definition: Instance Segmentation

The primary goal of the AI model is to provide a precise pixel-level location of every star streak and the target object, ensuring that even overlapping or closely spaced objects are treated as distinct entities. This requirement rules out simpler computer vision tasks. For instance, object detection would only provide bounding boxes, which are insufficient for accurate photometry as they include background pixels and merge multiple objects. Semantic segmentation, while providing pixel-level classification, would label all star streaks as a single class, making it impossible to separate individual objects.

The ideal task is, therefore, instance segmentation, see subsection 2.3.1. This advanced technique combines the strengths of both methods: it detects all objects in an image and, for each individual instance, generates a precise pixel-level mask. This one-to-one correspondence between an object and its mask is the essential prerequisite for the subsequent photometric analysis.

4.2.2. Choice of Architecture: Mask R-CNN

To perform the task of instance segmentation, this project utilizes the Mask R-CNN architecture. The selection of this specific model was driven by the following requirements: prioritizing precision, robustness, and training efficiency. The functionality of the model is described in subsection 2.3.2. Mask R-CNN is a state-of-the-art network, but the following factors make it an optimal choice for this application:

- Unparalleled Precision:
  The primary justification for the selection of the Mask R-CNN is the ability to generate highly accurate pixel masks. For a scientific tasks like photometry, the quality of the input data is important. A clean segmentation mask that precisely outlines the objects from the background is essential for minimizing noise and achieving an accurate flux measurement.

- Robustness in Complex Scenes:
  Astronomical images present a unique set of challenges, including faint objects, varying object scales, and the potential for overlapping star streaks. Mask R-CNN is designed to be robust in such scenarios. Its Region Proposal Network is effective at identifying potential objects across the entire image, regardless of their size, while at a later stage, it can separate and delineate even closely positioned or intersecting objects. This capability is crucial for ensuring that every reference star is individually segmented.

- Efficiency through Transfer Learning:
  A significant advantage of this model is the availability of pre-trained variations on large datasets like ImageNet. This project's training strategy hinges on transfer learning, see subsection 4.2.3. By starting with a pre-trained model, a foundation of learned visual features is used, reducing the required training time and improving final performance.

4.2.3. Training Strategy: Three-Phase Transfer Learning

Training a complex model like Mask R-CNN from scratch would require an immense dataset and computational resources. To achieve high performance, a three-phase transfer learning strategy is used:

1. Pre-training:
   The process begins with a Mask R-CNN model that has been pre-trained on the large-scale dataset ImageNet. This provides the model with a fundamental understanding of general visual features like edges, textures, and shapes.

2. Synthetic Dataset:
   The pre-trained model is then fine-tuned on the large, custom-generated synthetic dataset. During this phase, the model learns to identify unique features of star streaks and target objects.

3. Reality Adaptation:
   To reduce the sim-to-real gap, a final fine-tuning phase is performed, on a real dataset. This step allows the model to adjust its learned features to fine details and noise characteristics of the actual data, ensuring its robustness when deployed on real-world data.

4.2.4. Optimization: Hyperparameter Tuning

The performance of the training process is sensitive to the choice of hyperparameters (e.g., learning rate, optimizer, batch size, etc.). To find the optimal combination, a systematic and automated approach is used. The *Optuna* framework [3] is used to conduct a structured search, running numerous training trials with different parameter sets. This process identifies the configuration that yields the lowest validation loss, thereby maximizing the model's accuracy.

4.3. Photometric Analysis

The AI model provides the crucial step of identifying where the objects of interest are located. The photometric analysis aims to transform this output into useful data for later processing steps. This part is responsible for extracting two key parameters for every object: its precise centroid and its accurate instrumental brightness (flux).

4.3.1. Centroid Determination

The center coordinates of each reference star streak must be known for the following astronomic calibration, which is the process of mapping image pixel coordinates to absolute sky coordinates. This is achieved by calculating the geometric center, or centroid, of each individual mask predicted by the model. Mathematically, this is calculated using image moments. For a binary mask, where pixels inside the shape have a value of 1 and pixels outside have a value of 0, the moments are calculated as follows:

The zero moments (m_{00}) represent the total area (number of pixels) of the segmentation mask:

$$m_{00} = \sum_x \sum_y I(x,y) \tag{4.6}$$

The first-order moments $((m_{10})$ and $(m_{01}))$ represent the sum of all x and y coordinates within the shape:

$$m_{10} = \sum_x \sum_y x \cdot I(x,y) \tag{4.7}$$

$$m_{01} = \sum_x \sum_y x \cdot I(x,y) \tag{4.8}$$

Where $I(x,y)$ is the value of the pixel at coordinates (x, y). The centroid coordinates (C_x, C_y) are then found by dividing the first-order moments by the total area:

$$C_x = \frac{m_{10}}{m_{00}}, \quad C_y = \frac{m_{01}}{m_{00}} \tag{4.9}$$

This calculation provides an estimate of the streak's center, which serves as the input for the astronomy software.

4.3.2. PSF Estimation for Star Streaks

High-precision photometry requires an accurate model of the Point Spread Function. In a tracked image, all stars appear as streaks, meaning there are no point-like sources from which to directly measure the

PSF. Because the PSF is system-dependent and may vary across the field, it is derived locally from the streaks based on the principle that for a straight and constant rate, the cross-section of a star streak is a good approximation of the underlying PSF.

- Axis Determination via PCA:
  Principal Component Analysis (PCA) is used to determine the orientation of each streak. PCA is a statistical procedure that finds the primary axes of variance within a cloud of points. In this case, the pixel coordinates of the mask. For a streak, the first principal component aligns with its length, and the second is perpendicular to it, defining the axis cross-section. This is achieved by solving the eigenvalue problem for the covariance matrix of the point coordinates (X):

  $$\mathrm{Cov}(X) \cdot v = \lambda \cdot v \tag{4.10}$$

  Where v are the eigenvectors (principal components) and λ are the eigenvalues (variance along those components)

- FWHM Calculation: The brightness values of each streak's pixels are projected onto its perpendicular axis, creating a 1D brightness profile. A 1D Gaussian model is fitted to this profile to measure its Full Width at Half Maximum (FWHM). To create an estimate for the entire image and reject outliers from faint or noisy streaks, the FWHM from all detected star streaks is taken as the definite value for the image. This is also converted to the standard deviation σ of the PSF

4.3.3. Model Fitting

With an estimate of the psf_σ, the brightness of every object is measured using PSF photometry. This approach is more accurate than simple pixel summation (aperture photometry) because it separates the object's signal from the background noise and random sensor fluctuations. Instead, this method fits a mathematical model of the object's brightness profile to the data. The final flux is then derived from this ideal, noise-free model.

- For Star Streaks (1D Model Fitting):
  The measurement for each streak is performed on its 1D cross-section profile. A 1D Gaussian model is fitted to the noisy data points. A constraint is applied during this process: the σ (width) of the Gaussian model is fixed to the median value that was estimated for the entire image. By constraining the shape of the model, the fitting algorithm can focus on determining the free parameters: amplitude (peak brightness) and background offset. The final flux is calculated from the integral of this best-fitting Gaussian curve, providing a background-subtracted brightness value.

- For the Target Object (2D Model Fitting): The same principle is applied to the non-streak target object, in two dimensions. A 2D Gaussian model is fitted to the pixel data within the target's masks.

4.3.4. Final Output

The final output of this process is a structured list containing every detected object, each with its calculated centroid coordinates and instrumental flux. The accuracy of the entire process is quantitatively verified by comparing the measured flux values from the synthetic test against their corresponding ground through values. The validation results are listed in section 5.2.

4.4. Astrometric and Photometric Calibration

An important phase of the concept is to bridge the gap between the pipeline's internal instrumental measurements and the standardized astronomical coordinate and magnitude system. This calibration process connects the data to the physical world, allowing a scientific conclusion and comparisons.

4.4.1. Astrometric Fitting

The first step in this calibration process is to determine the celestial coordinates that correspond to every pixel mask in the image. This is achieved through a process called *plate solving*.

The list of (x,y) centroids, extracted by the photometric analysis, serves as the input. This list of coordinates, representing the geometric pattern of the stars in the image is fed into an astronomy tool such as Tetra3 [9]. The software compares this pattern against a star catalog (e.g., the Gaia catalog). By finding a unique match it solves the image's world coordinate system, which maps 2D pixel positions on the 3D celestial sphere.

4.4.2. Photometric Transformation

Once the image is astronomically solved, each detected star streak can be cross-referenced with the catalog to retrieve its known, true magnitude. This, however, presents a significant challenge to the project.

No optical filters are present in the telescope-camera path. In contrast, scientific star catalogs list magnitudes in specific photometric bands, such as the Johnson-System or the Gaia G-band. A direct comparison between the instrumental flux and a catalog magnitude is not physically valid, as stars of different color temperatures will register with different relative brightness on the sensor.

To overcome this challenge, a transformation between the instrumental and standard systems must be defined. The following two methods are presented to address this problem.

Method 1: Color-Term Transformation

This transformation method models the instrumental and standard system using the color of reference stars. It relies on a transformation equation established by fitting the data from all reference stars in the image. The relation is modeled as:

$$m_{\text{cat}} = m_{\text{inst}} + k \cdot X + ZP + C \cdot (B - V) \tag{4.11}$$

Where m_{inst} and m_{cat} are the measured instrumental and catalog magnitude. The atmospheric extinction is modeled through $(k \cdot X)$, where k is the extinction factor, which describes how much the atmosphere reduces the light, and X is the air mass. ZP is the zero-point, a calibration constant that represents the instrumental magnitude of a magnitude-0 star and accounts for factors like atmospheric transparency and exposure time. $(B-V)$ is the color index of the cataloged stars. It is the difference between the star's magnitude in the blue (B) and the visual (V) bands and approximates the star's temperature. C is the color term, this coefficient solves the unfiltered data problem. It models the unique spectral response of the camera system.

Determining all the factors above is difficult, mainly because factors, such as the atmospheric extinction, are dynamic and can change over time. However, the frame-based real-time calibration is beneficial. Due to the small field of view of the system, it can be assumed that the light of the observed object and the background star travels through the same air mass and interferes with the same disturbances. Because of this and the fact that the calibration is only valid for one frame at a time, effects like the atmospheric extinction can be neglected. Therefore, Equation 4.11 can be simplified to Equation 4.12:

$$m_{\text{cat}} = m_{\text{inst}} + ZP + C \cdot (B - V) \tag{4.12}$$

The calibration is performed by rearranging the equation into the form of a line, $y = mx + b$, see Equation 4.13.

$$\Delta m = m_{\text{cat}} - m_{\text{inst}} = C \cdot (B - V) + ZP \tag{4.13}$$

A linear least-squares fit is then performed on the reference stars, plotting $(\Delta m = m_{\text{cat}} - m_{\text{inst}})$ against $(B - V)$. The slope of the resulting line is the color term C, and the y-intercept is the zero-point ZP. Once determined, these two constants can accurately transform the measured instrumental magnitude of the observation target into a standard, calibrated magnitude.

Calculation Example This example demonstrates the color-term transformation method.

After the image analysis and the comparison with a star catalog, the following exemplary data is available, see Table 4.1.

Table 4.1.: Example values illustrating color-term transformation. The table lists the instrumental magnitude (m_{inst}), catalog magnitude (m_{cat}), and the color index $(BR - RP)$ for several reference stars. The difference $(\Delta m = m_{\text{cat}} - m_{\text{inst}})$ shows the deviation that correlates with the color index and is to be corrected by the transformation.

Object	measured m_{inst} [mag]	catalog m_{cat} [mag]	$BP{-}RP$	Δm [mag]
Satellite	−10.10	–	(Assumed: 0.82)	–
Star 1	−8.35	−11.20	0.55	−2.85
Star 2	−8.82	−10.95	0.81	−2.13
Star 3	−7.55	−11.60	0.41	−3.51
Star 4	−8.03	−10.72	1.15	−2.69
Star 5	−7.91	−11.00	0.72	−3.09

The difference Δm for each reference star is calculated, and this will be plotted against the color index. Using Equation 4.13, the best-fitted line describing the five data points is determined. Figure 4.4 shows the data point and the linear regression. Therefore, the following values for C and ZP are obtained:

- Slope (C): $C = -0.54$
- Y-intercept (ZP): $ZP = -2.57\,\text{mag}$

These values are only valid for a single frame. The ZP is the bias between the instrumental scale and the catalog scale, and the color-term C describes how the sensitivity of the system depends on the color. A positive value means that the system is more sensitive to redder stars than to blue ones. Finally, the calculated coefficients can be inserted into Equation 4.12 for the transformation:

$$m_{\text{cat,sat}} = m_{\text{inst,sat}} + ZP + C \cdot (\text{color index}_{\text{sat}})$$
$$m_{\text{cat,sat}} = -10.1\,\text{mag} + -2.57\,\text{mag} + 0.51 \cdot 0.82$$
$$m_{\text{cat,sat}} = -12.26\,\text{mag}$$

It should be mentioned that this method is less accurate than the second method (Spectral Integration, section 4.4.2). This is because the color index of the satellite has to be estimated. Here, it is simply estimated to reflect sunlight and thus show this color. In reality, this cannot be assumed, as various materials or structures (such as solar panels) mean that the reflected light does not necessarily correspond to the color index of the sun. This value is, therefore, only a first approximation.

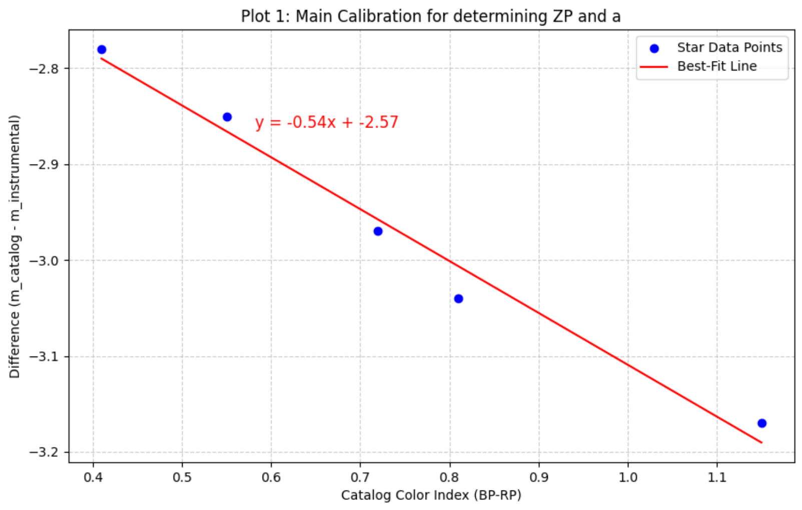


Figure 4.4.: Determination of the photometric zero-point ZP and the color-term C. The graph shows the difference between catalog and instrumental magnitude as a function of the color index $(BR-RP)$ of the reference stars (blue dots). The red line represents the linear regression, whose equation $(y = -0.54x - 2.57)$ provides the calibration parameters.

The result shows the calibrated value of $-12.26\,\text{mag}$ for the apparent magnitude of the satellite, in the given frame. This process must be done for every frame and each star in it.

Color Index Calibration Assuming the color index is less accurate and does not replicate the true index of the satellite, it is only a rough approximation. A better approach is to calibrate the index, too. For this, the hardware of a color camera can be used. Each color camera has a way to distinguish between red, blue, and green, the most common way for this is to use a Bayer-Matrix, which is a color filter array that can filter these colors. This can be used to calibrate the color index of a target object. Rather than just measuring the whole flux of an object, the separate flux for each pixel in the red, blue, and green filters of the Bayer-Matrix is measured. Therefore, this provides three instrumental brightnesses: $m_{\text{inst,red}}$, $m_{\text{inst,blue}}$, $m_{\text{inst,green}}$.

From this, an instrumental color index can be built, for example, $(m_{\text{inst,blue}} - m_{\text{inst,red}})$, which describes how the camera sensors see colors. As before, the linear relationship can be used for the calibration, but instead of plotting Δm against the catalog color index $(BP-RP)$, $(BP-RP)$ is plotted against the new instrumental color index $\Delta i = m_{\text{inst,blue}} - m_{\text{inst,red}}$, with this Equation 4.14 follows:

Table 4.2.: Example values illustrating color-index calibration. The table lists the instrumental magnitude (m_{inst}), instrumental magnitude with blue and red filters ($m_{\text{inst,blue}}$, $m_{\text{inst,red}}$), the instrumental color index (Δi), and the catalog color index ($BR - RP$) for several reference stars. The difference ($\Delta i = m_{\text{inst,blue}} - m_{\text{inst,red}}$) shows the instrumental color index.

Object	m_{inst} [mag]	$m_{\text{inst,blue}}$ [mag]	$m_{\text{inst,red}}$ [mag]	Δi [mag]	$BP-RP$
Satellite	-10.10	-10.21	-10.31	0.1	–
Star 1	-8.35	-8.81	-8.53	-0.28	0.55
Star 2	-8.82	-9.15	-8.98	-0.17	0.81
Star 3	-7.55	-8.39	-7.82	-0.57	0.41
Star 4	-8.03	-8.59	-8.23	-0.36	1.15
Star 5	-7.91	-8.52	-8.10	-0.42	0.72

$$BP - RP = c_1 \cdot \Delta i + c_2 \tag{4.14}$$

Through this linear regression, the transformation coefficients c_1, c_2 can be calculated. Figure 4.5 shows the regression for the example values in Table 4.2. Therefore, the following values for c_1 and c_2 are obtained:

- Slope (c_1): $c_1 = -1.86$
- Y-intercept (c_2): $c_2 = 0.06$

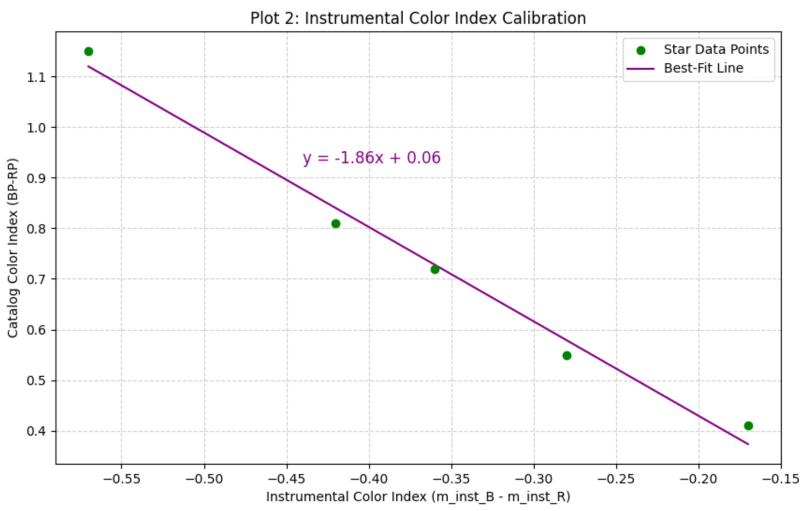


Figure 4.5.: Calibration of the instrumental color index. The graph shows the linear relationship between the measured instrumental color index ($m_{\text{inst,blue}} - m_{\text{inst,red}}$) and the true catalog color index ($BP - RP$) for the reference stars (green). The regression line (purple) provides the transformation equation ($y = -1.86x + 0.06$) for the calibration parameters.

These values can be inserted into Equation 4.14, and with this, the true color index of the satellite can be calculated via Equation 4.15:

$$\begin{aligned} BP - RP_{\text{sat}} &= -1.86 \cdot 0.1 + 0.06 \qquad (4.15)\\ BP - RP_{\text{sat}} &= 0.414 \end{aligned}$$

Therefore, the measured satellite has a color index of 0.414, which should be more accurate than the assumption of 0.82. The positive values also means that the camera measured more red color than blue color.

In the case, of the miniSLR® this approach is currently not possible because the installed camera is monochrome and, therefore, does not have any color filter to perform this index calibration.

Method 2: Spectral Integration Calibration

This second method is a more fundamental approach because it aims to model the physical process of photon detection. Instead of relying on fitting a model, it calculates the instrumental flux by integrating a known star spectrum against the camera's response curve.

The measured instrumental flux f_{inst} by the camera sensor is the total number of photons detected across all wavelengths. This can be expressed as follows:

$$f_{\text{inst}} = \int_{\lambda} S(\lambda) \cdot R(\lambda) \cdot T_{\text{atm}}(\lambda)\, d\lambda \qquad (4.16)$$

Where $S(\lambda)$ is the Spectral Energy Distribution (SED) of the star, this is the star's true spectrum, specifying its flux at each wavelength λ. This is to be obtained from spectral libraries. $R(\lambda)$ is the Total System Response Curve, this function represents the efficiency of the observation system at each wavelength. It is the product of the camera sensor's quantum efficiency (QE(λ)) and the transmittance of the telescope optics ($T_{\text{optic}}(\lambda)$), where the quantum efficiency curve of the camera sensors is the dominant factor. This relation is described in Equation 4.17:

$$R(\lambda) = QE(\lambda) \cdot T_{\text{optic}}(\lambda) \qquad (4.17)$$

The response curve for the miniSLR® is shown in Figure 4.6. $T_{\text{atm}}(\lambda)$ is the atmospheric transmission, but like in section 4.4.2, it can be neglected because the system has a small field of view and the calibration is only valid for one time-frame.

The calibration process of this method includes the following steps:

1. System Characterization: Measure or assume a standard response curve $R(\lambda)$ for the optical path.
2. Theoretical Flux Calculation: For each reference star, retrieve its SED $S(\lambda)$. Then compute the integral $f_{\text{theo}} = \int S(\lambda) \cdot R(\lambda)\, d\lambda$. This value is the theoretical flux that the system should have measured.
3. Zero Point Determination: Compare each reference star's measured instrumental flux to the calculated theoretical flux. The ratio provides a local-dependent calibration factor in a single image frame. This constant is the instrumental zero-point ($ZP_{\text{frame},i}$).
4. Object Calibration: Once $ZP_{\text{frame},i}$ is determined, it can be used to convert the measured flux of the observed object into a physically calibrated flux.

This method is more physically precise but more complex, as it relies on the availability of high-quality spectral data for the reference stars and an accurate characterization of the camera's response curve.

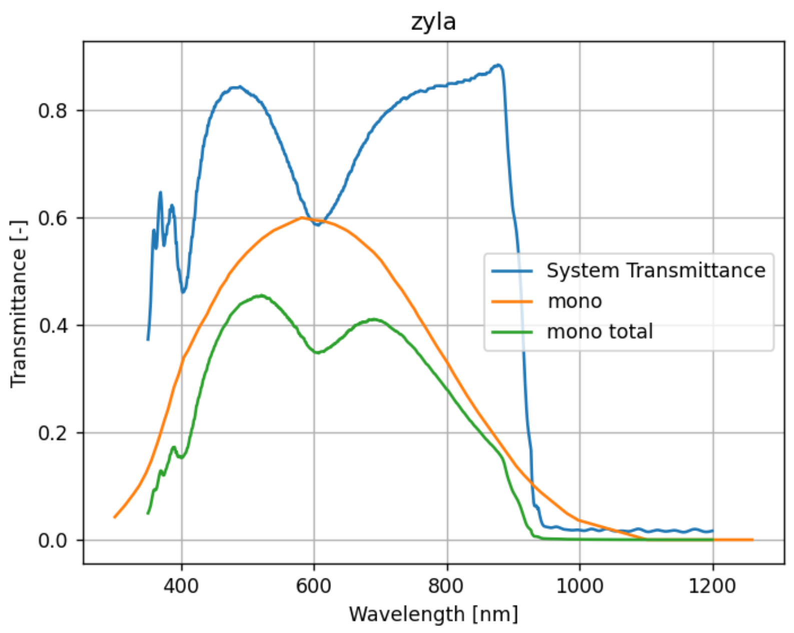


Figure 4.6.: Total System Response Curve of the miniSLR®. The plot shows the system's efficiency for detecting light at different wavelengths. The system transmittance (blue) shows the throughput of the optical components. The values are derived from the datasheets of the individual optical components and are not based on measurements.

Calculation Example This example demonstrates the spectral integration method.

First, the system parameters are defined. As a reference object, a star of the spectral type G2V (sun-like) is used, its SED $S(\lambda)$ can be taken from a spectral library (e.g., MILES [30]). Figure 4.7 shows an example of the SED from the sun-like star *HD 245*. The reference system is the G-band from the Gaia catalog. Its spectral transmission curve is referred to as $R_{\text{Gaia}}(\lambda)$, and is shown in Figure 4.8. The camera system of the miniSLR® $R_{\text{sys}}(\lambda)$ is described in Figure 4.6.

Table 4.3.: Example calculations for spectral integration. The table shows how the star spectrum $S(\lambda)$ is combined with the response curve of a reference system $R_{\text{Gaia}}(\lambda)$ and the measurement system $R_{\text{sys}}(\lambda)$ at different wavelengths (λ). The integrated product of these curves is used to calculate the expected flux in each system.

Wavelength λ [nm]	$S(\lambda)$	$R_{\text{Gaia}}(\lambda)$	$S(\lambda)\cdot R_{\text{Gaia}}(\lambda)$	$R_{\text{sys}}(\lambda)$	$S(\lambda)\cdot R_{\text{sys}}(\lambda)$
...	...	...	...	...	...
400	0.8	0.50	0.40	0.55	0.44
550	1.0	0.72	0.72	0.80	0.80
700	0.8	0.65	0.52	0.85	0.68
...	...	...	...	...	...

The measured flux is the integral of the star spectrum multiplied by the system's response curve (Equation 4.16). In practice, with discrete values, this becomes a sum, see Equation 4.18.

$$f = \sum S(\lambda_i) \cdot R(\lambda_i) \cdot \Delta\lambda_i \tag{4.18}$$

Where $\Delta\lambda_i$ is the step size in the wavelength.

Now, the flux for both spectra is calculated. Table 4.3 shows some example calculations for the flux in the Gaia G-band and Zyla camera..

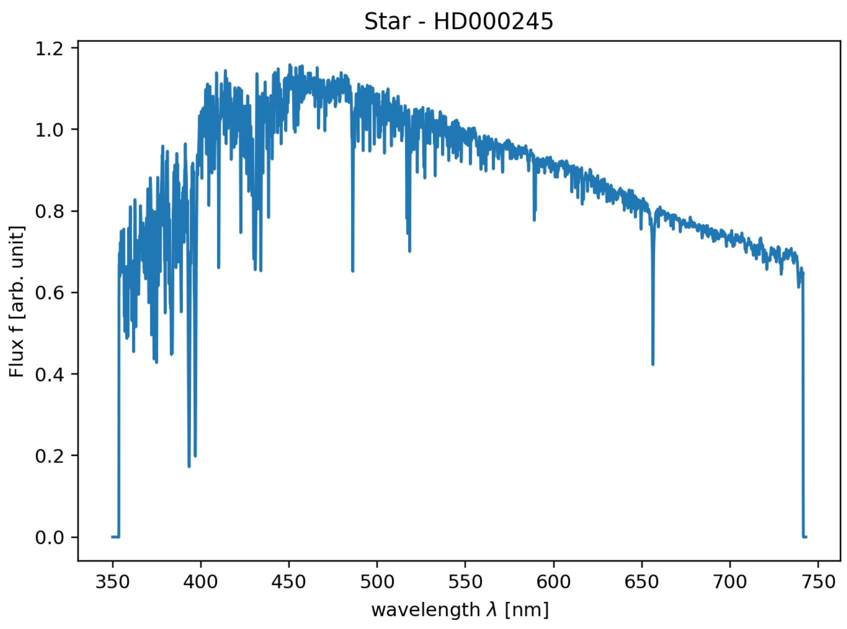


Figure 4.7.: Spectral Energy Distribution of the star HD 245, a star of spectral type G2V. The data comes from the MILES spectral library. Shown is the normalized stellar flux in arbitrary units as a function of wavelength in nanometers [30].

Based on the respective curves, the following fluxes are calculated for the G2V reference star (units are arbitrary). The camera system is more sensitive in red and blue as the G-band, therefore, the total flux will be higher. Assuming the sum is:

- Flux in Gaia G-band (f_{Gaia}):

$$f_{\text{Gaia}} = \sum S(\lambda) \cdot R_{\text{Gaia}}(\lambda) \cdot \Delta\lambda = 150.000\,\text{units}$$

- Flux in camera system (f_{sys}):

$$f_{\text{sys}} = \sum S(\lambda) \cdot R_{\text{sys}}(\lambda) \cdot \Delta\lambda = 210.000\,\text{units}$$

The conversion from flux to magnitude is performed using the standard Equation 2.7:

- Gaia G-band magnitude (m_{Gaia}):

$$m_{\text{Gaia}} = -2.5 \cdot \log_{10}(150,000) = -12.94\,\text{mag}$$

- Camera system magnitude (m_{sys}):

$$m_{\text{sys}} = -2.5 \cdot \log_{10}(210{,}000) = -13.31\,\text{mag}$$

The value m_{sys} is the theoretical expected brightness value, which the system should measure for this star. The next step is to determine the zero-point (ZP_{frame}). In one observation frame, the instrumental brightness depends on variable factors such as exposure time, gain, and atmospheric conditions:

- Measured brightness of the reference star ($m_{\text{star,measured}}$) = -8.5 mag

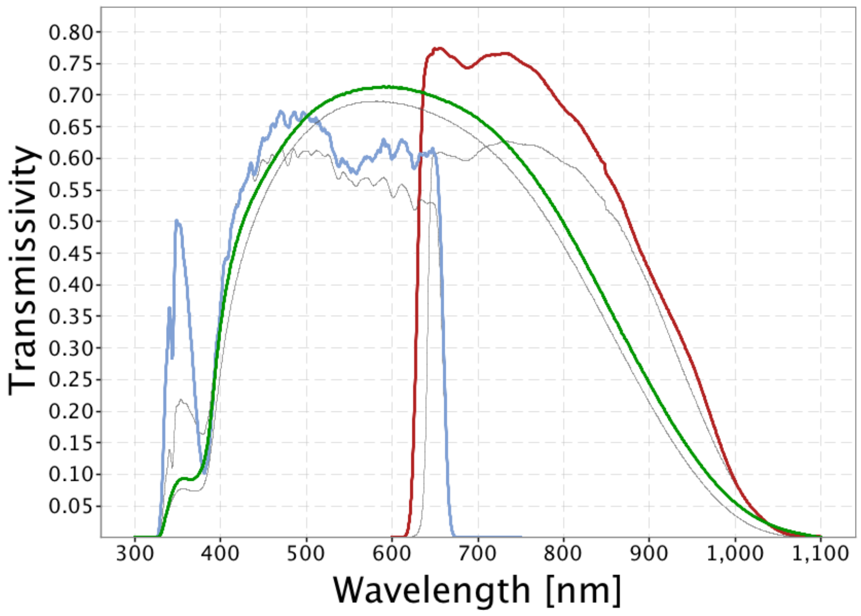


Figure 4.8.: Filter transmissivity of the Gaia space probe. The plot shows the transmission curves for Gaia's three photometric filters. The blue photometer (G_{BP}, blue), the red photometer (G_{RP}, red), and the main broadband G-filter (G, green) [1].

The ZP_{frame} is the difference between the measurement and the theoretically expected value

$$ZP_{\text{frame}} = m_{\text{star,measured}} - m_{\text{sys}}$$
$$ZP_{\text{frame}} = -8.5\,\text{mag} - (-13.31\,\text{mag}) = 4.81\,\text{mag}$$

This offset ZP_{frame} corrects all instrumental and atmospheric effects for the star's position in the image for that given frame and time. The instrumental brightness is also measured for the target object in that frame.

- Measured brightness of the target object ($m_{\text{targ,measured}}$) = -10.3 mag

To calibrate this value, the offset is subtracted:

- Calibrated brightness of the target object ($m_{\text{targ,calib}}$):

$$m_{\text{targ,calib}} = m_{\text{targ,measured}} - ZP_{\text{frame}}$$
$$m_{\text{targ,calib}} = -10.3\,\text{mag} - 4.81\,\text{mag} = -15.11\,\text{mag}$$

This value is the calibrated brightness of the target object in the camera's system and is free from disturbances. At last, to make the value comparable, it is converted into G-band's equivalent brightness. Therefore, the difference between the calibrated target brightness and the expected star brightness (both in the same system) is added to the standard magnitude of the star:

$$m_{\text{targ,G-equivalent}} = m_{\text{Gaia}} + (m_{\text{targ,calib}} - m_{\text{sys}})$$
$$m_{\text{targ,G-equivalent}} = -12.94\,\text{mag} + (-15.11\,\text{mag} - (-13.31\,\text{mag}))$$
$$m_{\text{targ,G-equivalent}} = -14.74\,\text{mag}$$

This result is the apparent magnitude of the target object with a magnitude of -14.74 mag in the Gaia G-band, for this given frame and time.

4.4.3. Comparison of Calibration Methods

Both methods provide a valid way to calibrate the data, but they differ in their approach, data requirements, and complexity. Table 4.4 shows a quick overview of the methods and shows the differences:

Table 4.4.: A comparative overview of two photometric calibration methods. The table outlines the core principle, data requirements, complexity, and source of error for the Color-Term Transformation and Spectral Integration Calibration.

	Color-Term Transformation	Spectral Integration Calibration
Core Principle	Empirical Fit: Assumes a linear relationship between color and system response	Fundamental Principle: Directly models physical process of photon detection
Data Requirements	Catalog magnitudes for reference stars (e.g., GAIA catalog contains billions of stars)	Full SED for reference stars (catalogs contain only a few hundred stars) and system's response curve
Complexity	Lower. Requires standard catalog queries and a linear least-square regression	Higher. Requires sourcing spectral data, defining system response curve and performing integration
Source of Error	Accuracy of linear fit, depends on the number and color range of reference stars	Accuracy of the cataloged SEDs and the assumed or measured system response curve

4.4.4. Spatial Correction Map and Visualization

In reality, instrumental and environmental effects can introduce spatial variations in the overall sensitivity. To address this, it is necessary to model and correct for these position-dependent errors. Sources of such variation include dust on the sensor or imperfections in the optical path, which create a subtle, fixed pattern of variations. A second source is environmental effects, such as clouds or atmospheric haze. This can cause parts of the image to be dimmer than others. These effects are not uniform across the sky and can be modeled as a gradient of extinction over the image.

To correct for the combined impact of these effects, the calculated correction values for all reference stars are used to generate a 2D spatial correction map. This map models the total instrumental and environmental signature for that specific exposure. This map allows for a final, photometric correction to be applied to any object in the image, ensuring a high measurement accuracy.

4.5. Implementation

This section provides an overview of the technical framework, libraries, and organizational structure used to implement the multi-stage pipeline.

4.5.1. Core Libraries

The entire project is developed in Python, leveraging its advanced ecosystem for scientific computing and machine learning. The used key libraries and frameworks are:

- PyTorch: The main deep learning framework used for training and running the Mask R-CNN model. Its extensive support for computer vision tasks makes it the ideal choice.
- OpenCV, Pillow, and NumPy: These libraries are used for creating and manipulating image arrays, drawing shapes, and performing mathematical operations required for photometric analysis.
- Optuna: This hyperparameter optimization framework is used to systematically search for the best training parameters for the AI model.
- Pandas: Used for managing and saving the ground truth data in the CSV format.
- Gin-config: A configuration framework used to manage all parameters for data generation, training, and evaluation. This ensures that all experiments are reproducible and easy to configure.

4.5.2. Project Structure

To ensure maintainability and clarity, the codebase is organized into a modular structure. Each module represents a specific part of the overall concept:

- data_generation/: Contains the scripts and logic for generating the synthetic dataset, including the images, masks, and ground truth files.
- star_trail_recognition/: Core module for the AI pipeline, containing logic for the input pipeline, model architecture, training, and evaluation.
- brightness_calibration/: Includes the final script that performs the photometric analysis, plate solving, and calibration.
- Output Directories (datasets/, runs/, tuning_runs): These output directories store the generated data, the trained models from various runs, and the results from the hyperparameter tuning.

4.5.3. Computational Resources

Training and optimizing a state-of-the-art DL model is computationally intensive. The training runs for this project were conducted on a high-performance workstation equipped with two NVIDIA A6000 GPUs. The training process heavily relies on matrix multiplications and convolutions. These operations are massively accelerated by the GPUs architecture via the CUDA platform. Furthermore, the A6000's substantial 48GB of VRAM per card allows the use of larger batch sizes during training , which leads to more stable gradients and can accelerate the training time of the model.

5. Test and Validation

This chapter aims to validate the functionality of each component of the developed pipeline and evaluate its performance and limitations based on the synthetic dataset. First, the Mask R-CNN is validated by training on synthetic data and fine-tuned on a real dataset, followed by the presentation of quantitative and qualitative results. Afterwards, the photometric pipeline is validated, beginning with the centroid estimation, continuing with the construction of the spatial correction maps, and concluding with the calibration validation across multiple observation sequences.

5.1. DL Model Performance Evaluation

The first part is to validate the instance segmentation model, as the accuracy of the following photometric measurements depends on the quality of the predicted masks. This section details the training process and presents an evaluation of the final model performance.

5.1.1. Training Process (Synthetic Dataset)

The model was trained using a transfer learning approach, starting with a Mask R-CNN architecture pre-trained on the ImageNet dataset. This uses the model's existing knowledge of general shapes and features, which is further described in subsection 4.2.3. The model is then fine-tuned on the synthetic dataset with 10,000 images. The key training parameters, optimized by the hyperparameter tuning, are shown in Table 5.1.

Table 5.1.: Hyperparameters used for training the Mask R-CNN model. The parameters for two training phases are listed. (1) Initial training on the synthetic dataset for convergence of the base weights and (2) following fine-tuning on the real dataset for domain adaptation.

Hyperparameter	Synthetic Dataset	Real Dataset
Optimizer	Adam	SGD
Learning Rate	1e-05	1e-03
Epochs	25	10
Batch Size	32	8
Model Layer	256	256
Scheduler	ReduceLROnPlateau	ReduceLROnPlateau

The learning process was monitored by tracking different components of the model's loss function. The loss curves, shown in Figure 5.1, provide detailed insights.

The analysis of the loss components shows a good training process:

- Total Loss: This is the sum of all other losses. The consistent decrease for training and validation indicates that the model is learning effectively and improving its overall performance with each epoch.

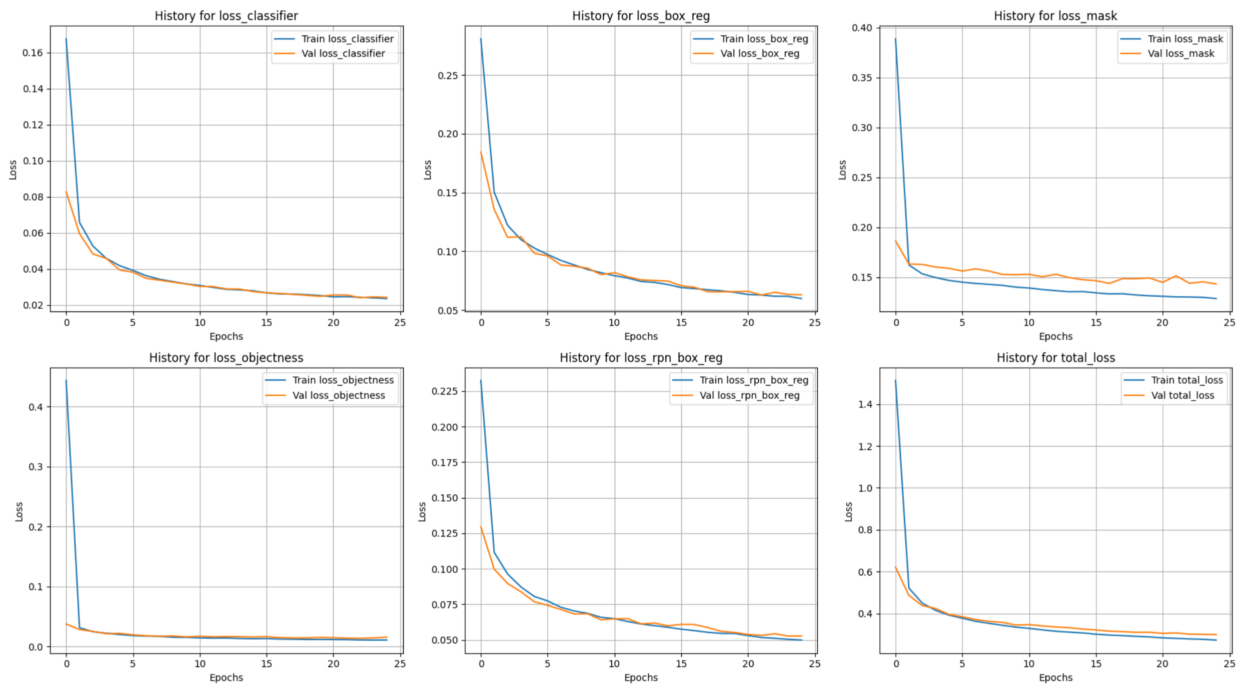


Figure 5.1.: Loss function during training of the Mask R-CNN model on synthetic data. The graphs show the training (blue) and validation loss (orange) over 25 epochs. The graphs represent the loss components of the Mask R-CNN model. The graph in the bottom right shows the total loss, the sum of all individual losses. The steadily decreasing line of both curves indicates a successful training process without severe overfitting.

- Mask Loss (loss_mask): This measures the pixel-level dissimilarity between the predicted mask and the ground truth. The steep drop shows that the model quickly learns the general shape of the streaks and target. The following convergence shows it refining the features of these blurry objects.
- Box Regression Loss (loss_box_reg and loss_rpn_box_reg): These losses penalize the model for inaccuracies in the location and size of the predicted bounding boxes. Their rapid convergence to low values shows that the model became proficient at localizing the objects early in the training process.
- Classifier Loss (loss_classifier): This measures the model's ability to correctly classify each detected object. The steady decline shows it has successfully learned to distinguish between the two categories.
- Objectness Loss (loss_objectness): This loss is associated with the RPN and measures the model's ability to distinguish between foreground objects and background. Like the other losses, it quickly drops, which confirms that the model learned to effectively identify RoIs in the noise images.

Across all six plots, the validation loss (orange line) closely tracks the training loss (blue line). This is the most important indicator of a successful training, as it shows that the model is not memorizing the training data. Instead, it learns a robust generalization of the features, which confirms that the model is not overfitting.

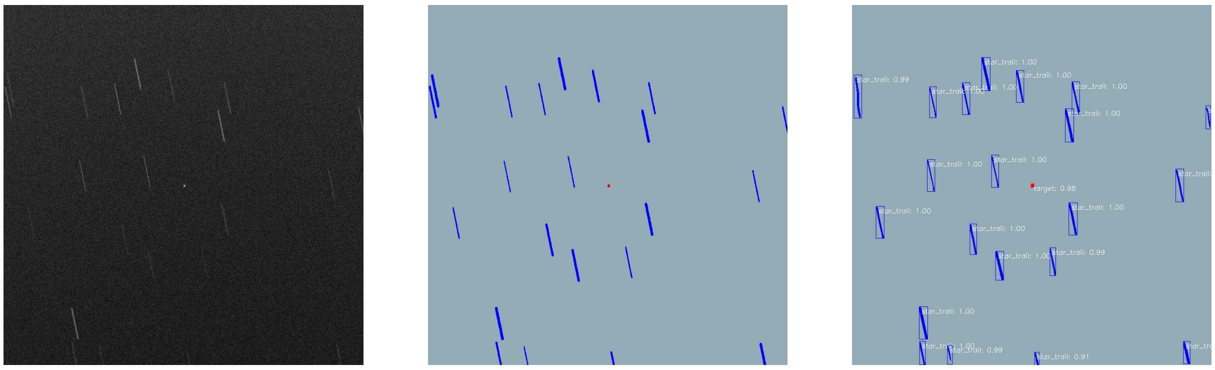
5.1.2. Evaluation Results (Synthetic Dataset)

The main metric for evaluating an instance segmentation model is the mAP, which is described in section 2.4. The final model, trained on a 10k image dataset, achieved the following performance on the synthetic data, shown in Table 5.2.

Table 5.2.: Evaluation results of the Mask R-CNN on synthetic data. Performance comparison of the object detection (bounding boxes) and segmentation, measured as mAP at different IoU thresholds.

IoU-thresholds	mAP	
	Bounding Box	Segmentation
mAP@IoU=.50:.95	0.814	0.128
mAP@IoU=.50	0.97	0.575
mAP@IoU=.75	0.925	0.001

These scores demonstrated the model's performance in locating the streaks and target, as shown by the bounding box mAP of 0.814. The score of 0.97 of the IoU=0.5 threshold confirms the model is reliable in finding the objects. The lower overall segmentation mAP of 0.128 reflects the difficulty of the tasks. There are two main reasons for this. The first is the metric sensitivity to small objects. Star streaks can be thin objects. The IoU metric is sensitive to small deviations. A shift of just one or two pixels in the predicted mask can cause a significant percentage drop in the overlap with the ground truth, penalizing the mAP score. The second reason is that the ground truth masks are generated from a perfect mathematical Gaussian, resulting in smooth, well-defined edges. However, the model's predictions are made on noisy image data where the faint and fuzzy edges of the streaks are difficult to distinguish. The segmentation mAP metric at the IoU=.75 threshold demands a pixel-perfect precision that is often physically not well-defined with noise.



(a) Synthetic Test Image (b) Real Synthetic Mask (c) Predicted Synthetic Mask

Figure 5.2.: Qualitative result of the Mask R-CNN model in object detection of synthetic images. 5.2a The synthetically generated image simulates an astronomical image. 5.2b The corresponding ground truth mask for the synthetic test image. 5.2c The model's prediction shows that most objects were correctly detected as star trails and target with high confidence (e.g., 1,00).

For this thesis, the primary metric is the model's ability to provide a mask that accurately captures the core signal of the streak. The mAP@IoU=.5 (0.575) and the qualitative results confirm that the model achieves this successfully.

Apart from the quantitative metric, a qualitative assessment is important. As shown in Figure 5.2, the model produces good segmentation masks for the streaks and the target object.

The visual shows that the model is effective at its primary task: locating and providing a well-fitting mask.

5.1.3. Training Process (Real Dataset)

The model, pre-trained on the 10k synthetic dataset, was further fine-tuned on the small real-world dataset of 400 images. The training process and the following evaluation show the challenges of this final step. The loss curves of the fine-tuning process are shown in Figure 5.3.

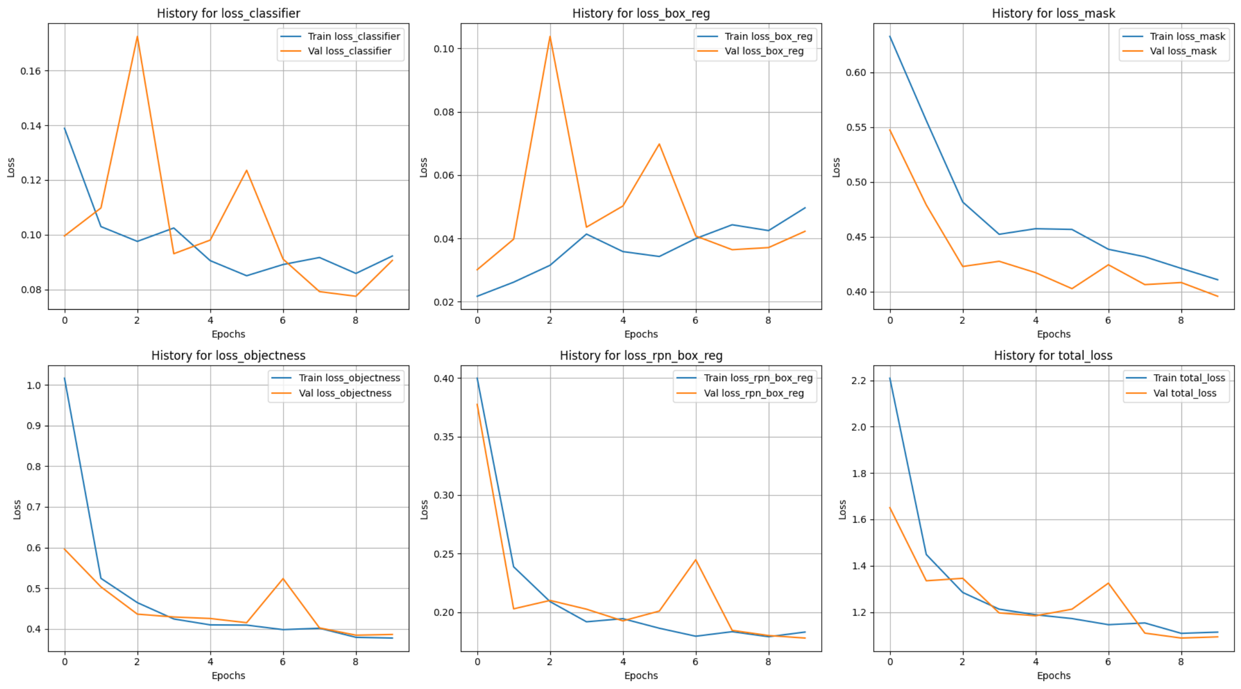


Figure 5.3.: Loss function during training of the Mask R-CNN model on real data. The graphs show the training (blue) and validation loss (orange) over 15 epochs. The graphs represent the loss components of the Mask R-CNN model. The graph in the bottom right shows the total loss, the sum of all individual losses.

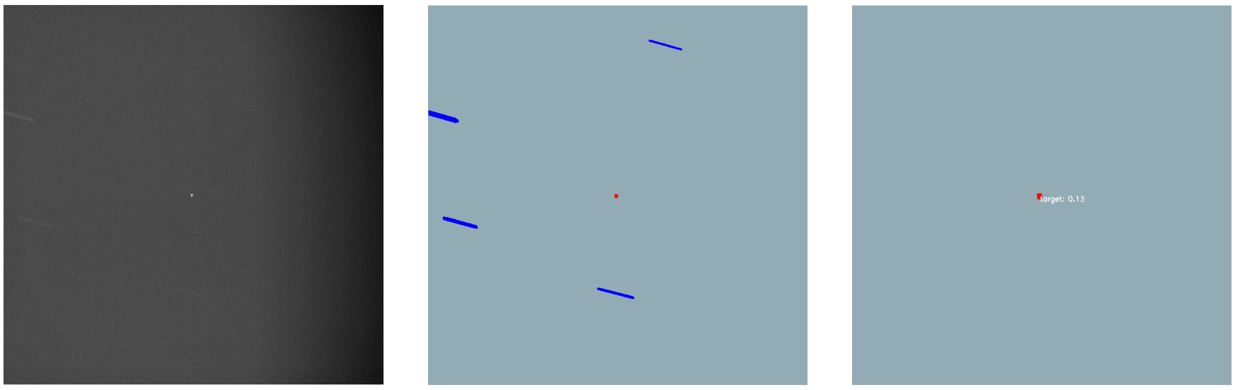
As the curves show, while the training loss (blue) decreases, the validation loss (orange) fails to converge. This indicates that the model is memorizing the limited features of the 400 real images but failed to learn a general representation. This is a consequence of the sim-to-real gap, the subtle but significant differences in noise profile, PSF shape, and camera artifacts between the synthetic and real data. The key training parameters, optimized by the hyperparameter tuning, are shown in Table 5.1.

5.1.4. Evaluation Results (Real Data)

The mAP scores, evaluated on a small test set of 10 images, confirm the model's inability to generalize, shown in Table 5.3.

Table 5.3.: Evaluation results of the Mask R-CNN on real data. Performance comparison of the object detection (bounding boxes) and segmentation, measured as mAP at different IoU thresholds.

	mAP	
IoU-thresholds	Bounding Box	Segmentation
mAP@IoU=.50:.95	0.088	0.045
mAP@IoU=.50	0.197	0.121
mAP@IoU=.75	0.042	0.000



(a) Real Image of the miniSLR® (b) Real Image Mask (c) Predicted Mask

Figure 5.4.: Qualitative result of the Mask R-CNN model in object detection of real images. 5.4a The real image shows several faint objects. 5.4b The corresponding ground truth mask for the real test image. 5.4c The model's prediction shows that none of the streaks were detected - only the target object, with a low confidence of 0.13.

Figure 5.4 shows that the model has learned to identify the target, even if only with low confidence of 13 %, but failed to learn the more complex features of the star streaks. This experiment confirms that while transfer learning from a synthetic dataset is a valid approach, its success is highly dependent on the synthetic data. For this model to achieve a higher performance on real data, the synthetic data would need to be further improved to more accurately replicate the characteristics of the real astronomical data. Furthermore, an increased training set of real images would help the model to learn more effectively and learn better representations of the features.

5.2. Photometric Pipeline Validation

With the performance of the model validated, the next step is to evaluate the photometric pipeline. Since the goal was to develop a complete end-to-end pipeline, in applying photometric transformation to real-world imagery, a significant challenge was encountered during the final transfer learning stage, which prevents the full validation of the photometric transformation on real images. As detailed in subsection 5.1.3, the model fine-tuned on a small real-world dataset struggles to generalize to these images due to the sim-to-real gap and insufficient, hand-labeled real-world data. Therefore, the model is not able to detect space objects, such as stars, that are required for the astrometric fit and the catalog cross-matching.

Without a set of instrumental magnitudes for the known stars in real images, it is not possible to perform the photometric transformation of the camera system. Overcoming this gap between simulation and reality requires further work and goes over the time frame of this thesis. Therefore, the photometric pipeline itself has been validated in a synthetic environment to show that the calibration concept works.

The validation is demonstrated by comparing a synthetic image of a clear sky and a cloudy sky. The clear sky represents perfect observation conditions, without major disturbances, while the cloudy image presents a disturbance factor in the form of a uniform cloud. This results in the objects appearing fainter and, therefore, reduces their brightness values, which changes the light curve.

5.2.1. Centroid Estimation

The first step for each frame is to detect and measure all objects. After the Mask R-CNN model detects all the star streaks and the target, the pipeline estimates the precise (x,y) coordinates of each object, based on the predicted mask. These centroid coordinates are the input for the following astrometric fitting. As shown in Figure 5.5, the model and centroid-finding algorithm are effective at locating the objects and building a segmentation mask (streaks - blue, target - orange), as well as their center point (red).

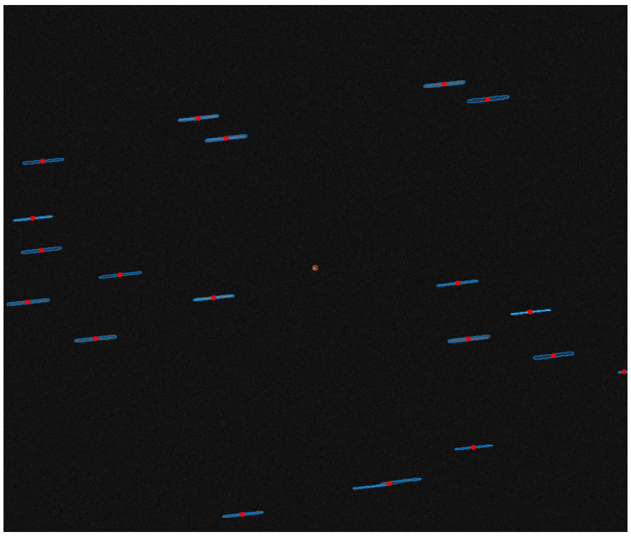


Figure 5.5.: Visualization of the centroid estimation for detected star streaks. After segmenting the objects (blue/orange), the center point (centroid, red dots) is calculated for each trail. This step reduces the object to a single (x,y) coordinate in the image, which is later used as the input for the astrometric fit.

Figure 5.5 shows that the centroid estimation is working across a sequence. To get a quantitative analysis of the measurement error, it is calculated across 15 synthetic test sequences. Two primary metrics were used to evaluate the performance: the Mean Absolute Percentage Error (MAPE) and the Median Absolute Percentage Error (MdAPE).

The MAPE is the average of the absolute percentage errors for N measurements, providing a measure of the overall error. Therefore, the MAPE is defined by Equation 5.1:

$$\mathrm{MAPE} = \frac{100}{N}\sum_{i=i}^{N}\left|\frac{x_{\mathrm{GT},i} - x_{\mathrm{Estimated},i}}{x_{\mathrm{GT},i}}\right| \tag{5.1}$$

The MdAPE is the median of the same errors. It is a more robust metric for characterizing the performance of the system, as it is less sensitive to individual outliers. Equation 5.2 defines the MdAPE as follows:

$$\mathrm{MdAPE} = \mathrm{median}\left(\left|\frac{x_{\mathrm{GT},i} - x_{\mathrm{Estimated},i}}{x_{\mathrm{GT},i}}\right|\right)\cdot 100 \tag{5.2}$$

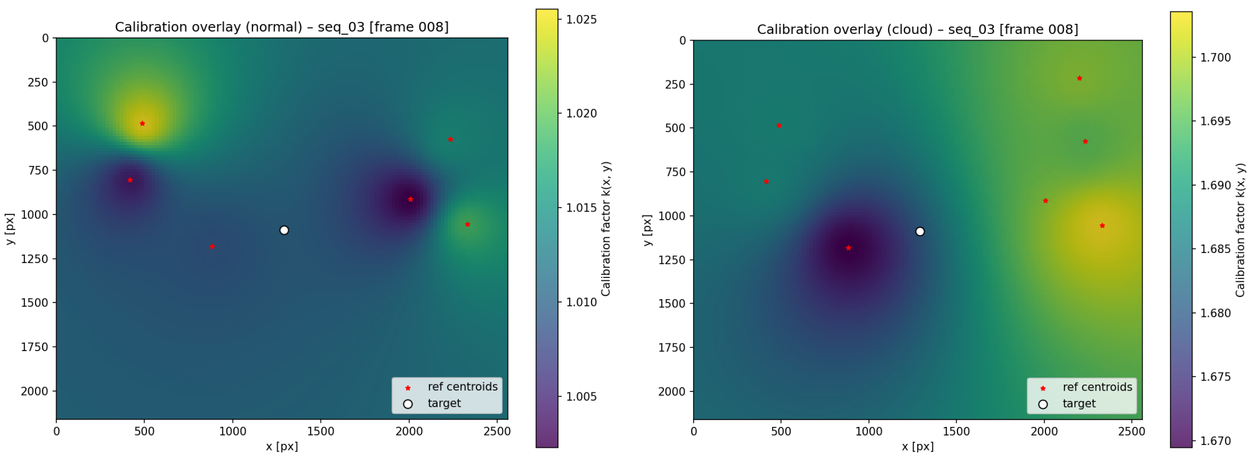
Table 5.4 presents the errors of the centroid validation. It shows that the MAPE and the MdAPE are below 0.5 %, therefore, the centroid estimation is working reliably and only produces a deviation of some pixels for the center coordinates. The MdAPEs for the streaks and target are lower than their corresponding MAPEs. This suggests that while the prediction is highly accurate, there are outliers in the estimation, possibly due to image artifacts or noise that can influence the prediction masks and therefore the centroid estimation.

Table 5.4.: Evaluation for centroid estimation. This table evaluates the centroid error for star streaks and the target object. It lists the MAPE and MdAPE for the centroid x- and y-coordinates. This table shows an overall precision, with a median error below 0.17 %.

Object	Axis	MAPE [%]	MdAPE [%]
Streaks	x	0.381	0.162
	y	0.445	0.162
Target	x	0.369	0.064
	y	0.441	0.098

5.2.2. Spatial Correction Map

For each frame, the measured instrumental magnitudes are transformed into standard magnitudes and compared against their true catalog magnitude (from a star catalog, like Gaia), providing the needed calibration coefficients that are then used to generate a spatial correction map. Due to the fact that the current model requires further refinement to achieve consistent object detection in real images, see subsection 5.1.3, the transformation is shown via a calculation example for the color-term transformation in section 4.4.2 and for the spectral integration calibration in section 4.4.2.

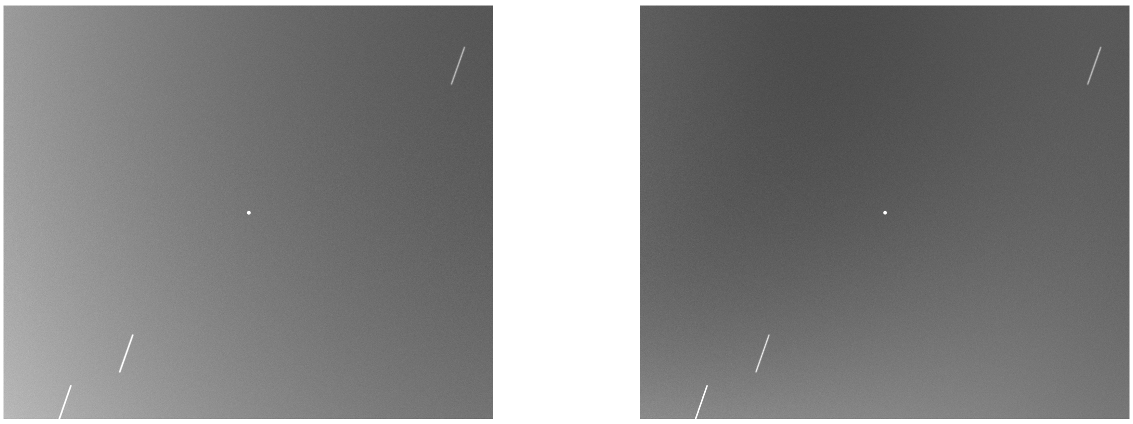


(a) Spatial Correction Map - Clear Sky

(b) Spatial Correction Map - Cloud

Figure 5.6.: Comparison of spatial correction maps. The map visualizes the location-dependent calibration factor $k(x,y)$, which is interpolated from the residual errors of the reference stars (red dots). (5.6a) Under normal conditions (clear sky), the correction map is relatively flat and shows only slight gradients. (5.6b) In the presence of clouds, the map shows stronger variations across the image.

While this provides a single calibration factor for each streak in the image, it does not account for the entire image. To map these spatial effects, the calculated residuals for each reference star are interpolated across the image using Inverse Distance Weighting to create a continuous spatial correction map. This visualizes the photometric calibration factors throughout the image. Figure 5.6 compares these maps for the clear and cloudy sky images. It should be noted that the colorbars are scaled differently for the two sky conditions, otherwise, the variations in the clear sky image would hardly be visible. The gradients of the clear sky are relatively low, which shows that the calibration does not correct much in normal conditions. When a disturbance, like a cloud, flies through the image, the gradients are much stronger. The comparison illustrates the need for location-dependent calibration to compensate for photometric errors caused by environmental conditions. Figure 5.7 shows example images between a clear sky and a cloudy image. In the cloudy image, it can be seen that the cloud is flying into the image from the top left and darkening the area.



(a) Clear Sky Image (b) Cloudy Image

Figure 5.7.: Comparison of images under clear sky and cloudy conditions. (5.7a) An image under clear sky, which has a more uniform brightness. (5.7b) An image under cloudy conditions, in which the darker region in the upper left corner visualizes the arrival of a cloud from the top left corner.

5.2.3. Light Curve Results and Performance

To show the calibration for a real-time event, two 15-frame sequences were generated: a clear sky sequence under clear and normal conditions, and a cloudy sequence where a semi-transparent cloud drifts across the frame. Figure A.1 shows the ground truth masks of one of the used sequences. The goal is to demonstrate that the calibration pipeline can correct for these environmental factors and recover the true, underlying brightness values of the target object.

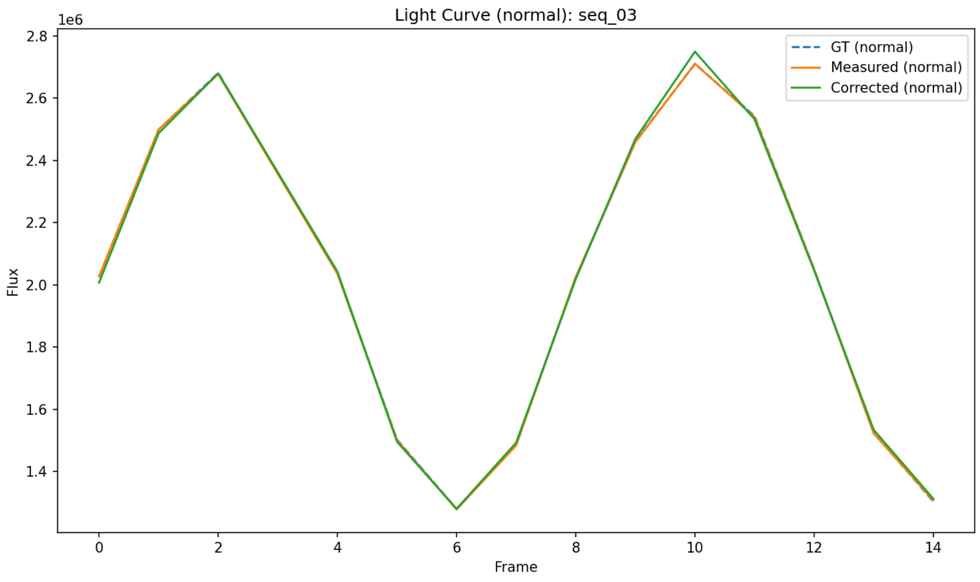


Figure 5.8.: Light curve of the target under clear sky conditions. Shown are the brightness curves (flux) over the frame number. The measured (orange) and the corrected (green) brightness follow the ground truth (blue) closely.

Under clear sky conditions, the instrumental measurements are already quite accurate. As shown in Figure 5.8, the measured flux (orange) closely follows the ground truth values (dashed blue line). After calibration, the corrected flux (green) aligns with the ground truth, but the measurements are sometimes overcorrected (second peak). This shows that the measured values are already good enough, and a calibration does not change them much.

Table 5.5.: Performance evaluation of the photometric pipeline under clear and cloudy conditions. The mean (MAPE) and median (MdAPE) absolute percentage errors before and after applying the correction are given. The results show that the pipeline reduces the measurement error in cloudy conditions, while the error in clear conditions is slightly affected. This demonstrated the method's effectiveness in non-ideal observation conditions.

	Measured error [%]		Corrected error [%]	
Sequence type	MAPE	MdAPE	MAPE	MdAPE
Clear Sky	2.61	0.30	2.63	0.77
Cloudy	39.53	39.92	2.34	1.03

Under cloudy conditions, the instrumental measurements are significantly biased. As the cloud passes, the brightness of all the objects is dimmer, this results in a higher correction factor. When the correction is applied, the effect of the cloud is removed. Figure 5.9 shows the corrected light curve (purple) close to the ground truth. It discards the environmental bias of the cloud (red) and recovers the true brightness values.

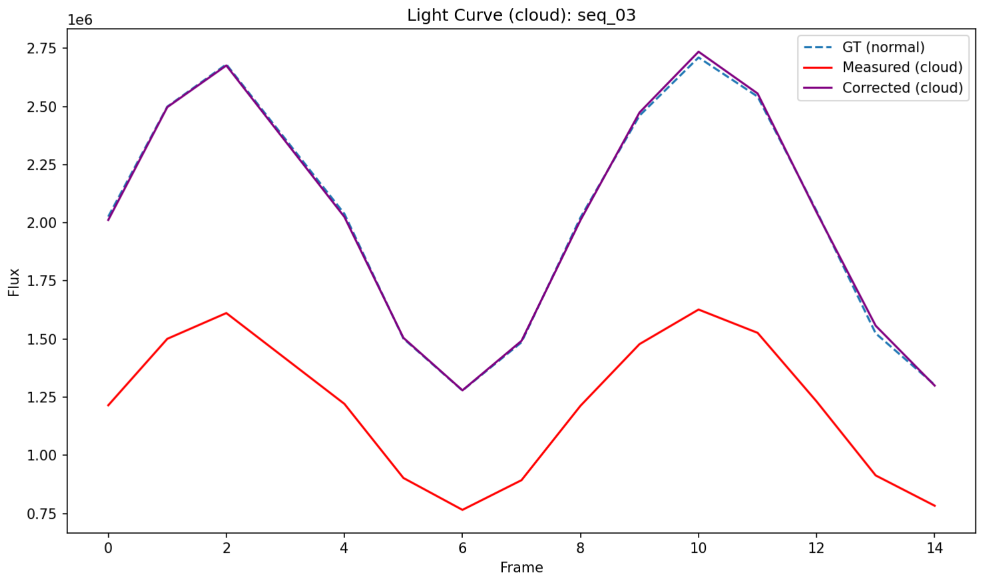


Figure 5.9.: Light curve of the target under cloudy conditions. Shown are the measured brightness (red) that is biased through the cloud in the image, while the corrected curve (purple) eliminates this bias and follows the ground truth (blue) closely.

Table 5.5 presents the results for the clear sky and cloudy sequences. Under clear sky conditions, the instrumental measurement is already precise, with a median error of 0.30 %. The calibration process yields a median error of 0.77 %. While the median error slightly increases, it remains below 1 %, showing that the calibration maintains a high level of precision. Under cloudy conditions, the pipeline is able to correct the raw measurement with a median error of nearly 39 % down to a median error of just 1.03 %. This shows that the calibration can bring the measurement in cloudy conditions nearly to the same level as in clear sky conditions, which makes the corrected data more scientifically useful.

Using Equation 2.7, the median flux error of 1.03 % can be converted into a magnitude error via Equation 5.3. This results in a median magnitude error of $-0.0109\,\mathrm{mag}$.

$$\Delta m = -2.5 \cdot \log_{10}(1+\delta) = -0.0109\,\mathrm{mag}, \quad \text{with } \delta = \frac{\Delta f}{f} = 0.01 \tag{5.3}$$

The final errors in the photometric measurements can be attributed to two main sources. On one hand, there are the imperfect model segmentation masks. Although the model provides highly accurate masks, these are not pixel-perfect. Minor discrepancies at the faint edges of the streaks can introduce minor errors into the profile extraction and the following model fitting. On the other hand, the curve fitting algorithm is a statistical process that finds the best possible fit to noisy data. The presence of random pixel noise means that even with a perfect mask, the fitted parameters will have a small statistical uncertainty, which contributes to the measurement error.

6. Discussion

This chapter evaluates the functionality, performance, and limitations of the DL model and the photometric pipeline. The first part is the evaluation of the Mask R-CNN model by its training and performance, for both datasets. Afterwards, the photometric pipeline is evaluated, starting with the centroid estimation, followed by the calibration process.

6.1. Discussion of the DL Model

The validation of the Mask R-CNN is split into two results. The first one is the training and performance validation of the synthetic dataset, in subsection 5.1.1 and 5.1.2. The model has a high quantitative and qualitative performance on the synthetic dataset. This shows that the concept for pre-training the model on synthetic data works, as it is able to detect and segment star streaks and the target object. It also justifies the use of an instance segmentation method to get precise pixel masks, on which the later calibration depends. However, the current weakness of this is the second training on real-world data, see subsection 2.1.4. The performance is much lower, and the model is not able to detect objects reliably, which is due to the sim-to-real gap. This is also the next step to work on. Bridging this gap is the next step for a completely working pipeline. This would require, among other things, expanding the real data set so that the model has more examples to learn from. The other step is to better align the synthetic data to the real scenarios and cover a wider range of scenarios, which would help the model to generalize better. Another approach would be to use another instance segmentation model, YOLO is a smaller and faster model, which could help to better adjust to the data distribution. However, this often comes with lower performance.

In the paper of Felt and Fletcher [12], they achieve a precision of 0.9 for their best model (Hybrid Task Cascade). This precision is based on a single threshold of eight pixels within a detected source that is classified as correct (True Positive). Therefore, this is a point-based metric [12]. In this thesis, the mAP is used, which is a more rigorous metric over more thresholds and considers precision and recall. A direct comparison between Felt and Fletcher's model is not possible. In terms of quality, it can be said that a mAP value in the range of 0.75 - 0.85 corresponds to a high level of precision. Thus, the performance of the model presented here is of the same range as the results in [12]. However, the differences in the data sets, evaluation metrics, and threshold definitions must be taken into account. Although the chosen approach of using a DL model for instance segmentation is state-of-the-art and superior to traditional methods such as the Source Extractor, its success depends on the quality of the training data, as demonstrated in this thesis.

The current optical path of the miniSLR® consists only of the telescope, deflecting mirrors, and the camera. From a hardware standpoint, it would be better to introduce corrective lenses into the system to focus the captured light. This would create sharper images, leading to smaller and brighter streaks, which would help to distinguish them from the background. Plans are already in place to install such lenses.

In conclusion, it can be said that the chosen method is valid and represented in the literature. Nevertheless, the validation showed that the practical success depends strongly on the task and enough high-quality data.

6.2. Discussion of the Photometric Pipeline

Validation of the photometric pipeline is solely based on a synthetic environment because the AI model was not able to detect objects in real-world data. However, the validation shows that the pipeline performed well in the synthetic environment and confirms the validity of the concept.

The centroid estimation achieves a MdAPE below 0.17 %. At first glance, this seems fine, but since astrometric fitting requires accurate data, even a deviation of a few pixels can cause the plate solving to fail. However, this only becomes clear once the model is usable for real-world data. In addition, the pipeline demonstrates its ability to correct environmental changes (e.g., synthetic clouds) and reduces the measurement error of nearly 39 % down to 1.03 %. This shows that the developed concept is validated and can compete with the current standard. The author B. Warner reports in [45] that differential photometry allows reaching a deviation of millimagnitudes (< 0.01 mag) [45]. The photometric pipeline achieves a median magnitude error of 0.0109 mag and is, therefore, viable and comparable with state-of-the-art methods.

Nevertheless, even if the photometric pipeline is successful, there are limitations. First, the missing validation of real-world data is the primary limitation. Although the pipeline works, the performance on real-world data is not validated. Disturbances, such as noise or camera effects, could have a higher impact in reality. Secondly, the concept describes two transformation methods, see subsection 4.4.2, to solve the problem that the optical path does not have any filters. However, the validation demonstrated the result of a successful calibration, but it cannot compare these methods, due to the failed detection in real-world images. A future implementation should validate the methods and compare them for their functionality.

In conclusion, the photometric pipeline shows that the theoretical concept works and is within the state-of-the-art methods. The synthetic validation shows that the implemented algorithm is correct and the calibration concept works as intended.

7. Conclusion and Future Work

This thesis aims to establish a photometric calibration process that minimizes environmental variables within the non-sidereal miniSLR® imagery of the German Aerospace Center. The project designed, implemented, and validated a complete multi-stage pipeline that uses the combination of Deep Learning and scientific algorithms to achieve this goal. The project achieved several significant results that demonstrate the feasibility of a data-driven approach for star streak detection. A core finding is that the AI instance segmentation model proved effective on synthetic data. The Mask R-CNN model, trained on a larger, synthetic dataset, demonstrated the ability to locate stellar objects with a bounding box mAP of 0.814. This confirms that a deep learning model can become an expert at identifying complex, faint features in noisy astronomical images. Furthermore, the validation of the measurement pipeline was a success. By combining the AI's predicted masks with a PSF-fitting algorithm, the system is able to measure the instrumental flux of objects with a median error of 0.77 % under clear sky conditions. Significantly, the system demonstrated its ability to correct for environmental factors. In simulated light curve sequences, the pipeline reduces a measurement error with a median of nearly 39 %, caused by a passing cloud, down to 1.03 %. This represents a performance improvement of almost 39 times and proves that the frame-by-frame differential photometry approach is effective at removing environmental and instrumental variations to recover the actual astronomical light signal.

While the project succeeded in its primary goal on the synthetic dataset, it also highlighted a crucial challenge that represents a significant limitation of the current work: the sim-to-real gap. As detailed in the transfer learning validation, in subsection 5.1.3, the model pre-trained on synthetic data struggled to generalize to a smaller set of real-world images, achieving a mAP of 0.088. The fine-tuned model failed to reliably detect star streaks necessary for a complete calibration, indicating that differences between the synthetic and real data were significant enough to affect performance. Because the model could not detect a sufficient number of reference stars in the images, the goal of performing a photometric transformation on real data could not be achieved within the scope of this thesis.

The findings and limitations of this project open up several directions for future research. The next step is to bridge the sim-to-real gap by improving the accuracy of the synthetic data generator, for instance, by incorporating more realistic noise and PSF models and increasing the real-world dataset. Once this gap is closed, the final step would be to integrate the entire pipeline with an astrometric solver and an online star catalog. This would create a complete system to take a raw sequence from the miniSLR® and produce a calibrated light curve. In conclusion, this thesis has successfully laid the theoretical and practical groundwork for a frame-by-frame real-time approach to photometric calibration. It has been proven that each component of the proposed system works with a high degree of accuracy in a controlled environment. It has identified its primary challenges, which must be overcome to transition this concept into a fully operational tool.

Bibliography

[1] E. S. Agency, *Gaia edr3 passband*, `https://www.cosmos.esa.int/web/gaia/edr3-passbands`, Accessed: 2025-10-05, 2020.

[2] E. S. Agency, *Clearspace-1: Active debris removal mission*, `https://www.esa.int/Space_Safety/ClearSpace-1`, Accessed: 2025-07-22, 2025. [Online]. Available: `https://www.esa.int/Space_Safety/ClearSpace-1`.

[3] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Accessed: 2025-10-05, 2019.

[4] J. Allworth, L. Windrim, J. Wardman, D. Kucharski, J. Bennett, and M. Bryson, "Development of a high fidelity simulator for generalised photometric based space object classification using machine learning," in *Proceedings of 70th International Astronautical Congress (IAC)*, Appeared as a pre-print in arXiv:2004.12270v1 [physics.space-ph], Washington D.C., 2019.

[5] Bertin, E. and Arnouts, S., "Sextractor: Software for source extraction," *Astron. Astrophys. Suppl. Ser.*, vol. 117, no. 2, pp. 393–404, 1996. DOI: `10.1051/aas:1996164`. [Online]. Available: `https://doi.org/10.1051/aas:1996164`.

[6] N. Cardiel et al., "Rgb photometric calibration of 15 million gaia stars," *Monthly Notices of the Royal Astronomical Society*, vol. 507, no. 1, pp. 318–329, Jul. 2021, ISSN: 1365-2966. DOI: `10.1093/mnras/stab2124`. [Online]. Available: `http://dx.doi.org/10.1093/mnras/stab2124`.

[7] F. R. Chromey, *To Measure the Sky: An Introduction to Observational Astronomy*, 2nd ed. Cambridge University Press, 2016.

[8] N. G. Corporation, *Spacelogistics: Satellite life-extension and on-orbit servicing services*, `https://www.northropgrumman.com/what-we-do/space/space-logistics-services`, Accessed on July 22, 2025, 2025. [Online]. Available: `https://www.northropgrumman.com/what-we-do/space/space-logistics-services`.

[9] European Space Agency, *Welcome to tetra3! — tetra3 0.1 documentation*, Legacy documentation; revision 19fac3f1, 2019. [Online]. Available: `https://tetra3.readthedocs.io/en/legacy/index.html`.

[10] European Space Agency, "Esa space environment report 2025," European Space Agency (ESA), Tech. Rep., 2025, Accessed on July 18, 2025. [Online]. Available: `https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf`.

[11] European Union Agency for the Space Programme (EUSPA), *Eu space programme – space situational awareness (ssa)*, `https://www.euspa.europa.eu/eu-space-programme/ssa`, Accessed: Jul. 18, 2025, 2025.

[12] V. Felt and J. Fletcher, "Seeing stars: Learned star localization for narrow-field astrometry," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2024, pp. 8297–8305.

[13] P. Fergus and C. Chalmers, *Applied Deep Learning: Tools, Techniques, and Implementation*. Springer, 2022. DOI: 10.1007/978-3-031-04420-5. [Online]. Available: https://link.springer.com/book/10.1007/978-3-031-04420-5.

[14] C. J. Fluke and C. Jacobs, "Surveying the reach and maturity of machine learning and artificial intelligence in astronomy," *WIREs Data Mining and Knowledge Discovery*, vol. 10, 2 2019. DOI: 10.1002/widm.1349.

[15] J. Goodeve, "Light streak photometry and streaktools," *The Astronomical Journal*, vol. 169, p. 151, Feb. 2025. DOI: 10.3847/1538-3881/ada950.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[17] D. Hampf, F. Niebler, T. Meyer, and W. Riede, "The minislr: A low-budget, high-performance satellite laser ranging ground station," *Journal of Geodesy*, vol. 98, no. 1, p. 8, 2024, ISSN: 1432-1394. DOI: 10.1007/s00190-023-01814-1. [Online]. Available: https://doi.org/10.1007/s00190-023-01814-1.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

[19] Z. He, B. Qiu, A.-L. Luo, J. Shi, X. Kong, and X. Jiang, "Deep learning applications based on sdss photometric data: Detection and classification of sources," *Monthly Notices of the Royal Astronomical Society*, vol. 508, no. 2, pp. 2039–2052, Aug. 2021, ISSN: 0035-8711. DOI: 10.1093/mnras/stab2243. eprint: https://academic.oup.com/mnras/article-pdf/508/2/2039/40535770/stab2243.pdf. [Online]. Available: https://doi.org/10.1093/mnras/stab2243.

[20] S. B. Howell, *Handbook of CCD Astronomy* (Cambridge Observing Handbooks for Research Astronomers), 2nd ed. Cambridge University Press, 2006.

[21] A. J. Inc., *Adras-j: Active debris removal by astroscale-japan*, https://astroscale.com/missions/adras-j/, Accessed on July 22, 2025, 2024. [Online]. Available: https://astroscale.com/missions/adras-j/.

[22] C. Jeffries and R. Acuña, "Detection of streaks in astronomical images using machine learning," *Journal of Artificial Intelligence and Technology*, vol. 4, no. 1, pp. 1–8, Aug. 2023. DOI: 10.37965/jait.2023.0413. [Online]. Available: https://ojs.istp-press.com/jait/article/view/413.

[23] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9396–9405. DOI: 10.1109/CVPR.2019.00963.

[24] T. Y. Lin et al., *Microsoft coco: Common objects in context*, https://cocodataset.org, Accessed 2025-09-22, 2014.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944. DOI: 10.1109/CVPR.2017.106.

[26] T.-Y. Lin et al., *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1405.0312.

[27] E. A. Magnier et al., “Pan-starrs photometric and astrometric calibration,” *The Astrophysical Journal Supplement Series*, vol. 251, no. 1, p. 6, Oct. 2020. DOI: 10.3847/1538-4365/abb82a. [Online]. Available: https://dx.doi.org/10.3847/1538-4365/abb82a.

[28] S. K. Meher and G. Panda, “Deep learning in astronomy: A tutorial perspective,” *The European Physical Journal Special Topics*, 2021. DOI: 10.1140/epjs/s11734-021-00207-9.

[29] T. Meyer et al., “High-fidelity light curve simulation and validation using empirical data,” in *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, AMOS, 2024.

[30] MILES Team. “Miles stellar library: The catalogue.” Accessed: 2025-10-06, Instituto de Astrofísica de Canarias (IAC). [Online]. Available: https://research.iac.es/proyecto/miles/pages/stellar-libraries/the-catalogue.php.

[31] F. Niebler et al., “Compact ground station for satellite laser ranging and identification,” in *73rd International Astronautical Congress (IAC)*, 2022.

[32] F. Niebler et al., “The minislr handbook,” 2025.

[33] S. I. Nikolenko, *Synthetic Data for Deep Learning*. Springer, 2021. DOI: 10.1007/978-3-030-75178-4.

[34] M. Nussbaum, *Simulation and measurement of multispectral space debris light curves*, Internal document, unpublished, 2021.

[35] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. DOI: 10.1109/TKDE.2009.191.

[36] Pöntinen, M. et al., “Euclid: Identification of asteroid streaks in simulated images using deep learning,” *A and A*, vol. 679, A135, 2023. DOI: 10.1051/0004-6361/202347551. [Online]. Available: https://doi.org/10.1051/0004-6361/202347551.

[37] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017. DOI: 10.1109/TPAMI.2016.2577031.

[38] W. Romanishin, *An Introduction to Astronomical Photometry Using CCDs*. CreateSpace Independent Publishing Platform, 2014, ISBN: 9781500772116. [Online]. Available: https://books.google.de/books?id=OnbMoQEACAAJ.

[39] Rubin Observatory, *Legacy Survey of Space and Time (LSST)*, https://rubinobservatory.org/explore/how-rubin-works/lsst, Zugriff am 18-August 2025, 2025.

[40] S. Russell and P. Norvig, *Künstliche Intelligenz Ein moderner Ansatz*. Pearson Deutschland, 2012, p. 1312, ISBN: 9783868940985. [Online]. Available: https://elibrary.pearson.de/book/99.150005/9783863265045.

[41] C. SA, *Clearspace-1: Active debris removal mission*, `https://clearspace.today/missions/clearspace-1`, Accessed: 2025-07-22, 2025. [Online]. Available: `https://clearspace.today/missions/clearspace-1`.

[42] K. Sharma et al., "Astreaks: Astrometry of neos with trailed background stars," *Monthly Notices of the Royal Astronomical Society*, vol. 524, no. 2, pp. 2651–2660, Jul. 2023, ISSN: 0035-8711. DOI: `10.1093/mnras/stad1989`. eprint: `https://academic.oup.com/mnras/article-pdf/524/2/2651/51652656/stad1989.pdf`. [Online]. Available: `https://doi.org/10.1093/mnras/stad1989`.

[43] R. Sun, P. Jia, Y. Sun, Z. Yang, Q. Liu, and H. Wei, "Pnet—a deep learning based photometry and astrometry bayesian framework," *The Astronomical Journal*, vol. 166, no. 6, p. 235, Nov. 2023. DOI: `10.3847/1538-3881/ad01b5`. [Online]. Available: `https://dx.doi.org/10.3847/1538-3881/ad01b5`.

[44] Ultralytics, *Yolov8: State-of-the-art computer vision model*, Accessed: 2025-10-06, 2025. [Online]. Available: `https://yolov8.com/`.

[45] B. D. Warner, *A Practical Guide to Lightcurve Photometry and Analysis* (The Patrick Moore Practical Astronomy Series). Springer, 2016, Accessed: 2025-07-23, ISBN: 978-3-319-32750-1. DOI: `10.1007/978-3-319-32750-1`. [Online]. Available: `https://link.springer.com/book/10.1007/978-3-319-32750-1`.

[46] D. Xu and Y. Zhu, "Surveying image segmentation approaches in astronomy," *Astronomy and Computing*, vol. 48, p. 100 838, 2024, ISSN: 2213-1337. DOI: `https://doi.org/10.1016/j.ascom.2024.100838`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S2213133724000532`.

A. Appendix

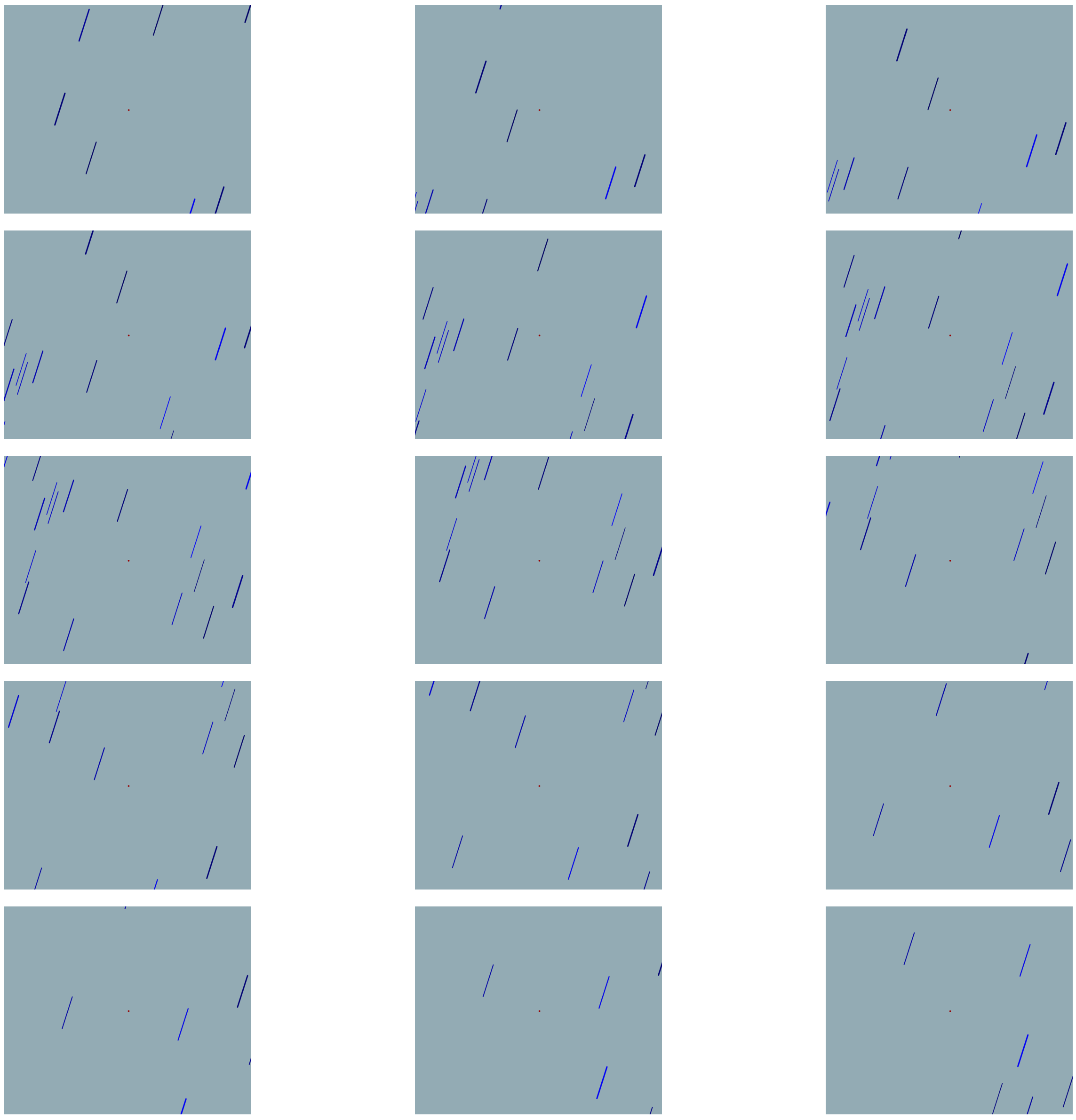


Figure A.1.: Visualization of an observation sequence using the corresponding ground truth masks. The 15 images show a non-sidereal observation sequence, read from left to right and from top to bottom. Instead of using the raw image, the masks are shown to visually highlight the position of the target object (red) and the background star trails (blue lines).

A.1. Zusammenfassung

Das Ziel dieser Arbeit ist es, einen photometrischen Kalibrierungsprozess zu etablieren, der Umwelteinflüsse innerhalb der nicht-siderischen miniSLR® Bilder des DLR minimiert. Im Rahmen des Projektes wurde eine vollständige mehrstufige Pipeline entworfen, implementiert und validiert, die eine Kombination aus DL und wissenschaftlichen Algorithmen nutzt, um dieses Ziel zu erreichen. Das Projekt erzielte mehrere bedeutende Ergebnisse, die die Machbarkeit eines daten-gesteuerten Ansatzes zur Erkennung von Sternspuren belegen. Eine zentrale Erkenntnis ist, dass sich das Instanzsegmentierungsmodell bei synthetischen Daten als wirksam erwiesen hat. Das Mask R-CNN Modell, das auf einen großen synthetischen Datensatz trainiert wurde, zeigt die Fähigkeit, Objekte mit einem Begrenzungsrahmen mAP von 0,814 zu lokalisieren. Dies bestätigt, dass ein DL Modell zu einem Experten für die Identifizierung komplexer, schwacher Merkmale in verrauschten astronomischen Bildern werden kann. Darüber hinaus war die Validierung der Pipeline, auf synthetische Daten, ein Erfolg. Durch die Kombination der von der KI vorhergesagten Masken mit einem PSF Algorithmus ist das System in der Lage, den instrumentellen Fluss von Objekten mit einem mittleren Fehler von 0.77 % unter klaren Himmelsbedingungen zu messen. Am bedeutendsten ist, dass das System seine Fähigkeit zur Korrektur von Umweltfaktoren unter Beweis gestellt hat. In simulierten Lichtkurvensequenzen reduzierte die Pipeline einen durch eine vorbeiziehende Wolke verursachten Messfehler mit einem Median von fast 39 % auf nur noch 1.03 %. Dies entspricht einer Leistungssteigerung um fast das 39-fache und beweist, dass der Ansatz der Frame-für-Frame Differenzphotometrie wirksam ist um Umgebung- und Instrumentalschwankungen zu beseitigen und das tatsächliche astronomische Lichtsignal wiederherzustellen.

Das Projekt hat zwar sein primäres Ziel in Bezug auf den synthetischen Datensatz erreicht, aber es hat auch eine entscheidende Herausforderung aufgezeigt, die eine erhebliche Herausforderung aufzeigte: die Diskrepanz zwischen Simulation und Realität. Wie in der Validierung des Transferlernens unter subsection 5.1.3 ausführlich beschrieben, hatte das auf synthetische Daten vortrainierte Model Schwierigkeiten, auf einen kleineren Satz realer Bilder zu generalisieren und erreichte einen mAP von 0,088. Das feinabgestimmte Modell konnte die für eine vollständige Kalibrierung erforderlichen Sternspuren nicht zuverlässig erkennen, was darauf hindeutet, dass die Unterschiede zwischen den synthetischen und den realen Daten so groß waren, dass sie die Leistung beeinträchtigten. Da das Modell nicht genügend Referenzsterne in den Bildern erkennen konnte, konnte das Ziel einer photometrischen Transformation auf reale Daten im Rahmen dieser Arbeit nicht erreicht werden.

Die Ergebnisse und Grenzen dieses Projekts eröffnet mehrere Richtungen für die zukünftige Forschung. Der nächste Schritt besteht darin, die Lücke zwischen Simulation und Realität zu schließen, indem die Genauigkeit des Generators für synthetische Daten verbessert wird, beispielsweise durch die Einbeziehung realistischerer Rausch- und PSF-Modelle. Sobald diese Lücke geschlossen ist, wäre der letzte Schritt die Integration der gesamten Pipeline mit einem astronomischen Solver und einem Online-Sternenkatalog. Dadurch würde ein komplettes System entstehen, die die Rohsequenzen aus dem miniSLR® aufnimmt und eine kalibrierte Lichtkurve erzeugt. Zusammenfassend lässt sich sagen, dass die Arbeit erfolgreich die theoretischen und praktischen Grundlagen für eine Frame-für-Frame Echtzeitkalibrierung gelegt hat. Es wurde nachgewiesen, da jede Komponente des vorgeschlagen System in einer kontrollierten, auf synthetischen Daten basierten, Umgebung mit hoher Genauigkeit arbeite. Die wichtigsten Herausforderungen, die bewältigt werden müssen, um dieses Konzept in ein funktionsfähiges Werkzeug umzusetzen, wurden identifiziert.