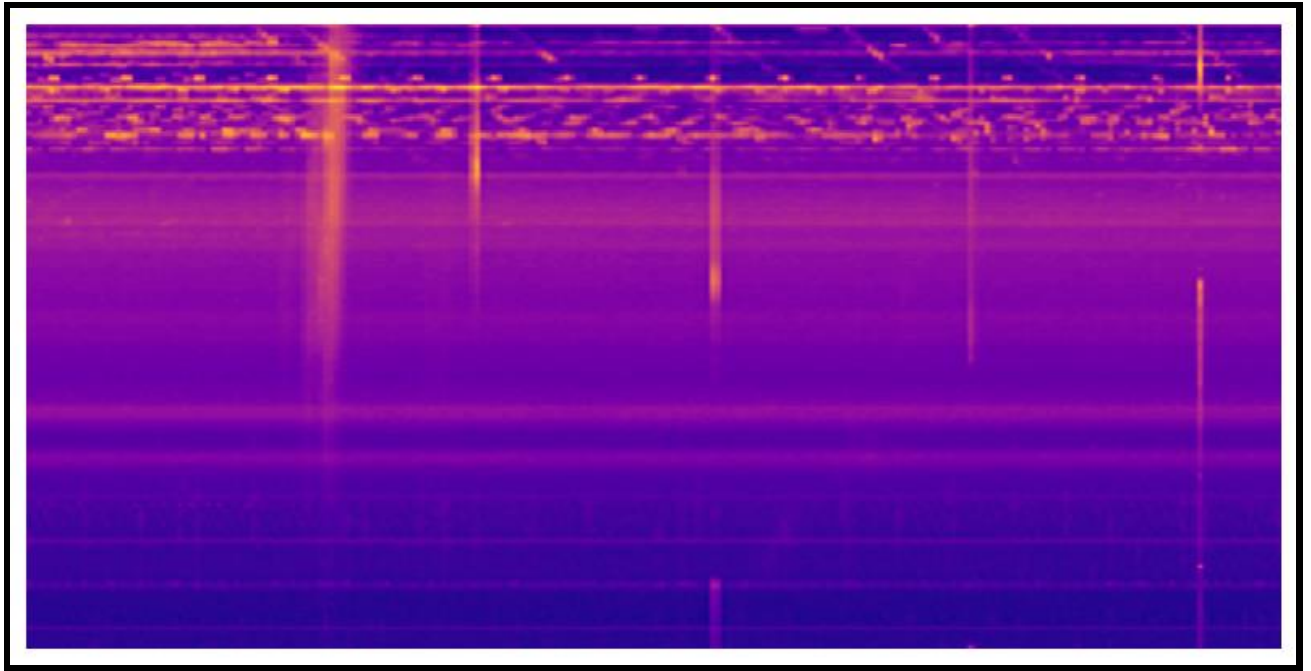


Klassifikation von Sonneneruptionen mittels Machine Learning und e-Callisto Netzwerk



Projektarbeit 5. Semester HS23

Auftraggeberschaft Fachhochschule Nordwestschweiz, Institut Data Science

Autoren Schürmann, Patrick
Torres Gamez, Gabriel

Betreuer: Csillaghy, André
Timmel, Vincenzo

Ort, Datum Windisch, 19. Januar 2024

Eigenständigkeitserklärung

Wir erklären hiermit,

- dass wir sämtliche nicht von uns selbst stammenden Textstellen gemäss gängigen wissenschaftlichen Zitierregeln korrekt zitiert und die verwendeten Quellen gut sichtbar erwähnt habe;
- dass wir in einem Verzeichnis alle verwendeten Hilfsmittel (z.B. KI-Assistenzsysteme wie Chatbots [z.B. ChatGPT], Übersetzungs-, Paraphrasier-Tools) oder Programmierapplikationen [z.B. GitHub Copilot] deklariert und ihre Verwendung bei den entsprechenden Textstellen angegeben haben;
- dass wir sämtliche immateriellen Rechte an von uns allfällig verwendeten Materialien wie Bildern oder Grafiken erworben haben oder dass diese Materialien von uns selbst erstellt wurden;
- dass das Thema, die Arbeit oder Teile davon nicht bei einem Leistungsnachweis eines anderen Moduls verwendet wurden, sofern dies nicht ausdrücklich mit der Dozentin oder dem Dozenten im Voraus vereinbart wurde und in der Arbeit ausgewiesen wird;
- dass wir uns bewusst sind, dass unsere Arbeit auf Plagiate und auf Drittautorschaft menschlichen oder technischen Ursprungs (künstliche Intelligenz) überprüft werden kann;
- dass wir uns bewusst sind, dass die Hochschule für Technik FHNW einen Verstoss gegen diese Eigenständigkeitserklärung bzw. die ihr zugrundeliegenden Pflichten der Studien- und Prüfungsordnung der Hochschule für Technik FHNW verfolgt. Zu den Pflichten gehört insbesondere die Wahrung von Urheberrechten und das Unterlassen von Plagiaten. Ein diesbezüglicher Verstoss wird mit der Note 1 bewertet und es können zusätzlich disziplinarische Folgen (Verweis/Ausschluss aus dem Studiengang) resultieren.

Windisch, 19. Januar 2024

Unterschriften

Abstract

Das Projekt "Klassifikation von Sonneneruptionen mittels Machine Learning und dem e-Callisto Netzwerk" zielt darauf ab, Sonneneruptionen automatisch zu klassifizieren, um das manuelle Erfassen der Labels zu ersetzen. Es wurden Daten des e-Callisto-Netzwerks verwendet, um verschiedene Deep-Learning-Modelle wie ResNet und eigens entwickelte CNNs für binäre Klassifikationen zu trainieren. Die besten Ergebnisse erzielte ResNet50 mit einem Recall von 0.88 und einer Precision von 0.69. Allerdings zeigte sich, dass die Modelle die Validierungsdaten nicht gut lernen konnten. Es wird empfohlen, die Untersuchung mit 15-Minuten-Daten fortzusetzen, insbesondere das Verhalten im Training der ersten Epoche zu untersuchen. Kann das Problem beim Training nicht verstanden und korrigiert werden, können alternative Methoden erkundet werden. Zudem wurde ein Datensatz mit gelabelten 15-Minuten-Bildern veröffentlicht, mit dem zukünftige Modelle trainiert werden können. Zusätzlich wurde eine API für die Sonneneruptionserkennung entwickelt, welche die Sonnenspektren und die von den Modellen gegebene Gewissheit visualisiert, um beim manuellen Labeling zu unterstützen.

Inhaltsverzeichnis

Eigenständigkeitserklärung	II
Abstract	III
Inhaltsverzeichnis.....	IV
1 Einleitung	1
1.1 Überblick über das Projekt.....	1
1.2 Bedeutung der Klassifikation von Sonneneruptionen.....	2
1.3 Unser Projekt.....	2
2 Theoretischer Teil	3
2.1 Data Science	3
2.2 Wissenschaftlicher Kontext von Sonneneruptionen	3
2.3 Herausforderungen in der Klassifikation	5
2.4 Deep Learning	5
2.5 ResNet-18	6
2.6 Weitere Deep Learning Modelle	7
2.7 Custom CNN	8
2.8 Custom CNN 2	8
2.9 Metriken.....	9
3 Praktischer Teil	10
3.1 Datenerfassung und -management	10
3.2 Datenanalyse und Vorverarbeitung	10
3.2.1 Labels.....	10
3.2.2 Spektren	12
3.3 Entwicklung des Modells.....	14
3.4 Resultate	15
3.4.1 ResNet50	15
3.4.2 ResNet18	15

3.4.3	CustomCNN	16
3.4.4	CustomCNN2	16
3.5	Evaluierung der Modelle	18
3.6	Wichtige Erkenntnisse	19
3.7	Sunburst Detection API	20
4	Fazit	21
4.1	Empfehlung für weitere Schritte	22
	Quellenverzeichnis	23
	Abbildungsverzeichnis	25
	Tabellenverzeichnis	26
	Hilfsmittelverzeichnis	27

1 Einleitung

1.1 Überblick über das Projekt

Das Projekt wurde unter dem Namen “Klassifikation von Sonneneruptionen mittels Machine Learning und e-Callisto Netzwerk” für das Herbstsemester 2023 ausgeschrieben. Es wurde für Studenten an der FHNW im Studiengang Data Science als Projekt 5 erstellt. Es dient zur Vorbereitung auf die Bachelorarbeit und wird in der Regel im zweitletzten Semester und als Zweierarbeit bearbeitet.

CALLISTO ist ein Spektrogramm, das Partikel von Sonneneruptionen, die auf die Erde treffen, messen kann und überwacht dabei standardmässig einen Frequenzbereich von 45 bis 870 MHz (*e-Callisto*, 2023). CALLISTO steht für “Compound Astronomical Low frequency Low cost Instrument for Spectroscopy and Transportable Observatory” und sämtliche Empfänger bilden zusammen das e-Callisto Netzwerk. Die Empfänger sind global verteilt und per 14. November 2023 senden 76 Geräte regelmässig Daten an e-Callisto. Abgespeichert werden die Daten als Zeitreihe, für die Auswertung werden sie aber häufig als Bild dargestellt. Abbildung 1 zeigt einer der verwendeten Antennen in Landschlacht (Monstein, 2023).



Abbildung 1: CALLISTO Long Wavelength Antenna in Landschlacht

Das Projekt wurde laut Projektbeschreibung ausgeschrieben, um eine Klassifikation von Sonneneruptionen mittels maschinellen Lernens zu erstellen. Die Datenbasis für das Projekt stammt vom E-Callisto-Netzwerk. Dieses sammelt global Daten zur Sonnenaktivität mittels Radiospektrometern. Wir können dabei auf eine Zeitreihendatenbank zugreifen, in der die Daten in Viertelsekundenintervallen zur Verfügung stehen. Die Labels für die Sonneneruptionen wurden manuell gelabelt und befinden sich in einzelnen Text-Dokumenten. Das Hauptziel ist es, die Sonneneruptionen automatisch zu klassifizieren. Falls ein Modell zuverlässige Klassifikationen generiert, könnte es die manuelle Erstellung von Labels ablösen.

1.2 Bedeutung der Klassifikation von Sonneneruptionen

Während Sonneneruptionen Menschen nicht direkt bedrohen, stellen sie für unterschiedliche Technologien ein Risiko dar. Mögliche Folgen sind Störungen und Zerstörungen vom Flugverkehr, Stromnetz, Handynetz, Satelliten, GPS und Astronauten, die sich ausserhalb des schützenden Erdmagnetfeldes befinden (*FAQ Sonnenstürme und Sonnenaktivität*, 2013b). Eine Sonneneruption hat 2022 40 von 49 neu gestarteten Starlink-Satelliten zerstört, was SpaceX bis zu USD 100 Mio gekostet haben könnte (Andrews, 2022).

Vor diesem Hintergrund ist es für unterschiedliche Branchen wichtig, dass Sonneneruptionen und deren Auftreffen auf die Erde vorhergesagt werden können. Gemäss Stefan Kraft sollen Vorlaufzeiten für Warnungen von mehreren Tagen möglich sein (*Weltraumwetter*, o. J.). Dies würde es ermöglichen, Vorsichtsmassnahmen zu treffen und Auswirkungen reduzieren zu können. So könnten Satelliten in einen sicheren Zustand versetzt und Energiesysteme auf die Störung vorbereitet werden.

1.3 Unser Projekt

Wir haben für diese Arbeit einen Datensatz mittels 15 Minuten Bilder erstellt, mit dem wir eine Selektion an Modellen trainieren. Diese Modelle basieren auf bekannten, sowie auch selbsterstellten CNN Architekturen. Diese Modelle kann man mittels der API ansteuern. Die API lädt mittels `ecallisto_ng` einen Zeitraum eines Australia-ASSA Instrumentes herunter und berechnet mittels des ausgewählten Modells die Wahrscheinlichkeit, dass in einer bestimmten Minute ein Flare vorhanden ist. Somit kann diese Darstellung die Erkennung von Flares erleichtern und bietet eine Basis dafür, mit einem besseren Modell die Erkennung zu automatisieren. Abbildung 2 zeigt die Erkennung und Markierung zweier Flares.

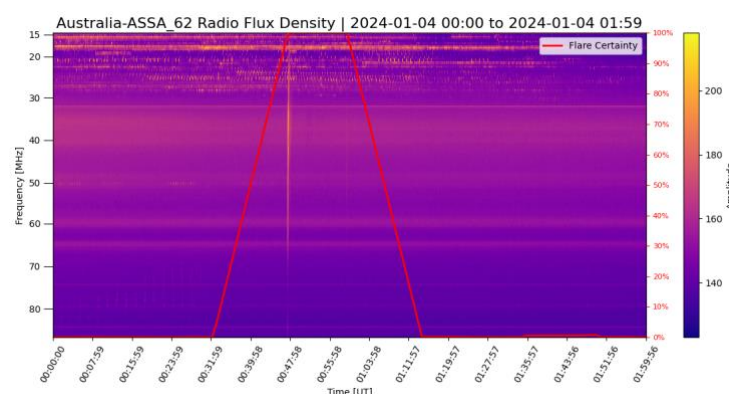


Abbildung 2: Erkennung und Markierung in der API

2 Theoretischer Teil

Im theoretischen Teil beschreiben wir den Hintergrund Data Science sowie von Sonneneruptionen und deren Klassifizierung und stellen unsere verwendeten Methoden vor.

2.1 Data Science

Data Science ist ein interdisziplinäres Feld, das Techniken und Theorien aus Bereichen wie Statistik, Informatik und Mathematik nutzt, um aus Daten Erkenntnisse zu gewinnen. In einer Welt, in der Daten in exponentiellem Masse wachsen, ist Data Science zum Schlüssel für das Verständnis komplexer Muster und Trends geworden. Es ermöglicht uns, aus unstrukturierten Datenmengen wertvolle Informationen zu extrahieren, die für Entscheidungsfindungen in verschiedenen Branchen wie Finanzen, Gesundheitswesen, Bildung und Astronomie entscheidend sind.

Der Kern der Data Science besteht darin, aus rohen Daten Wissen zu generieren. Dieser Prozess beginnt mit der Datensammlung, gefolgt von der Datenbereinigung, bei der ungenaue oder unvollständige Daten entfernt oder korrigiert werden. Anschliessend folgt die Datenexploration und -visualisierung, um Muster und Anomalien zu identifizieren. Mit statistischen Methoden und maschinellen Lernmodellen werden dann Vorhersagen getroffen oder Einsichten gewonnen.

Deep Learning, eine fortgeschrittene Form des maschinellen Lernens, ist ein wichtiger Aspekt der modernen Data Science. Es verwendet neuronale Netzwerke mit vielen Schichten (daher der Name "tiefes" Lernen), um komplexe Muster in grossen Datenmengen zu lernen und zu modellieren. Diese Technik ist besonders nützlich in Bereichen, in denen die Mustererkennung entscheidend ist, wie bei der Bild- und Spracherkennung oder der Vorhersage von Trends auf der Grundlage historischer Daten.

Data Science bietet eine leistungsstarke Methode, um die Herausforderungen bei der Klassifizierung von Sonneneruptionen mit E-Callisto zu bewältigen.

2.2 Wissenschaftlicher Kontext von Sonneneruptionen

Am ersten September 1859 konnte Richard Christopher Carrington erstmals den Zusammenhang zwischen Sonneneruptionen und Schwankungen im Magnetfeld der Erde feststellen (*Weltraumwetter*, o. J.). Er hat eine Explosion auf der Sonne festgestellt, die 20 Stunden später einen magnetischen Sturm auf der Erde ausgelöst hat.

Die Explosion auf der Sonne wird in der Zwischenzeit als Sonneneruption, resp. Solar Flare, genannt. Die Sonne sendet ständig Strahlung und geladene Teilchen in den Weltraum, was Sonnenwind genannt wird (*FAQ Sonnenstürme und Sonnenaktivität*, 2013a). Wird für eine kurze

Zeit und eine begrenzte Region auf der Sonne mehr Material ausgestossen, spricht man von einer Sonneneruption. Diese Regionen weisen eine hohe magnetische Feldstärke in den äussersten Schichten der Sonne (Chromosphäre und Korona) aus. In der Regel sind diese Feldlinien bogenförmig geschlossen, doch können sie aufbrechen, worauf die Plasmawolke (bestehend aus Protonen, Elektronen und weiteren elektrisch geladenen Atomen) entweicht. Die Veränderung der Feldlinien kann folgende drei Auswirkungen haben:

- Ausstossen von hochenergetischen Teilchen, welche die Erde in rund einer Stunde erreichen.
- Röntgenblitz, resp. Flare, weil die ausgestossenen Teilchen auch Schichten der Sonne treffen, die unterhalb des Ausstosses liegen.
- Koronaler Massenausstoss, resp. CME (coronal mass ejection), welcher zwar nicht immer auftritt, aber die zuvor erwähnten Schäden im Umfeld der Erde auslösen kann.

Unterteilt werden Sonnenruptionen in die fünf Klassen. Die Skala ist logarithmisch und die Klassen lauten A, B, C, M und X. X ist die stärkste Klasse und weist damit den höchsten Röntgenfluss auf (*FAQ Sonnenstürme und Sonnenaktivität*, 2013c). Während die Klasse C kaum Auswirkungen hat, können M-class zu Radiostörungen in den Polarregionen führen. X-class können global zu Störungen führen (*Weltraumwetter*, o. J.).

Radio-Emissionen und optische Folgen der Sonneneruptionen können von der Erde aus beobachtet werden, energetische Emissionen hingegen nur vom Weltall aus (*NASA Solar Flares*, o. J.). Zu den Satelliten, die die Sonne beobachten, gehören derzeit unter anderem Solar Orbiter, SOHO und Parker Solar Probe (*NASA Science Sun Exploration*, o. J.). Es gibt aktuell 19 laufende Missionen und weitere 13 sind in Planung. Abbildung 3 zeigt eine CME einer X-Class Sonneneruption, aufgenommen von SOHO (*SOHO (ESA & NASA) X-Class Solar Flare*, 2014).

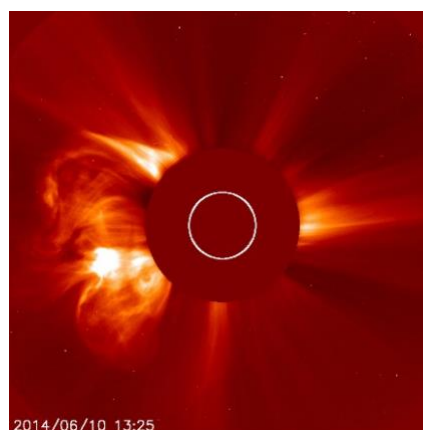


Abbildung 3: SOHO Aufnahme eines CME einer X-Class Sonneneruption

2.3 Herausforderungen in der Klassifikation

Die Daten von E-Callisto wurden bisher kaum verwendet, um automatische Klassifikationen vorzunehmen. In der Vorbereitung für das Projekt haben wir dazu fast keine Literatur gefunden. Singh et al (2019) haben einen Ansatz ausprobiert, bei dem sie einen Area Slope Index berechnet haben. Der Recall ist mit 95% relativ hoch, eine Schwäche von ihrer Methode besteht bei schwachen Signalen mit verhältnismässig hohem Rauschen. Weitere Untersuchungen sind uns von einem Team in Österreich und von Baris Bakirdöven und Vincenzo Timmel (2023) bekannt. Letztere haben einen ähnlichen Ansatz verfolgt, welcher auf 1-Minuten Bilder beruht. Wir möchten ihn aber ausbauen und wenn möglich auf 15-Minuten Bilder anwendbar machen.

Scully et al. (2023) haben Deep Learning Methoden angewendet, um Type III Solar Radio Bursts zu klassifizieren. Landa und Reuveni (2022) haben unterschiedliche Deep Learning Methoden auf Röntgenstrahlen angewendet und konnten X-Class mit einem Recall von über 90% klassifizieren.

Zur Vorhersage von Sonneneruptionen und Solar Flares (was nicht Teil dies Projekts ist, aber ein Verständnis der Methoden nützlich sein kann), wurden zuletzt vermehrt Deep Learning Methoden untersucht. Zur Vorhersage von Solar Flares haben Abdualah et al. (2023) eine neue Deep Learning Architektur entwickelt, die Convolution, LSTM und Transformer verbindet.

2.4 Deep Learning

Deep Learning ist eine Vertiefung von maschinellern Lernen, das in der Regel auf der Anwendung von Neuronalen Netzen aufbaut. Dazu werden diese Netze mit weiteren Layers verbunden, wodurch zum Beispiel ein Convolutional Neural Network (CNN) entsteht. CNNs sind eine Kategorie von Deep Learning-Modellen, die für die Analyse visueller Daten entwickelt wurden. Sie sind besonders effektiv bei der Mustererkennung in Bildern.

Folgendes sind die Schlüsselkomponenten von CNNs:

- Faltungsschichten (Convolutional Layers): Sie filtern das Bild, um bestimmte Merkmale wie Kanten, Texturen oder andere relevante visuelle Elemente zu identifizieren.
- Pooling-Schichten: Diese Schichten reduzieren die räumliche Grösse der Repräsentation, was eine Überanpassung des Modells (Overfitting) reduzieren kann.
- Aktivierungsfunktionen: Die Anwendung führt Nichtlinearität in das Netzwerk ein und ermöglicht es dem Modell, komplexe Muster zu lernen.
- Fully Connected Layers: Diese Schichten werden am Ende des Netzwerks eingesetzt, um die Merkmale, die durch die vorherigen Schichten extrahiert wurden, für die Klassifizierung oder Vorhersage zu nutzen.

Da wir unsere Daten als Bilder darstellen, eignen sich CNNs gut für unsere Aufgabenstellung.

2.5 ResNet-18

ResNet-18 ist ein Modell innerhalb der ResNet (Residual Network) Familie, die insbesondere in der Bildklassifizierung verwendet wird. ResNet wurde erstmals in dem Paper "Deep Residual Learning for Image Recognition" von He et al (2015) vorgestellt. Diese Architektur wurde entwickelt, um das Problem des Verschwindens von Gradienten in sehr tiefen neuronalen Netzen zu adressieren, ein häufiges Problem, das auftritt, wenn Netzwerke zu tief werden.

Folgendes sind die Schlüsselkomponenten von ResNet-18:

- **Residuale Blöcke:** Das Kernstück von ResNet sind die residualen Blöcke, die es ermöglichen, dass Signale direkt durch das Netzwerk fließen können. Dies wird durch sogenannte "Skip-Connections" erreicht, die es ermöglichen, dass die Ausgabe einer Schicht über eine oder mehrere Schichten hinweg 'übersprungen' wird.
- **18 Schichten:** ResNet-18 ist eine Version des ResNet-Modells, das 18 Schichten tief ist. Dies macht es vergleichsweise leichtgewichtig im Vergleich zu anderen Versionen wie ResNet-50 oder ResNet-101.
- **Effizienz:** Trotz seiner Tiefe bleibt ResNet-18 effizient in Bezug auf die Rechenleistung und den Speicherbedarf, was es für verschiedene Anwendungen, insbesondere für diejenigen mit begrenzten Ressourcen, praktikabel macht.

Abbildung 4 zeigt die Funktionsweise der Residuale Blöcke gemäss Paper von He et al:

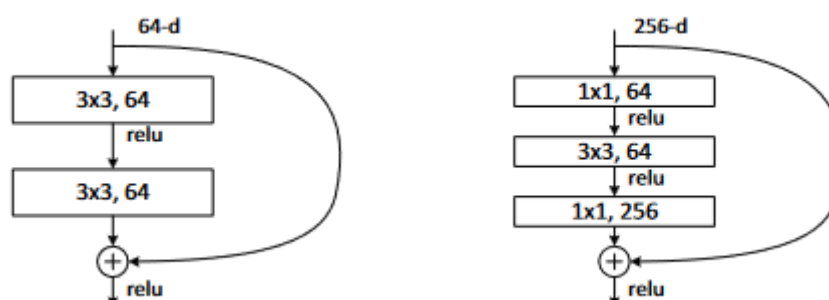


Abbildung 4: Funktionsweise der Residuale Blöcke

Innerhalb der ResNet Familie gibt es unterschiedlich grosse Modelle. Im ursprünglichen Paper wurden fünf unterschiedliche Grössen vorgestellt, sie haben zwischen 18 und 152 Layers. Die Architektur bleibt bei allen Varianten sehr ähnlich und wird in Abbildung 5 visualisiert.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Abbildung 5: Architektur der unterschiedlichen Modell-Größen

2.6 Weitere Deep Learning Modelle

Im Rahmen der Experimente haben wir weitere Deep Learning Modelle wie EfficientNetV2 (Tan & Le, 2021), ResNet50 (He et al., 2015) und ViT B16 (Dosovitskiy et al., 2021) getestet. Jedoch haben wir früh erkannt, dass diese Modelle unsere Trainingsdaten sehr stark overfitten. Deswegen haben wir zusätzlich zwei simplere und kleinere CNNs erstellt.

2.7 Custom CNN

Das CustomCNN Modell ist ein konvolutionelles neuronales Netz für binäre Klassifizierung. Es verwendet Parameter für Lernrate, Gewichtsabnahme und Dropout-Wahrscheinlichkeit. Das Modell hat vier konvolutionelle Schichten mit zunehmender Filterzahl (64, 128, 256, 512), jeweils gefolgt von Max-Pooling. Es folgen zwei lineare Schichten, die die Dimensionen zuerst auf 64 und dann auf 1 reduzieren. Eine Dropout-Schicht wird eingesetzt, um Overfitting zu verhindern. In der forward-Methode werden Eingabedaten durch diese Schichten geführt und ReLU sowie Sigmoid-Aktivierungsfunktionen für die Ausgabe verwendet. Abbildung 6 zeigt die Architektur.

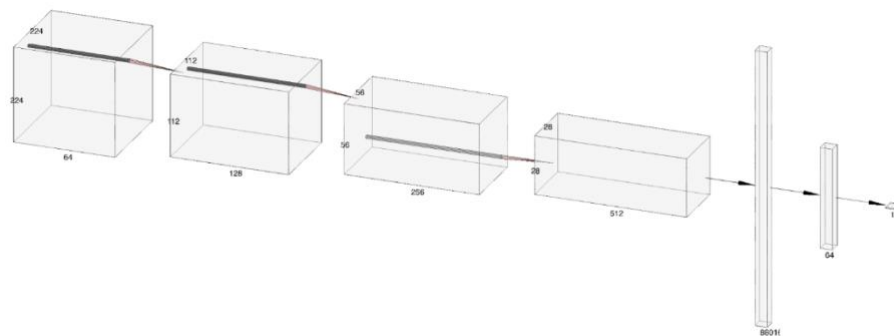


Abbildung 6: Architektur vom CustomCNN

2.8 Custom CNN 2

Das CustomCNN2 Modell wurde im Vergleich zum CustomCNN modifiziert, um Overfitting weiter zu reduzieren. Es verwendet kleinere Filtergrößen in den konvolutionellen Schichten, beginnend mit 32 und ansteigend bis 256, im Gegensatz zu den grösseren Filtern von 64 bis 512 im CustomCNN. Diese Anpassung verringert die Modellkomplexität. Zusätzlich integriert CustomCNN2 Batch-Normalisierung nach jeder konvolutionellen Schicht, was im ursprünglichen CustomCNN fehlt. Diese Ergänzungen tragen dazu bei, Overfitting zu mindern und könnten zu einer stabileren und effizienteren Trainingsleistung führen. Abbildung 7 zeigt die Architektur vom CustomCNN2.

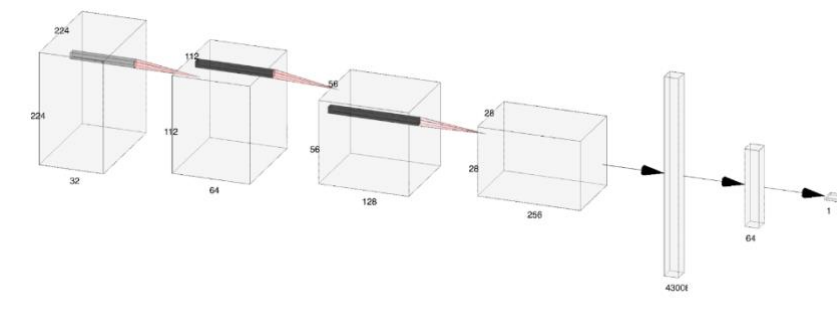


Abbildung 7: Architektur vom CustomCNN2

2.9 Metriken

Zur Evaluation der Resultate verwenden wir die Metriken Recall und Precision. Beide Metriken evaluieren die Qualität der Klassifikationen und eignen sich auch für Daten mit Imbalance.

Recall:

- Definition: Recall misst den Anteil der tatsächlich positiven Fälle, die vom Modell korrekt als positiv identifiziert wurden.
- Formel: $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$, wobei TP die Anzahl der wahren positiven Ergebnisse (True Positives) und FN die Anzahl der falsch negativen Ergebnisse (False Negatives) ist.

Precision:

- Definition: Precision misst den Anteil der vom Modell als positiv identifizierten Fälle, die tatsächlich positiv sind.
- Formel: $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$, wobei FP die Anzahl der falsch positiven Ergebnisse (False Positives) ist.

3 Praktischer Teil

Hier thematisieren wir den praktischen Teil der Arbeit, wie die Datengrundlage, das Training und die Evaluation unserer Modelle und unsere Erkenntnisse.

3.1 Datenerfassung und -management

Die erfassten Daten werden von den e-Callisto Stationen selbstständig auf einen Server der FHNW hochgeladen. Diese hostet die Daten und stellt sie zum Download via Browser (Monstein et al., 2023) oder via API (Timmel, 2023) zur Verfügung.

Die 1-Minuten-Bilder, bereitgestellt von Timmel, sind im "data" Ordner des FlareSense/FlareSense Repositories bei DagsHub zu finden (Torres Gamez & Schürmann, 2023). Der eigens erstellte Datensatz mit 15-Minuten-Spektren ist bei Hugging Face unter i4ds/ecallisto-bursts verfügbar (Torres Gamez, 2023).

3.2 Datenanalyse und Vorverarbeitung

In diesem Kapitel widmet sich die Analyse den Bursts und dem neu generierten Datensatz. Der initiale Datensatz, welcher von Vincenzo Timmel zur Verfügung gestellt wurde, findet in dieser Betrachtung keine Berücksichtigung. Der Grund hierfür liegt in der Entscheidung, Bilder in einem 15-Minuten-Intervall zu klassifizieren, was die Erstellung eines neuen Datensatzes erforderlich machte.

3.2.1 Labels

Im Rahmen der Untersuchung der Sonneneruptionen (Sunbursts) wird auf die Daten des i4DS zurückgegriffen, welche unter der folgenden Internetadresse abrufbar sind: <https://soleil.i4ds.ch/solarradio/data/BurstLists/>. Für jeden Monat eines Jahres stellt das i4DS eine Textdatei zur Verfügung, die detaillierte Informationen über die Sonneneruptionen enthält. Diese Dateien beinhalten das Datum, die Uhrzeit, den Typ der Sonneneruption sowie die Stationen, an denen eine bestimmte Sonneneruption detektiert wurde. Ein Ausschnitt davon wird in Tabelle 1 dargestellt.

Date	Time	Type	Stations
20230101	01:53-02:09	III	Australia-ASSA, ssRT
20230101	07:11-07:12	VI	ALMATY, Australia-ASSA, INDIA-GAURI, INDIA-OOTY, INDIA-UDAIPUR, ssRT
20230101	07:42-07:48	III	ALMATY, AUSTRIA-UNIGRAZ, Australia-ASSA, GERMANY-DLR, INDIA-GAURI, INDIA-OOTY, INDIA-UDAIPUR, SOUTHAFRICA-SANSA, ssRT, SWISS-Landschlacht, TRIEST
20230101	10:42-10:42	III	(AUSTRIA-UNIGRAZ), BIR, GERMANY-DLR, GLASGOW, INDIA-GAURI, INDIA-OOTY, SOUTHAFRICA-SANSA
...	...	...	...

Tabelle 1: Inhalte der Dateien (Ausschnitt)

Für die Vorbereitung der Labeldaten wurde von Vincenzo Timmel (2023) ein Repository erstellt. Dieses Repository dient dem Herunterladen der Labeldaten und der Korrektur von Fehlern wie inkorrekten oder unmöglichen Uhrzeiten sowie Rechtschreibfehlern in den Bezeichnungen der Stationen.

Bei der Analyse der exportierten Daten vom 1. Januar 2021 bis zum 12. Dezember 2023 zeigt sich, dass die Dauer der Sonneneruptionen grösstenteils zwischen 0 und 10 Minuten liegt. Allerdings gibt es auch einige Ausreisser, die bis zu 23 Stunden und 59 Minuten dauern, was darauf hindeutet, dass in diesen Fällen der gesamte Tag klassifiziert wurde. Abbildung 8 zeigt ein Histogramm der Dauer der gemessenen Bursts.

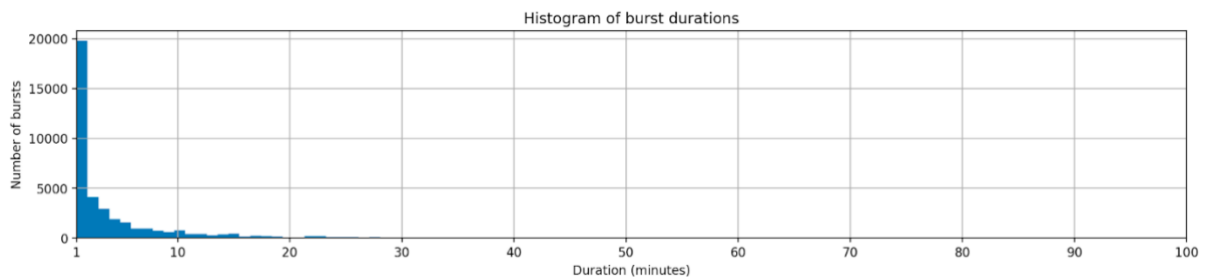


Abbildung 8: Histogramm der Dauer der Bursts

In der Analyse der Anzahl der Labels pro Station stellte sich heraus, dass die meisten Sonneneruptionen von der Station Australia-ASSA detektiert wurden. Angesichts dieser Erkenntnis wurde für den Umfang dieses Projekts beschlossen, ausschliesslich der Daten der Australia-ASSA Stationen zu berücksichtigen. Abbildung 9 zeigt die Anzahl Bursts pro Station.

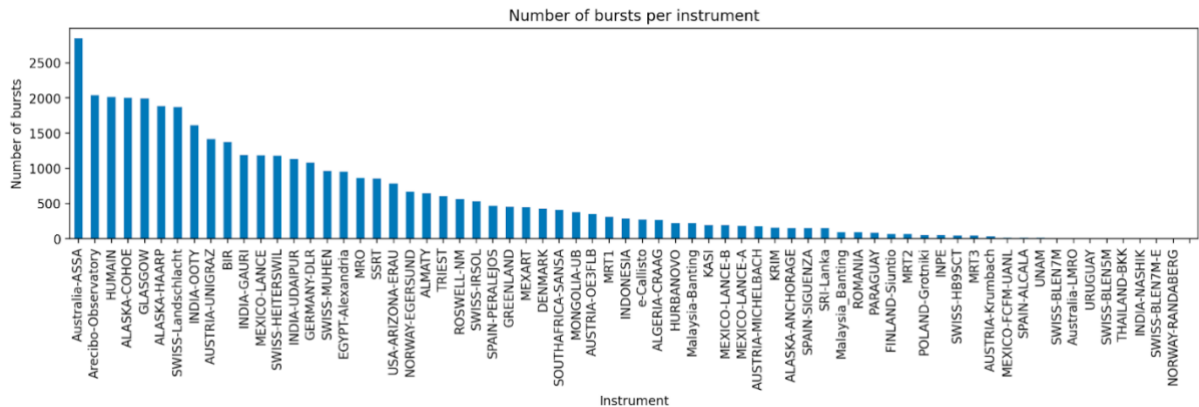


Abbildung 9: Anzahl Bursts pro Instrument

In unserem spezifischen Anwendungsfall haben wir uns dazu entschieden, die Dauer der Sonneneruptionen auf maximal 10 Minuten zu beschränken. Dabei gehen wir von der Annahme aus, dass der Burst zu Beginn des betrachteten Zeitintervalls auftritt und Burstlängen von über 10 Minuten als Labelingfehler betrachtet werden. Sonneneruptionen mit einer negativen Dauer oder einer Dauer von über 60 Minuten wurden vollständig aus dem Datensatz entfernt.

Bei der Visualisierung der Uhrzeiten der von der Station Australia-ASSA erkannten Bursts zeigt sich, dass diese Station während der Nachtstunden fast keine Bursts detektiert. Dies lässt sich dadurch erklären, dass in der Nacht keine direkte Sichtlinie zur Sonne besteht. Die Verteilung wird in Abbildung 10 dargestellt.

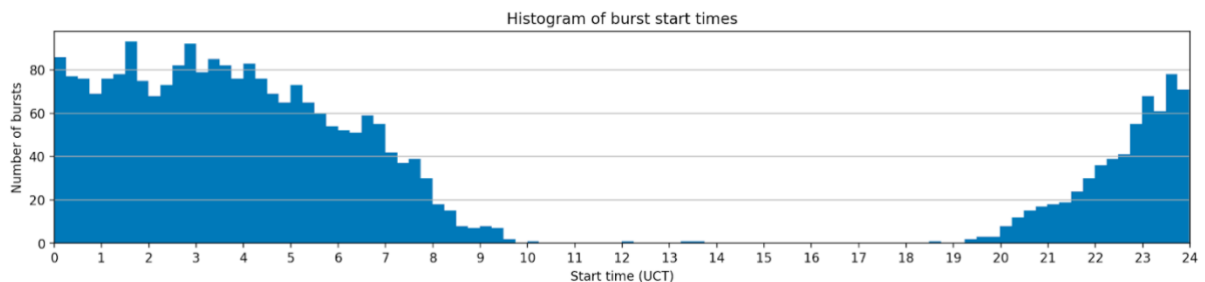


Abbildung 10: Histogramm der Startzeiten der Bursts

3.2.2 Spektren

Bei der Station Australia-ASSA zeigte sich, dass die Instrumente (Australia-ASSA)_01, _02, _62 und _63 besonders effektiv bei der Erkennung von Bursts waren. Basierend auf diesen Erkenntnissen wurde ein Datenexport mit 15-Minuten-Intervallen generiert. Der Export wurde mit der Hilfe vom i4DS Repository ecallisto_ng erstellt. Für diesen Export wurden die Bursts augmentiert, indem vor und nach jedem Burst verschiedene Paddings hinzugefügt wurden. Zeitpunkte ohne Burst-Ereignis wurden als 'no_burst' klassifiziert und ebenfalls heruntergeladen, jedoch nicht augmentiert.

Dieser Datensatz ist auf Hugging Face unter dem Namen “i4ds/ecallisto-bursts” oder der folgenden URL verfügbar: <https://huggingface.co/datasets/i4ds/ecallisto-bursts>. Bei den Instrumenten _01 und _02 konnten Daten vom 13. Februar 2021 bis zum 11. Dezember 2021 / 9. Dezember 2021 heruntergeladen werden, während für die Instrumente _62 und _63 Daten vom 10. Dezember 2021 bis zum 12. Dezember 2023 heruntergeladen wurden.

In Abbildung 11 wird ein Spektrum vom Instrument Australia-ASSA_01 visualisiert. Auf der x-Achse wird die Uhrzeit dargestellt, auf der y-Achse befindet sich der Frequenzbereich. Die Werte im Spektrum entsprechen der Amplitude des Signal pro Frequenz und Uhrzeit.

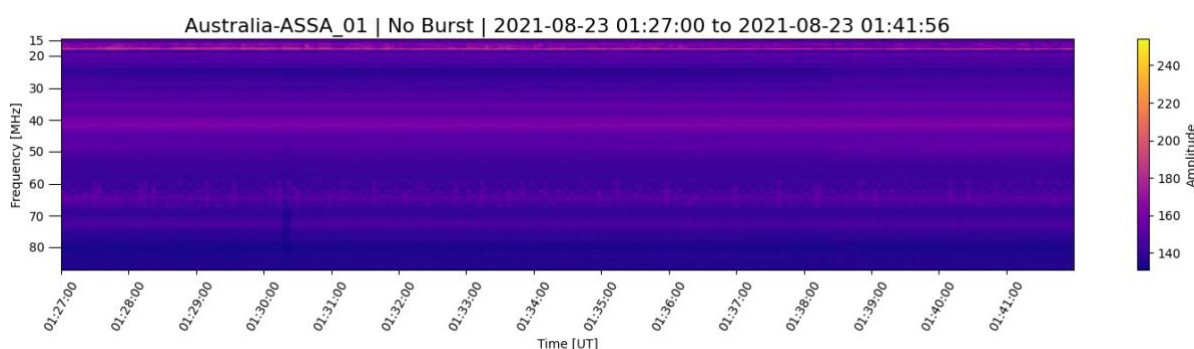


Abbildung 11: Spektrum vom Instrument Australia-ASSA_01

Wir erkennen, dass hier kein Burst vorhanden ist. Das Spektrum sieht in fast allen Zeitabschnitten genau gleich aus. In Abbildung 12 wird ein Spektrum mit einem kleinen Radioburst visualisiert.

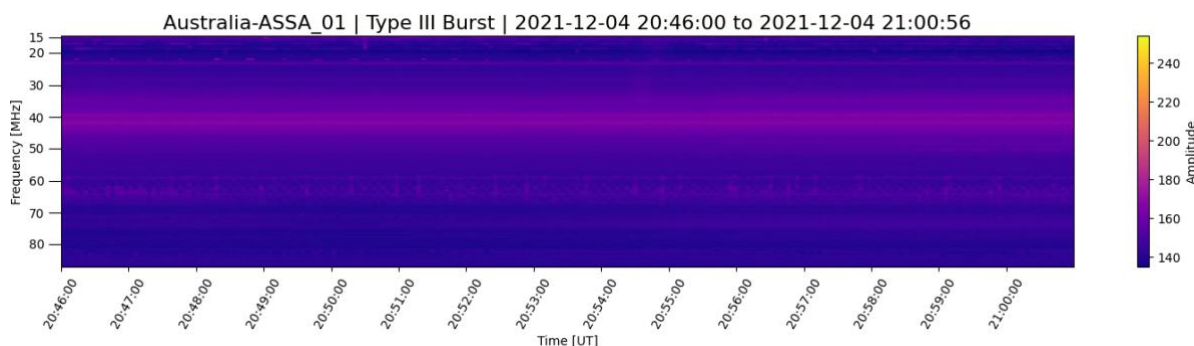


Abbildung 12: Beispiel von einem schwach erkennbaren Burst

Hier ist es deutlich schwieriger zu erkennen, dass ein Burst in den Daten versteckt ist. Jedoch sieht man zwischen den Minuten 20:54 und 20:55 einen geringen Burst. Zum Schluss wird in Abbildung 13 ein deutlicher erkennbarer Burst visualisiert.

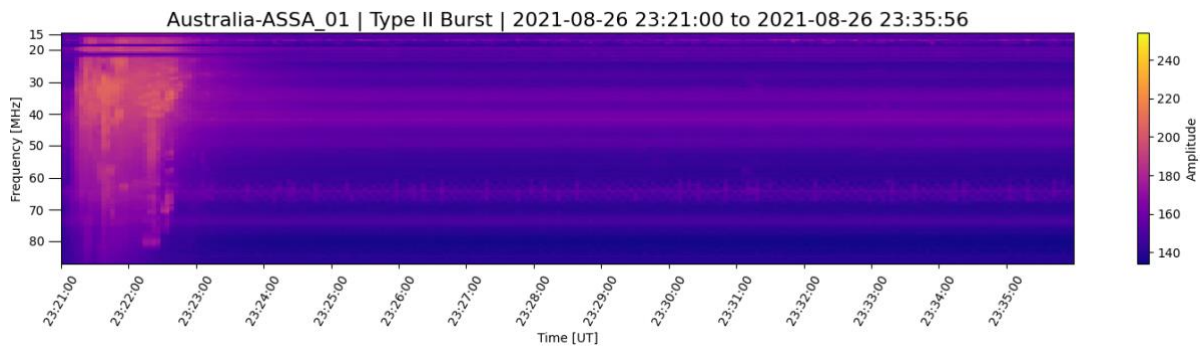


Abbildung 13: Beispiel von einem besser erkennbaren Burst

Hier erkennen wir einen sehr deutlichen Burst in den Minuten 23:21 und 23:22.

Als letztes schauen wir uns in Abbildung 14 die Distribution der Bursts (inkl. Augmentierten Daten) in unserem Datensatz an.

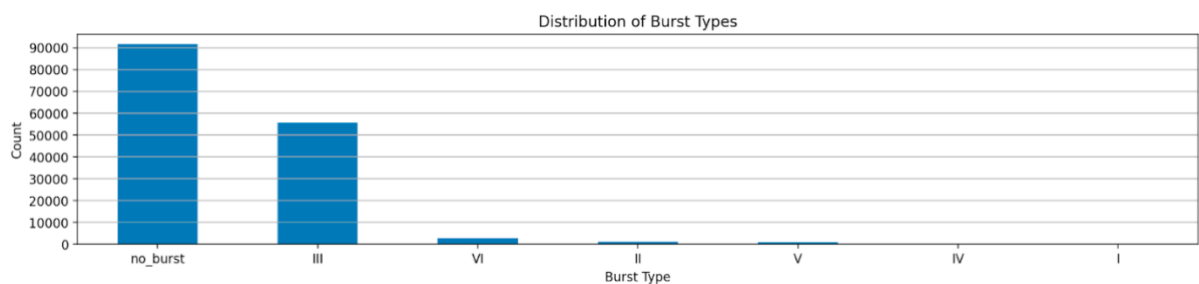


Abbildung 14: Verteilung der Burst Typen

In unserem Datensatz überwiegen die Sonneneruptionen der Kategorie III deutlich, während Daten zu anderen Klassen in geringerer Anzahl vorhanden sind. Da der Fokus unserer Arbeit jedoch auf der Klassifikation von Bursts und No Bursts liegt, stellt diese Ungleichverteilung kein wesentliches Problem für unsere Analyse dar.

3.3 Entwicklung des Modells

Wir haben uns für die Verwendung von PyTorch Lightning für das Training unserer Modelle entschieden und nutzen MLflow, um unsere Metriken zu protokollieren. Die verwendeten Jupyter Notebooks, Modell- und Dataloader-Quelldateien sind in unserem GitHub-Repository i4Ds/FlareSense (Torres Gamez & Schürmann, 2024) zu finden. Die Experimente, inklusive Metriken und Modellcheckpoints, sind unter DagsHub (Torres Gamez & Schürmann, 2023) verfügbar.

3.4 Resultate

Bei den Experimenten wurden für die unten genannten Modelle folgende Hyperparameter untersucht:

- Learning Rate: 1e-3, 3e-4, 1e-4, 3e-5
- L2-Regularisierung: 3e-3, 1e-3, 3e-4, 1e-4

Dies ergab pro Modell 16 Experimente, deren Resultat in der Folge vorgestellt und später diskutiert werden. Weiter wurde bei jedem Experiment Data Augmentierung durchgeführt.

In den folgenden Tabellen werden die jeweils besten und schlechtesten erzielten Werte pro Metrik, sowie der Median dargestellt. Die Metriken beziehen sich immer auf den Validierungsdatensatz.

3.4.1 ResNet50

Metrik	Bestes Resultat	Schlechtestes Resultat	Median
Recall	0.922	0.806	0.867
Precision	0.738	0.552	0.637

Tabelle 2: Resultate vom ResNet50

Tabelle 2 zeigt die wichtigsten Resultate vom ResNet50. Die Metrik Recall beträgt zwischen 0.806 und 0.922 und der Median beträgt 0.867. Der Median liegt damit über dem gemäss Projektvereinbarung angestrebten Wert von 0.85. Bei der Precision liegt die Bandbreite zwischen 0.552 und 0.738, mit einem Median von 0.637. Alle Werte liegen unter dem angestrebten Wert von 0.9.

3.4.2 ResNet18

Metrik	Bestes Resultat	Schlechtestes Resultat	Median
Recall	0.919	0.806	0.847
Precision	0.738	0.505	0.637

Tabelle 3: Resultate vom ResNet18

Tabelle 3 zeigt die wichtigsten Resultate vom ResNet18. Die Metrik Recall liegt mit 0.806-0.919 und einem Median von 0.847 ebenfalls zu einem grossen Teil über dem Zielwert von 0.85. Die Precision beträgt zwischen 0.505 und 0.738 mit einem Median von 0.637.

3.4.3 CustomCNN

Metrik	Bestes Resultat	Schlechtestes Resultat	Median
Recall	0.881	0.633	0.819
Precision	0.689	0.286	0.578

Tabelle 4: Resultate vom CustomCNN

Tabelle 4 zeigt die wichtigsten Resultate vom CustomCNN. Es drei Kombinationen von Learning Rate und Regularisierung, bei der das Modell nicht trainieren, resp. lernen, konnte. Wir haben diese Experimente deshalb entfernt und beachten sie in der Auswertung nicht. Der Recall liegt in einem Bereich von 0.633 bis 0.881 mit einem Median von 0.819. Die Precision liegt zwischen 0.286 und 0.689 mit einem Median von 0.578.

3.4.4 CustomCNN2

Metrik	Bestes Resultat	Schlechtestes Resultat	Median
Recall	0.897	0.633	0.827
Precision	0.708	0.341	0.639

Tabelle 5: Resultate vom CustomCNN2

Tabelle 5 zeigt die wichtigsten Resultate vom CustomCNN2. Die Bandbreite liegt zwischen 0.633 und 0.897 und der Median beträgt 0.827. Die Precision reicht von 0.341 bis 0.708 mit einem Median von 0.639.

Abbildung 15 vergleicht den maximalen Recall pro Modell.

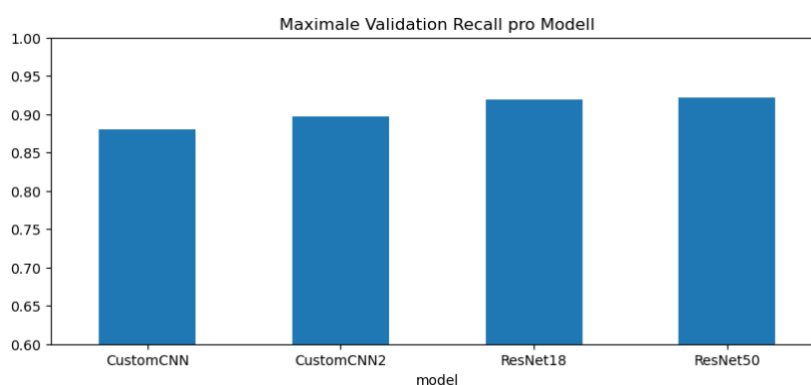


Abbildung 15: Maximaler Recall pro Modell

Die beiden ResNet Modelle haben mit über 0.9 den höchsten Recall erzielt. Bei allen Modellen liegt der Recall über dem ursprünglich angestrebten Wert von 0.85.

Abbildung 16 vergleicht die maximale Precision pro Modell auf den gleichen Achsen-Werten.

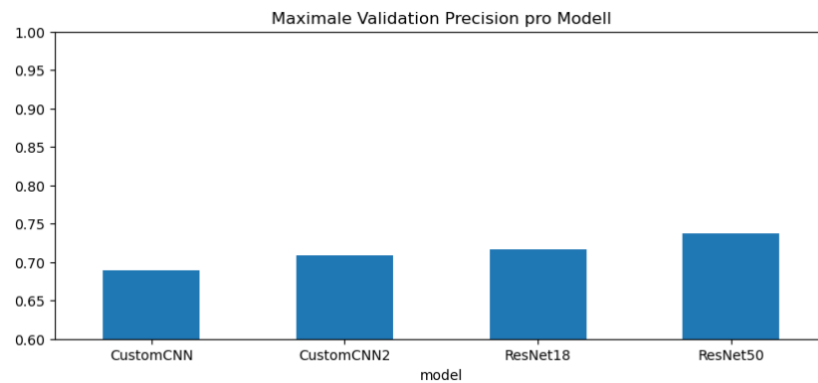


Abbildung 16: Maximale Precision pro Modell

Im Vergleich zum Recall liegen die maximal erzielten Werte bei der Precision tiefer. Der höchste Wert wurde vom ResNet50 erzielt, die anderen drei Modelle haben alle einen ähnlichen Wert erzielt. Insgesamt liegen die Resultate unter dem angestrebten Wert von 0.9.

Abbildung 17 zeigt das Verhältnis zwischen Precision und Recall.

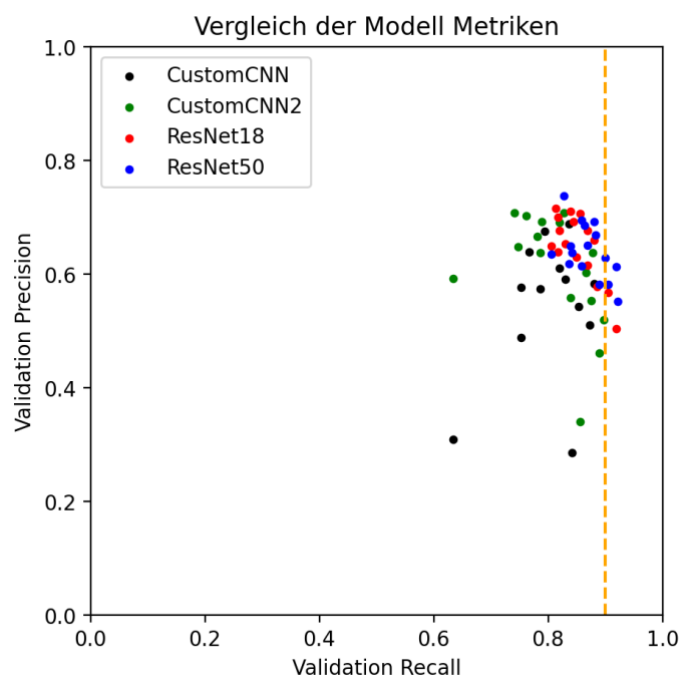


Abbildung 17: Vergleich aller Resultate

Es wird visuell ersichtlich, dass ResNet50 (blau) und ResNet18 (rot) tendenziell das beste Precision/Recall Verhältnis aufweisen. Dies lässt sich daran ablesen, dass ihre Experimente am stärksten gegen den oberen rechten Rand tendieren. Am tiefsten liegen die Werte beim CustomCNN

Orange eingezeichnet liegt die Schwelle von Validation Recall Wert 0.9. Für die Wahl des besten Modells und Hyperparameter wäre dies die Schwelle gewesen und bei den Modellen darüber wäre das mit der höchsten Precision gewählt worden. Da es aber ein paar Modelle mit Recall knapp darunter, aber eine höhere Precision gibt, sind wir von unserem Richtwert abgewichen und haben das Modell mit Recall 0.881 und Precision 0.692 entschieden.

Unser bestes Modell ResNet50 mit Recall 0.88 und Precision 0.69 wurde mit einer Learning Rate von $3e-3$ und einer L2-Regularisierung von $1e-4$ erzielt.

3.5 Evaluierung der Modelle

Während die Metriken Recall und Precision auf dem Validierungsdatensatz die primären Ziele erreicht haben, haben wir bei der vertieften Untersuchung der Resultate festgestellt, dass das Training der Modelle nicht gut verlaufen ist. Abbildung 18 zeigt beispielsweise für das als bestes definierte Modell, wie sich der Loss auf dem Trainings- und Validierungsdatensatz entwickelt hat.

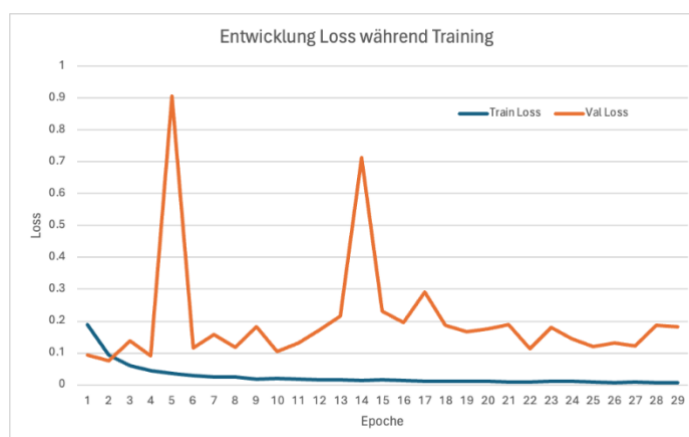


Abbildung 18: Entwicklung von Loss während Training

Während der Trainingsloss gemäss dem erwarteten Muster Richtung 0 gesunken ist, hat sich der Loss auf den Validierungsdaten insgesamt horizontal entwickelt. Er weist kein klassisches Muster vom sinkenden Loss oder Overfitting auf. Wir vermuten, dass das Modell (ResNet50) bereits nach der ersten Epoche vollständig gelernt hat und sich danach nur noch der Trainingsloss verbessert hat. Dies könnte sich dadurch erklären lassen, dass aufgrund der Batchgrösse innerhalb der ersten Epoche die Gewichte im Modell trainiert und verbessert wurden. Diese Annahme beruht darauf, dass sich Recall und Precision nach der ersten Epoche nicht mehr verbessert haben, sondern einigermaßen konstant geblieben sind, wie es in Abbildung 19 dargestellt wird.

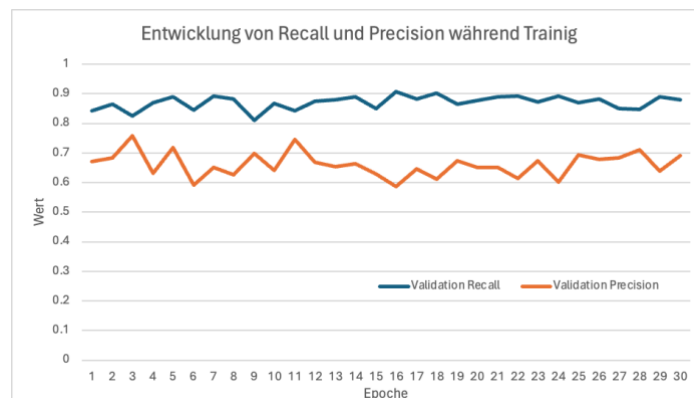


Abbildung 19: Entwicklung der Metriken während Training

Wir haben dieses Verhalten bei allen vier Modellen und den meisten Experimenten festgestellt. Bei den beiden eigenen Modellen (CustomCNN und CustomCNN2) haben wir vereinzelte Experimente gefunden, bei denen das Modell gemäss Entwicklung von Validation-Loss, Recall und Precision gelernt zu haben scheint. Diese weisen aber eine grössere Differenz zwischen Loss von Train und Validation auf und haben tiefere Recall und Precision Werte auf den Validierungsdaten.

3.6 Wichtige Erkenntnisse

Während die Resultate auf den ersten Blick zufriedenstellend waren, hat eine vertiefte Analyse ergeben, dass das Training der Modelle nicht zufriedenstellend verlaufen ist. Die Modelle weisen nicht erwartungsgemäss einen sinkenden Validierungs-Loss auf. Er bleibt auf einem hohen Niveau und entwickelt sich, abgesehen von Ausreissern, seitwärts. Dasselbe Muster ist bei den Metriken Recall und Precision zu erkennen. Sie liegen schon nach der ersten Epoche auf einem hohen Niveau und verbessern sich nicht.

Bei den eigenen, simpleren Modellen gibt es einzelne Kombinationen von Learning Rate und Regularisierung, bei denen die Modelle weniger schnell zu lernen scheinen. Bei ihnen entwickeln sich die Metriken wie es zu erwarten wäre. Allerdings weisen die Metriken tiefere Werte auf und der Validierungs-Loss sinkt deutlich weniger stark als Trainings-Loss. Allerdings gibt es bei den simpleren Modellen auch Experimente, bei denen das Modell nicht lernen konnte. Die Wahl der korrekten Hyperparameter scheint dort noch wichtiger zu sein.

3.7 Sunburst Detection API

Um unsere trainierten Modelle auch nutzen zu können, haben wir eine API entwickelt. Diese API wird im Notebook 16-API-Test.ipynb im FlareSense Repository (Torres Gamez & Schürmann, 2024) angesteuert. Zukünftig sollte diese auch im ecallisto-ng (Timmel, 2023) Repository integriert werden. Hier wird die originelle Visualisierung von ecallisto-ng mit der Sicherheit der Vorhersagen des Modells addiert. Für die Berechnung der Sicherheit wird ein Zeitraum mittels ecallisto-ng heruntergeladen, für jede Minute ein 15 Minuten Fenster erstellt, diese Fenster einzeln mit unserem Modell vorhergesagt und schlussendlich der Mittelwert jeder Minute gezogen. Abbildung 20 zeigt ein Beispiel der Detektion eines Bursts auf einen nicht in dem Datensatz vorhandenen Zeitpunktes.

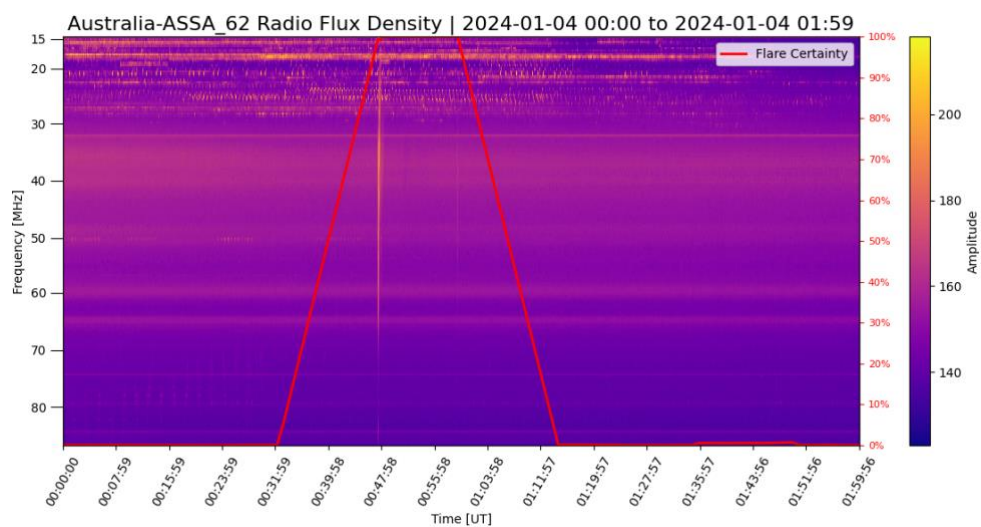


Abbildung 20: Erkennung und Markierung in der API

In diesem Zeitfenster befinden sich 2 verschiedene, unterschiedlich grosse, aufeinander folgende Bursts. Unser Modell gibt wie erwartet eine hohe Sicherheit bei der Vorhersage in dem Zeitpunkt der beiden Bursts.

4 Fazit

Im ersten Teil des Projekts konnten wir früher erzielte Resultate in unserem Deep Learning-Framework PyTorch replizieren. In dem damaligen Projekt wurden die Experimente auf 1-Minuten Daten durchgeführt.

In der Folge haben wir die Datenaufbereitung geändert und die Modelle mit 15-Minuten Fenstern trainiert. Wir haben festgestellt, dass bereits einfache Deep-Learning-Architekturen wie ResNet die Daten lernen können. Wir haben deshalb nur ResNet18, ResNet50 und zwei eigene, simple Architekturen verwendet. Bei den Resultaten haben wir festgestellt, dass es Modelle und Hyperparameter gibt, welche die definierten Erwartungen beim Recall erfüllt haben. Bei der Precision sind die Ergebnisse nicht so gut, wie bei den Daten auf 1-Minuten-Basis. Bei der vertieften Analyse haben wir festgestellt, dass die Modelle trotz der guten Metriken nicht wie erwartet gelernt haben. Der Validierungs-Loss sowie Recall und Precision auf den Validierungsdaten haben sich nach der ersten Epoche nicht mehr verbessert. Es scheint, als hätten die Modelle bereits in der ersten Epoche bestmöglich gelernt und sich in den restlichen Epochen lediglich der Trainings-Loss verbessert hat. Dieses Muster betrifft alle Modelle, bei den simpleren Modellen gab es aber Kombinationen von Hyperparametern, bei denen das Trainings-Muster eher den Erwartungen entsprach. Sie weisen aber schlechtere Metriken auf und haben eine deutliche Differenz zwischen Trainings- und Validierungs-Loss.

Zuletzt haben wir eine API erstellt, welche neue Daten anhand von «ecallisto_ng» und unseren Modellen herunterladen und verarbeiten kann, sodass man schlussendlich eine Übersicht über die potenziellen Bursts in einem gegebenen Zeitraum bekommen kann. Hier haben wir uns für einen Ansatz entschieden, welcher viele Vorhersagen macht und diese dann kombiniert. Dieser hilft somit, Fehler von einzelnen Vorhersagen zu mindern und somit eine akkuratere und robustere Vorhersage generiert.

Auch wenn wir im Rahmen dieses Projekts keine Trainings durchführen konnten, bei dem die Modelle gemäss dem klassischen Muster die Daten lernen, empfehlen wir, die Untersuchungen mit 15-Minuten Daten weiterzuführen. Gemäss den Bildern unterscheiden sich Sonneneruptionen vom Hintergrund, weshalb wir erwarten, dass Modelle diese Unterschiede lernen können.

4.1 Empfehlung für weitere Schritte

- Vertiefte Untersuchung der verwendeten Modelle. Da das Problem beim Training zu wenig verstanden wird, sollte das Verhalten in der ersten Epoche untersucht werden, bevor weitere Methoden getestet werden.
- Mehr Regularisierungsmethoden untersuchen: Regularisierung ist eine bewährte Methode, um Modelle robuster zu machen. Dies könnte helfen, die Precision zu verbessern und sie auf ein vergleichbares Niveau zu den 1-Minuten Daten zu bringen.
- Verwendung von Zeitreihen anstatt Bildern: Wir haben die Daten in Bilder umgewandelt und Methoden zur Klassifikation von Bildern angewendet. Da die Rohdaten als Zeitreihe vorliegen, könnten Deep Learning-Methoden zur Klassifikation von Zeitreihen verwendet werden. Dies wären zum Beispiel RNN oder LSTM.
- Ansätze mit tieferer Modellkomplexität anstelle von CNNs: Wir haben festgestellt, dass bereits einfache Architekturen die Sonneneruptionen erkennen können. Wir können es uns deshalb vorstellen, dass Methoden, die weniger komplex als Deep Learning sind, gute Resultate erzielen oder als Vorstufe dienen können. Möglichkeiten sehen wir zusätzlich im Bereich der Statistik, Recurrent Neural Networks oder Unsupervised Learning.

Quellenverzeichnis

- Abduallah, Y., Wang, J. T. L., Wang, H., & Xu, Y. (2023). Operational prediction of solar flares using a transformer-based framework. *Scientific Reports*, 13(1), 13665. <https://doi.org/10.1038/s41598-023-40884-1>
- Andrews, R. G. (2022, Februar 9). Solar Storm Destroys 40 New SpaceX Satellites in Orbit. *The New York Times*. <https://www.nytimes.com/2022/02/09/science/spacex-satellites-storm.html>
- Bakırdöven, B. (2023, August 28). *Radio Sunburst Detector*.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* (arXiv:2010.11929). arXiv. <https://doi.org/10.48550/arXiv.2010.11929>
- E-Callisto*. (2023, Oktober 26). <https://www.e-callisto.org/>
- ESA - Weltraumwetter: Die zerstörerische Kraft der Sonne. (o. J.). ESA - Weltraumwetter: Die Zerstörerische Kraft Der Sonne. Abgerufen 10. Dezember 2023, von https://www.esa.int/Space_in_Member_States/Germany/Weltraumwetter_Die_zerstoererische_Kraft_der_Sonne
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition* (arXiv:1512.03385). arXiv. <https://doi.org/10.48550/arXiv.1512.03385>
- Landa, V., & Reuveni, Y. (2022). Low Dimensional Convolutional Neural Network For Solar Flares GOES Time Series Classification. *The Astrophysical Journal Supplement Series*, 258(1), 12. <https://doi.org/10.3847/1538-4365/ac37bc>
- Missions to the Sun*. (o. J.). Abgerufen 10. Dezember 2023, von <https://science.nasa.gov/sun/exploration/>
- Monstein, C. (2023, Oktober 26). *Callisto Observatory Landschlacht*. e-Callisto. <https://www.e-callisto.org/HB9SCT/Landschlacht/Landschlacht.html>
- Monstein, C., Csillaghy, A., & Benz, A. O. (2023). *CALLISTO Solar Spectrogram FITS files* [dataset]. International Space Weather Initiative. <https://doi.org/10.48322/PMWD-MK15>

- Scully, J., Flynn, R., Gallagher, P., Carley, E., & Daly, M. (2023). Improved Type III solar radio burst detection using congruent deep learning models. *Astronomy & Astrophysics*, 674, A218. <https://doi.org/10.1051/0004-6361/202346404>
- Singh, D., Raja, K. S., Subramanian, P., Ramesh, R., & Monstein, C. (2019). Automated Detection of Solar Radio Bursts using a Statistical Method. *Solar Physics*, 294(8), 112. <https://doi.org/10.1007/s11207-019-1500-0>
- SOHO views X-class solar flare. (2014, Juni 13). https://www.esa.int/ESA_Multimedia/Images/2014/06/SOHO_views_X-class_solar_flare
- Solar Flares. (o. J.). Abgerufen 10. Dezember 2023, von <https://www.jpl.nasa.gov/nmp/st5/SCIENCE/flares.html>
- Tan, M., & Le, Q. V. (2021). *EfficientNetV2: Smaller Models and Faster Training* (arXiv:2104.00298). arXiv. <https://doi.org/10.48550/arXiv.2104.00298>
- Timmel, V. (2023). *Ecallisto-ng* (0.6.7) [Python; OS Independent]. Institute for Data Science. https://github.com/i4Ds/ecallisto_ng
- Torres Gamez, G. (2023, Dezember 25). *i4ds/ecallisto-bursts* · Dataset. Huggingface. <https://huggingface.co/datasets/i4ds/ecallisto-bursts>
- Torres Gamez, G., & Schürmann, P. (2023). *FlareSense/FlareSense*. DagsHub. <https://dagshub.com/FlareSense/FlareSense>
- Torres Gamez, G., & Schürmann, P. (2024). *i4Ds/FlareSense*. <https://github.com/i4Ds/FlareSense>
- Was ist ein Sonnensturm und wie entsteht er? (2013a, Januar). <https://www.mps.mpg.de/sonnenstuerme-sonnenaktivitaet-faq/1>
- Welche Auswirkungen können Sonnenstürme auf der Erde haben? (2013b, Januar). <https://www.mps.mpg.de/sonnenstuerme-sonnenaktivitaet-faq/4>
- Wie werden Sonnenstürme klassifiziert? (2013c, Januar). <https://www.mps.mpg.de/sonnenstuerme-sonnenaktivitaet-faq/3>

Abbildungsverzeichnis

Abbildung 1: CALLISTO Long Wavelength Antenna in Landschlacht	1
Abbildung 2: Erkennung und Markierung in der API	2
Abbildung 3: SOHO Aufnahme eines CME einer X-Class Sonneneruption	4
Abbildung 4: Funktionsweise der Residuale Blöcke.....	6
Abbildung 5: Architektur der unterschiedlichen Modell-Größen	7
Abbildung 6: Architektur vom CustomCNN	8
Abbildung 7: Architektur vom CustomCNN2	8
Abbildung 8: Histogramm der Dauer der Bursts.....	11
Abbildung 9: Anzahl Bursts pro Instrument	12
Abbildung 10: Histogramm der Startzeiten der Bursts	12
Abbildung 11: Spektrum vom Instrument Australia-ASSA_01	13
Abbildung 12: Beispiel von einem schwach erkennbaren Burst	13
Abbildung 13: Beispiel von einem besser erkennbaren Burst	14
Abbildung 14: Verteilung der Burst Typen.....	14
Abbildung 15: Maximaler Recall pro Modell	16
Abbildung 16: Maximale Precision pro Modell.....	17
Abbildung 17: Vergleich aller Resultate.....	17
Abbildung 18: Entwicklung von Loss während Training	18
Abbildung 19: Entwicklung der Metriken während Training	19
Abbildung 20: Erkennung und Markierung in der API	20

Tabellenverzeichnis

Tabelle 1: Inhalte der Dateien (Ausschnitt)	11
Tabelle 2: Resultate vom ResNet50	15
Tabelle 3: Resultate vom ResNet18	15
Tabelle 4: Resultate vom CustomCNN.....	16
Tabelle 5: Resultate vom CustomCNN2.....	16

Hilfsmittelverzeichnis

Hilfsmittel	Verwendung	Betroffene Stellen
Zotero	Erstellung der Kurzreferenzen im Text und des Literaturverzeichnisses.	Kurzbelege im Fliesstext Literaturverzeichnis
ChatGPT	Optimierung einzelner Textstellen auf Lesbarkeit.	Gesamte Arbeit
ChatGPT	Erstellung der Beschreibung der Metriken.	Kapitel 2.9
GitHub Copilot und ChatGPT	Unterstützung bei der Programmierung.	Quellcode
CookieCutter	Struktur des GitHub Repositories.	Quellcode