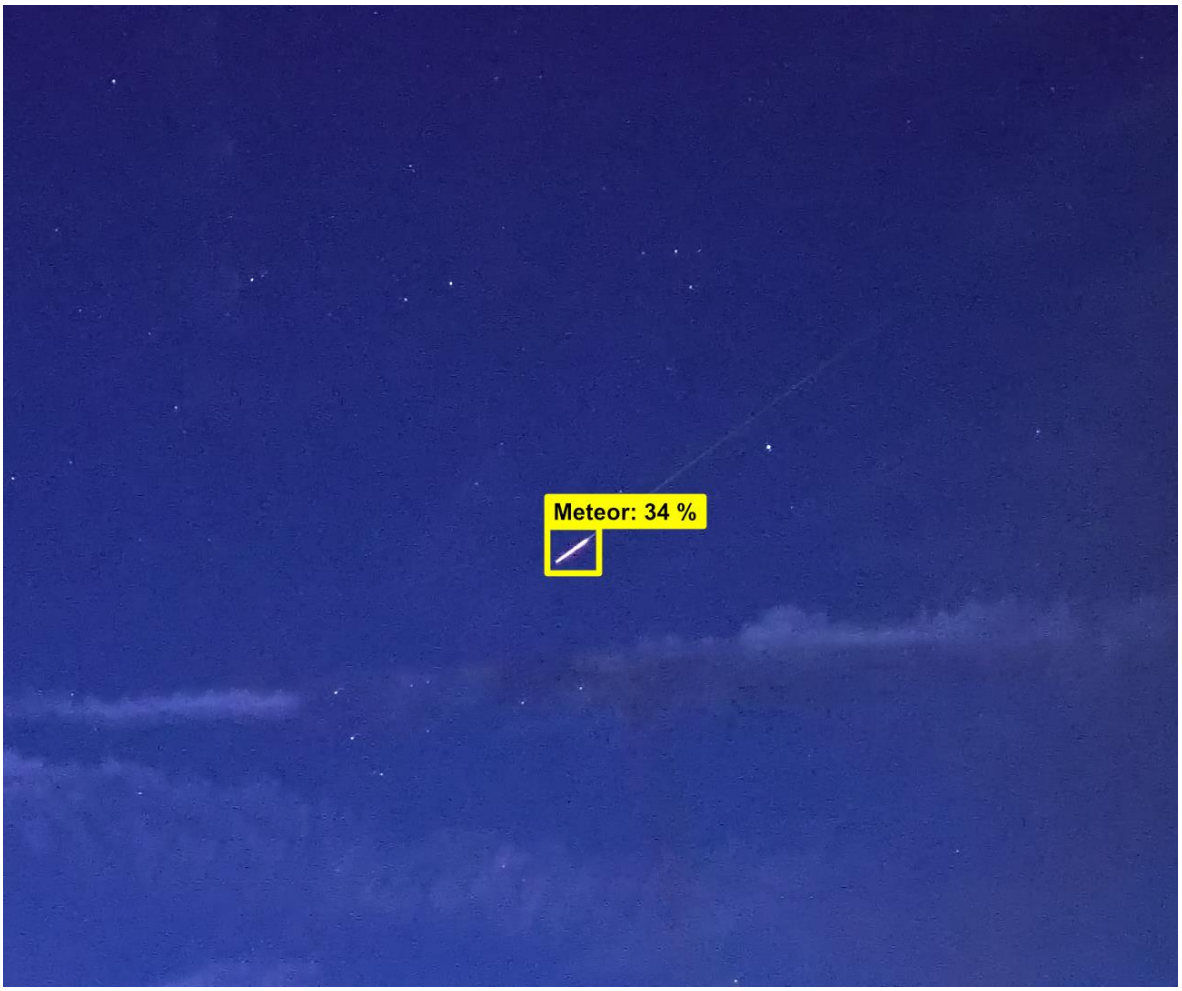
VAMOS

Entwicklung einer Software für Meteor-Erkennung
mit Künstlicher Intelligenz



Linus Sorg

Jugend forscht 2023

jugend forscht

VAMOS – Entwicklung einer Software für Meteor-Erkennung mit Künstlicher Intelligenz

Linus Sorg (16)

Progymnasium Rosenfeld, Schulstraße 9, 72348 Rosenfeld

Projektbetreuer:	Herr Dipl.-Phys. Till Credner
Thema des Projekts:	VAMOS – Entwicklung einer Software für Meteor-Erkennung mit Künstlicher Intelligenz
Fachgebiet:	Mathematik / Informatik
Wettbewerbssparte:	Jugend forscht
Bundesland:	Baden - Württemberg
Wettbewerbsjahr:	2023

Es gibt weltweit viele Kamerasysteme, die den Nachthimmel auf Meteore überwachen. Die Auswertung dieser Videos durch einen Menschen ist nicht nur monoton, sondern auch mit großem Arbeitsaufwand verbunden. Deshalb wurde in den letzten drei Jahren für die Automatisierung dieser Aufgabe die Software “VAMOS” entwickelt, die zuerst mittels Bewegungserkennung und später mit Künstlicher Intelligenz Meteore finden und verschiedene Daten erheben kann. Dabei erreicht das verwendete Neuronale Netzwerk mittlerweile dank verschiedener Optimierungen vor und nach dem Trainingsprozess mehr als 80 % der Leistungsfähigkeit des Auges bei einer starken Reduzierung des menschlichen Arbeitsaufwands. Außerdem konnte das Modell so umstrukturiert werden, dass es auf einem Prozessor ausführbar ist, der speziell für die Arbeit mit Künstlicher Intelligenz ausgelegt ist und deshalb deutlich effizienter arbeitet.

Gliederung der schriftlichen Arbeit

Einleitung

In dieser Jugend-forscht-Arbeit soll unter Einbezug der letzten drei Arbeiten untersucht werden, wie Softwarelösungen dabei helfen können, Meteore in Videos zu erkennen. Die vorangegangenen Arbeiten haben im Fachgebiet Geo- und Raumwissenschaften den Zusammenhang zwischen Radio- und Videosignalen von Meteoren untersucht. Für diesen Zweck mussten viele Stunden an Videomaterial der Meteore ausgewertet werden, wofür in den letzten beiden Jahren unterschiedliche Software – zuletzt eine künstliche Intelligenz – entwickelt und verwendet wurde. Diese Arbeit beschäftigt sich deshalb mit dem technisch-informatischen Teil der bestehenden Arbeiten und zeigt eine Weiterentwicklung und Verbesserung des verwendeten Neuronalen Netzes. Dementsprechend lautet die Leitfrage:

Welche Software-Lösung ist am besten dazu geeignet, Meteore in Videos zuverlässig und schnell zu erkennen?

Insbesondere geht es auch darum, wie man ein Neuronales Netzwerk zur Meteorerkennung so optimieren kann, dass es eine möglichst hohe Genauigkeit kombiniert mit einer möglichst großen Schnelligkeit aufweist. Interessant ist zusätzlich, ob man diese Künstliche Intelligenz auf spezialisierter Hardware effizienter und schneller ausführen kann, sodass die Erkennung letztendlich weniger Energie verbraucht und kompakter ist.

Bei den letzten drei Jugend-forscht-Arbeiten war mein Mitstreiter Till Eissler ebenfalls involviert. Da es in dieser Arbeit jedoch um Informatik und nicht um Geo- und Raumwissenschaften geht, hat er entschieden, sich nicht zu beteiligen.

Vorgehensweise, Materialien und Methoden

Videotechnik

Die aufgenommenen Videos zeigen den Perseidenstrom der Jahre 2019 – 2022 und wurden an unterschiedlichen Orten während dem jeweiligen Meteorhöhepunkt im August aufgenommen. Als Kamera wurde immer eine SONY Alpha 7s II (Abb. 1) verwendet, da diese sehr lichtstark ist und auch bei den kurzen Video-Belichtungszeiten (1/25 s) ein verhältnismäßig rauscharmes Bild aufnimmt. Es wurden unterschiedliche Objektive genutzt, die mit einer weit geöffneten Blende zwischen f/1,4 und f/1,3 ebenfalls dazu beitragen, ein möglichst rauscharmes Bild zu produzieren. Die Videos wurden mit einer Auflösung von 4K Ultra HD (3840 x 2160 px) und einer Bildwiederholrate von 25 FPS aufgenommen.

Abb. 1: SONY Alpha 7s II (Abgebildetes Objektiv ist nicht das verwendete Objektiv)
Foto: Linus Sorg

Visuelle Auswertung der Videos durch einen Menschen

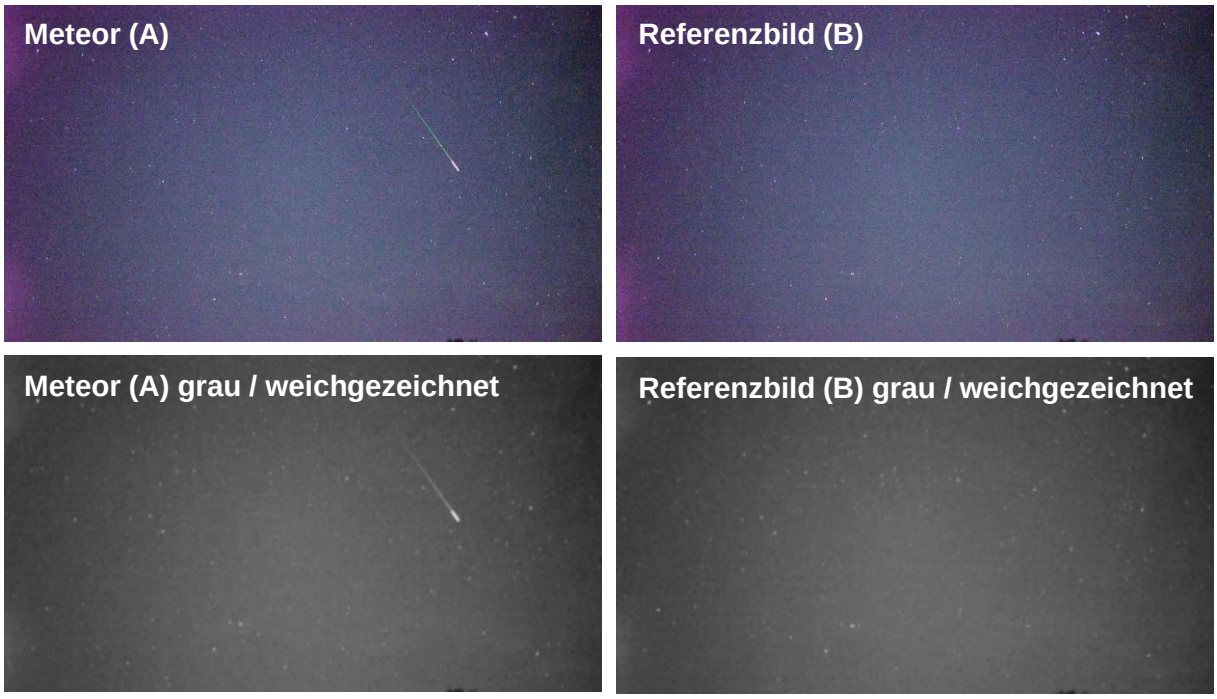
Im ersten Forschungsjahr wurden die Videos noch visuell am Bildschirm von Till Eissler und mir ausgewertet. Dadurch, dass für jeden gefundenen Meteor ein Eintrag in einer Excel-Tabelle erstellt werden musste, dauerte die Auswertung ca. 3-mal so lange wie die eigentliche Laufzeit des Videos.

Entwicklung einer Software für Meteor-Analyse durch Bewegungserkennung

Abb. 2: Das Logo von VAMOS
Grafik: Linus Sorg

Aufgrund des hohen Arbeitsaufwands und einer fehlenden Skalierbarkeit der menschlichen Auswertung wurde im zweiten Forschungsjahr die Software „VAMOS“ („Video-Assisted Meteor Observation System, Abb. 2) mit der Programmiersprache Python entwickelt, die mithilfe eines Differenz-Algorithmus Bewegungen im Video erkennen und dann anhand verschiedener Kriterien klassifizieren kann, ob es sich um einen Meteor handelt. Als Bibliothek für die Bild- und Videoverarbeitung wurde „OpenCV“[1] (Open Computer Vision) verwendet, was als Python-Paket z.B. das Auslesen von Einzelbildern aus Videos, die Subtraktion zweier Bilder voneinander oder das Hinzufügen von Text oder anderen Grafikelementen zu Bildern ermöglicht.

Zur Meteorerkennung wird von jedem Einzelbild ein Referenzbild subtrahiert. Dieses Referenzbild stammt aus demselben Video, allerdings von einigen Sekunden vor dem Meteor, sodass es den Himmel zeigt, wie er ohne Meteor aussieht. Gibt es einen neuen Meteor, dann war er vorher auf dem Referenzbild nicht enthalten und wird als Änderung der Helligkeit einzelner Pixel registriert. Um das Bildrauschen nicht als Erkennung zu erhalten, werden beide Bilder vor der Subtraktion noch zu Graustufen konvertiert und leicht weichgezeichnet. Nach der Differenzbildung erzeugt VAMOS ein „binäres“ Bild, also ein Bild, indem es aufgrund einer Schwellwertbildung nur die Farbe Schwarz und Weiß gibt, jedoch keine Graustufen dazwischen. Dieser Schwellwert liegt standardmäßig bei 20 Zählwerten (von 256 Graustufen, 8-bit). Im binären Bild kann man dann die einzelnen weißen Bereiche erkennen und deren Position und Fläche bestimmen (Abb. 3).



[1] https://opencv.org/

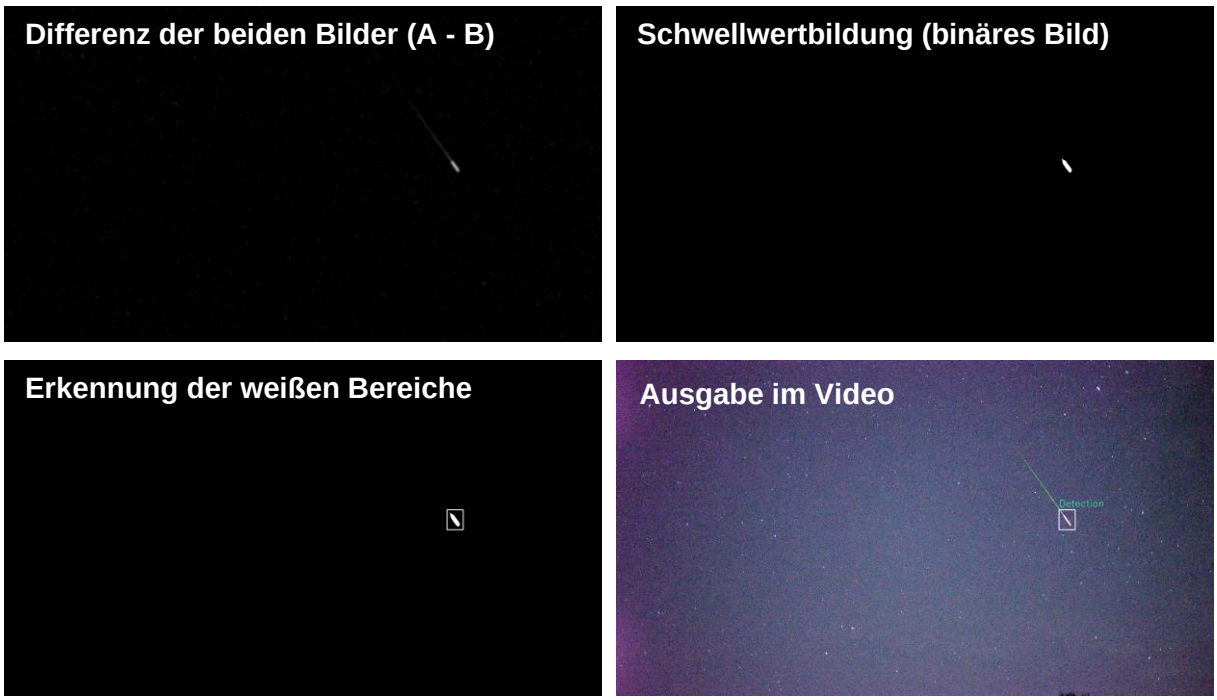


Abb. 3: Prozess zur Erkennung eines Meteors anhand eines Referenzbildes
Grafik: Linus Sorg

Diese Daten werden dann in einer Datenbank abgespeichert und der Prozess so lange wiederholt, bis alle Einzelbilder des Videos analysiert wurden.

Immer, wenn die Anzahl der Detektionen im Einzelbild einen bestimmten Schwellwert – in diesem Fall fünf erkannte Bereiche – überschreitet, wird das aktuelle Einzelbild als neues Referenzbild gesetzt. Dies verhindert beispielsweise, dass ein plötzlicher Kameraschwenk oder der sich langsam bewegende Sternenhimmel unerwünschte Erkennungen auslöst. Die Wahrscheinlichkeit, dass ein Meteor auf diesem Referenzbild zu sehen ist, besteht zwar, stellt jedoch keine Gefahr dar. Dies liegt daran, dass aufgrund der Subtraktion des Referenzbildes vom potenziellen Meteor-Bild nur die Bereiche zum Vorschein treten, die im Meteorbild heller sind als im Referenzbild – und nicht umgekehrt.

Algorithmus zur Nachverarbeitung der Signale

Ist die zeit- und rechenaufwendige Analyse eines Videos abgeschlossen, hat die Datenbank folgende Struktur:

```
{
        "signal_1": {
                "VideoID": "V-0001",
                "box_coordinates": [1146, 1999, 1197, 2038],
                "frame": [257],
                "area": 1989,
                "rotation": 69
        },
        "signal_2": {
                "VideoID": "V-0001",
                "box_coordinates": [1110, 2002, 1157, 2036],
                "frame": [398],
                "area": 1598,
                "rotation": 76
        },
        …
}
```

Diese Signale stellen immer eine einzelne Erkennung auf einem Einzelbild dar und müssen nun verschiedenen Meteoren zugeordnet werden. Hierzu wird jedes Signal einem anderen zugeordnet, falls die Positionen ähnlich sind und die Einzelbilder zeitlich gesehen nah beieinander liegen. Die Schwellwerte sind in der Software manuell anpassbar, für dieses Projekt wurde als maximaler räumlicher Abstand 200 px (gilt für 4K-Auflösung, für Full HD sind es 100 px) und als maximaler zeitlicher Abstand 0,8 s verwendet. Fällt ein Signal in diese beiden Schwellwert-Bereiche, werden dem bestehenden Meteor in der Datenbank die Nummer des Einzelbilds und die Box-Koordinaten hinzugefügt. Außerdem wird die Fläche der Box des Signals zur neuen Fläche des Meteor-Eintrags, sofern sie größer als die Bestehende ist.

Um statische Objekte wie sehr helle Sterne auszusortieren, die manchmal fälschlicherweise auch erkannt werden, wird der oben beschriebene Algorithmus zusätzlich nochmal mit einem anderen Schwellenwert als maximale zeitliche Differenz angewendet. Ist das aktuelle Signal nämlich weniger als 8 s, jedoch mehr als 0,8 s von einem bestehenden potenziellen Meteor entfernt, muss davon ausgegangen werden, dass es sich nicht um einen Meteor handelt, und der Eintrag wird für eine spätere Entfernung markiert. Auf diese Weise können im Laufe des Videos noch weitere Einzelbilder dem Meteor zugeordnet werden, die dann später zuverlässig entfernt werden.

Sind alle Meteore verarbeitet, berechnet die Software den Mittelpunkt aller Box-Koordinaten und berechnet so eine einzelne Koordinate, die die Position des Meteors bestmöglich beschreibt.

Nach der Verarbeitung hat die neue Meteor-Datenbank folgende Struktur:

```
{
	"M-0000001": {
		"VideoID": "V-0030",
		"box_coordinates": [
			[2713, 128, 2732, 152],
			[2722, 165, 2742, 197],
			[2735, 199, 2755, 232],
			[2753, 233, 2776, 268],
			[2770, 273, 2791, 308],
			[2786, 310, 2806, 345],
			[2799, 349, 2826, 387],
			[2818, 384, 2835, 416],
			[2832, 414, 2843, 443]
		],
		"frames": [26821, 26822, 26823, 26824, 26825, 26826, 26827, 26828, 26829],
		"area": 1026,
		"position": [2779, 289],
		"duration": 9,
		"beginning": [[1, 25, 18, 840000], [0, 17, 52, 840000]],
		"end": [[1, 25, 19, 200000], [0, 17, 53, 200000]],
		"date": [2021, 8, 13],
	},
	…
}
```

Entwicklung einer Benutzeroberfläche

Um VAMOS auch für Menschen benutzbar zu machen, die sich nicht mit Python-Programmierung auskennen, wurde eine Benutzeroberfläche entwickelt. Nachdem bei ersten Tests mit dem Python-Framework tkinter viele Limitationen auftraten, wurde Qt bzw. PyQt5 als Python-Version von Qt ausgesucht, was zwar komplexer aufgebaut ist, allerdings auch mehr Anpassungen des Layouts und des Designs ermöglicht. Öffnet man VAMOS, kann man ein kleines Fenster sehen, indem alle Optionen für die Analyse dargestellt werden (Abb. 4).

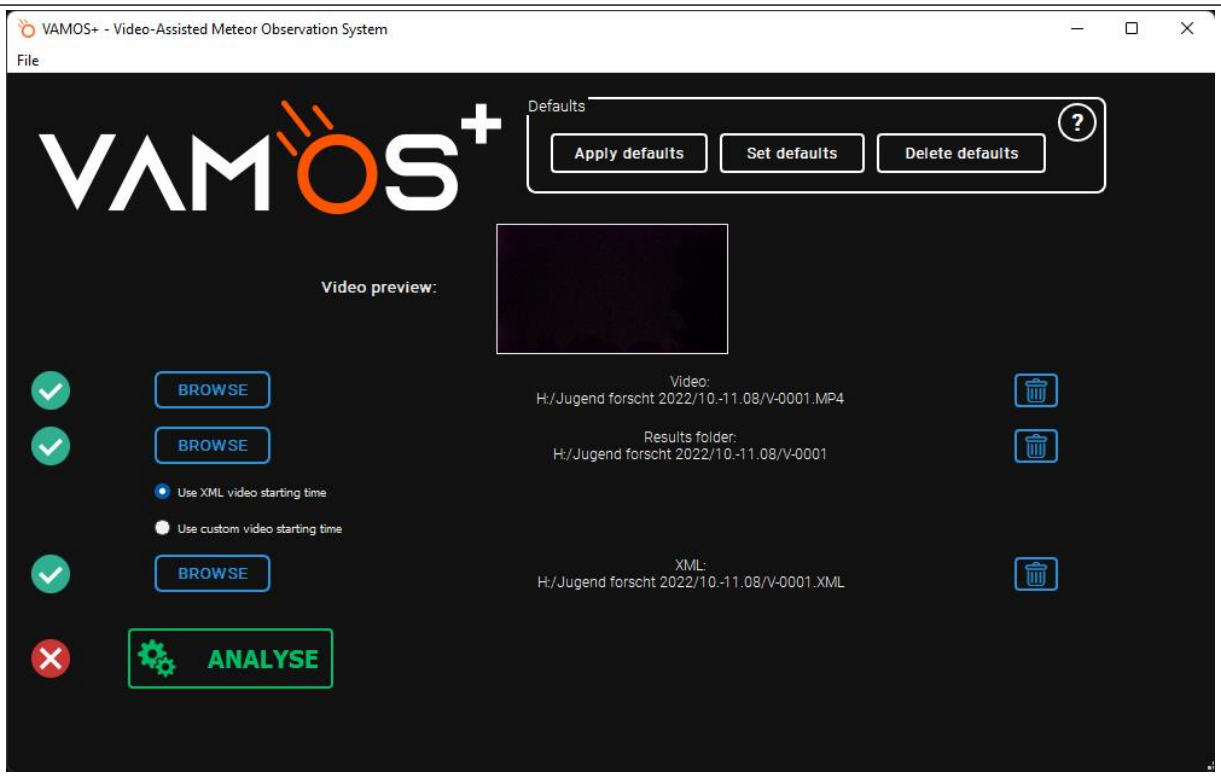


Abb. 4: Hauptoberfläche von VAMOS
Grafik: Linus Sorg

Über den obersten „Browse“-Buttons kann man als Benutzer das Video auswählen, welches man analysieren möchte. Um Dateien während der Analyse abspeichern zu können, benötigt VAMOS auch noch eine Ordner-Auswahl, die man mit dem mittleren Browse-Button tätigen kann. Zuletzt kann man Datum und Uhrzeit des Videobeginns festlegen, indem man entweder eine XML-Datei der Kamera hinterlegt oder die Daten manuell eingibt. Möchte man noch Analyse-Parameter anpassen, kann man das in den Einstellungen unter „File“ → „Settings“ tun. Folgende Standardwerte sind hinterlegt (Tab. 1):

Name	Beschreibung	Wert
Blur	Radius der Weichzeichnung (siehe Abb. 3)	20 px
Threshold	Schwellwert für „binäres“ Bild (siehe Abb. 3)	20
Min. area	Schwellwert für die minimale Fläche einer Meteor-Erkennung	20 px
Max. area	Schwellwert für die maximale Fläche einer Meteor-Erkennung	500 px
Min. length	Schwellwert für die minimale Dauer einer Meteor-Erkennung	0,04 s
Max. length	Schwellwert für die maximale Dauer einer Meteor-Erkennung	10 s
Max. distance	Schwellwert für den maximalen Abstand zweier Signale, um dem gleichen Meteor zugeordnet zu werden	100 px
Min. signal difference	Schwellwert für die minimale Flächendifferenz zweier Signale	80 px

Tab. 1: Standardwerte für die Analyse in VAMOS
Grafik: Linus Sorg

Über den großen „ANALYSE“-Button kann man die Analyse dann starten, worauf sich ein neues Fenster öffnet, welches das Video und darin erkannte potenzielle Meteore anzeigt (Abb. 5):

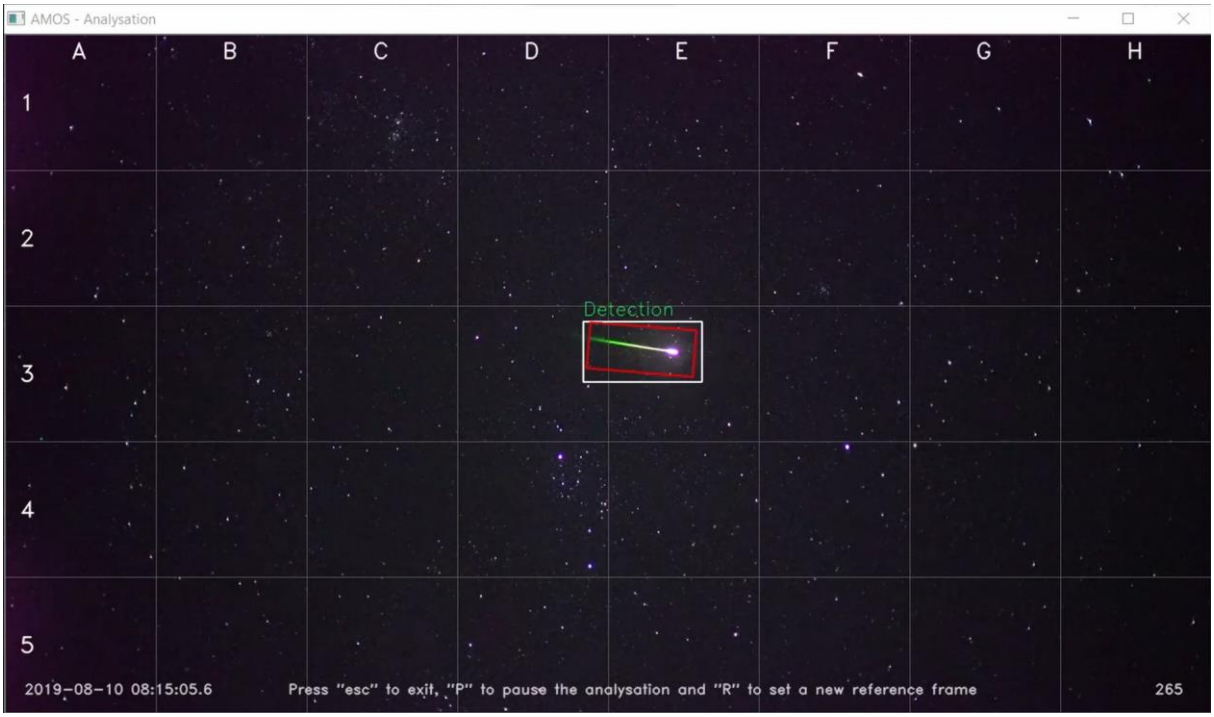


Abb. 5: Analyse-Fenster von VAMOS
Grafik: Linus Sorg

Ist die Analyse des Videos abgeschlossen, kann man die Ergebnisse exportieren, was über das Menü „File“ → „Save“ geschieht. Dabei läuft der im vorigen Abschnitt beschriebene Algorithmus über die Ergebnisse und schreibt eine „*.vamos“-Datei auf die Festplatte. Diese enthält die Meteor-Datenbank, die Dateipfade zum Video, der XML-Datei und dem Ordner und die FPS-Anzahl, Länge, Startuhrzeit und Auflösung des Videos. Das bedeutet, man kann sie jederzeit wieder in VAMOS öffnen, z.B. zum Exportieren in eine Tabellenkalkulation wie Microsoft Excel oder zur Durchsicht und Kontrolle der Meteordaten.

Dies kann man über „File“ → „Open“ tun und sieht anschließend die Ergebnisse in einem separaten Fenster (Abb. 6):

Abb. 6: Ergebnis-Fenster von VAMOS
Grafik: Linus Sorg

Der Bildschirm ist hier aufgeteilt in Details zum Projekt (1), Details zu den Videos (2), eine Videovorschau (3) und die Meteortabelle (4).

Im Bereich 1 gibt es neben einer Anzeige zum ausgewählten Projekt-Ordner zwei Buttons: Mit dem „Generate Spreadsheet“-Button kann man die ggf. auch modifizierte Meteortabelle jederzeit in ein Excel-Dokument exportieren. Mit „Save File“ kann man die VAMOS-Datei nach Änderungen wieder abspeichern. Im Bereich 2 sieht man alle analysierten Videos samt Dateipfad, zugehöriger XML-Datei, Anzahl aller Einzelbilder, FPS, Auflösung und Datum / Uhrzeit des Videobeginns. Wenn man auf ein Video klickt, wird es in Bereich 3 abgespielt und kann bei Bedarf mit den unteren Tasten pausiert oder stummgeschaltet werden. Um die erkannten Meteore zu überprüfen, werden diese in einer Tabelle in Bereich 4 dargestellt. Dabei gibt es folgende Informationen:

• MeteorID: „Name“ des Meteors von M-0000001 bis M-9999999

• VideoID: Entspricht dem Namen des Videos, von „V-0001“ bis „V-9999“

• Position: Position der Mitte des Meteors in Pixeln von der oberen linken Ecke des Bilds aus

• Frames: Einzelbilder, auf denen der Meteor erkannt wurde

• Beginning CET: Uhrzeit in MEZ, bei der der Meteor erstmalig erkannt wurde

• End CET: Uhrzeit in MEZ, bei der der Meteor zuletzt erkannt wurde

• Beginning Video: Zeit im Video, bei der der Meteor erstmalig erkannt wurde

• End Video: Zeit im Video, bei der der Meteor zuletzt erkannt wurde

• Duration: Dauer des Meteors in Einzelbildern

• max. Area: Größte erkannte Fläche des Meteors in Pixeln

• Date: Datum, bei dem der Meteor erstmalig erkannt wurde

Indem man auf einen der Meteore in der Tabelle klickt, springt das Video in Bereich 3 zu der Stelle, an der er auftaucht, sodass man die Meteore sehr einfach und schnell überprüfen kann. Zusätzlich wird auch ein Rechteck auf das Video gezeichnet, an der Stelle, an der der Meteor zu finden ist. Dies hilft dabei, auch schwache Meteore bei der Kontrolle schnell mit dem Auge zu erfassen.

Hat man eine oder mehrere Fehlerkennungen gefunden, kann man sie löschen, indem man auf „Delete Selected“ im Kontextmenü klickt. Sorgt ein statisches Objekt – wie z.B. ein sehr heller Stern – für viele Fehlerkennungen, kann man auch „Delete Similar“ verwenden, was alle Einträge löscht, die eine bis auf 75 px entfernte Position und eine bis auf 100 px ähnliche Fläche zum ausgewählten Meteor haben.

Verbesserung der Erkennung durch eine Künstliche Intelligenz

Eines der größten Probleme von VAMOS war jedoch die fehlende Zuverlässigkeit und die Tatsache, dass oft Satelliten, Flugzeuge, Sterne und sogar Wolken und Rauch fälschlicherweise als Meteore identifiziert wurden. Stattdessen sollte die Software Meteore anhand von Mustern erkennen, die einen solchen typischerweise ausmachen. Die Lösung dieses Problems liegt im Bereich der Künstlichen Intelligenz, die auf die Erkennung bestimmter Objekte trainiert werden und dann auch in neuem Material aufgrund der gelernten Zusammenhänge die Objekte erkennen kann. Das hier verwendete Modell ist ein Konvolutionelles Neuronales Netzwerk (engl. CNN für „Convolutional Neural Network“), welches mit dem TensorFlow Object Detection API[2] von Google trainiert wurde. Es handelt sich nicht um ein Modell zur Klassifikation, sondern zur Objekterkennung, sodass es neben der Klassifikation als Meteor auch möglich ist, die Position und Größe im Bild festzustellen.

[2] https://github.com/tensorflow/models/tree/master/research/object_detection

Für das Training des Neuronalen Netzwerks wurden Einzelbilder aus den Videos der Vorjahre verwendet, die Meteore beinhalten. Nach einem ersten erfolgreichen Test mit ca. 500 4K-Bildern wurde der Datensatz auf insgesamt ca. 1800 4K-Bilder erweitert, bei dem auf jedem die Position des Meteors im Bild von Hand mit dem Open-Source-Tool „LabelImg“[3] markiert wurde. Alle gesammelten Bilder waren Positiv-Samples, also Bilder mit Meteor. Negativ-Samples, also Bilder ohne Meteor, waren nicht nötig, da diese nur bei der Klassifikation benötigt werden. 90 % der Bilder werden zum Trainingsdatensatz, welcher für das Training des Modells verwendet wird, die restlichen 10 % bilden den Testdatensatz und dienen im Anschluss zur Evaluierung des Modells.

Das Modell wurde nicht von Grund auf neu trainiert, was bei Bilderkennungssoftware selbst auf leistungsstarken Servern Tage bis Wochen dauern kann. Stattdessen kam die Technik des „Transfer Learning“ zum Einsatz. Dabei wird bei einem bestehenden Modell nur die letzte Ebene des Neuronalen Netzes – die sich um die letztendliche Klassifikation als Meteor kümmert – neu trainiert. Alle anderen Ebenen funktionieren bspw. als Kantenerkennung und werden auch bei der Meteorerkennung unverändert benötigt. TensorFlow bietet im „TensorFlow 2 Detection Model Zoo“[4] verschiedene Modelle an, die zum Transfer Learning verwendet werden können. Aufgrund eines geeigneten Kompromisses zwischen Präzision und Geschwindigkeit wurde das Modell „SSD MobileNet V2 FPNLite 640x640“ ausgesucht, welches auf dem „COCO 2017“-Datensatz[5] trainiert wurde.

Mit folgenden Parametern wurde das Modell trainiert (Tab. 2):

Trainingsdatensatz	1632 4K-Bilder
Testdatensatz	182 4K-Bilder
Input-Größe	640 x 640 px
Learning Rate	0,01
Batch Size	2
Trainingsschritte	100.000

Tab. 2: Trainingsparameter des Neuronalen Netzwerks
Grafik: Linus Sorg

Mit dem Tool „TensorBoard“[6] kann der Fortschritt im TensorFlow-Trainingsprozess verfolgt werden, vor allem die Metriken zu „Loss“, „Precision“ und „Recall“. So kann man unter anderem verhindern, dass ein „Overfitting“ stattfindet, was bedeutet, dass das Neuronale Netz die Trainingsdaten so oft wiederholt hat, dass es sich die Bilder „merkt“ – und nicht die zu Grunde liegenden Muster, die einen Meteor identifizieren.

Hat man das Training eines Modells abgeschlossen, kann man es als „TensorFlow SavedModel“[7] exportieren. Diese Datei kann man anschließend in der Software verwenden, um das Modell via Tensorflow zu laden und die Einzelbilder des Videos zu analysieren. Das bedeutet, dass der Ablauf und Aufbau von VAMOS erhalten bleibt und ausschließlich der Differenz-Algorithmus durch das trainierte Neuronale Netz ersetzt wird. Auf Grund dieser Optimierung wurde die neue Software „VAMOS+“ genannt.

Für jedes analysierte Einzelbild gibt Tensorflow folgendes aus:

---

3 https://github.com/heartexlabs/labelImg

4 https://www.github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

5 https://cocodataset.org

6 https://www.tensorflow.org/tensorboard

7 https://www.tensorflow.org/guide/saved_model

```
{
        „detection_boxes“: [[0.5160102, 0.6480539, 0.5280073, 0.6536226], [0.88723135,
                0.54895407, 0.90596664, 0.5514361],],
        „detection_scores“: [0.30699536, 0.22868103],
        „detection_classes“: [1, 1],
        „num_detections“: 2,
}
```


Mit diesen Daten kann VAMOS+ dann genauso verfahren wie VAMOS. Nach einer Filterung für den „Score“ einer Erkennung werden überlappende Boxen erkannt und zu einer einzelnen kombiniert. Nach der Analyse läuft derselbe Nachverarbeitungsalgorithmus über die Daten, der auch schon beim Differenz-Algorithmus angewendet wurde.

Optimierung des Neuronalen Netzwerks

Nach dem Wettbewerb 2022 wurde das Neuronale Netzwerk noch deutlich verbessert. Dabei wurde die Anzahl der Trainingsschritte von 15.000 auf 100.000 angehoben und verschiedene Parameter beim Training variiert. Ein wichtiger Parameter ist hier die sog. „Learning Rate“, die angibt, wie stark die Gewichte an den Neuronen im Neuronalen Netzwerk pro Schritt angepasst werden dürfen. Ist diese zu niedrig gewählt, macht der Trainingsprozess kaum Fortschritte und dauert sehr lange. Ist sie jedoch zu hoch, verändert sich die Leistungsfähigkeit des Modells ständig und bleibt instabil. Um herauszufinden, welche Learning Rate am besten geeignet ist, wurde ein Modell mit exponentiellem Verlauf der Learning Rate trainiert (Abb. 7):

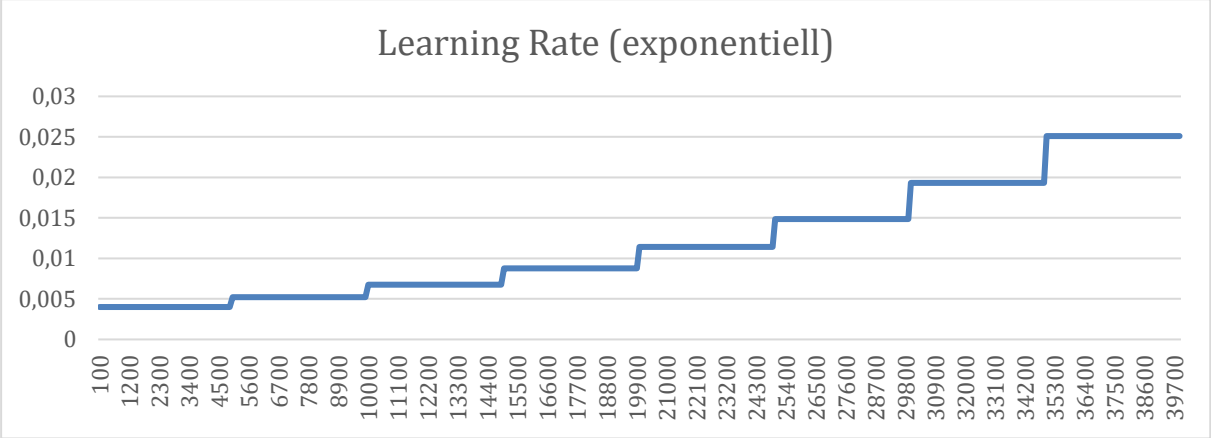


Abb. 7: Verlauf der Learning Rate bei einer in Abständen von 5000 Trainingsschritten exponentiell zunehmenden Kurve, die dazu dient, einen optimalen Wert für die Learning Rate zu finden
Grafik: Linus Sorg

Daraus ergab sich folgendes Diagramm zur Präzision (Abb. 8):

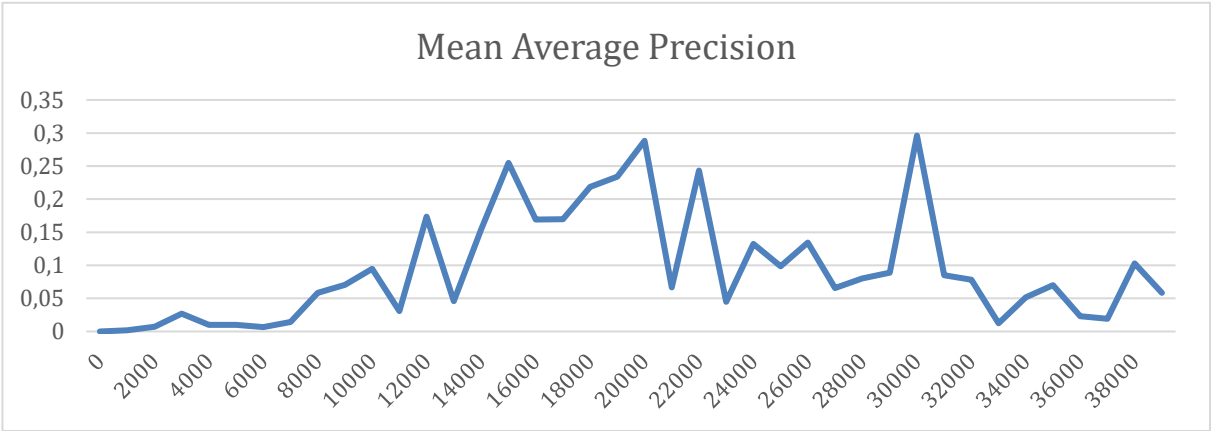


Abb. 8: Präzision des Modells, das mit der exponentiell zunehmenden Learning Rate (s. oben) trainiert wurde
Grafik: Linus Sorg

Wie man sehen kann, verläuft der Trainingsprozess bis zum Schritt 20.000 gut und die Präzision wird ständig gesteigert. Als jedoch die Learning Rate danach erhöht wird, nimmt die Präzision stark ab. Aus diesem Grund wurde eine Learning Rate ausgewählt, die sich zwischen Schritt 1 und Schritt 20.000 linear von 0,004 bis 0,01 steigert und dann in Form einer Kosinus-Kurve bis zum Ende des Trainingsprozesses auf 0 sinkt. Diese Senkung ist sinnvoll, da nach anfänglichen größeren Anpassungen nur noch feine Veränderungen notwendig sind, um die Leistungsfähigkeit des Modells weiter zu verbessern. Der genaue Verlauf kann so veranschaulicht werden (Abb. 9):

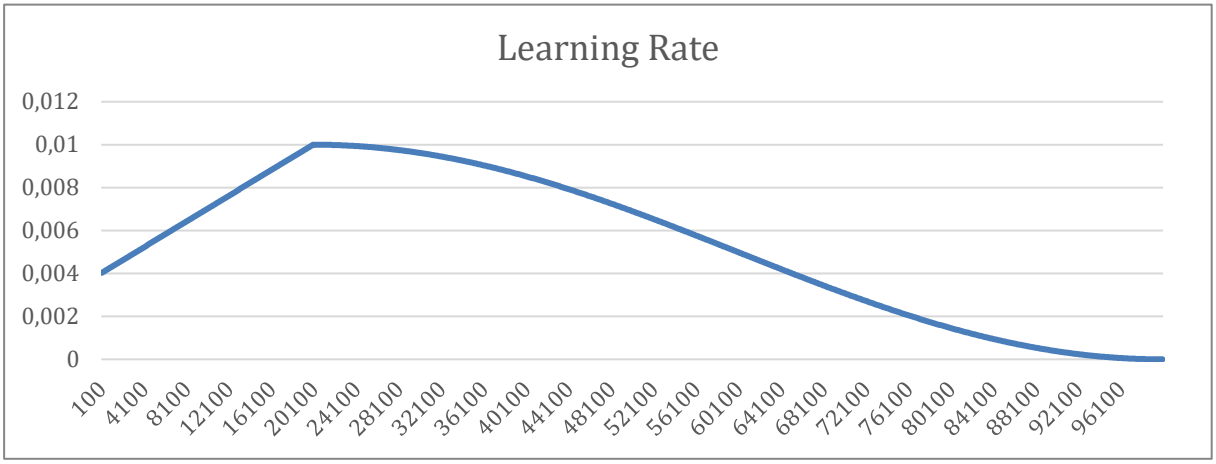


Abb. 9: Visualisierung des optimierten Learning-Rate-Verlaufs, der erst linear zunimmt und dann in einer Kosinus-Kurve abnimmt
Grafik: Linus Sorg

Durch diese Maßnahmen konnte die Leistungsfähigkeit des Modells seit dem Landeswettbewerb Jugend forscht im April 2022 etwa verdreifacht werden. Dies spiegelt sich nicht nur in der Zahl gefundener Meteore wider, sondern zu großen Teilen auch in der Rate der falsch positiven Erkennungen, die stark gesunken ist.

Ausführung auf einer Edge-TPU

Traditionelle Prozessorarchitekturen sind nicht dazu optimiert, Neuronale Netze oder andere KI-Modelle auszuführen. Deshalb werden hierzu häufig leistungsstarke Grafikkarten verwendet, die eine deutlich höhere Geschwindigkeit ermöglichen. Doch auch diese GPUs sind eigentlich für etwas anderes optimiert: die Arbeit mit Bildern, Videos oder anderer Grafikarbeit. Deshalb entwickelte Google speziell für den Zweck des Maschinellen Lernens die sog. „**T**ensor **P**rocessing **U**nit“ kurz TPU. Die Prozessorarchitektur dieser TPUs ist optimiert für die Ausführung von TensorFlow-Modellen und eignet sich deshalb auch besonders gut für dieses Projekt. TPUs erlauben nämlich eine Steigerung der Leistungsfähigkeit bei gleichbleibendem Stromverbrauch bzw. eine Senkung des Stromverbrauchs bei gleichbleibender Leistung. Dies macht es möglich, auf immer kleineren Geräten immer komplexere KI-Modelle auszuführen. Google selbst bietet mit der Tochterfirma „Coral“ Geräte an, die mit einem TPU-Chip ausgestattet sind und die sowohl für Prototypen als auch für marktreife Produkte verwendet werden können. Für dieses Projekt wurde das „Coral Dev-Board“[8] (Abb. 10) verwendet, ein Einplatinencomputer mit einer

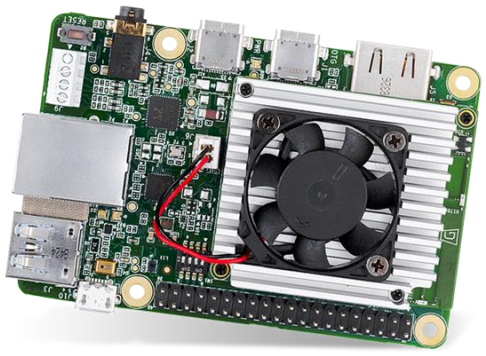


Abb. 10: Das Coral Dev-Board mit Edge-TPU
Foto: Google LLC

[8] https://coral.ai/products/dev-board

Edge-TPU[9] samt Kühlung, der in etwa die Dimensionen eines Raspberry Pi hat. Damit das trainierte Modell auf der Edge-TPU ausgeführt werden kann, muss es jedoch konvertiert werden. Ähnlich wie auf mobilen Geräten wird hier nämlich das Dateiformat „TensorFlow Lite“[10] verwendet, welches die Größe und Komplexität des Modells reduziert und dabei etwas an Präzision verliert. Zusätzlich muss das Modell auch noch „quantisiert“ und für die spezielle Prozessorarchitektur der Edge-TPU kompiliert werden[11]. Alle diese Werkzeuge stellt der Hersteller zur Verfügung, deshalb setzt sich diese Arbeit auch nicht genauer mit den zu Grunde liegenden Mechanismen auseinander.

Sind alle diese Schritte abgeschlossen, kann man das Modell auf das Dev-Board übertragen und dort ausführen. Das installierte Betriebssystem heißt „Mendel Linux“[12] und wurde ebenfalls von Google speziell für diesen Anwendungsfall entwickelt, basiert aber zu großen Teilen auf Debian, einer der bekanntesten Linux-Distributionen.

Nach einigen Tests konnte das Modell dort ausgeführt werden, war mit 90 ms pro Einzelbild allerdings nicht schnell genug für eine Echtzeitauswertung. Ein großes Problem war auch der verbaute USB-C-Anschluss, der die Videodaten nicht schnell genug von der angeschlossenen SSD-Festplatte auslesen konnte. Deshalb kam das Dev-Board bei bisherigen Videoauswertungen noch nicht zum Einsatz, was aber in Zukunft geplant ist. Dabei geht es ganz besonders um eine Live-Auswertung, mehr dazu später im Ausblick. Alle Auswertungen in dieser Arbeit wurden stattdessen auf einem PC mit leistungsstarkem Prozessor und Grafikkarte ausgeführt.

Ergebnisse

Vergleich der verschiedenen Auswerte-Methoden

Um die Genauigkeit und Schnelligkeit verschiedener Auswerte-Methoden zu untersuchen, wurden alle Methoden auf die gleichen zwei Stunden Videomaterial angewendet und anschließend die gefundenen Meteore überprüft und verglichen. Konkret verglichen werden die visuelle Auswertung aus dem ersten, VAMOS aus dem zweiten, VAMOS+ aus dem dritten und das seitdem weiter optimierte Modell aus dem vierten Forschungsjahr. Dabei ergibt sich folgende Verteilung (Abb. 11):

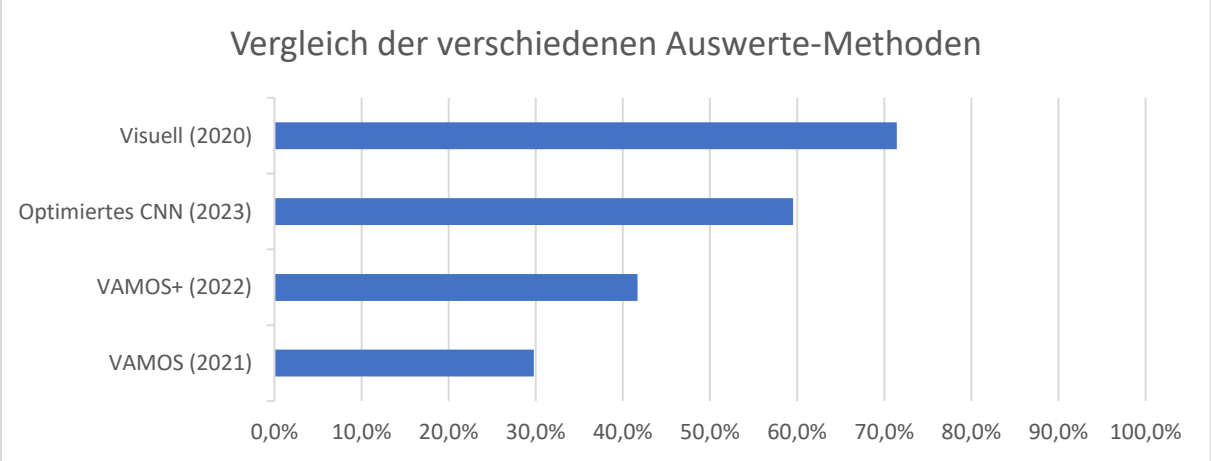


Abb. 11: Vergleich der Auswerte-Methoden: Prozentsatz der gefundenen Meteore im Referenz-Zeitraum (100% stellen alle Meteore dar, die kombiniert von allen Methoden gefunden wurden)
Grafik: Linus Sorg

[9] Der Begriff „Edge“ wird bei Geräten verwendet, die KI-Modelle lokal ausführen können, anstatt eine Verbindung zu einem leistungsstärkeren Server zu benötigen (Quelle: https://www.fierceelectronics.com/electronics/what-edge-machine-learning)

[10] https://www.tensorflow.org/lite

[11] https://coral.ai/docs/edgetpu/models-intro/

[12] https://coral.googlesource.com/docs/+/refs/heads/master/ReadMe.md

Eine der großen Verbesserungen des neuesten Neuronalen Netzwerks gegenüber dem Stand von VAMOS+ ist die Verbesserung der Präzision, was im Bereich der Objekterkennung u.a. als eine Reduktion von Fehlerkennungen definiert ist. Im folgenden Diagramm wird dargestellt, wie viele der erkannten Objekte tatsächlich Meteore sind (Abb. 12):

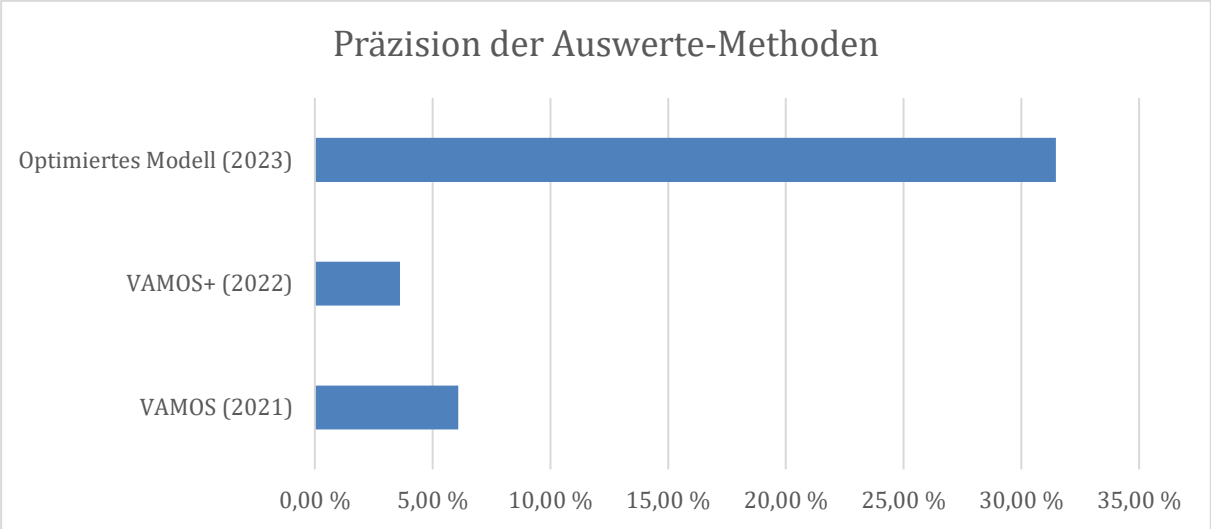


Abb. 12: Vergleich der Präzision von VAMOS, VAMOS+ und dem neuesten und optimierten Modell. Präzision ist definiert als das Verhältnis aller korrekten Meteor-Erkennungen zur Gesamtzahl aller Erkennungen
Grafik: Linus Sorg

Leistungs-Metriken des Neuronalen Netzwerks

Der Recall-Wert eines Neuronalen Netzwerks beschreibt, wie viele der Meteore im Testdatensatz vom Modell tatsächlich gefunden wurden. Im Laufe des Trainings stieg der Wert bis auf 42,0 % an (Abb. 13):

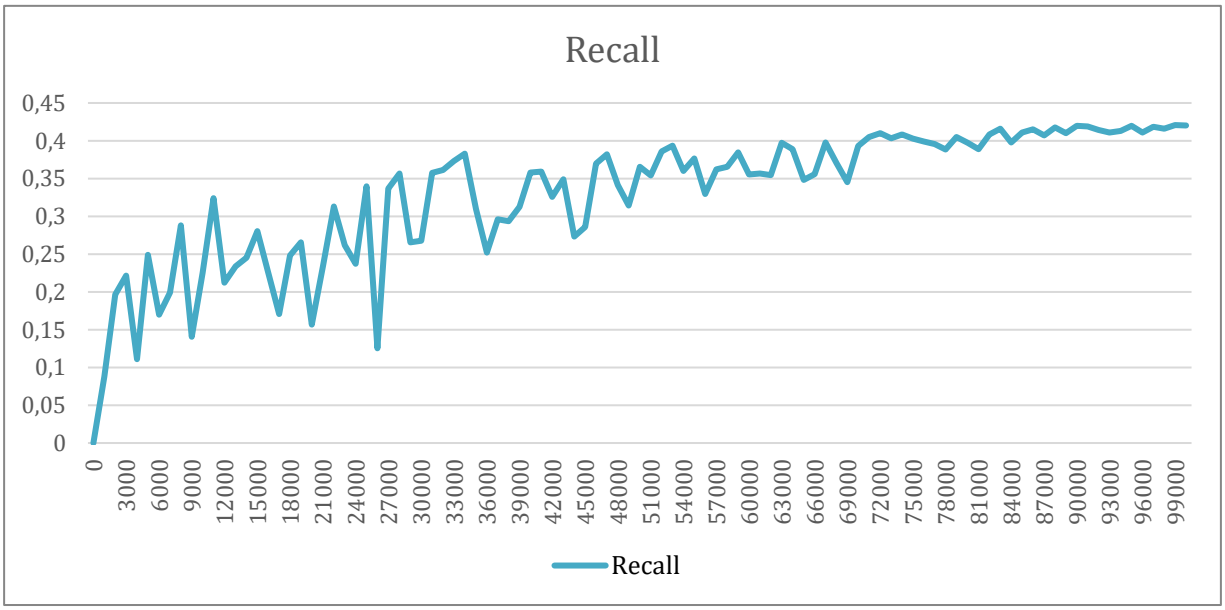


Abb. 13: Verlauf des Recall-Wertes des Neuronalen Netzwerks während des Trainingsprozesses
Grafik: Linus Sorg

Mean Average Precision (kurz: mAP) beschreibt, wie viele der vom Modell gefundenen Objekte auch wirklich Meteore waren. Am Ende des Trainingsprozesses (nach 100.000 Schritten) lag dieser Wert bei ca. 57,4 % (Abb. 14):

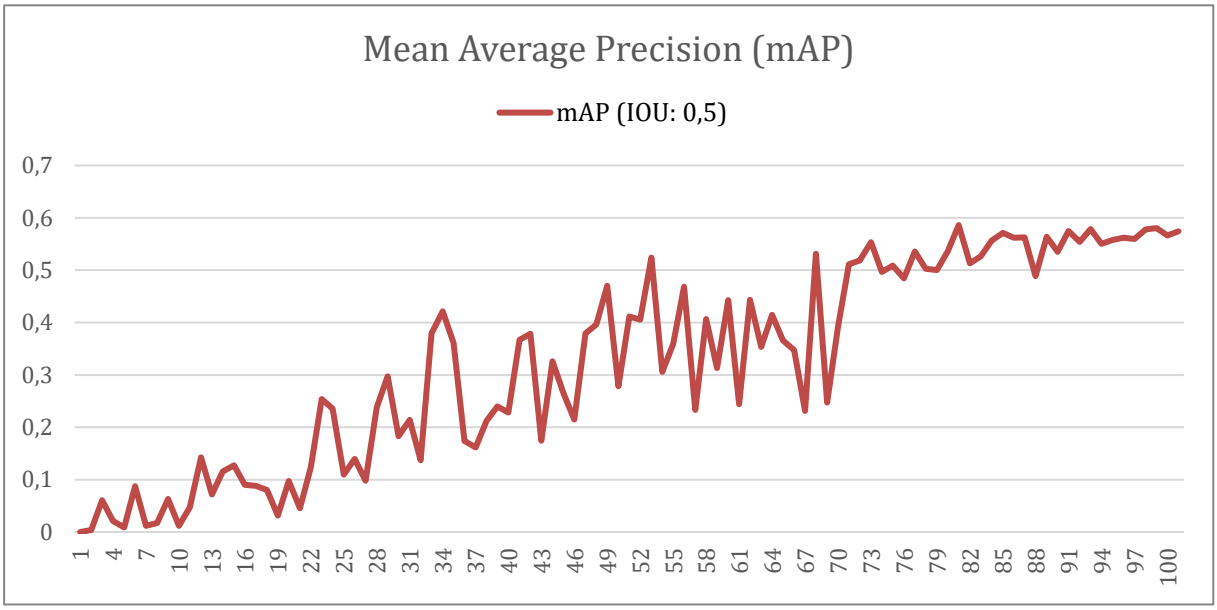


Abb. 14: Verlauf des mAP-Wertes (Mean Average Precision) des Neuronalen Netzwerks während dem Training „IOU" = „Intersection over Union": Schwellwert für die Evaluierung von Objekterkennungs-Modellen
Grafik: Linus Sorg

Ergebnisdiskussion

Der Vergleich der verschiedenen Methoden zur Meteor-Erkennung in Videos hat gezeigt, dass eine Automatisierung dieser Aufgabe möglich ist. Während ein Differenz-Algorithmus zwar Meteore an ihrer Eigenschaft der Bewegung erkennen kann, ergibt sich hier besonders das Problem fehlender Klassifizierungs-Fähigkeiten, was in einer hohen Falsch-Positiv-Rate resultiert. Ebenso kann dieser Algorithmus nicht zuverlässig erkennen, ob es sich um einen Meteor handelt oder nicht und benötigt noch einiges an manueller Arbeit in der Nachverarbeitung der Ergebnisse.

Im Gegensatz dazu zeigt sich, dass ein Konvolutionelles Neuronales Netzwerk für diese Aufgabe besser geeignet ist, da hier eine Klassifizierung möglich ist. Das Machine-Learning-Modell kann mehr Meteore in den Videos erkennen und zeichnet sich auch durch eine deutlich geringere Falsch-Positiv-Rate aus. Um jedoch diese Ergebnisse erzielen zu können, war eine aufwendige Optimierung des Modells nötig, um die Parameter so anzupassen, dass das Modell die grundlegenden Merkmale eines Meteors bestmöglich erkennt und damit auch Meteore in anderen Videos erkennen kann. Das optimierte Neuronale Netzwerk findet bereits 83 % der mit dem Auge erkennbaren Meteore und findet im Vergleich zu VAMOS+ 43 % mehr Meteore, im Vergleich zu VAMOS sogar das Doppelte.

Durch die weitere Optimierung des Neuronalen Netzwerks im letzten Jahr konnte die Häufigkeit von Fehlerkennungen deutlich reduziert werden. Dies liegt vor allem an der Anpassung der Parameter für das Training, aber auch an einer Verbesserung des Algorithmus zur Nachverarbeitung der Signale. Der Anteil der korrekt erkannten Meteore an der Gesamtzahl aller Erkennungen konnte verglichen zu den anderen Methoden verfünf- bzw. verneunfacht werden (siehe Abb. 12).

Betrachtet man die Leistungsfähigkeit des menschlichen Auges bei der Meteor-Erkennung, fällt auf, dass auch dieses manchmal Meteore nicht erkennt, welche eine Software aber erkennen kann. Dabei handelt es sich zwar teilweise um schwache Meteore, oft sind es jedoch auch helle Meteore, die man bei einem zweiten Durchlauf vielleicht sogar sehen würde. Dies liegt daran, dass man sich beim Ansehen der Videos nicht auf den gesamten Bildbereich gleichzeitig fokussieren kann und

so leicht Meteore übersieht, die sich außerhalb dieses vom Auge fokussierten Bereichs befinden. Software hat dieses Problem selbstverständlich nicht, was einen Vorteil gegenüber der menschlichen Arbeit darstellt. Im Referenz-Zeitraum von zwei Stunden (siehe Ergebnisse) hat ein menschlicher Beobachter z.B. 60 Meteore gefunden, allerdings 24 Meteore übersehen, die stattdessen vom Differenz-Algorithmus und / oder dem Neuronalen Netzwerk gefunden wurden.

Ein großer Vorteil der Verwendung einer Software-Lösung ist außerdem die Einsparung von Arbeitsaufwand. Um eine halbe Stunde Videomaterial von Meteoren visuell auszuwerten und dabei verschiedene Daten zu den gefundenen Meteoren zu sammeln, benötigt ein Mensch im Normalfall ca. 90 Minuten. Eine Software wie VAMOS+ kann ein halbstündiges 4K-Video in 30 – 45 Minuten analysieren, wenn ein leistungsstarker Computer verwendet wird. Die anschließende Überprüfung der Ergebnisse und das Löschen der Fehlerkennungen dauert dann in der Regel nur ca. 1 – 2 Minuten, da auch hier viele automatisierte Werkzeuge zur Verfügung stehen. Da die Software die Videos unbeaufsichtigt analysieren kann, beschränkt sich der menschliche Arbeitsaufwand durch eine Software-Lösung lediglich auf diesen letzten Teil.

Ein mögliches Problem wiederum entsteht bei der Skalierung der Video-Einzelbilder auf die Auflösung, die das Neuronale Netzwerk zum Verarbeiten benötigt – in diesem Fall auf 640 x 640 px. Das dabei angewendete „Binning“ – also die Kombination mehrerer Pixel zu einem Einzigen – kann unter Umständen dafür sorgen, dass kleinere und schwächere Meteore verschwinden oder im Bildrauschen untergehen. Auch der Kompressions-Algorithmus der Videokamera kann ein Problem darstellen, da Details möglicher Meteore verloren gehen können. An dieser Stelle muss man einen Kompromiss zwischen der Geschwindigkeit und der Genauigkeit des Modells eingehen. Erhöht man nämlich die Auflösung, mit der das Modell die Bilder verarbeitet, erhöht sich gleichzeitig auch die Verarbeitungszeit. Es wäre beispielsweise möglich, jedes Einzelbild in vier Teile zu teilen (mit einem kleinen Überlapp, um auch Meteore am Rand der Teilbereiche identifizieren zu können) und jeden Teil separat zu analysieren. Durch die vierfache Analyse-Auflösung würden vermutlich mehr Meteore gefunden werden, jedoch würde die Analyse eines Videos damit auch ca. viermal länger dauern.

Auch wenn die Erkennung von Meteoren durch die Edge TPU bisher aus oben genannten Gründen noch nicht zum Einsatz kam, lässt sich trotzdem sagen, dass die Ausführung des Modells auf Hardware möglich ist, die besonders spezialisiert auf die Arbeit mit Künstlicher Intelligenz ist. Der erfolgreiche Testdurchlauf mit 90 ms pro Einzelbild zeigt das Potenzial, zukünftig Meteor-Erkennung auch auf Geräten mit kleinerem Formfaktor – wie zum Beispiel dem Dev-Board – durchführen zu können.

Ausblick

Die Auswertung mit einem Konvolutionellen Neuronalen Netzwerk ist zwar die gängigste Vorgehensweise bei der Analyse von Videodaten, bietet jedoch nur die Möglichkeit, jedes Einzelbild separat zu analysieren. Besser wäre es, bei der Analyse mehrere Einzelbilder auf einmal zu betrachten. Ein Mensch würde in manchen Situationen einen Meteor auch nur daran erkennen, dass er sich über die Einzelbilder hinwegbewegt. Bei der Analyse alleinstehender Einzelbilder ohne Kontext ergibt sich also das Problem, dass unter Umständen Meteore nicht erkannt werden, die man aber unter Einbezug der vorhergehenden Bilder aus dem Video erkennen könnte.

Im Feld der künstlichen Intelligenz gibt es spezialisierte Modelle, die beispielsweise auch bei Chatbots angewendet werden, um im Kontext der vorhergegangenen Konversation antworten zu können. Diese sog. „RNNs“ („**R**ecurrent **N**eural **N**etworks“) bzw. „LSTMs“ („**L**ong **S**hort-**T**erm **M**emory Networks“)[13] können Informationen aus mehreren Einzelbildern verwenden, um eine bessere Leistungsfähigkeit zu erreichen.

[13] LSTMs stellen eine Weiterentwicklung der RNNs dar, indem sie mehr Kontext als RNNs miteinbeziehen können, allerdings benötigen sie dafür auch mehr Rechenleistung, was indirekt eine längere Ausführzeit bedeutet Quelle: https://builtin.com/data-science/recurrent-neural-networks-and-lstm

In Zukunft soll ein KI-Modell trainiert werden, dass die Anwendung eines RNNs oder LSTMs zur Erkennung von Meteoren über mehrere Einzelbilder hinweg implementiert. Anschließend soll es hinsichtlich der Präzision und Geschwindigkeit mit der Implementierung eines CNNs verglichen werden. Besonders im Feld der Meteorerkennung mit RNNs gibt es zum aktuellen Zeitpunkt noch sehr wenige wissenschaftlichen Arbeiten, es handelt sich also um ein wichtiges und interessantes Forschungsthema.

Die vorher beschriebene Verwendung einer Edge-TPU bietet für die Meteor-Erkennung ganz besonders das Potenzial, eine portable, energieeffiziente und schnelle Auswertung von Videomaterial zu ermöglichen. Geplant ist in Zukunft, eine Kamerastation zu entwickeln, die das Bildsignal direkt von der Kamera über eine Capture Card – also ein Gerät, das das HDMI-Signal der Kamera in ein USB-Webcam Signal konvertieren kann – an das Dev-Board weitergibt. Dieses kann in Echtzeit eine Auswertung vornehmen und das Videomaterial auf einer externen SSD-Festplatte abspeichern. Damit wäre es theoretisch möglich, jede Nacht den Himmel auszuwerten und sogar andere Himmelsphänomene zu erfassen. Ein herkömmlicher PC mit Grafikkarte würde zu viel Strom verbrauchen und einen zu großen Formfaktor haben.

Eine Limitation des aktuellen Trainings- und Testdatensatzes ist, dass nicht alle Meteore auf den Bildern gefunden und markiert werden können. Wie es bei jeder Signalerkennung der Fall ist, gibt es auch hier Meteore, die im Bildrauschen untergehen, weil sie zu schwach sind, und Meteore, die von einem menschlichen Beobachter übersehen wurden. Eine mögliche Problemlösung hierfür wäre die Erstellung eines künstlichen Datensatzes, also das Hineinmontieren von Meteoren unterschiedlicher Helligkeiten in bestehende Videos, um die genaue Anzahl aller im Video vorhandenen Meteore zu kennen und dementsprechend auch die Leistungsfähigkeit der Auswerte-Methoden besser beurteilen zu können.

Quellen- und Literaturverzeichnis:

Sorg, Linus; Eissler, Till: Meteore – Video- und Radiobeobachtungen des Perseidenstromes, Schüler experimentieren 2020; https://jugend-forscht.linus-sorg.com/files/Schriftliche-Arbeit-2020.pdf, Letzter Zugriff am 14.01.2023

Sorg, Linus; Eissler, Till: Meteore – Video- und Radiobeobachtungen von Meteorströmen, Jugend forscht 2021; https://jugend-forscht.linus-sorg.com/files/Schriftliche-Arbeit-2021.pdf, Letzter Zugriff am 14.01.2023

Sorg, Linus; Eissler, Till: Meteore – Synchrone Video- und Radiobeobachtungen des Perseidenstromes, Jugend forscht 2022; https://jugend-forscht.linus-sorg.com/files/Schriftliche-Arbeit-2022.pdf, Letzter Zugriff am 14.01.2023

Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K: Speed/accuracy trade-offs for modern convolutional object detectors - CVPR 2017

Towards data science, Sumit Saha: A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way; https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, Letzter Zugriff am 14.01.2023

Github: labelImg; https://github.com/heartexlabs/labelImg, Letzter Zugriff 14.01.2023

Datasolut: Transfer Learning – Grundlagen und Einsatzgebiete; https://datasolut.com/was-ist-transfer-learning/, Letzter Zugriff am 14.01.2023

TensorFlow: Transferlernen und Feinabstimmung; https://www.tensorflow.org/tutorials/images/transfer_learning, Letzter Zugriff am 14.01.2023

Github: Tensorflow Models; https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md, Letzter Zugriff am 14.01.2023

COCO: Common Objects in Context; https://cocodataset.org/#home, Letzter Zugriff am 14.01.2023

Fierce Electronics: What is Edge Machine Learning; https://www.fierceelectronics.com/electronics/what-edge-machine-learning, Letzter Zugriff am 14.01.2023

Sorg, Linus; Eissler, Till: VdS-Journal für Astronomie Nr.84: „Jugend forscht: Synchrone Video- und Radiobeobachtung von Perseiden-Meteoren", in press and accepted 1/2023

Unterstützung durch Personen und Institutionen

Credner, Till: Projektbetreuer, Beratung Schriftliche Arbeit, Unterstützung bei den Videoaufnahmen der Perseiden, Progymnasium Rosenfeld
Eissler, Till: Mitstreiter der letzten drei Jugend-forscht-Projekte
Gaiselmann, Marianne und Sorg, Armin: Beratung Schriftliche Arbeit
Blickle Räder+Rollen Rosenfeld: Teil-Kostenübernahme der verwendeten Geräte und Materialien
Sparkasse Zollernalb: Teil-Kostenübernahme der verwendeten Geräte und Materialien
Verein der Freunde und ehemaligen Schüler des Progymnasiums Rosenfeld e.V.: Teil-Kostenübernahme der verwendeten Geräte und Materialien
Baden-Württemberg Stiftung: Kostenübernahme des Kamerabody der SONY Alpha 7s II
Projekt Sternenpark Schwäbische Alb: Veranstalter des Meteorcamps

VIELEN DANK!