

BACHELOR THESIS
Steffen Lossau

Automatische Generierung von Lichtsteuerdaten aus Videos von Bühnenauftritten

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

Steffen Lossau

Automatische Generierung von Lichtsteuerdaten aus Videos von Bühnenauftritten

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Peer Steldinger
Zweitgutachter: Prof. Dr. Stephan Pareigis

Eingereicht am: 10. September 2024

Steffen Lossau

Thema der Arbeit

Automatische Generierung von Lichtsteuerdaten aus Videos von Bühnenauftritten

Stichworte

Computer Vision, Machine Learning, Licht, Video, Steuerdaten, Bühne, Veranstaltungstechnik

Kurzzusammenfassung

Das Ziel dieser Arbeit ist das Erstellen von Lichtsteuerdaten aus Videos von Bühnenauftritten durch die Verwendung geeigneter Modelle der Computer Vision. Diese Zielsetzung wird durch die Bearbeitung von fünf Kernpunkten erreicht. So werden unterschiedliche Modelle gegeneinander abgegrenzt, damit geeignete Modelle für diese zu erfüllende Aufgabe gefunden werden können. Des Weiteren muss der Datensatz für das Training der Modelle aufbereitet werden. Der Fokus bei dieser Aufbereitung wird auf der Abstraktion der einzelnen Lichtparameter für das Training der Modelle liegen. Aufgrund der Größe des zur Verfügung stehenden Datensatzes, wird eine Anwendung entwickelt, welche die Aufbereitung des Datensatzes automatisch vornimmt. Vor dem Training der einzelnen Modelle werden zudem die unterschiedlichen Trainingsparameter untersucht. Bevor die Trainingsergebnisse der Modelle und die erstellten Lichtsteuerdaten ausgewertet werden, muss eine Anwendung für die Erstellung der Steuerdaten auf Basis der trainierten Modelle erstellt werden. Die generierten Lichtsteuerdaten werden nachfolgend mit den tatsächlichen Lichtsteuerdaten desselben Datensatzes verglichen. In Bezugnahme auf die Ergebnisse wird abschließend ein Ausblick für die Erweiterung und Optimierung dieser Aufgabe vorgeschlagen.

Steffen Lossau

Title of Thesis

Automatic generation of lighting control data from videos of stage performances

Keywords

Computer Vision, Machine Learning, Lighting, Video, Control Data, Stage, Event technology

Abstract

The aim of this thesis is to create lighting control data from videos of stage performances by using suitable computer vision models. This objective is achieved by working on five key points. Different models are differentiated from each other so that suitable models can be found for this task. Furthermore, the data set must be prepared for training the models. The focus of this preparation will be on the abstraction of the individual light parameters for training the models. Due to the size of the available data set, an application will be developed that automatically prepares the data set. Before training the individual models, the different training parameters are also examined. Before the training results of the models and the generated lighting control data are evaluated, an application must be created to generate the control data on the basis of the trained models. The generated lighting control data is then compared with the actual lighting control data from the same data set. Finally, with reference to the results, an outlook for the expansion and optimization of this task is proposed.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
1 Einleitung	1
1.1 Problemstellung	1
1.2 Ziele	2
2 Stand der Forschung	4
2.1 Modelle der Computer Vision	4
2.2 Machine Learning in der Veranstaltungstechnik	6
3 Model-Auswahl	7
3.1 Model-Abgrenzung	7
3.2 Alternativer Ansatz	8
4 Datensatz	10
4.1 Aufbereitung	11
4.1.1 Unterteilung	11
4.1.2 Datensatz für das Detection-Model	12
4.1.3 Datensatz für die Classification-Models	13
4.2 Manuelle Überarbeitung	18
5 Training	19
5.1 Activation Functions	19
5.2 Loss Functions	20
5.3 Batch Size	20
5.4 Image Preperation	21
5.5 Segmentation Mask	21
5.6 Data Augmentation	22
5.7 Learning Rate	22
5.8 Optimizer	23

5.9	Regularization	24
6	Anwendung	25
7	Ergebnisse	26
7.1	Auswertung des Fixture Detection Models	26
7.2	Auswertung des ParFect Classification Models	28
7.3	Auswertung des Beam Classification Models	30
7.4	Auswertung des OnePatt Classification Models	32
7.5	Auswertung der generierten Steuerdaten	34
8	Erweiterung und Optimierung	37
9	Fazit	39
	Literaturverzeichnis	40
A	Anhang	43
A.1	Verwendete Hilfsmittel	43
A.2	Anleitung für die Anwendung	44
A.3	Anwendungen in der Veranstaltungstechnik	44
A.4	Lichtsteuerdaten in PIX-Darstellung	45
A.5	Datenstruktur der PKL-Dateien	46
A.6	YOLO Datenstruktur	46
A.7	Beispielbilder aus dem Datensatz	49
A.8	Beispielbilder aus dem Training	51
B	Datenblätter	53
B.1	Robe LEDBeam 150	53
B.2	Robe OnePatt	58
B.3	Robe ParFect	63
B.4	Robe TetraBar RGB	68
	Glossar	73
	Selbstständigkeitserklärung	75

Abbildungsverzeichnis

4.1	Ein Frame aus dem Trainingsdatensatz des „Detection Models“ mit manuell gesetzten „Bounding Boxes“ für die automatisierte Weiterverarbeitung.	12
4.2	Übersicht der Verteilung der Fixtures in dem Trainingsdatensatz für das „Detection Models“.	13
4.3	Parameterverteilung im Datensatz für das Robe Parfect „Classification Model“.	16
4.4	Parameterverteilung im Datensatz für das Robe LEDBeam 150 „Classification Model“.	16
4.5	Parameterverteilung im Datensatz für das Robe OnePatt „Classification Model“.	17
7.1	Verlauf der Precision bzw. des Recalls zu der Confidence des „Detection Models“.	26
7.2	Die normalisierte „Confusion Matrix“ für die Trainingsergebnisse des „Detection Models“.	27
7.3	Der Trainingsverlauf der letzten zehn Epochen für das Robe ParFect „Classification Model“. Es zeigt die Veränderung des Verlustes und der Genauigkeit.	28
7.4	Die normalisierte Confusion-Matrix für die Trainingsergebnisse des Robe ParFect „Classification Model“.	29
7.5	Der Trainingsverlauf der letzten 20 Epochen für das Robe LEDBeam 150 „Classification Model“. Es zeigt die Veränderung des Verlustes und der Genauigkeit.	30
7.6	Die normalisierte Confusion-Matrix für die Trainingsergebnisse des Robe LEDBeam 150 „Classification Model“.	31
7.7	Der Trainingsverlauf der letzten 20 Epochen für das Robe OnePatt „Classification Model“. Es zeigt die Veränderung des Verlustes und der Genauigkeit.	32

7.8	Die normalisierte Confusion-Matrix für die Trainingsergebnisse des Robe OnePatt „Classification Model“	33
7.9	Ein Frame aus der verwendeten Videosequenz mit erkannten Fixtures des „Detection Models“	34
7.10	Die normalisierte Confusion Matrix für die Klassifikationsergebnisse der automatisch erstellten Steuerdaten.	35
7.11	Die Verteilung der Intensitäten aus der Sequenz die für die automatische Generierung von Steuerdaten verwendet wurde.	36
A.1	Abstraktion der Steuerdaten durch eine zeilenweise Darstellung der Parameter der Fixture Gruppen in einem Pixelformat.	45
A.2	Beispielhafter Aufbau der Datenstruktur der vorliegenden PKL-Dateien.	46
A.3	Der Aufbau der Datenstruktur für das Training eines „Yolov8 Detection Models“	47
A.4	Der Aufbau einer „Label“ TXT-Datei für das Training eines „Yolov8 Detection Models“	47
A.5	Der Aufbau der Datenstruktur für das Training eines „Yolov8 Classification Models“	48
A.6	Unmodifizierte Bilder aus dem Datensatz für das „Fixture Detection Model“	49
A.7	Unmodifizierte Bilder aus dem Datensatz für das Robe ParFect „Classification Model“	49
A.8	Unmodifizierte Bilder aus dem Datensatz für das Robe LEDBeam „Classification Model“	50
A.9	Unmodifizierte Bilder aus dem Datensatz für das Robe OnePatt „Classification Model“	50
A.10	Beispielbilder aus dem Training des „Fixture Detection Models“	51
A.11	Beispielbilder aus dem Training des Robe ParFect „Classification Models“	52
A.12	Beispielbilder aus dem Training des Robe LEDBeam 150 „Classification Models“	52
A.13	Beispielbilder aus dem Training des Robe OnePatt „Classification Models“	52

1 Einleitung

Die Computer Vision wird in vielen Bereichen der Wirtschaft und Gesellschaft als Werkzeug verwendet. Durch ihren Einsatz ist es beispielsweise möglich, dass Autos autonom fahren können, indem in Echtzeit Steuerdaten aus Bilddaten extrahiert werden. In der Medizin können mittels ihr erste Einschätzungen über mögliche Erkrankungen erstellt und Tumore erkannt werden, indem durch Computer Vision Aufnahmen ausgewertet werden. In der Landwirtschaft wird sie eingesetzt, um den Wachstumsprozess von Pflanzen zu überwachen und einen Krankheitsbefall zu identifizieren.

Bei diesen Anwendungsbeispielen werden Informationen aus Bilddaten durch den Einsatz von Computer Vision gewonnen. Da die Methode in diesen Bereichen gut funktioniert, kann die Frage gestellt werden, ob der Einsatz von Computer Vision auch in der Domäne der professionellen Veranstaltungstechnik einen sinnvollen Anwendungsfall findet. Dabei beruht der Anwendungsfall dieser Bachelorthesis auf derselben Methode wie die zuvor benannten Beispiele: Die Bilddaten aus den Videos von Bühnenauftritten sollen genutzt werden, um Informationen über die Steuerdaten des Lichtpultes zu gewinnen.

1.1 Problemstellung

Die Herausforderung bei der Erreichung des gesetzten Zieles lässt sich in vier zu bearbeitende Aufgabenfelder unterteilen: Die Aufbereitung des Videodatensatzes(1), die Abstraktion der Steuerdaten eines Lichtpultes(2), die Auswahl geeigneter Modelle für die Bildverarbeitung(3), sowie die Entwicklung einer Anwendung für die Erstellung von Lichtsteuerdaten(4).

Aufgabe 1: Als Datensatz stehen mehrere Videos von Bühnenauftritten zur Verfügung. Eine manuelle Bearbeitung dieses Datensatzes würde den Einsatz einer großen zeitlichen Ressource bedeuten, weshalb es notwendig ist eine Form der automatisierten Aufbereitung des Datensatzes zu erstellen.

Aufgabe 2: Die Steuerdaten eines Lichtpultes bestehen aus unterschiedlichen Parametern für die einzelnen Channels eines Bühnenscheinwerfers, dem so genannten Fixture. In der Regel besitzt ein Fixture mehrere Channels. Die Werte eines Channels liegen im 8-bit Zahlenraum. Die Parameter für jeden Channel können als Zustand für das Fixture angenommen werden. Zudem sind bei dem Lichtaufbau in dem dieser Bachelorthesis zugrunde liegendenm Datensatz unterschiedliche Fixture-Typen verwendet worden. Es ist anzunehmen, dass der daraus resultierende Zustandsraum als Klassifizierungsmenge für ein Modell der Computer Vision zu groß sein wird. Um den Zustandsraum der Parameter zu verringern, wird eine Form der Abstraktion der Steuerdaten nötig sein.

Aufgabe 3: Für die Klassifizierung der einzelnen Fixture-Parameter müssen geeignete „Classification Models“ gefunden werden. Eine Klassifizierung der einzelnen Fixture-Typen wird nicht für die Generierung von Steuerdaten ausreichen. Es ist wichtig, dass die Position der Fixtures erkannt wird, damit Rückschlüsse auf die Fixture-Gruppen geschlossen werden können. Daher wird es nötig sein, ein „Detection Model“ für die Erkennung der einzelnen Fixtures zu verwenden.

Aufgabe 4: Abschließend muss eine Anwendung für die Aufbereitung des Datensatzes und der Abstraktion der Steuerdaten erstellt werden. Nach dem Training des einzelnen Modelle wird eine Anwendung für das Zusammenführen der Ergebnisse in eine Form der Steuerdaten nötig sein.

1.2 Ziele

Das Hauptziel dieser Bachelorthesis ist die automatische Generierung von Lichtsteuerdaten durch die Verwendung von „Computer Vision Models“. Dieses Ziel kann in zwei Unterziele aufgeteilt werden:

- Die Auswahl, das Training und die Evaluation von geeigneten „Models“ für die Erkennung und Klassifizierung der Fixtures.
- Das Erstellen von Anwendungen für die Aufbereitung des Datensatzes und die Zusammenfassung der Vorhersagen der „Models“ in Form von Lichtsteuerdaten.

Da in der Domäne der Generierung von Lichtsteuerdaten durch „Computer Vision Models“ keine weiteren Arbeiten bekannt sind, lassen sich weitere Ziele für die Qualität der Ein- und Ausgabedaten ableiten: Es wird untersucht, welche Form von Aufbereitung der

Eingabedaten sinnvoll sein könnte, sowie die Qualität der generierten Steuerdaten mit den tatsächlichen Steuerdaten von Bühnenlichtshows verglichen. Des Weiteren wird ein Ausblick für die Weiterentwicklung des Projektes gegeben.

2 Stand der Forschung

Seit mindestens zwei Jahren sieht man durch den Erfolg von OpenAI, dass Machine Learning im Allgemeinen, und die Computer Vision im Speziellen, deutliche entwicklungstechnische Fortschritte erzielt haben. Mittlerweile gibt es eine Vielzahl an unterschiedlichen Modellen und Modellversionen mit eigenen Merkmalen und Stärken. Durch ihre unterschiedlichen Eigenschaften, ist es wichtig, das richtige Modell für einen Anwendungsfall auszuwählen. Um einen guten Überblick über die einzelnen Modelle zu bekommen, ist es sinnvoll, den aktuellen Stand der Forschung einzubeziehen.

2.1 Modelle der Computer Vision

CenterNet: Im Gegensatz zu anderen „Models“, die Objekte durch „Bounding Boxes“ klassifizieren, führt CenterNet eine Klassifizierung durch „Center Points“ ein. Bei dieser Methode werden Mittelpunkte des Objektes erkannt. Von diesen Mittelpunkten können weitere Merkmale des Objektes, wie die Größe, Position oder Orientierung abgeleitet werden. Dies verspricht eine hohe Genauigkeit und Geschwindigkeit. Durch diese Methode können Objekte der gleichen Klasse in unterschiedlichen Positionen gut erkannt werden. Des Weiteren ist das Modell echtzeitfähig.[25]

MobileNet-SSD: Bei diesem „Model“ handelt es sich um eine Kombination aus dem vortrainierten Deep-Learning Modell MobileNet mit dem „Single Shot Multi-Box Detector“ (SSD). Das Ziel dieses Modells ist es, eine ressourcensparende und effiziente Lösung für Objekterkennung in Echtzeit zu erstellen. Es wird beispielsweise im Bereich des autonomen Fahrens und der Videoüberwachung eingesetzt.[4]

EfficientDet: Hierbei handelt es sich um eine Familie an „Models“ für die Objekterkennung. Als besonderes Merkmal wurde bei diesen „Models“ ein gewichtetes bidirektionales Feature-Pyramidennetzwerk (BiFPN) integriert. Das BiFPN sorgt für eine effizientere und schnellere Merkmalsfusion des zugrunde liegenden neuronalen Netzes. Des Weiteren

wurde eine neue Methode für eine gleichmäßige Skalierung der Auflösung, Tiefe und Breite des Backbones, Feature Networks und der Box/Class Prediction Networks eingeführt. Diese Erweiterungen führen zu einer erhöhten Effizienz im Bereich der durchschnittlichen Präzision.[13]

cGANs: „Conditional Generative Adversarial Networks“ (cGANs) bieten die Möglichkeit einer Bild zu Bild-Übersetzung. Ein GAN besteht aus zwei neuronalen Netzen: Dem Diskriminator und dem Generator. Die beiden neuronalen Netze werden gegeneinander trainiert, wodurch eine Generierung von nahezu realistischen Daten möglich ist. Das „conditional“ Prefix beschreibt die Möglichkeit, die Art der generierten Daten besser zu steuern. Durch die Verwendung dieses Modells ist es möglich, Bilder zwischen zwei Domänen zu konvertieren.[5]

YOLO: Der Ansatz für die „You Only Look Once“ (YOLO) „Models“ ist, ein schnelles und leichtgewichtiges System für die Objekterkennung zu entwickeln. Insbesondere soll der Kompromiss aus langsamen und akkuraten „Models“ gegenüber schnellen und weniger akkuraten Modellen in Echtzeitanwendungen verbessert werden. Ein weiterer Fokus dieses Modells liegt auf der Benutzerfreundlichkeit. Die meisten anderen „Models“ sind nicht echtzeitfähig und benötigen eine hohe Anzahl an GPUs für das Training. Das Ziel der „YOLO-Models“ ist das Training und das Erzielen von zufriedenstellenden Ergebnissen in Echtzeit unter der Verwendung einer einzelnen, konventionellen GPU.[2]

Oriented R-CNN: Der Fokus von diesem Modell ist eine Erhöhung der Effizienz zur Erkennung von Objekten in unterschiedlicher Ausrichtung. Das Modell besteht aus zwei Hauptkomponenten: Die erste Komponente ist das „Oriented Region Proposal Network“ (oriented RPN). Es benötigt weniger Parameter als andere RPNs oder „Region of Interest“ (RoI) Transformer. Die zweite Komponente ist der „Oriented R-CNN Detection Head“. Er ist für die „Feature-Extraction“ und Klassifizierung der Objekte zuständig.[24]

2.2 Machine Learning in der Veranstaltungstechnik

Das Machine Learning im Allgemeinen und die Computer Vision im Speziellen, finden bisher kaum Verwendung in der professionellen Veranstaltungstechnik. Für die Vorhersage des Verhaltens und der Anzahl an Besuchenden werden im organisatorischen Bereich der Veranstaltungsbranche Technologien verwendet, die auf „Machine Learning Models“ beruhen[15]. Des Weiteren können im Bereich der Sicherheit und der Überwachung von Veranstaltungen Systeme angewandt werden, die Anomalien im Verhalten einer Menschenmenge feststellen und melden[1]. Diese Technologien wurden nicht für die Veranstaltungsbranche entworfen, können aber in ihr Verwendung finden.

Im Bereich der Veranstaltungstechnik ist die Verwendung von Machine Learning noch nicht sehr genau erforscht. Einzelne Technologien sind kommerziell veröffentlicht worden, finden aber praktisch keine Anwendung in der Veranstaltungstechnik. Beispiele können dem Anhang A.3 entnommen werden. Auf Basis von „Large Language Models“ kann eine Echtzeitübersetzung in Form von „Speech-to-Text“ von Reden realisiert werden. Im Bereich des „Audio-Engineering“ können die Frequenz-Parameter der einzelnen Audiospuren unter der Verwendung eines „Machine Learning Models“ in Echtzeit voreingestellt werden. Des Weiteren ist es möglich in Echtzeit automatische Lichtsteuersignale auf Basis von Audiospuren zu erstellen.

3 Model-Auswahl

Vor dem Beginn der Aufbereitung des Video-Datensatzes und dem Training muss ein geeignetes „Detection Model“ für die Erkennung der einzelnen Fixtures, sowie geeignete „Classification Models“ für die Klassifizierung der Parameter der Fixture-Typen ausgewählt werden. Dabei sollen die erkannten Fixtures des „Detection Models“ den „Classification Models“ als Eingabe für die Klassifikation gegeben werden. Somit sollten diese „Models“ als wichtige Voraussetzung aufeinander abgestimmt sein. Des Weiteren wird die Verarbeitung von Bilddaten zu Steuerdaten voraussichtlich auf konventionellen Endgeräten wie Laptops oder PCs stattfinden. Diese technische Anforderung sollte bei der Auswahl der „Models“ ebenfalls betrachtet werden. Insgesamt ist es wichtig, dass die gewählten „Models“ zu der Problemstellung und den Zielen dieser Bachelorthesis passt.

3.1 Model-Abgrenzung

Für das gestellte Problem ist das „CenterNet Model“ nicht zielführend: Denn der Ansatz von „CenterNet“, die gleichen Objekte in unterschiedlichen Ausrichtungen anhand eines Center Points zu klassifizieren, ist für diese Anwendung nicht wichtig. Bei den Fixtures handelt es sich um kompakte Gegenstände, die eher nicht aus unterschiedlichen Perspektiven betrachtet werden. Daher ist eine differenzierte Betrachtung des Objektes, in Bezug auf die Perspektive, nicht notwendig. Analog zu „CenterNet“ ist das „Oriented R-CNN“ zu betrachten. Es verspricht ebenfalls eine bessere Klassifizierung von Objekten mit einer unterschiedlichen Ausrichtung. Lediglich der „Region of Interest“ Transformer ist für diesen Anwendungsfall interessant, um die Fixtures möglicherweise schneller zu erkennen und zu identifizieren. Dieses Merkmal reicht aber nicht, um als Model für diese Anwendung verwendet zu werden.

Das „MobileNet-SSD Model“ könnte als geeignetes Model für den beschriebenen Anwendungsfall angenommen werden, obwohl die Kernfunktion für die definierte Problemstellung nicht relevant sind. Denn die beiden Kernmerkmale von „MobileNet“ sind für diese

Anwendung nicht kritisch: Es wird keine ressourcensparende Lösung benötigt, da die Anwendung eigenständig läuft und nicht in einem eingebetteten Kontext funktionieren muss. Des Weiteren ist die Echtzeitfähigkeit keine Anforderung dieser Anwendung.

Deshalb bieten sich als geeignete „Models“ zur Lösung der Aufgabe insbesondere die „YOLO-Models“ und die „EfficientDet-Models“ an. Denn die Erkennung der einzelnen Fixtures durch ein „Detection Model“ ist eine Kernkomponente dieser gestellten Aufgabe. Durch die Verwendung von BiFPN der „EfficientDet-Models“ wird eine höhere Genauigkeit bei der Objekterkennung erzielt. Für die Klassifizierung der Fixtures muss jedoch ein anderes „Model“ verwendet werden. Da die unterschiedlichen Models in dieser Anwendung miteinander verbunden sind, bieten sich die „EfficientNet-Models“ an, da beide „Models“ vom Großkonzern Google entwickelt wurden.

Den „EfficientNet-Models“ stehen die „YOLO-Models“ gegenüber. Diese „Models“ bieten sowohl beim Training, als auch bei den Vorhersagen eine hohe Geschwindigkeit und Genauigkeit der Ergebnisse. Des Weiteren sind sie für Anwendungen auf einer einzelnen, konventionellen GPU konzipiert. Das ist sowohl für die Erstellung dieser Bachelorthesis, als auch für die spätere Verwendung der Anwendung, hilfreich. Ein weiterer Vorteil der „YOLO-Models“ ist die Leichtigkeit und Benutzerfreundlichkeit des „Frameworks“, sowie eine gute und übersichtliche Dokumentation[23]. Aus diesen Gründen wurden die zuletzt, veröffentlichten „YOLOv8-Models“ für die Problemlösung verwendet. Als „Object Detection Model“ wurde das „YOLOv8s Model“ verwendet.

Damit nicht grundsätzlich die „Object Detection“ von dem „Model“ gelernt werden muss, wurde ein auf dem „Common Objects in Context“ (COCO) Datensatz vortrainiertes „Model“ verwendet[12]. Für die Klassifikationen der Fixtures wurde das „YOLOv8s-cls Model“ angewandt. Alle verwendeten „Classification Models“ wurden bereits auf dem „ImageNet“ Datensatz vortrainiert[6].

3.2 Alternativer Ansatz

Eine besondere Erwähnung findet die Modell-Familie der „Generative Adversarial Networks“ (GANs). Als Ground-Truth für die Validierung der generierten Daten liegen sowohl die Parameter für die Lichtsteuerung, als auch eine Abstraktion der Parameter in Bildform vor, wie im Anhang A.4 dargestellt. Somit wäre ein Bild zu Bild-Ansatz für

diese Anwendung möglich. Da die generierten Bilder die Parameter des Lichtpultes repräsentieren und abstrahieren, wird eine Validierung der Ergebnisse schwieriger, da durch die Abstraktion mit einem Informationsverlust zu rechnen ist. Des Weiteren weisen diese Eingabe -und Ausgabebilder bei GANs eine ähnliche Struktur auf. Beispielsweise kann aus einer Skizze eines Schuhs ein Bild eines realistisch aussehenden Schuhs generiert werden. Dadurch ähnelt die Struktur des Ausgabebildes der Struktur der Skizze. Für den Anwendungsfall in dieser Bachelorthesis müssten Bildaufnahmen einer Bühne in abstrakte Pixel und Farbformen umgewandelt werden, wodurch keine ähnliche Struktur zwischen den Ein- und Ausgabebildern mehr vorhanden bleibt. Aus diesen Gründen wurde sich gegen eine Realisierung durch einen Bild zu Bild-Ansatz entschieden.

4 Datensatz

Der dieser Bachelorthesis zugrunde liegende Video-Datensatz wurde 2022 in der HAW-Hamburg von der Fakultät Medientechnik erstellt. Er besteht aus insgesamt 21 MP4-Videos mit einer Auflösung von 1920 x 1080 Pixeln, die jeweils eine Länge von ca. 30-60 Sekunden haben. Die Videos zeigen einen Bühnenaufbau mit eingebautem Bühnenlicht. Das Bühnenlicht besteht insgesamt aus 27 Fixtures. Die Fixtures lassen sich in fünf Gruppen aufteilen: Floor, Riser, Front-Truss, Mid-Truss und Back-Truss. Die Floor-Gruppe besteht aus acht Robe Parfect 150 RGBW Fixtures (#701 - #708). Die Riser-Gruppe besteht aus sieben Robe OnePatt Fixtures (#801 - #807). Die Mid-Truss-Gruppe besteht aus vier Robe TetraBar RGBPart Fixtures (#301, #401, #501, #601). Für die Front-Truss (#101 - #104) und die Back-Truss (#201 - #204) Gruppen wurden jeweils vier Robe LEDBeam150 Fixtures aufgehängt. Die technische Daten der Fixture können den Datenblättern im Anhang B entnommen werden. Jedes Video zeigt eine vorprogrammierte Lichtshow für einen Ausschnitt eines Musikstücks. Die Videos wurden von einer stationären Kamera aufgenommen, wodurch sich ein einheitlicher Bildausschnitt auf jedem Video ergibt.

Zu jedem Video gibt es eine eigene PKL-Datei. Aus diesen Dateien lässt sich der „Ground-Truth“ für das Training ableiten. Ein beispielhafter Aufbau der Datenstruktur der PKL-Dateien kann dem Anhang A.5 entnommen werden. Für jedes verwendete Fixture wurden, mit einer Abtastrate von 30 Einträgen pro Sekunde, die Ausgabeparameter des Lichtpultes für die einzelnen Fixture-Channels aufgezeichnet. Die aufgezeichneten Channels entsprechen nicht der gesamten Anzahl der Channels eines Fixtures. Hier wurde bereits eine Vorauswahl getroffen.

Je nach Fixture-Typ unterscheiden sich die aufgezeichneten Parameter. Für jedes Fixture sind die Werte für die „Hue“, die „Saturation“, die „Brightness“ und für „White“ vorhanden. Für die Robe LEDBeam 150 Fixtures der Front- und der Back-Truss wurden zusätzlich die Werte für Pan und Tilt, sowie „Zoom“-Werte aufgezeichnet. Durch die PKL-Dateien

lassen sich somit für jedes Frames des Videos die tatsächlichen Parameter des Lichtpultes nachvollziehen.

4.1 Aufbereitung

Für eine Weiterverarbeitung des Datensatzes durch neuronale Netze muss der Datensatz aufbereitet werden. Da es sich bei dem verwendeten „Detection Model“ und den „Classification Models“ um Modelle für Objekterkennung und Klassifizierung auf Bildern handelt, müssen die Videos zunächst in Bilder umgewandelt werden. Da die PKL-Dateien mit einer Abtastrate von 30 Einträgen pro Sekunde erstellt wurden, wurden die Videos dementsprechend mit 30 Bildern pro Sekunde umgewandelt. Bei einer durchschnittlichen Länge von ca. 45 Sekunden pro Video ergibt das 1350 Bilder pro Video.

Das Training der „Models“ wird mit einer Cloud-Lösung realisiert werden. Daher war es naheliegend, die Bilder ebenfalls in einer Cloud zu speichern. Um den Speicherbedarf in der Cloud und die Verarbeitungszeit der einzelnen Bilder für das neuronale Netz möglichst gering zu halten, wurden die Bilder im JPEG-Format abgespeichert. Beispielbilder können dem Anhang A.7 entnommen werden.

4.1.1 Unterteilung

Da das „Labelling“ der Bilder durch eine selbst geschriebene Routine automatisiert erfolgt ist, war es wichtig sicherzustellen, dass der Bildausschnitt bei allen Videos identisch ist. Die verwendete Kamera besitzt eine Art Autofokus-Funktion, die bei einer direkten Beleuchtung der Linse oder starken Kontrasten zu einer Abweichung des Bildausschnittes geführt hat. Da die Front-Truss und Back-Truss Gruppen durch bewegtes Licht, so genannte Moving Heads, realisiert wurden, trat dieser Effekt bei 13 der 21 Videos auf. Diese Videos waren nicht mehr für eine automatisierte Weiterverarbeitung als Datensatz geeignet. In vier der 13 Videos waren die Abweichungen jedoch minimal. Somit wurden diese Videos als mögliche Validierungsvideos für die späteren Ergebnisse zurückgehalten. Die restlichen acht Videos waren für eine automatisierte Weiterverarbeitung geeignet und bildeten somit die Grundlage für das Training der „Models“.

4.1.2 Datensatz für das Detection-Model

Für das Training des „Detection Models“ müssen die Bilder durch „Bounding Boxes“ mit den vorhandenen Fixture-Typen gelabelt werden. Aufgrund der Anzahl der zu labelnde Bilder wurde eine Routine geschrieben, die das „Labelling“ automatisiert übernimmt.



Abbildung 4.1: Ein Frame aus dem Trainingsdatensatz des „Detection Models“ mit manuell gesetzten „Bounding Boxes“ für die automatisierte Weiterverarbeitung.

In Abbildung 4.1 ist der „Frame“ aus einem Video zu sehen, dessen „Bounding Boxes“ manuell erstellt wurden. Auf Basis dieser „Bounding Boxes“ wurden alle weiteren Bilder markiert. Da die Videos aus derselben Perspektive aufgenommen wurden und somit der Bildausschnitt auf allen Videos identisch ist, konnten die Koordinaten der einzelnen „Bounding Boxes“ für alle weiteren Bilder verwendet werden.

Die verarbeiteten Bilder wurden automatisch in die für YOLO „Detection Models“ benötigte Ordnerstruktur abgelegt. Dabei wurden 80% der Bilder als Trainingsdaten und 20% der Bilder als Testdaten klassifiziert. Des Weiteren wurden die Labels in dem für YOLO „Detection Models“ benötigten TXT-Format generiert. Eine Darstellung der Datenstruktur kann dem Anhang A.6 entnommen werden.

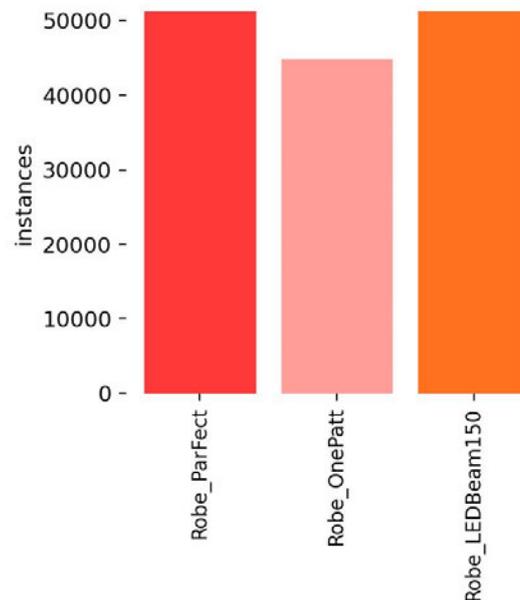


Abbildung 4.2: Übersicht der Verteilung der Fixtures in dem Trainingsdatensatz für das „Detection Models“.

Insgesamt ergibt dies einen Datensatz von 6.393 Trainingsbildern und 1.604 Testbildern. Auf allen Bildern sind drei unterschiedliche Fixture-Typen vorhanden: Acht Robe LED-Beam150, acht Robe Perfect 150 RGBW und sieben Robe OnePATT. Eine Verteilung der einzelnen Instanzen lässt sich Abbildung 4.2 entnehmen.

4.1.3 Datensatz für die Classification-Models

Auf Basis der „Bounding Boxes“ für das „Labelling“ des Datensatzes für das „Detection Model“ wurde der Datensatz für die „Classification Models“ erstellt. Für jeden Fixture-Typen wurde ein eigenes „Classification-Model“ verwendet. Somit wurde jeder Bildausschnitt einer „Bounding Box“ automatisch für das entsprechende „Model“ als Bild des Datensatzes verwendet. Auf diese Weise wurde die Zuordnung zu den unterschiedlichen Models automatisiert realisiert. Für das „Labelling“ und somit der Klassifikation der Bilder musste der Zustandsraum der Fixtures näher betrachtet werden.

Für eine Annäherung an den realen Zustandsraum werden die Channels für die Intensität und die RGB-Farbmischung betrachtet. Jeder Channel besteht aus einem 8-bit Zahlenraum. Für die vier Channels (Intensität, Rot, Grün und Blau) ergibt das einen

Zustandsraum von $256^4 = 4.294.967.296$ Zuständen. Somit müsste jedes „Classification Model“ über vier Milliarden einzigartige Objekte klassifizieren können, die sich teilweise marginal unterscheiden. Experimente über eine realistische Anzahl an einzigartigen Klassen, die von einem „Model“ klassifiziert werden können, liegen im niedrigen fünfstelligen Bereich[9]. In den Videos ist nicht der gesamte Zustandsraum abgebildet, dennoch ist die Zahl der einzigartigen Objekte zu groß. Des Weiteren sollte die Limitierung durch den Datensatz nicht eine ausschlaggebende Größe für die Art der Klassifizierung sein. Deshalb musste eine Abstraktion des Zustandsraumes stattfinden.

Analog zum Beispiel für den gesamten Zustandsraum wurden hierfür die Channels für die Intensität und die Farben verwendet. Für die Intensität wurde, anstatt einer 8-bit Abstufung, der Zustand in 20% Schritten definiert. Somit kann ein Fixture sechs unterschiedliche Zustände für die Intensität annehmen: 0%, 20%, 40%, 60%, 80% und 100%. Der Zustandsraum der Farben ergibt sich einerseits aus den üblichen Grundfarben von Effektlicht: Rot (R), Grün (G), Blau (B), Cyan (C), Magenta (M), Gelb (Y), Weiß (W) und Amber (A) und andererseits aus den tatsächlich wahrnehmbaren Unterschieden der Farbe. Als Beispiel könnte ein Fixture die Tatsächlichen RGB-Werte von $R = 0$, $G = 10$, $B = 255$ haben. Dem Gegenüber könnte ein anderes Fixture die RGB-Werte $R = 0$, $G = 0$, $B = 255$ angenommen haben. Das menschliche Auge nimmt jedoch beide Fixtures als blau war. Konkret musste deshalb ein Threshold für die Farbmischung festgesetzt werden. Wieviel Grün darf bei einem Fixture mit einem vollen Blau Wert hinzugemischt werden, damit es weiterhin als Blau definiert ist und noch nicht als Cyan gilt? Als Threshold wurde hier ein Faktor von 0,6 festgesetzt. Damit ein Fixture weiterhin von den „Classification Models“ als Blau klassifiziert wird darf bei einem Wert von $B=255$ der Farbwert für Grün nicht $255 \cdot 0,6 = 153$ überschreiten, denn sonst wird es als Cyan klassifiziert. Durch diese Abstraktion ergeben sich acht einzigartige Zustände für den Farbraum.

Somit kann ein Fixture $6 \text{ Intensitätszustände} \cdot 8 \text{ Farbzustände} = 48$ einzigartige Zustände annehmen. Diese Art der Abstraktion kann auf drei der vier Fixture-Typen angewendet werden (Robe LEDBeam150, Robe Parfect 150 RGBW und Robe onePatt). Dieser Zustandsraum und somit die Anzahl an einzigartigen Objekten, die die jeweiligen „Models“ klassifizieren müssen, ist annehmbar. Das endgültige Klassifizierungslabel für die Parameter eines Fixture-Typen, zusammengesetzt aus den abstrahierten Intensitäts- und Farbwerten, wurde beispielhaft in folgender Form realisiert: B_100 . Dieses Label repräsentiert die Parameter für die Farbe Blau (B) mit einer Intensität von 100% (100). Ein anderes Beispiel könnte R_40 sein, das die Parameter für die Farbe Rot mit einer Inten-

sität von 40% repräsentiert. Der Zustand in dem ein Fixture kein Licht emittiert wird durch das Label *NC_0* (No Color) dargestellt.

Die Bilder wurden automatisch in die für ein YOLO „Classification Model“ benötigte Ordnerstruktur abgelegt. Eine Darstellung der Ordnerstruktur kann dem Anhang A.6 entnommen werden. Des Weiteren wurden 80% der Bilder jeder Klassifikation als Trainingsbilder- und 20% als Testbilder verwendet. Der Datensatz für das Training des „Classification Models“ für die Robe LEDBeam150 Fixtures besteht aus 33.450 Trainingsbildern und 8400 Testbildern. Der Datensatz für das Training des „Classification Models“ für die Robe Parfect 150 RGBW Fixtures besteht aus 50.188 Trainingsbildern und 12.607 Testbildern. Der Datensatz für das Training des „Classification Models“ für die Robe onePATT Fixtures besteht aus 39.713 Trainingsbildern und 9923 Testbildern.

Insgesamt ist die quantitative Verteilung der einzelnen Zustände in den Trainingsdatensätzen unausgeglichen. Einen Großteil der Trainingsdaten machen die Zustände mit einer Intensität von 0% und 100% aus. Da die Trainingsdaten auf tatsächlichen Lichtsteuerdaten beruhen, ist die Verteilung nachvollziehbar. Kontrastreiche Effekte mit einem stufenlosen Wechsel zwischen einer Intensität von 0% und 100% werden häufig in der Programmierung von Lichtsteuerdaten verwendet. Stellenweise werden diese Effekte durch Sequenzen mit beispielsweise sinusförmigen Dimmerkurven erweitert. In diesen Sequenzen treten Zustände mit einer Intensität von 20% bis 80% auf. Daher ist es nachvollziehbar, dass diese Zustände in dem Datensatz unterrepräsentiert sind. Teilweise sind weniger als 50 Bilder für Zustände in diesem Intensitätsbereich vorhanden. Das könnte zu einem Problem für das Lernen dieser Zustände durch die „Classification Models“ führen, auch wenn es sich um vergleichsweise „simple“ Objekte handelt[27].

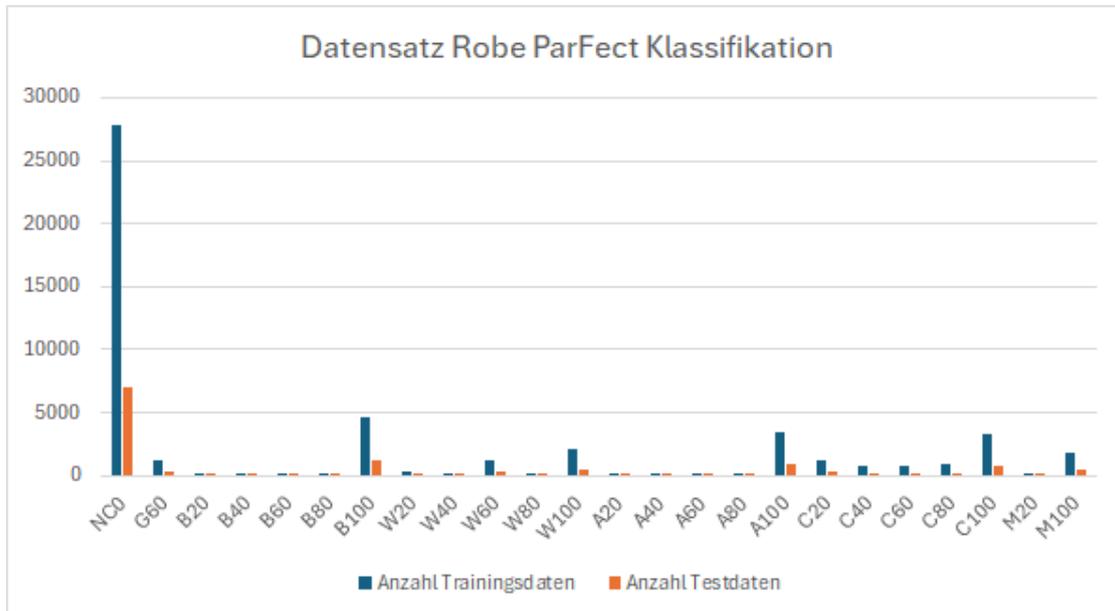


Abbildung 4.3: Parameterverteilung im Datensatz für das Robe ParFect „Classification Model“.

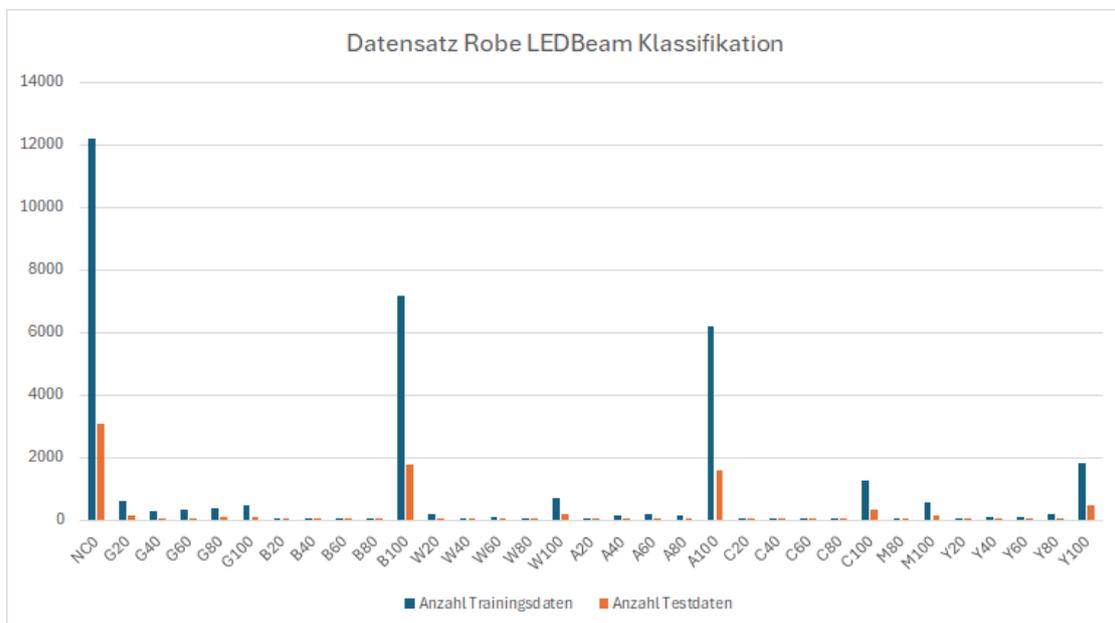


Abbildung 4.4: Parameterverteilung im Datensatz für das Robe LEDBeam 150 „Classification Model“.

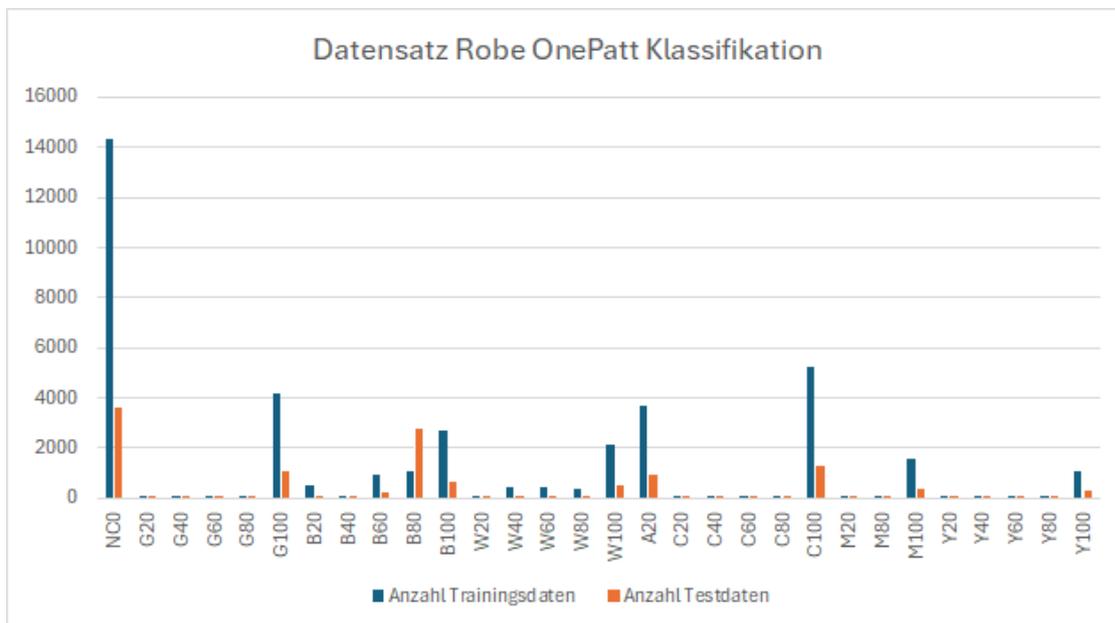


Abbildung 4.5: Parameterverteilung im Datensatz für das Robe OnePatt „Classification Model“.

4.2 Manuelle Überarbeitung

Bei der Überprüfung des automatisch erstellten Trainings- und Testdatensatzes sind Mängel in der Qualität der Klassifikation festgestellt worden. Der Datensatz für das Training des „Detection Models“ wurde ohne auffällige Mängel automatisch erstellt. Ebenso konnte der automatisch erstellte Datensatz für die Robe Parfect 150 RGBW Fixtures übernommen werden. Die Datensätze für die Robe OnePatt Fixtures und die Robe LEDBeam150 Fixtures mussten jedoch manuell überarbeitet werden.

Die Robe OnePatt Fixtures besitzen einen „Vintage Look“, wodurch sie älteren Lampenmodellen optisch nachempfunden wurden. In älteren Lampenmodellen sind konventionelle Leuchtmittel mit einer Wendel verbaut. Diese Leuchtmittel besitzen oft eine Trägheit in ihrer Intensität und somit eine flachere Dimmer-Kurve. Die Robe OnePatt Fixtures besitzen ein LED-Leuchtmittel, verfügen aber über eine künstlich eingestellte Dimmer-Kurve, um diesen Effekt nachzubilden. Wenn die Steuerdaten dieses Fixtures in einem Frame eine Intensität von 100% ausgeben und im nächsten Frame die Intensität auf 0% gesetzt wird, wird es einige Momente dauern, bis es wirklich kein Licht mehr emittiert. Tatsächlich ist in dem nächsten Frame ein Fixture zu sehen, dass beispielsweise eine reale Intensität von ca. 80% hat, während die Steuerdaten eine Intensität von 0% ausgeben. Da das „Labelling“ für die Bildklassifikation auf Basis der Steuerdaten geschehen ist, musste dieser Datensatz manuell überarbeitet werden.

Weiterführend waren die Steuerdaten für die Robe LEDBeam150 Fixtures in den PKL-Daten lückenhaft. In unregelmäßigen Abständen waren die Werte für die Farbe und die Intensität auf 0 gesetzt, obwohl die Fixtures in den Frames Licht emittierten und eine Farbe besaßen. Durch die automatische Erstellung der Bilder auf Basis der Steuerdaten wurden somit falsche Klassifikationen der Trainingsdaten erstellt. Aus diesem Grund musste dieser Datensatz ebenfalls manuell überarbeitet werden.

5 Training

Das Training des „Detection Models“ und der „Classification Models“ ist ein zentraler Punkt dieser Bachelorthesis. Durch eine YAML-Datei können bei den „Ultralytics Yolov8 Models“ insgesamt 106 Parameter vor dem Beginn des eigentlichen Trainings eingestellt werden[22]. Die Parameter beinhalten Meta-Daten, wie übergeordnete Projektnamen und Speicherpfade. Andere Parameter können zu logischen Gruppen zusammengefasst werden, die als gesonderte Unterkapitel betrachtet werden. Beispielbilder für das Training der „Models“ können dem Anhang A.8 entnommen werden.

5.1 Activation Functions

Die „Activation Function“ ist eine mathematische Funktion, die zur Berechnung der Ausgabewerte der einzelnen Neuronen in einem neuronalen Netz verwendet wird. Durch eine nicht lineare Transformation ermöglicht die „Activation Function“ das Lernen von komplexen und abstrakten Mustern und Merkmalen der Eingabedaten. Dabei sollte sich die Komplexität der Berechnung der Ausgabewerte nicht erhöhen. Mittlerweile gibt es eine Vielzahl an unterschiedlichen „Activation Functions“ mit eigenen Vor- und Nachteilen, die je nach Anwendungsfall verwendet werden können.

Die „YOLOv8 Detection Models“ verwenden eine „Sigmoid-weighted Linear Unit Activation Function“ (SiLU) bzw. die „Swish“-Funktion. Die beiden Funktionen sind hierbei als äquivalent anzusehen. Im Gegensatz zu vorherigen Versionen der „Models“, die eine „Leaky Rectified Linear Unit“ (Leaky ReLU) Funktion verwendet haben, wurde durch die Verwendung der „Swish“-Funktion der Gradientenfluss und die Merkmalsextraktion verbessert. Die ursprüngliche „Activation Function“ wurde nicht verändert, da ein bereits trainiertes Model für diese Anwendung verwendet wurde[14][18] .

5.2 Loss Functions

Die „Loss Function“ ist eine mathematische Funktion die zur Berechnung des Unterschieds zwischen den numerischen Vorhersagen eines „Models“ und der auf reellen Zahlen abgebildeten tatsächlichen Werte. Der Unterschied zwischen diesen Werten soll möglichst gering sein. Daher ist eine Minimierung der „Loss Function“ ein wichtiges Ziel beim Training eines „Models“. Es gibt eine Vielzahl an unterschiedlichen „Loss Functions“. Die Auswahl der „Loss Function“ sollte auf die Art des zu lösenden Problems, die verfügbaren Daten und die Art des verwendeten „Models“ abgestimmt werden[3].

Die „YOLOv8 Detection Models“ verwenden die „Complete-Intersection-over-Union“ (CIoU) Funktion. Durch die Verwendung von „CIoU“ wird die Erkennung von „Bounding Boxes“ verbessert, sowie die Klassifikation der Objekte präziser. Die ursprüngliche „Loss Function“ wurde nicht verändert, da ein bereits trainiertes Model für diese Anwendung verwendet wurde[14].

5.3 Batch Size

Die „Batch Size“ gibt an, wie viele Trainingsdaten in einer Iteration in das „Model“ gegeben werden. Eine große „Batch Size“ führt zu einer erhöhten Trainingsgeschwindigkeit, da mehr Trainingsdaten in einem Iterationsschritt verarbeitet werden. Gleichzeitig führt eine große „Batch Size“ zu einem erhöhten Speicherbedarf und somit zu einer größeren Ressourcenanforderung. Somit ist die Wahl der „Batch Size“ abhängig von der Verfügbarkeit der Ressourcen und der Größe des Trainingsdatensatzes. Des Weiteren führt eine größere „Batch Size“ zu einer höheren Genauigkeit des Models und eine kleinere „Batch Size“ zu einem höheren Rauschen in dem Lernprozess des Models[8]. Das Rauschen bei einer kleineren „Batch Size“ kann einen regulierenden Effekt auf das Training haben und eine Maßnahme gegen „Overfitting“ sein[17].

Als Standardwert war eine „Batch Size“ von 16 in der Konfigurationsdatei eingetragen. Durch die Größe des Trainingsdatensatzes konnte eine Trainingsepoche bei Google Colabs mehrere Stunden dauern. Durch eine Erhöhung der „Batch Size“ auf 128 konnte die Trainingsdauer durchschnittlich auf weniger als eine Stunde pro Epoche reduziert werden.

5.4 Image Preperation

Die Bilder können für das Training eines „Models“ vorbereitet werden. Damit der Trainingsdatensatz konsistent ist, sollten die Eingabebilder für das Trainings eines neuronalen Netzes dieselbe Größe haben. Weiterführend ist es sinnvoll die Farbwerte der Bilder zu normalisieren. Dazu werden die Farbwerte der Bilder in einen Bereich zwischen 0 und 1 umgewandelt. Durch die Verkleinerung der Werte verringert sich die numerische Darstellung der unterschiedlichen Merkmale der Bilder. Das führt zu einem verbesserten Lernprozess des „Models“[11].

Für das verwendete „YOLOv8 Detection Model“ wurde die vorgeschlagene Bildergröße von 640x640 Pixeln in der Konfigurationsdatei beibehalten. Die Auflösung der Videodaten wurde bei dem Umwandeln in Bilddaten beibehalten. Somit besaß jedes Bild des Trainingsdatensatzes eine Auflösung von 1920x1080 Pixeln. Durch ein Beibehalten des Parameters wurde jedes Bild automatisch beim Training auf eine Größe von 640x640 herunterskaliert.

Bei den YOLOv8 Classification-Models wurde ebenfalls die vorgeschlagene Größe von 64x64 Pixeln beibehalten. Die ursprüngliche Größe der Bilder des Datensatzes variieren, da sie abhängig von den gesetzten „Bounding Boxes“ sind.

Die Normalisierung der Bilder wird bei dem Beginn des Trainings von den „YOLO Models“ automatisch durchgeführt.

5.5 Segmentation Mask

Durch die Verwendung einer Segmentierungsmaske beim Training eines „Models“ wird das Bild in einzelne Segmente aufgeteilt. Die Verarbeitung der einzelnen Segmente lässt das „Model“ markante Merkmale eines Objektes besser lernen. Durch das Lernen der einzelnen Segmente erhöht sich die Genauigkeit des Models beim Erkennen und Klassifizieren von Objekten. Vor allem Objekte, die nur teilweise sichtbar sind, können besser erkannt und klassifiziert werden. Das ist besonders bei einer Überschneidung von zwei zu erkennenden Objekten hilfreich, da beide Objekte besser erkannt werden[16].

Bei dem Training der „Models“ wurde jedes Bild in vier Segmente aufgeteilt. Die „Bounding Boxes“ der einzelnen Fixtures sind disjunkt, wodurch keine Überschneidung von zwei zu erkennenden Objekten stattfinden kann. Dennoch ist es vorgekommen, dass der

Lichtkegel eines Fixtures teilweise ein anderes Fixture verdeckt. Daher war es sinnvoll eine Segmentierungsmaske für diese Anwendung zu verwenden.

5.6 Data Augmentation

Durch die Verwendung von „Data Augmentation“ wird durch das Verändern, Erweitern und Wegnehmen von Bildparametern der Datensatz diversifiziert. Es können Änderungen an der Perspektive des Bildes vorgenommen werden, indem die Bilder gespiegelt, neu skaliert, gedreht oder verschoben werden. Die Farbwerte des Datensatzes können durch eine Veränderung des Kontrastes, der Helligkeit oder der Sättigung vorgenommen werden. Teilweise können auch die einzelnen Farbwerte verändert werden. Des Weiteren kann die Qualität des Bildes verändert werden, indem zufällige Bereiche des Bildes ausgeschnitten werden. Bei Datensätzen, die nicht unter realen Bedingungen der Anwendung aufgenommen wurden, kann ein hinzufügen von Rauschen zu einem realistischeren Bild führen. Die Verwendung von Data-Augmentation bietet eine einfache Möglichkeit der Erweiterung eines Datensatzes. Darüber hinaus kann dem „Overfitting“ eines Modells entgegengewirkt werden und die Genauigkeit eines Modells erhöht werden[26].

Sowohl beim Training des „Detection Models“ als auch beim Training der „Classification Models“ wurde „Data Augmentation“ verwendet. Es wurden Änderungen an den Farb-, Kontrast-, und Helligkeitswerten vorgenommen. Des Weiteren wurden die Bilder gespiegelt, neu skaliert und zufällige Bereiche des Bildes abgeschnitten oder geschwärzt.

5.7 Learning Rate

Die „Learning Rate“ beschreibt die Größe der Schritte, die bei der Anpassung der Gewichte beim Training eines „Modells“ vorgenommen werden. Der Wert der „Learning Rate“ ist ein wichtiger Parameter beim Training eines „Modells“. Wird der Wert zu klein gewählt kann sich das Training eines „Modells“ erheblich verlangsamen. Durch eine zu hohe „Learning Rate“ können die Schritte bei der Gewichtsanzpassung zu groß sein, wodurch das „Modell“ instabil werden kann. Somit sorgt die Auswahl einer geeigneten „Learning Rate“ für eine schnellere Konvergenz des „Modells“[19].

Für das Training der „Models“ wurde die vorgeschlagene „Learning Rate“ von 0,01 beibehalten. Für eine verbesserte Konvergenz des „Models“ wurde im Laufe der Trainingsdurchläufe die „Learning Rate“ um den Faktor 0,01 verkleinert. Überdies gibt es eine „Warmup Phase“, in der die „Learning Rate“ für drei Epochen langsam erhöht wird, damit das Training stabilisiert wird.

5.8 Optimizer

Der „Optimizer“ ist eine wichtige Komponente für das Training eines neuronalen Netzes. Dabei handelt es sich um einen zentralen Algorithmus, der die Gewichte des „Models“ aktualisiert, um ein Minimum in der Verlustfunktion zu erreichen. Für die Verbesserung der Performance eines „Models“ sollte der Optimizer sorgfältig ausgewählt werden[21].

Bekannte „Optimizer“ sind die „Stochastic Gradient Descent“ (SGD), „AdaGrad“, „RMSProp“ oder „Adaptive Moment Estimation“ (Adam) Algorithmen. Der „SGD“-Algorithmus verwendet pro Iteration eine zufällige Instanz aus dem Datensatz zur Aktualisierung des Gradienten. Durch die zufällige Auswahl wird die Aktualisierungszeit bei großen Datenmengen verringert, sowie redundante Berechnungen vermieden. Eine Weiterentwicklung des „SGD“ ist der „AdaGrad“-Algorithmus. Anstatt einer festen „Learning Rate“ wurde bei diesem eine dynamische „Learning Rate“ eingeführt, die sich auf Basis der vergangenen Gradienten nach jeder Iteration aktualisiert. Durch die Verwendung des „AdaGrad“-Algorithmus muss die „Learning Rate“ nicht mehr manuell angepasst werden. Es ist jedoch möglich, dass sich die adaptive „Learning Rate“ zu schnell verkleinert. Der „RMSProp“-Algorithmus wirkt dem Effekt entgegen. Für die Berechnung der „Learning Rate“ werden lediglich eine bestimmte Anzahl an vergangenen Gradienten verwendet. „Adam“ ist ein weiterer „Optimizer“ der eine adaptive „Learning Rate“ verwendet. Für die Berechnung der aktuellen Gradienten Anpassung wird ein Momentum verwendet. Durch das Momentum kann eine schnellere Konvergenz erzielt werden[21].

Für das Training der Models wurden der „AdamW“, eine Weiterentwicklung von „Adam“-Algorithmus und der „SGD“-Algorithmus verwendet. Für die ersten 10.000 Iterationen wird „AdamW“ verwendet. Anschließend wechselt das „Model“ zum „SGD-Optimizer“. In einer frühen Trainingsphase führt die Verwendung von „AdamW“ zu einer hohen Konvergenz. Für eine bessere Feinabstimmung und Leistung wechselt der „Optimizer“ am Ende zu „SGD“. Diese Entscheidung beruht auf den Erfahrungen von Experimenten der Entwickler des YOLOv8 „Models“[7].

5.9 Regularization

„Regularization“-Methoden werden beim Training eines „Models“ eingesetzt, um eine verbesserte Generalisierung zu erreichen. Des Weiteren können sie einem möglichen „Overfitting“ des „Models“ entgegenwirken. Die „L2-Regularization“ erweitert die Verlustfunktion um einen Term der zu hohe Gewichtswerte bestraft. Dadurch wird das Risiko eines „Overfittings“ des Models verkleinert. Eine weitere Methode der „Regularization“ ist der „Dropout“, bei dem mit einer Wahrscheinlichkeit einzelne Neuronen aus dem neuronalen Netz während des Trainings deaktiviert werden. Durch die Verwendung von „Dropout“ kann ebenfalls einem möglichen „Overfitting“ entgegengewirkt werden[10].

Um überdies ein mögliches „Overfitting“ zu verhindern, wurde eine „L2-Regularization“ beim Training der Models verwendet. Aufgrund der Größe des Trainingsdatensatzes wurde kein „Dropout“ verwendet, da von keiner Verbesserung der Genauigkeit ausgegangen werden kann[20].

6 Anwendung

Die Anwendung teilt sich hauptsächlich in zwei Python-Skripte auf, die jeweils über eine eigene YAML-Datei konfiguriert werden können.

Das „Dataset_preprocessor.py“-Skript erstellt aus einer Sammlung an Bildern und der dazugehörigen PKL-Datei, analog zu Kapitel 4.1, automatisch einen Datensatz für das „Detection Model“ und die „Classification Models“. Der erstellte Datensatz entspricht den Voraussetzungen für das Training von YOLOv8 „Models“.

Das „Control_data_generator.py“ Skript erstellt aus einer Sammlung von zusammenhängenden Frames von Bühnenlichtern mit Hilfe eines trainierten YOLOv8 „Detection Model“ und „Classification Models“ eine Lichtsteuerdatei in Form einer PKL-Datei. Neben den Pfaden für den Datensatz und den „Models“ müssen lediglich die Koordinaten der einzelnen Fixture-Gruppen in die Konfigurationsdatei eingetragen werden. Die Koordinaten werden benötigt, damit die erkannten Fixtures des „Detection Models“ den richtigen Gruppen zugeordnet werden können. Die Anzahl der Fixtures pro Gruppe und deren Positionen innerhalb der Gruppe werden auf Basis von durchschnittlichen Koordinaten pro erkanntem Fixture vorgenommen. Sollte ein Fixture vom „Detection Model“ nicht erkannt werden, befindet sich in der erstellten PKL-Datei für dieses Fixture ein Dummy-Eintrag mit negativen Werten. Diese Einträge stellen sicher, dass die Anzahl der Einträge pro Fixture der gesamten Anzahl an Bildern entspricht. Des Weiteren bleiben durch die Dummy-Einträge die Steuerdaten insgesamt synchron.

Eine Anleitung sowie ein Link zu einer vorbereiteten Umgebung können dem Anhang A.2 entnommen werden.

7 Ergebnisse

Die Beurteilung der Ergebnisse dieser Arbeit besteht aus zwei Schritten. Im Folgenden werden zunächst die Trainingserfolge der einzelnen „Models“ ausgewertet. Anschließend werden die automatisch erstellten Steuerdaten auf Basis eines für die „Models“ unbekanntes Datensatzes durch Verwendung der „Ground Truth“ PKL-Datei validiert.

7.1 Auswertung des Fixture Detection Models

Das „Fixture Detection Model“ wurde für acht Epochen auf einem Datensatz mit 6.393 Bildern trainiert. Insgesamt sollten drei unterschiedliche Fixture Typen gelernt werden. Auf den Testdaten erzielt das „Model“ eine Genauigkeit von 99,6%.

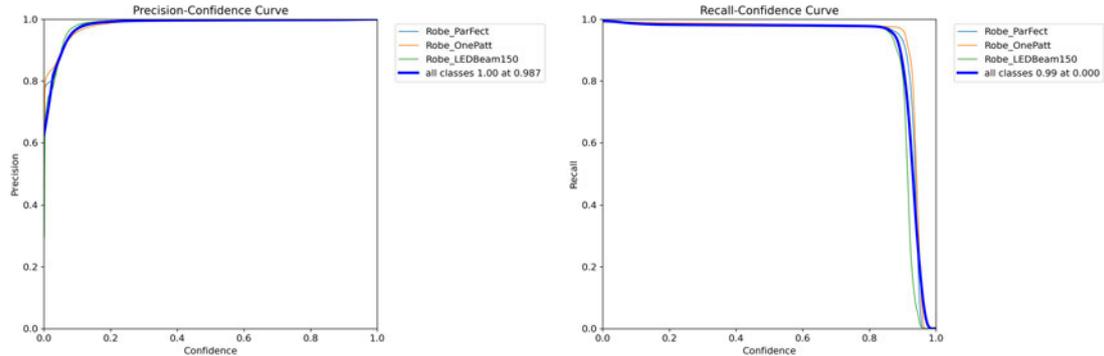


Abbildung 7.1: Verlauf der Precision bzw. des Recalls zu der Confidence des „Detection Models“.

Aus Abbildung 7.1 lässt sich die „Precision“ und der „Recall“ gegenüber der „Confidence“ ablesen. Ab einer „Confidence“ von ca. 20% erzielt das „Model“ eine „Precision“, also der Anteil an tatsächlich richtigen Vorhersagen unter allen als richtig markierten Vorhersagen, von mehr als 95%. In dem Bereich der „Confidence“ von 0% bis ca. 20% erreicht das „Model“ eine „Precision“ von ca. 60% bis 95%. Der „Recall“, der Anteil an richtigen

Vorhersagen unter allen möglichen richtigen Instanzen, liegt bis zu einer „Confidence“ von ca. 85% bei über 95%. Ab dieser „Confidence“ beginnt der „Recall“ deutlich zu sinken.

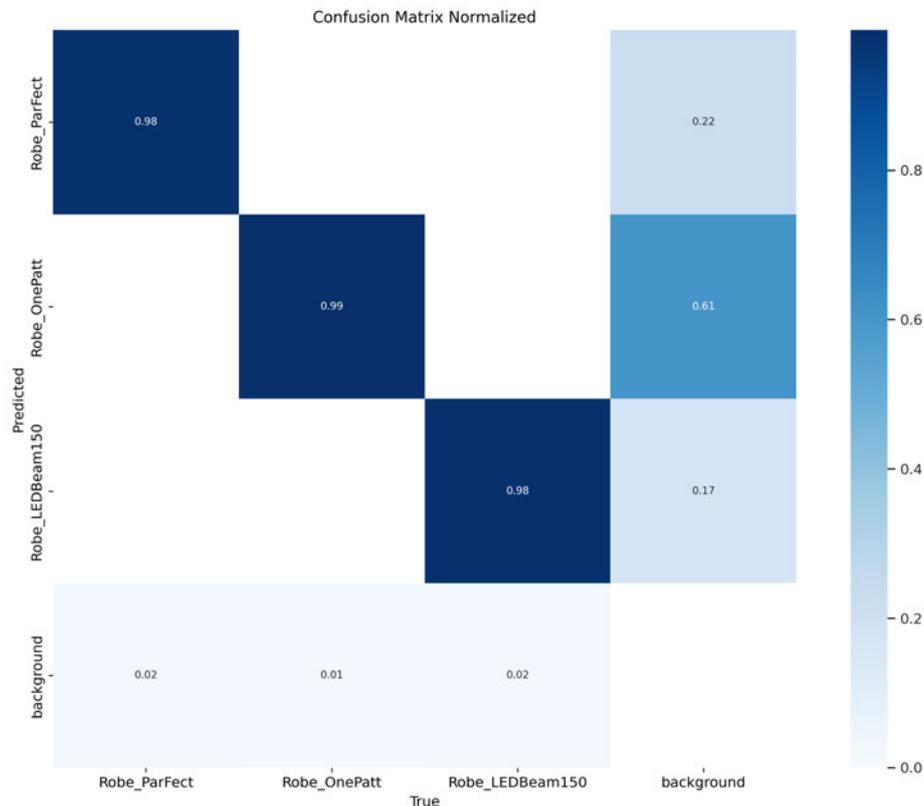


Abbildung 7.2: Die normalisierte „Confusion Matrix“ für die Trainingsergebnisse des „Detection Models“.

Die Werte der „Confusion Matrix“ aus Abbildung 7.2 sind, unter Betrachtung der Ergebnisse, nicht überraschend. Lediglich die falschen Vorhersagen der Robe OnePatt Fixtures sind genauer zu betrachten. Zu 61% hat das „Model“ diese Fixtures fälschlicherweise im Hintergrund erkannt. Die Robe OnePatt Fixtures befinden sich in der Mitte des Bühnenbildes, wodurch sie häufig von den Lichtkegeln der anderen Fixtures überstrahlt werden. Da diese Situation nahezu unvermeidbar ist und um die Robustheit des „Models“ zu erhöhen, befinden sich Bilder mit dieser Situation in dem Trainingsdatensatz des „Models“. Das könnte fälschlicherweise zu einer Bestimmung dieses Fixtures führen, obwohl lediglich der Hintergrund des Bildes zu sehen ist.

Dieses „Model“ ist eine wichtige Komponente in der erstellten Anwendung, da lediglich solche Fixtures von den anderen „Models“ klassifiziert werden können, die zuvor von

diesem „Model“ erkannt wurden. Daher ist eine hohe Genauigkeit insbesondere bei diesem „Model“ wichtig und im vorliegenden Fall zufriedenstellend.

7.2 Auswertung des ParFect Classification Models

Das „Classification Model“ für die Robe ParFect Fixtures wurde für 18 Epochen auf einem Trainingsdatensatz von 50.188 Bildern mit 24 unterschiedlichen Klassen trainiert. Aus Abbildung 7.3 lässt sich der Trainingsverlauf der letzten zehn Epochen ablesen. Auf den Testdaten erzielte das „Model“ eine Top1-Genauigkeit von 95,9% und eine Top5-Genauigkeit von 99,9%. Darüberhinaus ist zu sehen, wie der Verlust bei fortlaufendem Training abgenommen hat.

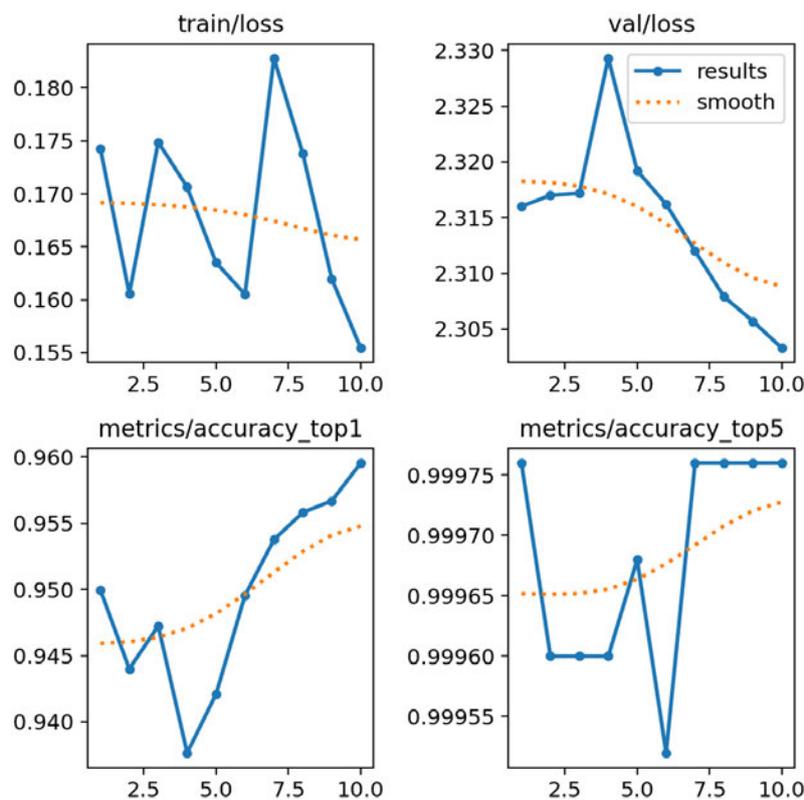


Abbildung 7.3: Der Trainingsverlauf der letzten zehn Epochen für das Robe ParFect „Classification Model“. Es zeigt die Veränderung des Verlustes und der Genauigkeit.

Die Ergebnisse der Confusion Matrix aus Abbildung 7.4 lassen sich mit Hilfe des Datensatzes erklären. Bei einem Großteil der Daten besaßen die Robe ParFect Fixtures eine feste Farbe mit einer Intensität von 100% oder sie waren ausgeschaltet. Der ausgeschaltete Zustand (NC_0) wurde zu 100% erkannt. Die Zustände mit einer Intensität von 100% haben mindestens eine Genauigkeit von 95% erreicht. Eine Ausnahme bildet die Familie der Farbe Weiß, bei der lediglich eine Genauigkeit von 84% bei einer Intensität von 100% erreicht wurde. Es ist auffällig, dass die Zustände mit einer Intensität von 80% als der Zustand mit derselben Farbe aber mit einer Intensität von 100% klassifiziert worden sind. Das könnte ein Indiz für ein mögliches „Overfitting“ des „Modells“ in den hohen Intensitätszuständen sein.

Grundsätzlich kann das „Modell“ die Farben gut vorhersagen. Ausgenommen von den Zuständen mit einer Intensität von 100% zeigt das „Modell“ aber Schwächen bei der genauen Bestimmung der Intensitäten.

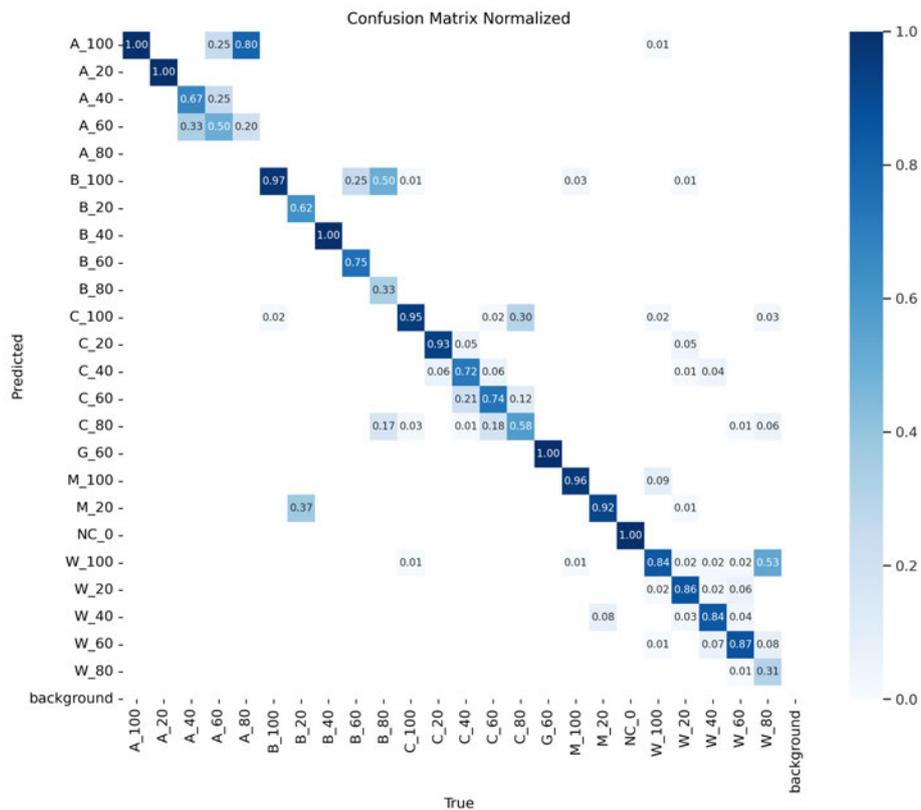


Abbildung 7.4: Die normalisierte Confusion-Matrix für die Trainingsergebnisse des Robe ParFect „Classification Model“.

7.3 Auswertung des Beam Classification Models

Das „Classification Model“ für die Robe LEDBeam150 Fixtures wurde für 20 Epochen auf einem Trainingsdatensatz von 33.450 Bildern mit 33 unterschiedlichen Klassen trainiert. Aus Abbildung 7.5 lässt sich der Trainingsverlauf der 20 Epochen ablesen. Auf den Testdaten erzielte das „Model“ eine Top1-Genauigkeit von 95,7% und eine Top5-Genauigkeit von 99,8%.

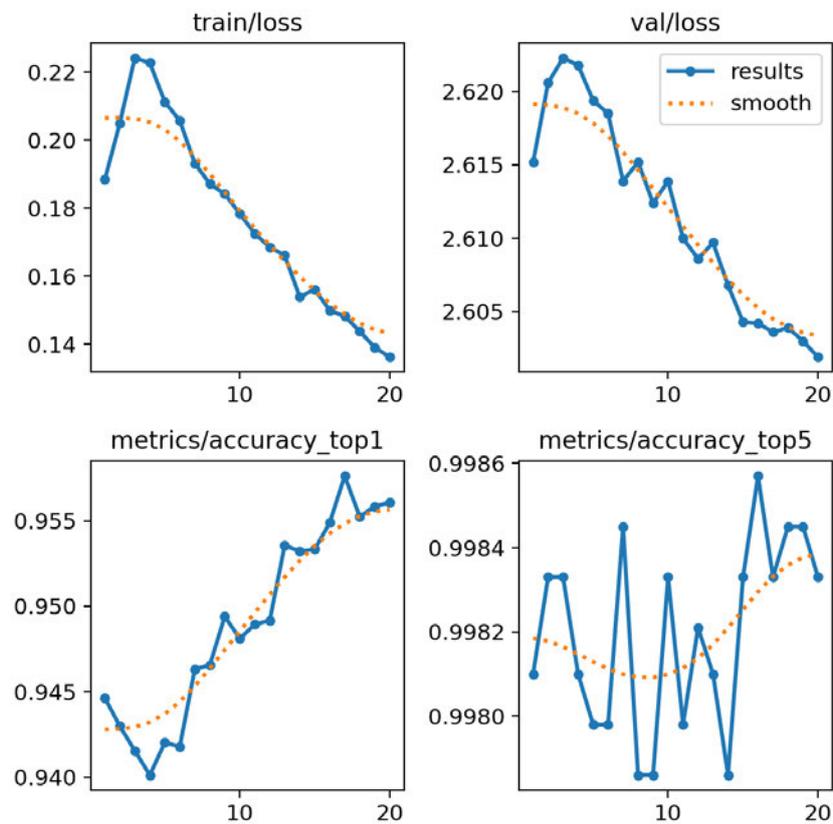


Abbildung 7.5: Der Trainingsverlauf der letzten 20 Epochen für das Robe LEDBeam 150 „Classification Model“. Es zeigt die Veränderung des Verlustes und der Genauigkeit.

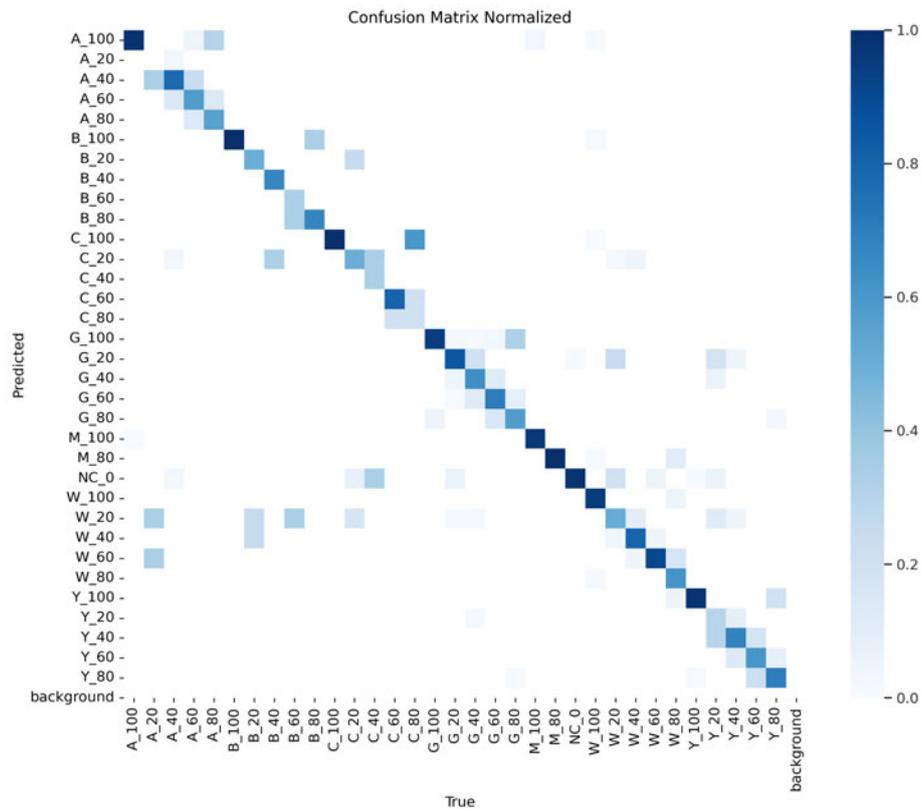


Abbildung 7.6: Die normalisierte Confusion-Matrix für die Trainingsergebnisse des Robe LEDBeam 150 „Classification Model“.

Grundsätzlich war der Datensatz für das „Classification Model“ anspruchsvoller zu lernen, da es sich bei den Robe LEDBem150 Fixtures um Moving Heads, also bewegtes Licht, handelt. Die Fixture-Köpfe mussten folglich aus unterschiedlichen Perspektiven klassifiziert werden. Die Ergebnisse der Confusion-Matrix aus Abbildung 7.6 lassen sich zunächst analog zu den Ergebnissen des Robe ParFect „Classification Models“ erklären. Ein Großteil des Datensatzes bestand aus Zuständen mit einer Intensität von 100% oder einem ausgeschalteten (NC_0) Fixture. In diesen Fällen liegt die Genauigkeit des „Models“ bei über 90%. Weiterhin ist auffällig, dass einige Zustände mit einer Intensität von 80% fälschlicherweise als ein Zustand mit einer Intensität von 100% klassifiziert wurden. Da die Farben bei den Klassifikationen korrekt vorausgesagt wurden, könnte dies ebenfalls ein Indiz für ein mögliches „Overfitting“ bei hohen Intensitäten sein. Auffällig sind die falschen Voraussagen des Zustandes W_20. Hier wurden vor allem die Farben falsch klassifiziert. Der Grund könnte die geringe Menge an Daten in dem Intensitäts-

bereich in Kombination mit den unterschiedlichen Perspektiven durch die Bewegung der Fixture-Köpfe sein.

7.4 Auswertung des OnePatt Classification Models

Das „Classification Model“ für die Robe OnePatt Fixtures wurde für 28 Epochen auf einem Trainingsdatensatz von 39.713 Bildern mit 30 unterschiedlichen Klassen trainiert. Aus Abbildung 7.7 lässt sich der Trainingsverlauf der letzten zehn Epochen ablesen. Auf den Testdaten erzielte das „Model“ eine Top1-Genauigkeit von 93,1% und eine Top5-Genauigkeit von 99,8%.

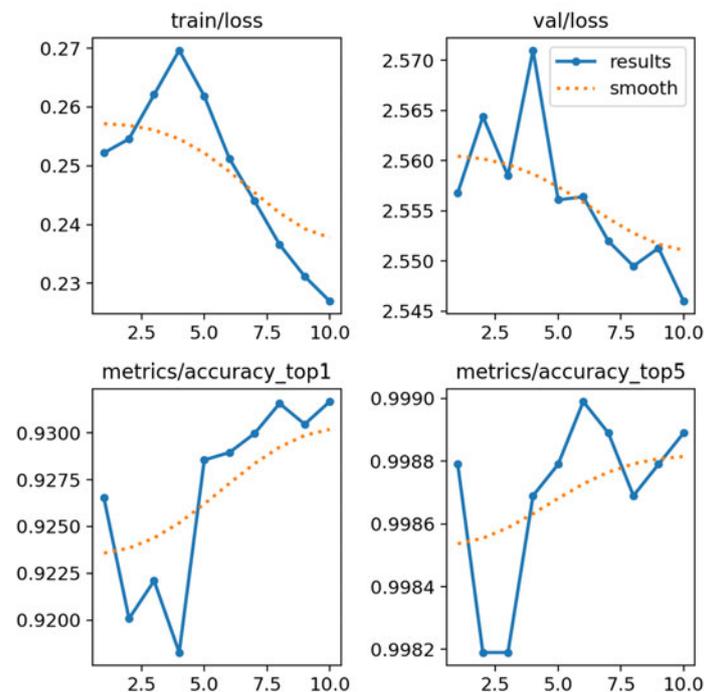


Abbildung 7.7: Der Trainingsverlauf der letzten 20 Epochen für das Robe OnePatt „Classification Model“. Es zeigt die Veränderung des Verlustes und der Genauigkeit.

Analog zu den anderen „Classification Models“ wurde bei den Zuständen mit einer Intensität von 100% und dem NC_0 Zustand eine Genauigkeit von über 90% erzielt. Das lässt sich ebenfalls durch die Verteilung der Klassen im Datensatz begründen. In Abbildung 7.8 sind die Häufungen an falschen Voraussagen der Zustände B_100, C_100 und

Y_100 auffällig. Aufgrund der künstlichen Trägheit der Leuchtmittel des Fixtures und der automatischen Erstellung der Trainingsdaten war der Datensatz für das Training dieses „Models“ der unsauberste. Auch nach einer manuellen Überarbeitung konnte es nicht komplett ausgeschlossen werden, dass Bilder falsch für das Training gelabelt wurden. Neben der Verteilung der Klassen im Datensatz könnte das ein Hauptgrund für die Ungenauigkeit in dem Bereich sein.

Des Weiteren ist das Cluster im Bereich der Farbe Weiß mit niedrigen Intensitäten auffällig. In der Realität ist in diesem Intensitätsbereich nur ein Glimmen des Leuchtmittels zu sehen, mit der Folge, dass die Farben für das „Model“ schwer erkennbar ist. Ein analoger Effekt scheint bei dem „Model“ aufgetreten zu sein.

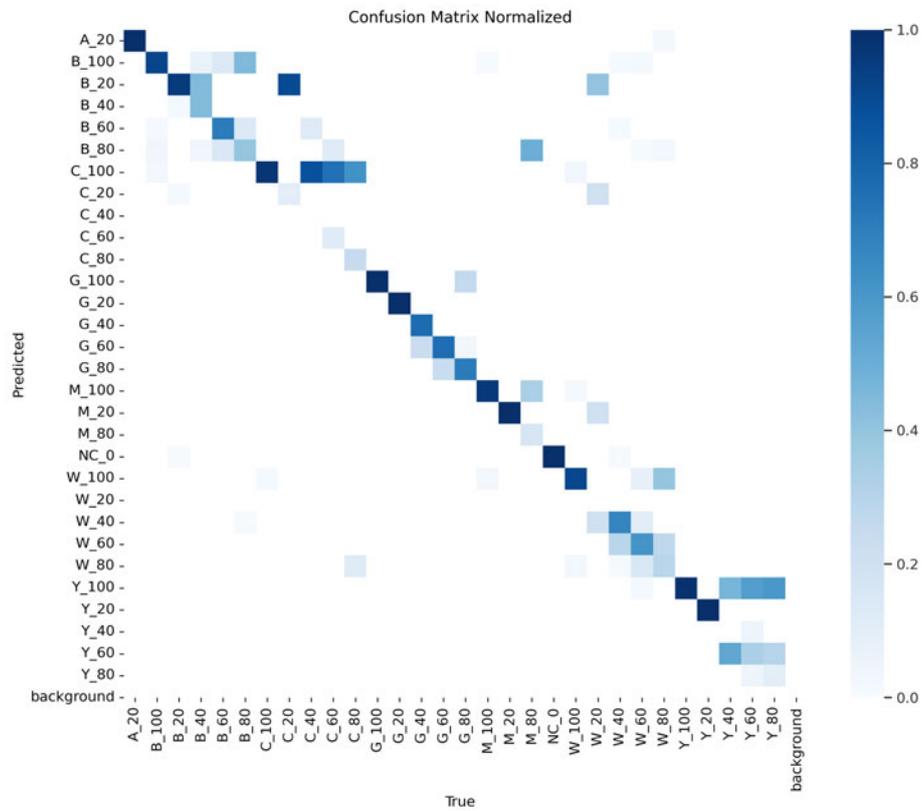


Abbildung 7.8: Die normalisierte Confusion-Matrix für die Trainingsergebnisse des Robe OnePatt „Classification Model“.

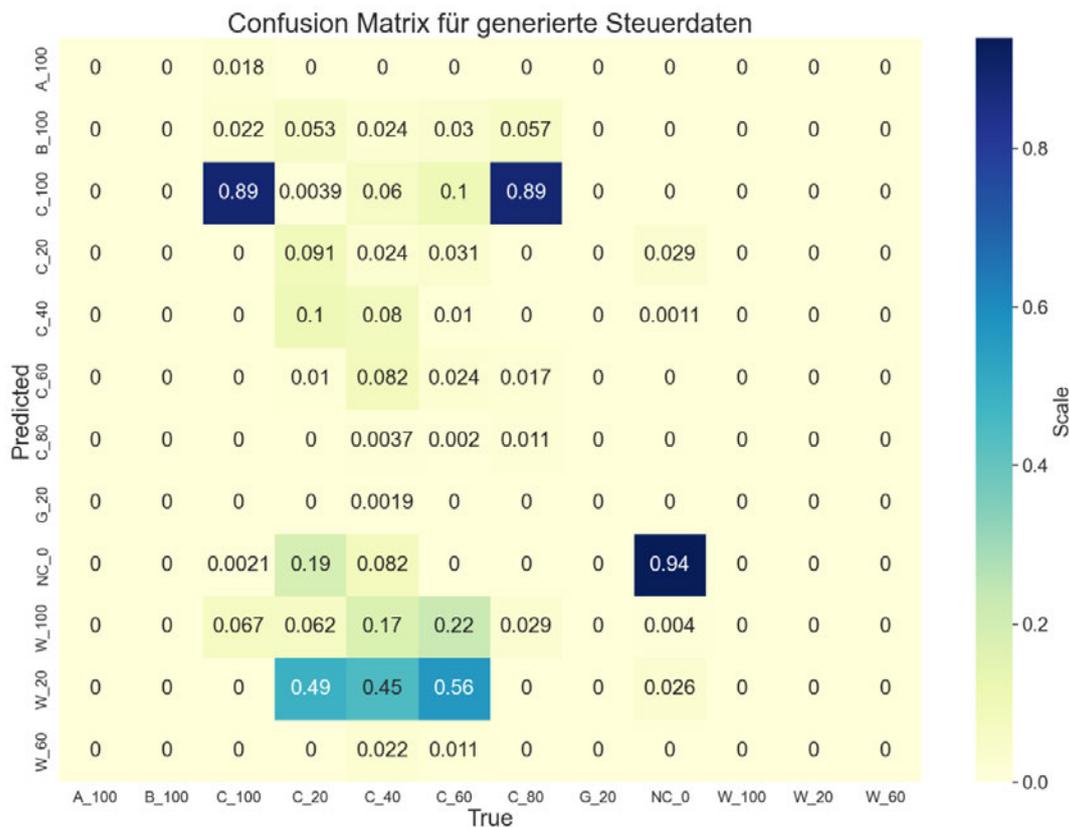


Abbildung 7.10: Die normalisierte Confusion Matrix für die Klassifikationsergebnisse der automatisch erstellten Steuerdaten.

Anschließend wurden die Parameter für die Steuerdaten der von dem „Detection Model“ erkannten Fixtures von den „Classification Models“ bestimmt. Insgesamt wurde 11.505 Fixtures klassifiziert. Dabei wurde eine Genauigkeit von 72,5% erreicht. Der Wert liegt unterhalb der Erwartung nach der Evaluation der Trainingsergebnisse der einzelnen „Classification Models“.

Aus Abbildung 7.10 kann entnommen werden, dass analog zu den Trainingsergebnissen der „Classification Models“ eine hohe Genauigkeit bei den Parametern mit einer Intensität von 100% erzielt wurde. Des Weiteren wurde mit einer Genauigkeit von 94% die „Fixtures“ im ausgeschalteten Zustand erkannt. Schwierigkeiten gab es bei der Klassifizierung mit einer Intensität von 80%. Diese Zustände wurden zu 89% als ein Zustand mit einer Intensität von 100% klassifiziert. Weniger ausgeprägt war dieser Effekt bereits bei den Trainingsergebnissen der einzelnen „Classification Models“ zu beobachten. Dem

„Classification Model“ für die Robe LED Beam Fixtures scheint eine Klassifikation der Farbe Cyan in diesem Intensitätsbereich ein Problem darzustellen. Das war ebenfalls bei den Trainingsergebnissen diese „Models“ erkennbar.

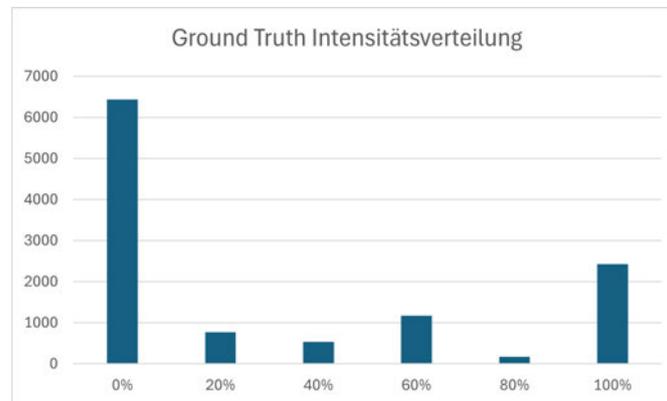


Abbildung 7.11: Die Verteilung der Intensitäten aus der Sequenz die für die automatische Generierung von Steuerdaten verwendet wurde.

Die Bereiche mit einer Intensität von 20% bis 60% wurden als ein Zustand mit einer geringen Intensität erkannt. Diese Zustände wurden zu ungefähr 50% als W_20 klassifiziert. Die Ungenauigkeit im Bereich der niedrigen Intensitäten deckt sich mit den Trainingsergebnissen und lässt sich auf die wenigen Trainingsdaten in diesem Bereich zurückführen[27]. In Abbildung 7.11 ist zu sehen, dass ca. 20% der „Ground Truth“ Parameter der Sequenz eine Intensität von 20% bis 60% haben. Die Anzahl an Zuständen in diesem Intensitätsbereich erklären den Verlust von Prozentpunkten in der Genauigkeit der „Classification Models“. Interessant ist die falsche Klassifizierung der Farbe. Dieser Effekt trat bei den Trainingsergebnissen nicht in dieser Ausprägung auf.

Bei der Betrachtung Der Confusion-Matrix in Abbildung 7.10 lässt sich die Klassifizierung der Farbe für das menschliche Auge nachvollziehen. Für die Klassifizierung wurde lediglich der Bereich der einzelnen „Bounding Boxes“ verwendet. Das austretende Licht in diesem Bereich kann fälschlicherweise als Weiß angenommen werden. Erst bei der Betrachtung der Lichtstrahlen lässt sich eindeutig die Farbe Cyan erkennen. Da sich dieses Problem nicht in den Trainingsergebnissen widerspiegelt scheinen die „Classification Models“ in dieser Lichtstimmung größere Schwierigkeiten in der Klassifikation zu haben. Eine Maßnahme gegen diesen Effekt könnte eine Vergrößerung der „Bounding Boxes“ für die Klassifizierung sein. Dadurch wäre ein größerer Teil des Lichtstrahles und somit eine größere Abbildung der Farbe für die Klassifizierung vorhanden.

8 Erweiterung und Optimierung

Verbesserter Datensatz: Der zugrunde gelegte Datensatz für das Training der Models könnte verbessert werden. Denn die quantitative Aufteilung der einzelnen Zustände der Fixtures in dem verwendeten Datensatz war unausgeglichen. Ebenso war die Verteilung der Farben in dem Datensatz unausgeglichen, wobei die Farbe Rot gar nicht in dem Datensatz abgebildet wurde. Zudem waren vor allem die Zustände mit einer geringeren Intensität unterrepräsentiert. Für dieses Training könnte ein Datensatz erstellt werden, der nicht aus tatsächlichen Lichtsteuerdaten besteht. Ein Fixture-Typ könnte einen festen Farb- und Intensitätswert haben, während die anderen Fixtures durch ihre unterschiedlichen Parameter ein Umgebungsrauschen in den Datensatz hineingeben. Dadurch können mehr Daten für die unterrepräsentierten Zustände mit einer niedrigen Intensität erstellt werden. Des Weiteren würde das Training von „Models“ für Fixtures mit einem trägen Leuchtmittel, wie die Robe OnePatts, davon profitieren.

Experimente mit Trainingsparametern: Die Trainingsergebnisse der einzelnen „Models“ waren zufriedenstellend. Für eine mögliche Verbesserung der Genauigkeit könnten weitere Experimente mit unterschiedlichen Kombinationen der in Kapitel 5 beschriebenen Parametern durchgeführt werden.

Models für weitere Parameter: Die Anwendung könnte um „Models“ für weitere Parameter erweitert werden. Fixture wie die Robe LEDBeams150 gehören zu der Familie der Moving Heads. Daher könnte ein „Model“ für die Pan- und Tilt-Werte dieses Fixtures trainiert und implementiert werden. Häufig haben Fixtures eingebaute Gobos mit denen eine Effektstruktur in ihrem Lichtkegel erzeugt werden kann. Es ist ebenfalls denkbar für diesen Parameter ein weiteres „Model“ zu trainieren.

Generische Models: Momentan sind die „Models“ für einzelne Fixture-Typen trainiert worden. Es ist denkbar, dass die „Models“ für Fixture-Familien verallgemeinert werden könnten. Häufige Fixture-Familien sind beispielsweise Wash, Spot oder Beam. Anstatt ein „Model“ für jeden Fixture-Typen der Familie Wash zu trainieren könnte ein „Model“

auf die Klassifizierung von Wash-Fixtures trainiert werden. Von Ausnahmen ausgenommen unterscheiden sich die Fixtures meistens nur in Form, Größe, Farbdarstellung und Intensität. Inwieweit eine Verallgemeinerung der „Models“ mit Blick auf die Genauigkeit sinnvoll ist, muss untersucht werden.

Eine analoge Überlegung kann bei dem „Detection Model“ angestellt werden: Das „Model“ könnte entweder Fixtures einer Familie erkennen oder weiterhin auf einzelne Fixtures trainiert werden. Dann müsste die Anwendung um eine Bibliothek an Fixture-Familien erweitert werden, damit die „Classification Models“ die erkannten Fixtures klassifizieren können.

Feinere Parameterabstufungen: Für eine verbesserte Qualität der generierten Steuerdaten könnten Experimente mit feineren Parameterabstufungen durchgeführt werden. Vor Allem ist es denkbar, dass kleinere Intensitätsintervalle zu verbesserten und realistischeren Effekten führen. Überdies ist eine erweiterte Abstufung der Farben denkbar. Für diese Erweiterung wäre eine Verbesserung des Datensatzes sinnvoll.

Model für die Robe TetraBar RGBPart: Dieses Fixture unterscheidet sich in seinen Eigenschaften von den anderen Fixtures dieser Bachelorthesis. Es besteht aus 18 einzelnen, nebeneinander liegenden LEDs. Jede LED kann einzeln angesteuert werden und somit eine eigene Intensität und Farbe haben. Das Erkennen dieser Fixtures durch das „Detection Model“ sollte problemlos funktionieren. Die Klassifizierung eines einzelnen „Classification Models“ wäre nur möglich, wenn alle LEDs dieselben Farb- und Intensitätsparameter haben. Eine Möglichkeit wäre die Verwendung eines „Classification Models“ für jede LED des Fixtures. Da dieses Fixture zur Familie des bewegten Lichts gehört, sind in der Praxis nicht immer alle LEDs zu sehen. Daher wird eine Klassifizierung mit einer zufriedenstellenden Genauigkeit schwierig sein. Es könnte ein „Model“ für die Bestimmung der Muster an unterschiedlichen Intensitäten der einzelnen LEDs trainiert werden. Weitere „Models“ könnten unterschiedliche Farben der Muster bestimmen.

Algorithmik in der Anwendung: Die Anwendung könnte algorithmisch weitergedacht werden. Sollte in einem Bild ein Fixture nicht erkannt werden, könnte anstatt eines Dummy-Eintrages beispielsweise der vorherige Wert des Fixtures in die Steuerdaten eingetragen werden. Ob diese Erweiterung sinnvoll ist, muss mit weiteren Validierungen der Ergebnisse ermittelt werden. Da die Abtastrate der Bilder und der generierten Steuerdaten 30 Frames pro Sekunde betragen, sollte dieses Vorgehen zu einer Verbesserung der Steuerdaten führen. Da die Erstellung der Steuerdaten hauptsächlich durch die trainierten YOLOv8 „Models“ geschehen sollte wurde von dieser Erweiterung abgesehen.

9 Fazit

Die Ergebnisse dieser Bachelorthesis zeigen, dass es grundsätzlich möglich ist Lichtsteuerdaten in einer einfachen Form unter der Verwendung geeigneter „Models“ der Computer Vision zu erstellen. Die Herausforderungen bei der Erstellung der Steuerdaten wurden jedoch erst im Verlauf der Bearbeitung sichtbar. Da der verfügbare Datensatz auf tatsächlichen, kontrastreichen Lichtsteuerdaten beruht, sind die Schwächen der „Classification Models“ im unterrepräsentierten, niedrigen Intensitätsbereich nachvollziehbar.

Die einheitliche Perspektive der Videos des zugrunde liegenden Datensatzes war für die automatisierte Aufbereitung eine dankbare Eigenschaft. Die automatische Synchronisation der erstellten Bilder aus dem Video-Datensatz mit den zur Verfügung gestellten Steuerdaten war eine Herausforderung, die nicht vollständig automatisch realisiert werden konnte. Daher war eine manuelle Überarbeitung der Trainingsbilder nötig.

Die einzelnen Trainingsergebnisse der „Models“ waren zufriedenstellend. Die Arbeit mit YOLOv8 „Models“ war eine gute Entscheidung mit Blick auf die Leichtgewichtigkeit des „Frameworks“ sowie der erzielten Trainingsergebnisse.

Eine Erweiterung der Anwendung für die Generierung von Lichtsteuerdaten bedarf weiteren Aufwand der eine Überarbeitung des Trainingsdatensatzes übersteigt. Die in dem Datensatz verwendeten Fixtures waren „simpel“ in ihrer Form und Funktion. Die Betrachtung der Robe TetraBars hat gezeigt, dass dasselbe Vorgehen nicht für alle Fixture-Typen in der professionellen Veranstaltungstechnik funktionieren wird. Analog dazu sind auch komplexere Bühnenaufbauten denkbar, die neue Anforderungen an die Sortierung und Gruppierung der Fixtures mit sich bringen würden.

Abschließend sind die Ergebnisse als erster Versuch der Generierung von Lichtsteuerdaten durch die Verwendung von „Models“ der Computer Vision zufriedenstellend. Die beschriebenen Erkenntnisse und Herausforderungen können als Grundlage für die Weiterentwicklung der Anwendung verwendet werden.

Literaturverzeichnis

- [1] AKSHARA, A. ; U.A.: *ABNORMAL EVENT DETECTION BY MACHINE VISION USING DEEP LEARNING*. 2020. – URL <https://www.ijeast.com/papers/188-192,Tesma412,IJEAST.pdf>
- [2] BOCHKOVSKIY, A. ; U.A.: *YOLOv4: Optimal Speed and Accuracy of Object Detection*. 2020. – URL <https://arxiv.org/pdf/2004.10934>
- [3] CIAMPICONI, L. ; U.A.: *A survey and taxonomy of loss functions in machine learning*. 2023. – URL <https://arxiv.org/pdf/2301.05579>
- [4] HOWARD, A. ; U.A.: *Searching for MobileNetV3*. 2019. – URL <https://arxiv.org/pdf/1905.02244>
- [5] ISOLA, P. ; U.A.: *Image-to-Image Translation with Conditional Adversarial Networks*. 2018. – URL <https://arxiv.org/pdf/1611.07004>
- [6] JIA, D. ; U.A.: *ImageNet: A Large-Scale Hierarchical Image Database*. 2016. – URL https://www.image-net.org/static_files/papers/imagenet_cvpr09.pdf
- [7] JOCHER, G.: *About the Optimizer*. 2024. – URL <https://github.com/ultralytics/ultralytics/issues/3360#issuecomment-1606054520>
- [8] JUAN, R. ; U.A.: *A COMPREHENSIVE REVIEW OF YOLO ARCHITECTURES IN COMPUTER VISION: FROM YOLOV1 TO YOLOV8 AND YOLO-NAS*. 2024. – URL <https://arxiv.org/pdf/2304.00501v6>
- [9] KRIZHEVSKI, A. ; U.A.: *ImageNet Classification with Deep Convolutional Neural Networks*. 2017. – URL <https://dl.acm.org/doi/pdf/10.1145/3065386>
- [10] KUKAČKA, J.: *Regularization for Deep Learning: A Taxonomy*. 2017. – URL <https://arxiv.org/pdf/1710.10686>

- [11] LEI, H. ; U.A.: *Normalization Techniques in Training DNNs: Methodology, Analysis and Application*. 2020. – URL <https://arxiv.org/pdf/2009.12836>
- [12] LIN, T. ; U.A.: *Microsoft COCO: Common Objects in Context*. 2015. – URL <https://arxiv.org/pdf/1405.0312>
- [13] MINGXING, T. ; U.A.: *EfficientDet: Scalable and Efficient Object Detection*. 2020. – URL <https://arxiv.org/pdf/1911.09070>
- [14] MUHAMMAD, H.: *YOLOV5, YOLOV8 AND YOLOV10: THE GO-TO DETECTORS FOR REAL-TIME VISION*. 2024. – URL <https://arxiv.org/pdf/2407.02988v1>
- [15] NEUHOFFER, B. ; U.A.: *The impact of artificial intelligence on event experiences: a scenario technique approach*. 2019. – URL <https://link.springer.com/article/10.1007/s12525-020-00433-4>
- [16] PRABU, S. ; U.A.: *A Study on Image Segmentation Method for Image Processing*. 2021. – URL <https://elar.khmnu.edu.ua/server/api/core/bitstreams/f0657f1d-5d38-4aa4-a30a-dff37d86e93c/content>
- [17] RADIUK, P. M. ; U.A.: *Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets*. 2017. – URL <https://elar.khmnu.edu.ua/server/api/core/bitstreams/f0657f1d-5d38-4aa4-a30a-dff37d86e93c/content>
- [18] RAMACHANDRAN, P.: *SEARCHING FOR ACTIVATION FUNCTIONS*. 2017. – URL <https://arxiv.org/pdf/1710.05941>
- [19] SENIOR, A. ; U.A.: *AN EMPIRICAL STUDY OF LEARNING RATES IN DEEP NEURAL NETWORKS FOR SPEECH RECOGNITION*. 2013. – URL <https://static.googleusercontent.com/media/research.google.com/de/pubs/archive/40808.pdf>
- [20] SRIVASTA, N. ; U.A.: *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. 2014. – URL <https://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf>
- [21] SUN, S. ; U.A.: *A Survey of Optimization Methods from a Machine Learning Perspective*. 2019. – URL <https://arxiv.org/pdf/1906.06821>

- [22] ULTRALYTICS: *Train Settings*. 2024. – URL <https://docs.ultralytics.com/modes/train/#train-settings>
- [23] ULTRALYTICS: *Where to start*. 2024. – URL <https://docs.ultralytics.com/#where-to-start>
- [24] XIE, X. ; U.A.: *Oriented R-CNN for Object Detection*. 2021. – URL https://openaccess.thecvf.com/content/ICCV2021/papers/Xie_Oriented_R-CNN_for_Object_Detection_ICCV_2021_paper.pdf
- [25] XINGYI, Z. ; U.A.: *Objects as Points*. 2019. – URL <https://arxiv.org/pdf/1904.07850>
- [26] YANG, S. ; U.A.: *Image Data Augmentation for Deep Learning: A Survey*. 2023. – URL <https://arxiv.org/pdf/2204.08610>
- [27] YILMAZ, E. ; U.A.: *Analysis of the effect of training sample size on the performance of 2D CNN models*. 2021. – URL <https://publish.mersin.edu.tr/index.php/igd/article/view/489/345>

A Anhang

A.1 Verwendete Hilfsmittel

In der Tabelle A.1 sind die im Rahmen der Bearbeitung des Themas der Bachelorarbeit verwendeten Werkzeuge und Hilfsmittel aufgelistet.

Tabelle A.1: Verwendete Hilfsmittel und Werkzeuge

Tool	Verwendung
L ^A T _E X	Textsatz- und Layout-Werkzeug verwendet zur Erstellung dieses Dokuments
Git Lab	Verwaltung und Versionierung des Quellcodes
Google Colabs	Umgebung für das Training der Modelle
Google Drive	Verwaltung des Trainingsdatensatzes und der Modelle
Microsoft Office	Schreiben der Thesis und Erstellung von Abbildungen
PyCharm	Entwicklungsumgebung für die Erstellung der Anwendung

A.2 Anleitung für die Anwendung

Dataset_preprocessor.py: Erstellt aus Bildern einen Trainingsdatensatz für das Training von YOLOv8 „Classification Models“ und „Detection Models“.

Control_data_generator.py: Erstellt Lichtsteuerdaten in Form einer PKL-Datei unter Verwendung der trainierten YOLOv8 „Models“.

Single_image_predictor.py: Erweitert ein Bild um die „Bounding Boxes“ der Erkennungen und Klassifikationen der YOLOv8 „Models“.

Video_converter.py: Erstellt aus einem MP4-Video einen Bilddatensatz mit einer Abtastrate von 30 Bildern pro Sekunde.

Validator.py: Validierungswerkzeug für die Auswertung der generierten Steuerdaten.

Jedes Python-Skript besitzt eine eigene YAML-Konfigurationsdatei.

Die erstellten Python-Skripte können durch Hochladen der 'lighting_control_data.rar' Datei und der anschließend Ausführung der Code Blöcke in einer vorbereiteten Google Colabs Umgebung angewandt werden: <https://colab.research.google.com/drive/1BJJuly2UEIjUMnYMj8J4cVSY-cAIADfM#scrollTo=wL5i-UrAU32Q>

A.3 Anwendungen in der Veranstaltungstechnik

Automatischer Frequenz Equalizer auf Basis von Machine Learning Models:

<https://www.soundandrecording.de/equipment/sonible-veroeffentlicht-live-eq-plugin-in-smarteq-mit-echtzeit-ki-engine/> (Stand 14.08.2024)

Steuerung von Bühnenlicht in Echtzeit durch Machine Learning Modelle:

<https://www.youtube.com/watch?v=pZZiAXjMOsw> (Stand 14.08.2024)

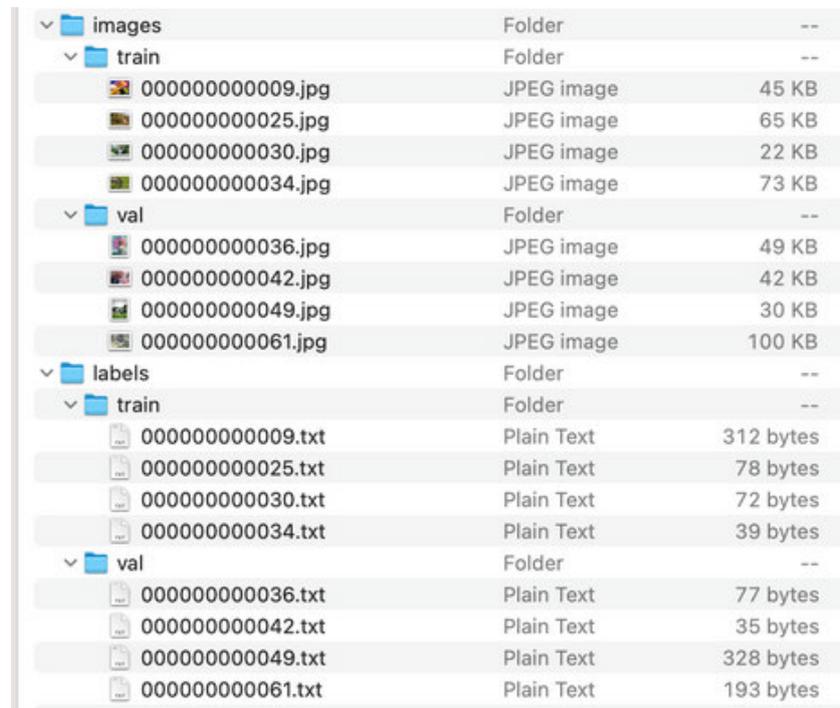
Echtzeitanwendung für Speech-to-Text Übersetzungen:

<https://www.interprefy.com/de/> (Stand 14.08.2024)

A.4 Lichtsteuerdaten in PIX-Darstellung



Abbildung A.1: Abstraktion der Steuerdaten durch eine zeilenweise Darstellung der Parameter der Fixture Gruppen in einem Pixelformat.



Item	Type	Size
images	Folder	--
train	Folder	--
000000000009.jpg	JPEG image	45 KB
000000000025.jpg	JPEG image	65 KB
000000000030.jpg	JPEG image	22 KB
000000000034.jpg	JPEG image	73 KB
val	Folder	--
000000000036.jpg	JPEG image	49 KB
000000000042.jpg	JPEG image	42 KB
000000000049.jpg	JPEG image	30 KB
000000000061.jpg	JPEG image	100 KB
labels	Folder	--
train	Folder	--
000000000009.txt	Plain Text	312 bytes
000000000025.txt	Plain Text	78 bytes
000000000030.txt	Plain Text	72 bytes
000000000034.txt	Plain Text	39 bytes
val	Folder	--
000000000036.txt	Plain Text	77 bytes
000000000042.txt	Plain Text	35 bytes
000000000049.txt	Plain Text	328 bytes
000000000061.txt	Plain Text	193 bytes

Abbildung A.3: Der Aufbau der Datenstruktur für das Training eines „Yolov8 Detection Models“

```
# Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt
path: ../datasets/coco8 # dataset root dir
train: images/train # train images (relative to 'path') 4 images
val: images/val # val images (relative to 'path') 4 images
test: # test images (optional)

# Classes (80 COCO classes)
names:
  0: person
  1: bicycle
  2: car
  # ...
  77: teddy bear
  78: hair drier
  79: toothbrush
```

Abbildung A.4: Der Aufbau einer „Label“ TXT-Datei für das Training eines „Yolov8 Detection Models“

```
cifar-10-/  
  
-- train/  
  |-- airplane/  
    |-- 10008_airplane.png  
    |-- 10009_airplane.png  
    |-- ...  
  
    |-- automobile/  
      |-- 1000_automobile.png  
      |-- 1001_automobile.png  
      |-- ...  
  
    |-- bird/  
      |-- 10014_bird.png  
      |-- 10015_bird.png  
      |-- ...  
  
    |-- ...  
  
-- test/  
  |-- airplane/  
    |-- 10_airplane.png  
    |-- 11_airplane.png  
    |-- ...  
  
    |-- automobile/  
      |-- 100_automobile.png  
      |-- 101_automobile.png  
      |-- ...  
  
    |-- bird/  
      |-- 1000_bird.png  
      |-- 1001_bird.png  
      |-- ...
```

Abbildung A.5: Der Aufbau der Datenstruktur für das Training eines „Yolov8 Classification Models“

A.7 Beispielbilder aus dem Datensatz



Abbildung A.6: Unmodifizierte Bilder aus dem Datensatz für das „Fixture Detection Model“

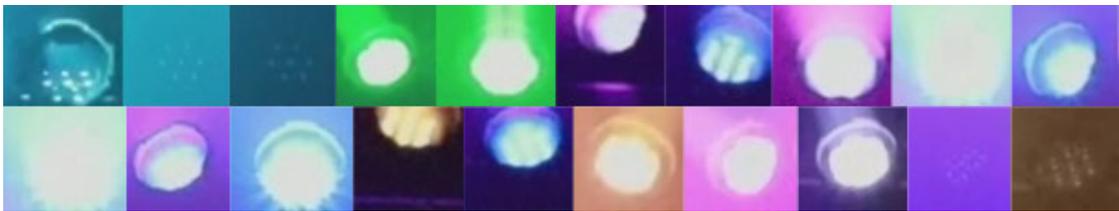


Abbildung A.7: Unmodifizierte Bilder aus dem Datensatz für das Robe ParFect „Classification Model“

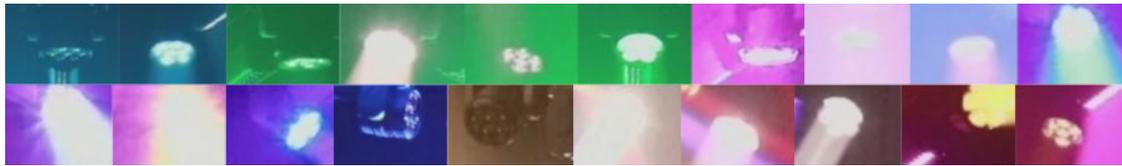


Abbildung A.8: Unmodifizierte Bilder aus dem Datensatz für das Robe LEDBeam „Classification Model“



Abbildung A.9: Unmodifizierte Bilder aus dem Datensatz für das Robe OnePatt „Classification Model“

A.8 Beispielbilder aus dem Training

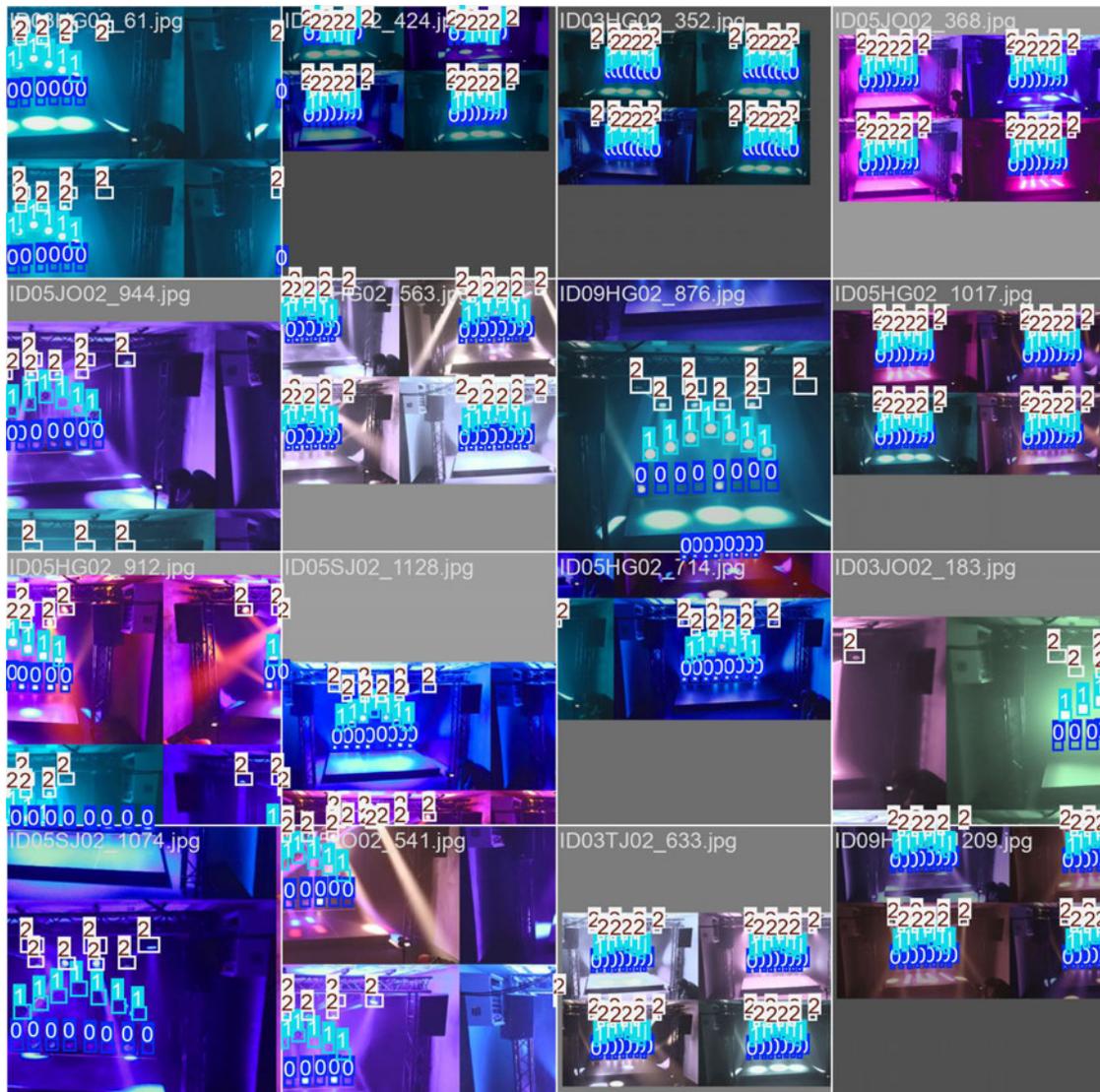


Abbildung A.10: Beispielbilder aus dem Training des „Fixture Detection Models“



Abbildung A.11: Beispielbilder aus dem Training des Robe ParFect „Classification Models“



Abbildung A.12: Beispielbilder aus dem Training des Robe LEDBeam 150 „Classification Models“

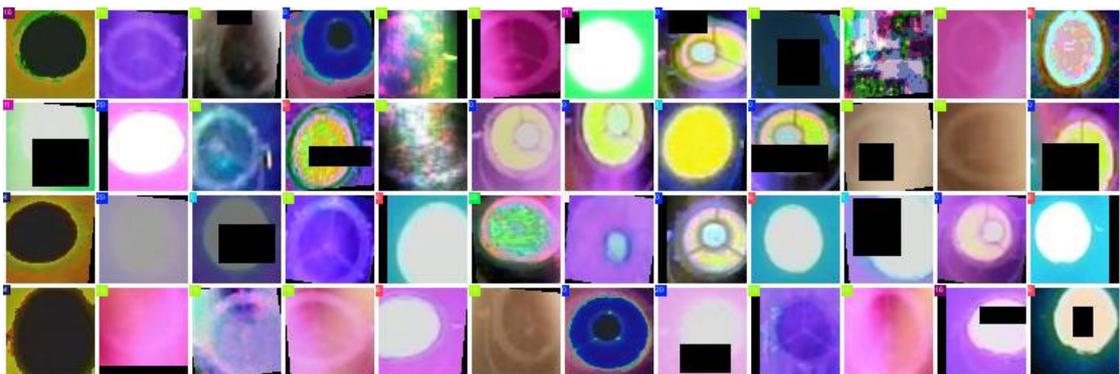


Abbildung A.13: Beispielbilder aus dem Training des Robe OnePatt „Classification Models“

B Datenblätter

B.1 Robe LEDBeam 150



LEDBeam 150™

The best things come in small packages, now with zoom included. LEDBeam 150 is the answer with fast sweeping beams and a wide far-reaching quality wash in compact housing.

**Light source**

7x 40W RGBW multichips

**Light output**

3.550 lm, 12.200 lx @ 5m

**Zoom range**

3,8° - 60°



The spectacular zoom range of 3.8° to 60° uses an eminent custom-designed optical system, now equipped with Robe's innovative lens coating technology which brings benefits such as bright and clear lenses, no scratches or marks, higher light output and longer intervals between cleaning. The unique LEDBeam 150 FW offers Fresnel-Wash type of light output for even smooth edges and better color homogenization.

Attractive colorful chases and smooth transitions are powered by a cluster of high power multichip 40W RGBW LEDs.

A highly optimized motorised control produces speedy pan and tilt movement. Besides intense strobing capabilities, both LEDBeam 150 and LEDBeam 150 FW also offer gentle 18-bit dimming, including Tungsten lamp effects.



Technical Specification

Source

- Light source type: 7x 40W RGBW multichips
- LED life expectancy: min. 50.000 hours
- Typical lumen maintenance: L70/B50 @ 50.000 hours

Optical system

- Robe's proprietary optical design
- High - efficiency zoom optical system, ratio: 15,5:1
- Zoom range: 3,8° - 60°
- Fixture total lumen output:
 - 3.550 lm (integrating sphere)
 - 2.842 lm (goniophotometer)
- Illuminance: 12.200lx @ 5 m
- (RLCT™) Innovative lens coating technology (Patent pending)

Dynamic Effects and Features

- Colour mixing mode RGBW or CMY
- Variable CTO: 2.700K - 8.000K
- Virtual Colour Wheel: with 66 preset swatches
- Tungsten lamp effect: 750W, 1.000W, 1.200W, 2.000W, 2.500W lamp emulation for whites from 2.700K to 4.200K (red shift and thermal delay)
- Motorized zoom
- Pre-programmed random strobe & pulse effects
- High resolution electronic dimming: 0 - 100%

Control and programming

- Setting & Addressing: two-row LCD display & 4 control buttons, stand-alone operation with 3 editable programs (each up to 40 steps)
- Protocols: USITT DMX-512, RDM
- Wireless CRMX™ technology from Lumen Radio - on request
- DMX Protocol modes: 2
- Control channels: 22, 16
- Pan/Tilt resolution: 8 or 16 bit
- RGBW/CMY: 8 or 16 bit
- Zoom: 8 or 16 bit



- Dimmer: 8 or 16 bit (internal 18 bit)

Movement

- Pan movement: 450°
- Tilt movement: 228°
- Movement control: Standard and Speed
- Controllable speed of Pan/Tilt movement
- Automatic Pan/Tilt position correction

Thermal specification

- Maximum ambient temperature: 45°C (113°F)
- Maximum surface temperature: 80°C (176°F)
- Minimum operating temperature: -5°C (23°F)

Noise Levels

- Sound pressure level:
 - 18 dB(A) at 1 m (quiet mode)
 - 34 dB(A) at 1 m (auto mode)
- Sound power level:
 - 26 dB(A) (quiet mode)
 - 42 dB(A) (auto mode)

Electrical specification and connections

- Power supply: Electronic auto-ranging
- Input voltage range: 100-240 V, 50/60 Hz
- Power consumption: Max. 220 W
- Power in/out connector: Neutrik powerCON in/out
- DMX and RDM data in/out: Locking 5-pin XLR

Approvals

- CE Compliant
- cETLus Compliant

Mechanical specification

- Height: 337 mm (13.3")
- Width: 244 mm (9.6")
- Depth: 149 mm (5.9")



- Weight: 5.7 kg (12.6 lbs)
- Ingress protection rating: IP20

Rigging

- Mounting points: 1 pair of ¼-turn locks
- 1x Omega adaptor with ¼-turn quick locks
- Universal operating position
- Safety cable attachment point

Included items

- User Manual
- Omega Adaptor CL-regular

Optional accessories

- Diffusion filter 2° FW: 10980423
- Clear lens cover: 10980604
- Wireless DMX external module: 10980127
- EggCrate: 10980346
- Doughty Trigger Clamp: 17030386
- Safety wire 36 kg: 99011963
- Mains Cable PowerCON In/Schuko 2m: 13051724
- Mains Cable PowerCON In/CEE 16A 2m: 13051725
- Mains Cable PowerCON In/US 2m: 13051726
- Mains Cable PowerCON In/open ended 2m : 13051731
- Daisy Chain PowerCON In/Out EU 2m : 13051727
- Daisy Chain PowerCON In/Out US 2m : 13051728
- Single Top Loader Case: 10120214
- Quad Top Loader Case: 10120215
- Eight Pack Top Loader Case: 10120216
- Dual Foam Shell: 20020300

Legal

- LEDBeam 150™ is a trademark of Robe lighting s.r.o.
- LEDBeam 150™ is patented by Robe lighting s. r. o. and is protected by one or more pending or issued patent

B.2 Robe OnePatt



onePATT™

Der onePATT ist eine neue Leuchte im Retro-Stil aus der PATT Familie. Er verfügt über einen RGBW hochleistungs-LED-Multichip, der in einem wunderbar schlanken und leichten Gehäuse aus gebürstetem Metall eingebaut ist.

**Lichtquelle**

40W RGBW LED Multichip

**Effekte**

Individuelle Ansteuerung der einzelnen RGBW Pixel, Tungsten (Glühlampen-)Effekt



Er überzeugt mit seiner hohen Lichtausbeute gleichermaßen in großen Matrix-Anwendungen, die über Pixel-Mapping angesteuert werden, wie auch als einzelnes Beleuchtungsgerät.

Der onePATT wird mit einem separaten Netzteil betrieben. Die integrierte elektronische Ansteuerung sorgt für flickerfreien Output, Schutz vor Überhitzung, Fernadressierung und -kalibrierung sowie Statusrückmeldung, wodurch sie einfachen Service und langfristige Zuverlässigkeit garantiert. Um die Installation zu erleichtern kann der onePATT auch unter Last vom Netzteil getrennt oder mit ihm verbunden werden.

Insgesamt 7 onePATTs können mit einem PATT-Treiber betrieben werden - einem speziell für die pixelPATT, PATT2017 und onePATT-Familie entwickelten Netzstrom- und Steuerungs-Gerät.



Technische Spezifikation

Lichtquelle

- Lichtquellenart: 40W RGBW Multichip
- Lebenserwartung der LEDs: min. 20.000 Stunden
- Typische Helligkeit nach 20.000 Stunden: 70%

Optisches System

- 200mm parabolischer Aluminium-Reflektor mit spezieller gold-brauner Beschichtung
- Der halbkugelförmige Diffusor-Dom sorgt für gleichmäßige Lichtverteilung von der RGBW LED Lichtquelle im Reflektor

Dynamische Effekte und Ausstattungsmerkmale

- Farbmischungsmodi: RGBW oder CMY
- Individuelle Ansteuerung der einzelnen RGBW Pixel bei Einsatz mehrerer onePATTs am Treiber
- Variable Farbtemperatur (CTO): 2700K - 8000K
- Virtuelles Farbrad: mit 66 vorprogrammierten Farben
- Vorprogrammierte Pixel Effekte mit Farben, Dimmung und Strobo, mit Wellen und Puls in Geschwindigkeit und Richtung regelbar
- Shutter: elektronisch mit variabler Strobo-Geschwindigkeit (max. 20 Blitze pro Sekunde)
- Vorprogrammierte, zufallsgenerierte Stroboskop- & Pulseffekte
- Hochauflösende elektronische Dimmung: 0 - 100%

Steuerung und Programmierung

- PATT Treiber: extern
- Einstellung und Adressierung: 2reihiges LCD Display & 4 Steuertasten
- Protokolle: USITT DMX-512, RDM, ArtNet, Kling-Net, sACN
- Drahtlose CRMX™-Technologie von Lumen Radio (optional)
- DMX Protokoll Modi: 6
- Anzahl Steuerkanäle: 13, 32, 40, 47, 53, 60
- Farbmischungsmodi RGBW oder CMY: Auflösung 8 oder 16 Bit (intern 18 Bit)
- Dimmer: Auflösung 8 oder 16 Bit (intern 18 Bit)

Thermische Spezifikation

- Maximale Umgebungstemperatur: 45°C (113°F)
- Maximale Gehäusetemperatur: 70°C (158°F)
- Minimale Betriebstemperatur: -5°C (23°F)



Elektrische Spezifikationen und Anschlüsse

- PATT Treiber:
 - Netzteil: Elektronisch mit automatischer Bereichswahl
 - Versorgungsspannung: 100-277 V, 50/60 Hz
 - Leistungsaufnahme: 240 W
 - Netzstrom-Verbinder: Neutrik PowerCon TRUE1 in/out
 - DMX und RDM Daten: verriegelbare 3-Pol & 5-Pol XLR in/out
 - Integrierter Netzwerk-Switch 10/100 Mbps: 1x in/1x out
 - Netzwerk-Schnittstelle: RJ45 in/out
 - Netzstrom und Daten out: verriegelbare 4-Pol XLR (von PATT Treiber zum onePATT)
- onePATT:
 - Netzstrom und Daten: verriegelbare 4-Pol XLR in

Freigaben

- CE Übereinstimmung
- cETLus Übereinstimmung (schwebend)

Mechanische Spezifikationen

- onePATT:
 - Höhe: 321 mm (12.6")
 - Breite: 309 mm (12.1")
 - Tiefe: 150 mm (5.9")
 - Gewicht: 1,6 kg (3.5 lbs)
- PATT Treiber:
 - Höhe: 358 mm (14.1")
 - Breite: 82 mm (3.2")
 - Tiefe: 162 mm (6.4")
 - Gewicht: 3,0 kg (6.6 lbs)
- IP-Schutzklasse: IP20



Rigging

- Befestigungspositionen:
 - Treiber: 1x Half-Coupler Klemme, 1x Ankerpunkt für Sicherheitskabel
 - onePATT: 1x 13mm Loch für die Montage einer Klemme
- Betriebsposition:
 - PATT Driver: Standmontage auf dem Boden oder an Traverse in vertikaler Position befestigt
 - onePATT: jede Betriebs- und Befestigungsposition

Im Lieferumfang enthalten

- englische Betriebsanleitung

Zubehör

- PATT Driver:
 - Standard Version: 10055002
 - Drahtlose Version: 10055004
- Doughty Trigger Clamp: 17030386
- Sicherheitsdrahtseil 36 kg: 99011963
- Netzkabel powerCON TRUE1 In/Schuko, 2m, Indoor: 13052405
- Netzkabel powerCON TRUE1 In/US, 2m, Indoor: 13052406
- Netzkabel powerCON TRUE1 In/CEE 16A, 2m, Indoor: 13052445
- Netzkabel powerCON TRUE1 In/open ended, 2m, Indoor: 13052407
- Daisy Chain powerCON TRUE1 In/Out, EU, 2m, Indoor: 13052439
- Daisy Chain powerCON TRUE1 In/Out, US, 2m, Indoor: 13052440
- Daisy Chain powerCON TRUE1 In/Out, EU, 5m, Indoor: 13052444
- Datenkabel 4-Pol XLR 10m, Ferrit, EU: 13052797
- Datenkabel 4-Pol XLR 5m, Ferrit, EU: 13052798
- Datenkabel 4-Pol XLR 10m, Ferrit, US: 13052799
- Datenkabel 4-Pol XLR 5m, Ferrit, US: 13052800
- onePATT 7fach Top Loader Case Set: 10120231

Rechtliches

- onePATT™ ist ein eingetragenes Warenzeichen der ROBE lighting s. r. o.
- onePATT™ ist ein Patent der ROBE lighting s. r. o. und ist geschützt durch ein oder mehrere Patente oder laufende Patentverfahren

B.3 Robe ParFect



ParFect 150™ RGBW

An LED source ACL beam at an affordable price made in Europe, now with zoom included.

**Light source**

7x 40W RGBW multichips

**Light output**

3.550 lm, 12.200 lux @ 5m

**Zoom range**

3,8° - 60°



ParFect 150 is a static, zoomable version of the successful ParFect 100 and retains all the key features including both CMY & RGBW colour control, 18-bit dimming, Tungsten emulation, selectable & variable colour temperatures and the powerful punchy beam ranging from 3,8° to 60°.

The compact and lightweight moulded composite unit includes a combined hanging bracket/floor stand. Accessories such as barn-doors and the gel frame give ParFect 150 the features required for TV and theatre.

ParFect 150™ RGBW is now equipped with Robe's innovative lens coating technology which brings benefits such as bright and clear lenses, no scratches or marks, higher light output and longer intervals between cleaning.



Technical Specification

Source

- Light source type: 7x 40W RGBW multichips
- LED life expectancy: min. 50.000 hours
- Typical lumen maintenance: L70/B50 @ 50.000 hours

Optical system

- Robe's proprietary optical design
- (RLCT™)Innovative lens coating technology (Patent pending)
- High - efficiency zoom optical system, ratio 15,5:1
- Zoom range: 3,8° - 60°
- Fixture total lumen output:
 - 3.550 lm (integrating sphere)
 - 2.842 lm (goniophotometer)
- Illuminance: 12.200lx @ 5 m

Dynamic Effects and Features

- Colour mixing mode RGBW or CMY
- Variable CTO: 2.700K - 8.000K
- Virtual Colour Wheel: with 66 preset swatches
- Tungsten lamp effect at whites: 2.700K and 3.200K (red shift and thermal delay)
- Motorized zoom
- Pre-programmed random strobe & pulse effects
- High resolution electronic dimming: 0 - 100%

Control and programming

- Setting & Addressing: two-row LCD display & 4 control buttons, stand-alone operation with 3 editable programs (each up to 25 steps)
- Protocols: USITT DMX-512, RDM
- Wireless CRMX™ technology from Lumen Radio - on request
- DMX Protocol modes: 2
- Control channels: 17, 11
- RGBW or CMY: 8 or 16 bit
- Zoom: 8 or 16 bit
- Dimmer: 8 or 16 bit (internal 18 bit)



Thermal specification

- Maximum ambient temperature: 40°C (104°F)
- Maximum surface temperature: 80°C (176°F)
- Minimum operating temperature: -5°C (23°F)

Electrical specification and connections

- Power supply: Electronic auto-ranging
- Input voltage range: 100-240 V, 50/60 Hz
- Power consumption: Max. 220 W
- Power in/out connector: Neutrik powerCON in/out
- DMX and RDM data in/out: Locking 5-pin XLR

Approvals

- CE Compliant
- cETLus Compliant

Mechanical specification

- Height: 192 mm (7.6")
- Width: 240 mm (9.4")
- Depth: 392 mm (15.4") without hanging bracket / 498 mm (19.6") with hanging bracket
- Weight: 5 kg (11.0 lbs)
- Ingress protection rating: IP20

Rigging

- Mounting horizontally or vertically via mounting yoke
- Universal operating position
- Safety cable attachment point

Included items

- User Manual
- Accessory frame adaptor

Optional accessories

- Wireless DMX external module: 10980127
- Gel frame: 10980372
- Diffusion filter:



- for fixed installation under optics 2° FW 10980423
- for sliding into Gel Frame 2° FW 10980373
- Clear lens cover: 10980604
- Barndoor module: 10980349
- EggCrate: 10980346
- HALO frame adaptor: 10980348
- Doughty Trigger Clamp: 17030386
- Safety wire 36 kg: 99011963
- Mains Cable PowerCon In/Schuko 2m: 13051724
- Mains Cable PowerCon In/CEE 16A 2m: 13051725
- Mains Cable PowerCon In/US 2m: 13051726
- Mains Cable PowerCon In/open ended 2m: 13051731
- Daisy Chain PowerCon In/Out EU 2m: 13051727
- Daisy Chain PowerCon In/Out US 2m: 13051728
- Single Top Loader Case: 10120217-02
- Six Pack Top Loader Case: 10120218-02
- Eight Pack Top Loader Case: 10120219-02
- Dual Foam Shell: 20020301-01

Legal

- ParFect 150™ is a trademark of Robe lighting s.r.o.
- ParFect 150™ RGBW is patented by Robe lighting s. r. o. and is protected by one or more pending or issued patent

B.4 Robe TetraBar RGB



Tetra2™

TETRAgnatha's are linear bars building on the Spiider and Tarrantula technology with TETRA2™ the uncompromising leader of the Tetra family.

**Light source**

18x 40 W RGBW LED multichips

**Light output**

10.500 lm

**Zoom range**

4° - 45°

**Effects**

2x MCFE™ - Multi-Coloured Flower Effects - creating spectacular multicolour beam effects in the air rotating in both directions at variable speed (patented), pixel control, virtual colour wheel, tungsten lamp effect



Generating an ultra-tight 4° beam from each of the 18 pixels, they combine to produce a bright, defined "sheet" of light, desired by Lighting Designers. Seamless curtains of light can be constructed using several fixtures as the detailed design allows any combination of TETRA1™ and TETRA2™ to be placed end to end on stage or truss, whilst maintaining equal spacing between pixels. With the addition of two exclusive Robe patented MCFE™ - Multi-Coloured Flower Effects, the pixel-driven Tetra1 sets itself apart from others by projecting charismatic in- air animations.

The homogenised beams, together with the smooth 11:1 zoom, provide; a wash out to 45°, a footlight, a wall graze or dynamic in-air effects with fast paced sweeping movements. Utilizing our latest L3™ (Low Light Linearity) dimming system for an imperceptible fade to black, the 18-bit control provides ultra-smooth colour mixing across the full colour spectrum. An embedded Ethernet switch and wide range of protocols (sACN, Art-Net or Kling-Net) allow a quick network installation and ease of control from media servers, DMX or the internal effects engine.



Technical Specification

Source

- Light source type: 18x 40W RGBW LED multichips
- LED life expectancy: min. 50.000 hours
- Typical lumen maintenance: L70/B50 @ 50.000 hours

Optical system

- Robe's proprietary optical design
- Zoom range: 4°- 45°
- Highly efficient component optics
- Fixture total lumen output
 - 10.500 lm (integrating sphere)
 - 8.439 lm (goniophotometer)

Dynamic Effects and Features

- Colour mixing mode RGBW or CMY
- Individual control of each RGBW pixel
- Variable CCT: 2.700K - 8.000K
- Tungsten lamp effect: 750W, 1.000W, 1.200W, 2.000W, 2.500W lamp emulation for whites from 2.700K to 4.200K (red shift and thermal delay)
- DataSwatch™ filters: pre-programmed 237 colours and tones including most used whites 2.700K, 3.200K, 4.200K, 5.600K and 8.000K
- Pre-programmed pixel effects with colour, dimming and strobe chases, waves and pulses at variable speed and direction
- 2x MCFE™ - Multi-Coloured Flower Effects - creating spectacular multicolour beam effects in the air rotating in both directions at variable speed (patented)
- Motorized zoom
- 2 independent zoom zones
- Pre-programmed random strobe & pulse effects
- Electronic strobe effect with variable speed up to 20 Hz
- High resolution electronic dimming: 0 - 100%
- L3™ - (Low Light Linearity) Imperceptible 18 bit dimming for ultra smooth fade to black



Control and programming

- Setting & Addressing: QVGA Robe touch screen with battery backup, gravitation sensor for auto screen positioning, operation memory service log with RTC, stand-alone operation with 3 editable programs (each up to 88 steps), built-in analyser for easy fault finding
- Protocols: USITT DMX-512, RDM, Art-Net, MA Net, MA Net2, sACN, Kling-Net
- REAP™ - Robe Ethernet Access Portal
- Wireless CRMX™ technology from Lumen Radio - on request
- DMX Protocol modes: 6
- Control channels: 34, 56, 97, 115, 110, 128
- RGBW / CMY: 8 or 16 bit
- Zoom: 8 or 16 bit
- Dimmer: 8 or 16 bit (internal 18 bit)

Movement

- Tilt movement: 191°
- 16 bit movement resolution
- Controllable speed of PTilt movement

Thermal specification

- Maximum ambient temperature: 40°C (104°F)
- Maximum surface temperature: 70°C (158°F)
- Minimum operating temperature: -5°C (23°F)
- Total heat dissipation: max. 2047 BTU/h (calculated)

Electrical specification and connections

- Power supply: Electronic auto-ranging
- Input voltage range: 100-240 V, 50/60 Hz
- Power consumption: 600 W at 230 V / 50 Hz
- Power in/out connector: Neutrik powerCON TRUE1 in/out
- DMX and RDM data in/out: Locking 5-pin XLR
- Ethernet port in/out: RJ45
- Embedded Ethernet switch 10/100 Mbps

Mechanical specification

- Height: 279 mm (10.98") - head in vertical position
- Width: 1007 mm (39.6")
- Depth: 143 mm (5.62") - head in vertical position



ROBE lighting s. r. o., Hazovice 2090
75661 Roznov pod Radhostem
Czech Republic



- Weight: 18.1 kg (39.9 lbs)
- Ingress protection rating: IP20

Rigging

- Mounting points: 2 pairs of 1/4-turn locking points
- 2x Omega adaptors with 1/4-turn quick locks
- Universal operating position
- Safety cable attachment point

Included items

- User Manual
- Variable Omega Adaptor: 2 pcs 99016241-02
- Power cord including powerCON TRUE1 In connector

Optional accessories

- Diffusion filter: 2° 10980593
- Clear lens cover: 10980606
- Safety wire 36 kg: 99011963
- Daisy Chain powerCON TRUE1 In/Out, EU, 2m, Indoor: 13052439
- Daisy Chain powerCON TRUE1 In/Out, US, 2m, Indoor: 13052440
- Daisy Chain powerCON TRUE1 In/Out, EU, 5m, Indoor: 13052444
- Single Top Loader Case: 10120259
- Triple Top Loader Case: 10120264
- Foam Shell: 20020371
- Omega Adaptor CL-variable 2pcs in box: 10980550

Legal

- Tetra2™ is a Trademark of Robe lighting s. r. o.
- Tetra2™ is patented by Robe lighting s. r. o. and protected by one or more pending or issued patents

Glossar

Beam Ein Scheinwerfertyp, der Licht in Form eines klar definierten Strahles emittiert. Wird häufig als Effektlicht verwendet.

Channel Ein Fixture besitzt einen -oder mehrere 8-bit Channels. Ein Channel beschreibt eine Funktion eines Fixtures. Beispielsweise könnte ein Fixture ein Channel für die Intensität haben, sowie jeweils einen Channel für die RGB-Werte.

Fixture Ein Bühnenscheinwerfer in der Veranstaltungstechnik. Meistens schriftlich durch eine „#“, gefolgt von einer dreistelligen Nummer, gekennzeichnet. #204 steht beispielsweise für den vierten Scheinwerfer (x04) der zweiten Gruppe (2xx).

Floor Eine Position von Fixtures auf der Bühne. Meistens an der Bühnen Hinterkante oder an der Bühnenseite.

Gobo Eine Art Schablone die in den Strahlengang eines Fixtures eingeführt werden kann. Durch dessen Verwendung verändert sich der Lichtkegel zu gewünschten Mustern.

Moving Head Eine Art von Scheinwerfer dessen Köpfe sich horizontal und vertikal bewegen können. Dadurch kann Licht in unterschiedliche Richtungen emittiert werden.

Pan Der Fixture-Channel für die horizontale Position des Fixture-Kopfes.

Riser Eine Erhöhung auf einer Bühne. Wird verwendet, um Mitglieder einer Band im hinteren Bereich zu erhöhen (Bspw. Das Schlagzeug oder das Keyboard) oder um Fixtures aus Gründen des Licht-Designs nicht auf den Floor zu stellen.

Spot Ein Scheinwerfertyp, der überlicherweise Licht in Form eines klassischen Lichtkegels emittiert.

Tilt Der Fixture-Channel für die vertikale Position des Fixture-Kopfes.

Truss Eine Aluminiumtraverse. Wird in diesem Kontext als Anschlagort für Fixtures verwendet. Einzelne Traversen werden als Front-, Mid- oder Backtruss, je nach Positionierung bezeichnet.

Wash Ein Scheinwerfertyp, der diffuses Licht über eine größere Fläche streut.

Erklärung zur selbständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original