Hochschule Darmstadt
Fachbereich Mathematik und Naturwissenschaften
&
Fachbereich Informatik

Optimierung von Bildklassifikatoren im Bereich Biodiversität und Citizen-Science mithilfe von Object-Detection

Abschlussarbeit zur Erlangung des akademischen Grades Master of Science (M. Sc.) im Studiengang Data Science

Simon Köhler

18. April 2024

Das Thema stellte
Prof. Elke Hergenröther

Korreferent
Prof. Andreas Weinmann

In den kleinsten Dingen
zeigt die Natur
die allergrössten Wunder.

Carl von Linné

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, 18. April 2024

Simon Köhler

Abstract

With web applications like iNaturalist, naturgucker, Flora Incognita or BirdNET it is possible for naturalists to determine the biological species of wild plants or animals. These so called citizen scientists support biodiversity research with their reports and determination of observed organisms. With the help of those datasets and scientific long-term observations, the global abundance and diversity of life can be evaluated. The project "WildLIVE!" of the Senckenberg Forschungsinstitut allows citizen scientists to determine species on photos of camera traps - installed automatic cameras in the wild - in the Bolivian region of Chiquitano and the South African region Baviaanskloof.

In research areas, which depend on photos, the establishment of image classification and object detection processes is essential. The work of citizen scientists can be supported and accelerated with Computer Vision (CV) and Machine Learning (ML). For many years there have been efforts to design image classification algorithms with Convoluational Neural Networks (CNN) and Transformers, which can classify images even for a high number of classes and very detailed visual differences. Especially in biology, those Fine Grained Visual Classifications can be helpful to spot differences on blurry images and differentiate between visually similar species. Existing work points out that it could be relevant to first localize important regions in images to boost classification performance. This aspect is even more important for camera trap images, which cannot focus on an object of interest. It stands to reason that classification performance is worse on large scale raw images compared to image cutouts where only an animal is visible.

This thesis evaluates whether machine-learning-based object detection can improve the performance of image classifications by localizing objects of interest on camera trap images. Image classification models trained on raw images, expert bounding box cutouts, and object detection cutouts are compared. In a future machine-learning-based citizen science platform, the best image classificator can be integrated to automatically determine the species of animals in Bolivia and other regions.

The experiment results show that MegaDetector can find correct (IoU=.75) bounding boxes for 90% of images. An image classification model trained on those bounding boxes and cutouts reaches a similar performance (Average F_1: 0.889, Precision: 97.7%, Recall: 92.0%, Accuracy: 95.0%) to a model trained on expert bounding boxes and cutouts (F_1: 0.909, P: 98.3%, R: 92.5%, A: 95.6%) and better results than a model trained on raw images (F_1: 0.873, P: 97.0%, R: 91.8%, A: 94.3%). Experiments on a subset of the data show that models based on cutouts (F_1: 0.856) generalize far better on photos of other locations than models based on raw images (F_1: 0.709).

Keywords: Machine Learning, Computer Vision, Image Classification, Object Detection, MegaDetector, Biodiversity, Wildlife, Citizen Science

Zusammenfassung

Mithilfe von Web-Anwendungen wie iNaturalist, naturgucker, Flora Incognita oder BirdNET werden naturinteressierte Laien unterstützt, beobachtete Wildpflanzen und Wildtiere biologisch zu bestimmen. Diese „Citizen-Scientists“ unterstützen die Biodiversitätsforschung durch die Meldung und Bestimmung beobachteter Organismen. Auf Basis der gesammelten Daten und wissenschaftlicher Langzeitbeobachtungen kann weltweit die Fülle und Diversität des Lebens beforscht werden. Beim Projekt “WildLIVE!” des Senckenberg Forschungsinstituts unterstützen Laien bei der Bestimmung von Arten auf Fotos von Fotofallen - fest installierten automatischen Kameras - aus der bolivischen Region Chiquitano und der südafrikanischen Region um den Fluss Baviaanskloof.

In Forschung, die auf Fotos angewiesen ist, sind Bildklassifikations- und Object-Detection-Verfahren nicht weit. Die Tätigkeiten der Citizen-Scientists können mit Computer Vision (CV) und Machine-Learning (ML) unterstützt und beschleunigt werden. Seit vielen Jahren werden Bemühungen angestellt, Bildklassifikatoren auf Basis von Convolutional-Neural-Networks (CNN) und Transformern zu entwerfen, die auch bei großer Klassenmenge und sehr detaillierten Unterschieden erfolgreiche Klassifikationen liefern. Gerade in der Biologie, wo die Herausforderung besteht, auf unscharfem Datenmaterial die Unterschiede zwischen ähnlichen Spezies zu finden, können Algorithmen für diese Fine-Grained-Visual-Classification hilfreich sein. In einigen Arbeiten wurde gezeigt, dass bei der Klassifikation von Fotos wichtig ist, relevante Bereiche im Bild zu lokalisieren. Dieser Aspekt ist bei Fotofallenfotos umso wichtiger, da die auslösende Kamera fest montiert ist. Es liegt nahe, dass ein Bildklassifikator auf großformatigen Fotos weniger erfolgreich sein dürfte, als auf dem Ausschnitt des Fotos, auf dem nur das Tier zu sehen ist.

In dieser Arbeit wird betrachtet, ob mithilfe von ML-basierter Object-Detection, wie MegaDetector, die Lokalisierung von Objekten auf Fotofallenfotos dazu beiträgt, die Güte von Bildklassifikatoren zu verbessern. Dazu werden Bildklassifikatoren verglichen, die auf Basis von rohen Bildern, händisch gesetzten Ausschnitten von Citizen-Scientists sowie automatisch gefundenen Ausschnitten trainiert wurden. In einer zukünftigen machine-learning-gestützten Citizen-Science-Plattform könnten MegaDetector und der beste Bildklassifikator eingesetzt werden, um automatische Bestimmungen der gefundenen Tiere durchzuführen.

Die Ergebnisse zeigen, dass MegaDetector bei 90% der Fotos richtige (IoU=.75) Bildausschnitte findet. Ein auf solchen Ausschnitten trainiertes Modell erzielt fast gleichwertig gute Ergebnisse (Mittlerer F_1: 0.889, Precision: 97.7%, Recall: 92.0%, Accuracy: 95.0%) wie ein auf von Experten gesetzten Ausschnitten trainiertes Modell (F_1: 0.909, P: 98.3%, R: 92.5%, A: 95.6%) und bessere Ergebnisse als ein auf Rohbildern trainiertes Modell (F_1: 0.873, P: 97.0%, R: 91.8%, A: 94.3%). Versuche mit einem Teildatensatz zeigen, dass Ausschnittklassifikatoren (F_1: 0.856) besser generalisieren und damit besser auf Fotos anderer Standorte anwendbar sind, als Rohbildklassifikatoren (F_1: 0.709).

Stichworte: Machine-Learning, Computer-Vision, Image-Classification, Object-Detection, MegaDetector, Biodiversität, Artenschutz, Citizen-Science

Danksagung

Diese Arbeit wäre nicht möglich gewesen ohne die Unterstützung der Senckenberg Gesellschaft für Naturforschung und den vielen fleißigen Citizen-Scientists im Projekt WildLIVE!.

Besonders möchte ich Vanessa danken, die mich bei allen Fragen und Sorgen unterstützt hat.

Mein Dank geht auch an Prof. Hergenröther und Prof. Weinmann, die mir die Chance gegeben haben, dieses Projekt in die Wege zu leiten und durchzuführen.

Mein Dank geht auch an die Accso - Accelerated Solutions GmbH, namentlich Valentin, Xenija, Volker und Florian, die mich mit Expertise, Freiraum und Hardware unterstützt haben. "Jarvis", der Machine-Learning-Rechner, hat - bis auf einen größeren Aussetzer - unzählbar viele Bits jongliert und damit ermöglicht, die Forschungsfragen überhaupt zu beantworten.

Zuletzt gilt großer Dank meinen Freunden, die mich überhaupt erst auf die Spur der großen und kleinen tierischen Freunde gelenkt haben.

Inhaltsverzeichnis

Abkürzungsverzeichnis

AP Average Precision

AR Average Recall

AUC Area under curve

BoF Bag of features

COCO Common Objects in Context

CVPR Conference on Computer Vision and Pattern Recognition

DETR Detection Transformer

FN False Negative

FP False Positive

FGVC Fine Grained Visual Classification

GPU Graphical Processing Unit

HOG Histogram of oriented gradients

IoU Intersection over Union

CNN Convolutional Neural Network

CV Computer Vision

mAP mean Average Precision

ML Machine Learning

NAS Neural Architecture Search

R-CNN Region-Based-Convolutional-Neuronal-Network

ReLU Rectified Linear Unit

ROC Receiver operation statistic

SIFT Scale-invariant feature transform

SURF Speeded-up robust features

TP True Positive

TN True Negative

ViT Vision Transformer

YOLO You Only Look Once

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

1.1 Motivation

Neben der anthropogenen Klimaerwärmung, die schon heute direkt Auswirkungen auf das Leben auf der ganzen Welt hat, ist die Biodiversitätskrise mit dem anthropogenen Artensterben eine weitere Herausforderung für die internationale Staatengemeinschaft und jeden einzelnen Menschen. Das Leben und alle Ressourcen, die für unser Leben und Zusammenleben nötig sind, beziehen wir aus dem, was gemeinhin als Natur bezeichnet wird. Die Natur setzt sich aus lebendigen Organismen zusammen, die in ihrer Diversität und Fülle so reichhaltig sind, dass sie fast jeden Winkel der Erde besiedeln. Die Diversität und Fülle des Lebens sinken weltweit (vgl. [Bro+19], XIV) und dieser Trend beschleunigt sich (vgl. [Bro+19], S. XVI), was auf die menschliche Transformation seiner Lebensräume im gegenwärtigen sogenannten Anthropozän zurückzuführen ist (vgl. ebd., S. XV). Anders als bei Klimaveränderungen, die prinzipiell reversibel sind, ist das Aussterben von Arten irreversibel. Zudem lässt sich, anders als bei der Klimaerwärmung, die mit der globalen mittleren Oberflächentemperatur gemessen werden kann (vgl. [Lee+23], V), die Stabilität von Ökosystemen nicht an wenigen Kennzahlen ablesen. Wichtige Einflussfaktoren und Kennzahlen von stabilen Ökosystemen sind deren Biodiversität, Fülle der Organismen (vgl. [Cle11]) sowie Konstanz, Resistenz, Resilienz und Performanz (vgl. [GSW92], 146).

Trotz der Schwierigkeit, ein stabiles Ökosystem im Angesicht der vielfältigen menschlichen Ausbreitungen und Einflüsse zu messen oder gar zu erhalten, versuchen Wissenschaftler - vornehmlich Biologen, Ökologen, Soziologen und Geologen - die ökologischen Zusammenhänge besser zu verstehen und daraus Empfehlungen für politisches Handeln abzuleiten (vgl. [Bro+19], S. IV). Eine momentane oder verlaufsmäßige Übersicht über den ökologischen Zustand von beforschten Regionen ist somit maßgeblich für die Bestandsaufnahme und die Evaluation von politischen, wirtschaftlichen und ökologischen Veränderungsprozessen.

In den letzten Jahren wurden häufig Laien in die Arbeit von Biologen, Ökologen und Zoologen eingebunden (vgl. [HGB18], S. 193). Insbesondere bei Schwerpunktsetzung, Methodik und Durchführung von wissenschaftlichen Projekten können Laien - auch Citizen-Scientists oder Bürgerwissenschaftler genannt - mit Know-How, Zeit, Perspektive und Reichweite wertvolle Hilfe leisten. Gerade im Bereich der Biologie besteht bezogen auf Biodiversität, Ökosysteme und Artenkenntnis ein Potential, das sich aus zwei Bereichen speist: Erstens liegt ein über Freunde oder Familie tradiertes Fachwissen vor, das nicht professionell eingerahmt ist und über Citizen-Science-Projekte genutzt werden könnte. Zweitens kann die Beschäftigung mit Naturphänomenen bei vormals naturfremden Unterstützern viel Nichtwissen ausräumen und damit zu mehr Problembewusstsein, systemischem Naturwissen, methodischem Verständnis, Partizipation, Reichweite und letztlich politischem Gewicht führen, das die Entschärfung der Biodiversitätskrise dringend benötigt.

Abbildung 1.1: Fotos aus dem Projekt “WildLIVE!”

Im Wissenschaftsbetrieb, sowohl unter Einbezug von Citizen-Scientists, als auch ohne, liegen Fragestellungen vor, in denen Daten gesammelt, erfasst, katalogisiert und ausgewertet werden. Die Biodiversitätsforschung, insbesondere das Biomonitoring, mit ihren multiplen Parametern und dynamischen Einflussfaktoren, ist stark mit dieser Herausforderung des Datenüberflusses konfrontiert (vgl. [Bee21], S. 15). Es liegt nahe, die Tätigkeiten des Sammelns, Erfassens, Katalogisierens und Auswertens an möglichst vielen Stellen zu automatisieren, wo Risiken durch maschinelle Störungen minimal sind.

In vielen Bereichen kommen für das Biodiversitäts-Monitoring Kamerafallen zum Einsatz (vgl. [Cos20], S. 106-7). Im Rahmen des Projekts “WildLIVE!” des Senckenberg Forschungsinstituts werden tropische Ökosysteme in der Provinz Chiquitano im Osten Boliviens und subtropisch-mediterrane Ökosysteme um den Fluss Baviaanskloof in Südafrika beobachtet. Durch eine Beobachtung über mehrere Jahre können klimatische und wirtschaftliche Veränderungen in den Regionen mit der Biodiversität in einen Zusammenhang gesetzt werden. Das Fotomaterial (s. Abbildung 1.1) wird im weiteren Verlauf von Biologen und Citizen-Scientists ausgewertet (vgl. [Wil14], [WJ02]).

Die Beteiligung am Citizen-Science-Projekt “WildLIVE!” , lässt sich nach Daum (2023) als ein “Beispiel der Fürsorge gegenüber Biodiversität unter Einbezug von ästhetischen Elementen und Technik” (vgl. [Dau23], S. 240) betrachten. Citizen-Scientists können sich einbringen, vernetzen, Wissen erlangen und wunderbare Fotos und Situationen aus der Wildnis entdecken. Nach dem Bewertungsrahmen von Gorke (vgl. [Dau23], S. 234) lassen sich für dieses Projekt sehr geringe Eingriffe in die Natur feststellen, die nicht über die Art der Eingriffe hinausgeht, die für das wissenschaftliche Projekt ohnehin nötig wäre. Das Senckenberg bietet somit eine Plattform für eine “Praxis der Fürsorge ohne Berührung”.

Computer-Vision- und Machine-Learning-Verfahren können hier eingesetzt werden, um Bestimmung/Klassifizierung der beobachteten Arten oder Anonymisierung von Menschen oder Fahrzeugen im Bild durchzuführen. Eine wichtige Aufgabe im Projekt “WildLIVE!” stellt dar, Tiere, die auf Fotofallen-Fotos abgebildet sind, korrekt einer biologischen Art oder Gruppe zuzuordnen. Bildklassifikatoren basierend auf tiefen neuronalen Netzen, wie einem Convolutional Neural Network (CNN) oder einem Transformer, sind seit einigen Jahren in der Lage, bei solchen Problemstellungen zu unterstützen (vgl. [Wei+16], [van+20], [BvP18],

[Nor+18], [Che+], [Par+], [Bee21], [Han+20], [He+21], [Che+19], [Rig+23]). Mithilfe von solchen maschinellen Klassifikatoren kann die Arbeit von Forschern beschleunigt werden.

1.2 Ziele

Die Arbeitsgruppe des Fine Grained Visual Classification (FGVC) beschäftigt sich seit mehr als zehn Jahren mit der Frage, wie Klassifikatoren - auch im Bereich Biodiversität - optimiert werden können, auch feine optische Unterschiede in Datensätzen mit einer Vielzahl von Zielklassen zu unterscheiden (vgl. [FGV31]). Der Phänotyp der Organismen - also die Menge chemisch-biologischer Merkmale - ist ein wesentliches Kriterium für die Unterscheidung von Arten (vgl. [Kun18], S. 171). Bei vielen Arten lässt sich eine Bestimmung anhand äußerer Merkmale jedoch nicht vornehmen. Oft liegen nur - wenn überhaupt - wenige äußere Merkmale vor, die zu einer Distinktion verwendet werden können. Für diese feinen Unterscheidungen ist es wichtig, dass die optischen Merkmale im verwendeten Bild nicht verwaschen oder komprimiert werden. Da der Rechenaufwand durch größere Auflösungen überproportional steigt, müssen große Bilder entweder herunterskaliert oder zugeschnitten werden.

In dieser Arbeit Klassifikatoren für die Bestimmung von Tierarten auf Basis von Fotos aus der Natur entwickelt, die auf nicht reduzierten Rohbildern und Bounding-Box-Ausschnitten dieser Rohbilder trainiert sind.

In einem ersten Experiment werden diese Klassifikatoren verglichen, die einen zentral abgebildeten Organismus finden und bestimmen können. Gemäß der Arbeiten von Beery et al., Rigoudy et al. und Miele et. al ([BvP18], [Rig+23], [Mie+21]) lässt sich vermuten, dass solche Klassifikatoren, die auf Bildausschnitten (s. Abbildung 1.2) trainieren und arbeiten, bessere Performanz liefern, als Klassifikatoren, die auf Rohbildern arbeiten. Diese These soll für Fotos aus dem “WildLIVE!”-Projekt validiert werden.

In einem zweiten abgewandelten Experiment werden Klassifikatoren verglichen, die auf unterschiedlichen Ausschnitten trainiert wurden. Häufig stehen keine fachlich gesicherten Daten zur Verfügung, um auf Fotos korrekte Ausschnitte des abgebildeten Organismus zu schneiden. Vom Projekt “WildLIVE!” stehen jedoch Bounding-Box-Daten zur Verfügung, um die Objekte von Interesse auf Fotos zu finden und zu klassifizieren. Andere Ausschnitte werden mithilfe von Bounding-Boxes erzeugt, die aus machine-learning-basierten Object-Detection-Verfahren wie Faster R-CNN oder You Only Look Once (YOLO), resultieren (vgl. [Ren+15], [Red+15]). Hier liegt also der Schwerpunkt auf dem Vergleich von Klassifikatoren, die auf Expertendaten und machine-learning-gestützten Daten trainiert wurden. Sollte ein Verfahren gefunden werden, das vollautomatisiert ablaufen kann und ähnlich gute Performanz liefert, wie ein citizen-science-gestütztes Verfahren, wäre dies ein hilfreiches Werkzeug für die Skalierung von weiteren Forschungsvorhaben des Senckenberg Forschungsinstituts und anderer Biomonitoring-Projekte auf der ganzen Welt.

Zudem besteht das Ziel, zu prüfen, ob die entwickelten Klassifikatoren auf neue Umstände und variiertes Fotomaterial übertragbar sind, denn Studien zeigen, dass Klassifikatoren für Fotofallenfotos häufig eine geringe Generalisierbarkeit aufweisen (vgl. [BvP18]).

Abbildung 1.2: Von Citizen-Scientists gesetzte Bildausschnitte aus dem Projekt “WildLIVE!”

1.3 Abgrenzung

Diese Arbeit hat den Fokus, auf Basis einer Trainingsstruktur einen Vergleich zwischen verschiedenen Klassifikatoren zu ziehen. Es wird kein tief gehendes Optimieren des Klassifikators oder Object-Detectors vorgenommen. Die relative Güte der Klassifikatoren untereinander ist somit im Gegensatz zur absoluten Güte zentral.

Manchen automatischen Bestimmungssystemen wird einverleibt, bei unsicherer Inferenz auch höhere Taxa, wie Gattung, Familie oder Ordnung vorzuschlagen (vgl. [Gom+21]). In dieser Arbeit wird für eine solche Kombination Vorarbeit geleistet. Entgegen aktueller Forschungsschwerpunkte bei Vision Transformer (ViT) und vielversprechenden Ergebnissen dieser Modelle bei Bildklassifikation, werden ViT in dieser Arbeit nicht eingesetzt. Es lässt sich annehmen, dass jedoch die vorgestellten Werkzeuge und Vorgehensweisen auch für ViT anwendbar sind.

1.4 Struktur

Nach dieser Einleitung werden in einem Grundlagenkapitel die Hintergründe zu Biologie und Ökologie, sowie Citizen-Science erläutert, die für diese Arbeit zentral sind. Zudem werden Zusammenhänge und Algorithmik der eingesetzten Klassifikations- und Objekterkennungs-Verfahren, sowie Datenstrukturen und Metriken skizziert. Im dritten Kapitel findet eine Besprechung des Forschungsstands zu FGVC und Machine-Learning im Bio-Monitoring statt. Im vierten Kapitel werden die Datensätze aus dem Projekt “WildLIVE!” vorgestellt. Darauf folgt das Kapitel über die Versuchsaufbauten der zwei Experimente, an die sich ein Auswertungskapitel anschließt. Im letzten Teil werden die Ergebnisse diskutiert und in einen über diese Arbeit hinausgehenden Ausblick eingeordnet.

2 Grundlagen

In diesem Kapitel werden wichtige Themengebiete, die für das Verständnis dieser Arbeit zentral sind, vorgestellt. Beginnend mit den fachlichen Domänen Citizen-Science (2.1) und Biodiversität (2.2) wird ein Überblick über die Möglichkeiten und Herausforderungen von kooperativer Biodiversitätsforschung beleuchtet. In Abschnitt 2.3 werden die verschiedenen Teilgebiete der Computer-Vision erläutert, auf welchen diese Arbeit technisch und methodisch basiert. Im darauffolgenden Kapitel 3 werden Forschungsergebnisse aus dem Bereich Bildklassifikation und Object-Detection insbesondere im Bereich Biodiversität vorgestellt.

2.1 Citizen-Science

Viele Menschen bringen Zeit und Wissen in ihrer Freizeit im Rahmen von Bürgerwissenschaft in Forschungsprojekte ein. Dies kann die Formulierung von Forschungsfragen, die Sammlung von Daten, die Annotierung von Metadaten, die Einbringung von persönlichen Ressourcen, die Aufbereitung von Forschungsergebnissen oder die Kommunikation an die Öffentlichkeit umfassen. Viele Citizen-Scientists verfolgen Bildung als Ziele ihrer Partizipation (vgl. [BLB05]). Ein Antrieb ist zudem die spielerische Lösung von definierten Aufgaben und Problemstellungen. Oft genug reicht aber schlicht die Freude an der Beteiligung an einem echten Forschungsprojekt aus (vgl. [Jan+24], S.6). Ein Fokus von Citizen-Science-Projekten liegt auf dem Bereich Natur- und Umweltschutz.

Die stärkere Verbreitung von Citizen-Science ist unter anderem auf die Verbreitung von Digitaltechnologie zurückzuführen (vgl. [Sil09], S. 470), wie auch das Projekt “WildLIVE!” anschaulich macht. Die fortschreitende Entwicklung von digitalen Sensoren führt aber gleichzeitig gerade im Bereich des Biodiversitäts-Monitoring zu einem Anstieg der gesammelten Datenmengen, die dann wiederum in Citizen-Science-Projekten geordnet werden können.

In auf Daten angewiesenen Forschungsbereichen muss berücksichtigt werden, die Datenqualität der resultierenden Datensätze zu prüfen, die aus der Sammlung und Annotierung entstehen. Riesch et al. schlagen vor, die Datenqualität durch Trainings zu verbessern, eine Validierung durchzuführen und Aufgaben simpel zu halten (vgl. [RP14]). Auf der Plattform iNaturalist, wo weltweit Amateure Lebewesen auf Basis von Fotos bestimmen, wird die Validierung der Genauigkeit mit Bestimmungen von Biologen durchgeführt (vgl. [iNa06]).

2.2 Biodiversität

Biodiversität beschreibt die Diversität der Natur auf den Ebenen der Gene, der Organismen und der ökologischen Systeme (vgl. [WN14], S. 4 ff.). Gene bestimmen den Phänotyp einer Art, der morphologische und physiologische Merkmale und Verhaltensmerkmale ausmacht. Die genetische Diversität gibt an, wie viele verschiedene Genvarianten aller betrachteten

Individuen innerhalb der Arten existieren. Fortpflanzungsweise und Mobilität bestimmen unter anderem die Diversität eines Genpools. Ein biologisches Taxon beschreibt eine Gruppe von beschriebenen Lebewesen. Taxa ordnen sich in eine hierarchische biologische Taxonomie ein. Sie besteht aus allgemeinen bis spezifischen Taxa, wie Domäne, Reich, Klasse, Ordnung, Familie, Gattung, Art, Unterart und Rasse. Die Diversität der Arten formt sich aus der Quantität (Anzahl) und Qualität (Unterschiedlichkeit und Eigenschaften) der Spezies in einem System. Unter einem Ökosystem versteht man "ein Wirkungsgefüge von Lebewesen und deren anorganischer Umwelt, das offen und bis zu einem gewissen Grad zur Selbstregulation befähigt ist" ([WN14], S. 17). Alle Lebewesen sind auf Strukturen und Zusammensetzungen von Ökosystemen angewiesen (vgl. [WN14], S. 7-17).

Die Gefährdung der Biodiversität geht auf die "Übernutzung durch direkten Zugriff, anthropogene Standortveränderungen inklusive Lebensraumvernichtung und Nutzungswandel von Ökosystemen durch den Menschen" ([WN14]) zurück. Genetische Vielfalt wird in begrenzten Systemen beforscht. Die starke Abnahme von Artendiversität und insbesondere auch der Fülle der Arten (vgl. [Bro+19]) wird in den letzten Jahren durch Monitoring der Bestände dokumentiert. Die Zerstörung von Ökosystemen durch Industrialisierung aller Wirtschaftszweige ist offensichtlich.

Diese Arbeit widmet sich Forschungsprojekten, welche die Vielfalt und die Fülle von Arten in einem begrenzten Gebiet optisch überwachen, denn "Bestandsaufnahmen und deren regelmäßige Wiederholungen (Monitoring) sind die wichtigste und die einzige unangreifbare Methode zur Ermittlung des Gefährdungsgrades von Arten und Unterarten [...]" ([WN14]). Die Veränderungen von Ökosystemen spielen dabei gerade in Regionen eine Rolle, wo noch vom Menschen weitestgehend unberührte Landschaft existiert.[1]

Das Monitoring von Arten in einem System bedingt, dass Arten eindeutig erkannt werden können. Es steht außer Frage, dass auch Experten sich bei der Bestimmung von Arten täuschen können (vgl. [Cul+03]). Die Artbestimmung hat aber prinzipielle Grenzen, die auf die Taxonomie selbst zurückgehen. Wo die Grenze zwischen zwei Arten verläuft, wird vielfältig diskutiert. Der visuelle Phänotyp ist jedoch beileibe nicht immer entscheidend. Zudem mag es bei manchen Arten möglich sein, visuelle Unterschiede bei Tageslicht und mit bloßem Auge zu erkennen. Andere Distinktionen benötigen jedoch Fernglas, Lupe, Mikroskop oder sogar eine invasive oder letale Sezierung. Das Biomonitoring, ob durch akustische, optische oder genetische Methoden, erzeugt große, teuer zu verarbeitende Datenmengen (vgl. [Nor+18]), die häufig auch noch sehr viel Ausschuss enthalten. Leere Fotofallenfotos, die durch Fehlauslösung der Kamera entstehen, sind hier ein passendes Beispiel. Diese Datenmengen schnell und systematisch auszuwerten, stellt eine große Herausforderung dar. Trotz aller Herausforderungen lassen sich viele Anwendungsfälle in der Biodiversitätsforschung mit automatischen optischen Verfahren unterstützen, um Experten die Möglichkeit zu geben, sich auf die bisher maschinell unlösbaren Probleme zu fokussieren.

[1] Die Veränderungen des in dieser Arbeit betrachteten Ökosystems in Chiquitano (Bolivien, [One21]) wurden unter anderen von Jansen et al. ([JGK09], [Jan+11], [Bál+18], [Rom+19], [Jan+20]) untersucht.

2.3 Computer-Vision

Die Felder Objekterkennung, Objektklassifizierung und Objektidentifikation in der Computer-Vision begannen in den 1950er- und 1960er-Jahren unter anderem mit der Problemstellung der Zeichen- und Fingerabdruckerkennung (vgl. [JM06], S. 4). Weitere Problemstellungen wurden im Forschungsfeld der Computer-Vision Ende der 1960er-Jahre mit künstlichen neuronalen Netzen am Vorbild des menschlichen Sehens begonnen zu bearbeiten. Aber auch ohne neuronale Netze wurde versucht, zweidimensionale und dreidimensionale Strukturen und Inhalte auf Basis von zweidimensionalen Bildern zu erkennen. Dafür wurden zumeist Objektkanten gesucht und extrahiert.

In den 2000er-Jahren etablierte sich ein Trend, Objekterkennung mithilfe von feature-basierten Verfahren zu lösen. Machine-learning-basierte Ansätze begannen, die Computer-Vision-Forschung zu dominieren, was auch auf die Bereitstellung neuer großer Datensätze und steigende Rechenleistung zurückzuführen ist. Dieser Trend beschleunigte sich in den 2010er-Jahren nochmals (s. [Sze22], S. 18ff.).

Nun befasste man sich mit dem Finden und Einordnen von automatisch extrahierten Bildmerkmalen (Features), wie Farbverläufen, Oberflächenstrukturen, o.ä., in vorgegebene Kategorien. Diese Bildmerkmale werden in der Regel mithilfe von Kontrastvergleichen oder Kantendetektion gefunden. Die Features werden dann mithilfe eines Feature-Descriptors in einen Feature-Vektor, der alle Features in einem hochdimensionalen Raum verteilt, übersetzt. Features, die in diesem Raum in der Nähe sind, sind sich ähnlich. Um aus den Features Rückschlüsse auf das abgebildete Objekt, das durch die Features ausgezeichnet ist, zu schließen, wird ein Klassifikator optimiert, der einen Zusammenhang zwischen Feature-Vektoren und Klassen (Labels) herstellt (vgl. [LW07], S. 826). Dies ist nur möglich, wenn genug annotierte Daten vorliegen, die einen Zusammenhang zwischen Bildern und Klassen begründen.

Die Herangehensweise lässt sich in zwei Ansätze teilen: Klassische/Traditionelle Computer-Vision ohne Machine-Learning und machine-learning-basierte Ansätze (vgl. [OMa+]), die in den folgenden Abschnitten vorgestellt werden.

2.3.1 Klassische Computer-Vision-Ansätze

Im Bereich Computer-Vision können klassische Ansätze als solche beschrieben werden, die ein Problem analytisch in Einzelteile zerlegen und mit algorithmischen Verfahren lösen. Es wird versucht, ohne eine große Menge an annotierten Bildbeispielen, Strukturen zu finden und zu extrahieren. Zu diesen Ansätzen lassen sich Verfahren wie Scale-invariant feature transform (SIFT), Speeded-up robust features (SURF) oder Bag of features (BoF) zählen.

Ein rudimentärer Ansatz für eine Ermittlung von wichtigen Bildmerkmalen (Features), wie der Oberflächenstruktur und Farbe eines Tierfells, wäre, Vorlagen des gewünschten Objekts im Bild zu suchen, indem das Bild mit der visuellen Schablone (bzw. Faltung, engl. Convolution-Matrix oder Kernel) abgefahren und verglichen wird. Verfahren mit diesen Ansätzen sind jedoch nicht robust gegenüber Merkmalen, die wegen affiner Transformationen, wie Skalierungen, Scherungen oder Rotationen, verändert sind oder anders beleuchtet sind.

Gefundene Features können im zweiten Schritt für verschiedene Zwecke herangezogen werden. Die Features in Form von Feature-Vektoren, also numerischen Abbildungen von Bildbereichen, spannen einen sogenannten Feature-Space[2] auf, der visualisiert, gruppiert oder gemäß der Koordinaten klassifiziert werden kann. Vor dem Durchbruch von Deep-Learning für Bildklassifikation wurden Features aus Bildern mit Verfahren wie SIFT, SURF oder Bag-of-Features extrahiert (vgl. BoF:[SB11]).

Am Problem der Bildklassifikation wurde mit allen bis dahin bekannten Ansätzen wie Decision-Trees, Support-Vector-Machines, regelbasierten Verfahren sowie Supervised- und Unsupervised-Verfahren gearbeitet (vgl. [LW07], S. 830 ff.). Erst aus den faltungsbasierten Ansätzen (Convolutions) in Kombination mit großen annotierten Datensätzen entwickelten sich Verfahren, die in den 2010er-Jahren in vielen Bereichen eine mit menschlicher Leistung vergleichbare Genauigkeit erreichten, indem die Parametrisierung und Ermittlung der signifikanten Faltungen mithilfe von Machine-Learning-Ansätzen automatisiert wurde (vgl. [LKF10]). Wo bei der maßstäbesetzenden "Large Scale Visual Recognition Challenge" ([Sta20]) die Fehlererkennungsrate im Jahr 2011 noch bei etwa 25% lag, erreichte sie durch Aufkommen von Deep-Learning in Form von CNNs in weniger als fünf Jahren niedrige einstellige Prozentpunkte (vgl. [Rob16]).

2.3.2 Computer Vision mit Convolutional-Neural-Networks

Machine-Learning-Ansätze, oder als Teil davon sogenannte Deep-Learning-Ansätze, sind auf eine große repräsentative Menge an Trainingsdaten angewiesen. Im Machine-Learning werden mithilfe von Supervised- oder Unsupervised-Learning Muster in den Daten erkannt und maschinell verinnerlicht, um diese auf ungesehenes Material zu übertragen und über dieses Aussagen treffen zu können.

Den Durchbruch bei Computer-Vision-Aufgaben lieferten mit Verfügbarkeit von leistungsfähigen Grafikkarten (Graphical Processing Unit (GPU)) und größeren offenen Trainingsdatensätzen, die sogenannten CNNs.

CNNs sind tiefe neuronale Netze, die sich primär durch den Einsatz von konvolutionellen Schichten (Convolutional-Layern) auszeichnen.[3] Diese Convolutions bzw. Faltungen werden auf einen Vektor oder ein vektorisiertes Bild angewandt und abstrahieren diesen. Die Parameter der Faltungsmatrizen[4] bestimmen die Art der Abstraktion. Nach diesem Schritt liegen Vektoren vor, die Abstraktionen lokaler Bereiche im Bild darstellen. Hinter die Faltungen werden Max-Pooling-Layer (vgl. [Yam+90], S. 1077) geschaltet, die diese Bereiche mithilfe eines lokalen Maximums zusammenfassen und damit auch die Vektordimension reduzieren. Durch eine iterative Verschaltung von mehreren dieser Abschnitte, die aus Convolutional-Layern und Max-Pooling-Layern bestehen, und der Integration von Aktivierungsfunktionen[5] werden die wesentlichen Features des Bildes kondensiert und werden resilient gegenüber affinen Transformationen (vgl. Abbildung 2.1, vgl. [SJM18], S. 378 f.). Nach dem letzten

[2] Auch Latent-Space, Latent-Feature-Space oder Embedding-Space genannt

[3] Erste Durchbrüche von LeCun et al. 1998 (vgl. [Lec+98]).

[4] Ein Beispiel für eine Faltung ist der Gausssche Glättungsfilter, der eine Weichzeichnung eines Bildes vollzieht. Diese wird erreicht, indem jeder Pixel mit seiner Umgebung anhand einer Mittelung per Standardabweichung vermengt wird.

[5] Meistens Derivate der Rectified Linear Unit (ReLU), aber auch Sigmoid, tanh oder Softmax.

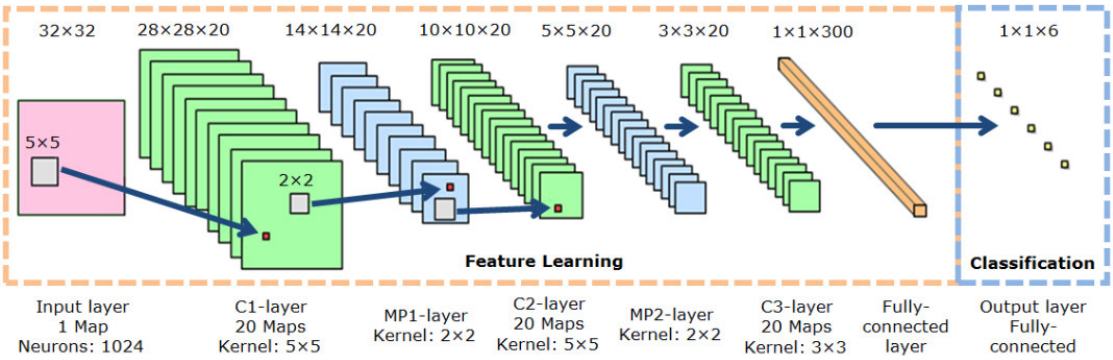

Abbildung 2.1: Ein Convolution-Neural-Network mit Convolutional-Layern, Max-Pooling-Layern und einem Kopf mit Fully-Connected-Layern (aus Nagi et al. [Nag+11]).

Abschnitt liegt eine abstrahierte vektorisierte Form der wesentlichen Bildmerkmale (Features) vor, die für verschiedene Zwecke genutzt werden können. Jedes Bild, dargestellt durch seine vektorisierte Form, liegt nun in einem Feature-Space vor und kann beispielsweise mithilfe von Clustering gruppiert, von Dimensionsreduktion visualisiert oder von Fully-Connected-Layern klassifiziert werden.

Seit LeCun et al. 1998 haben sich einige Architekturen für tiefe CNNs entwickelt (vgl. AlexNet [KSH12], VGGNet [SZ04], GoogleNet/InceptionV3 [Sze+15]/[Sze+16], ResNet [He+10], Xception [Cho10], NASNet [Zop+21], ConvNeXt [Liu+22], EfficientNet/EfficientNetV2 [MQ21]), die in der Praxis häufig wiederverwendet oder nur leicht angepasst werden. Die tatsächliche Zusammenstellung der einzelnen Layer, die bei heutigen Architekturen aus dutzenden Layern und dutzenden Millionen Parametern bestehen, wird häufig nicht händisch vorgenommen. Seit wenigen Jahren wird auch immer stärker daran geforscht, die Struktur der Layer selbst automatisch auf Basis von Neural Architecture Search (NAS) zu optimieren (vgl. [ZL17], [Zop+21]).

Machine-Learning-Verfahren können vielfältig eingestellt werden. Die Aspekte Datenaugmentierung, Optimizer, Netzstruktur, Trainingsverfahren und viele weitere, können mithilfe von Hyperparametern konfiguriert werden. Das Finden der optimalen Hyperparameter für ein Machine-Learning-Verfahren stellt eine wesentliche Aufgabe in Machine-Learning-Projekten dar. Optimale Hyperparameter werden meist durch Vergleiche mit ähnlichen Projekten oder durch zielgerichtetes Ausprobieren ermittelt.

2.3.3 Image-Classification

Bildklassifikatoren auf Basis von CNNs nutzen den extrahierten Feature-Space, um in der Kombination der Feature-Vektoren eine optimale Zuordnung der Bild-Features zu den Zielklassen zu finden. Dafür werden sogenannte Fully-Connected-Layer verwendet. Ein erster eher umfangreicher Fully-Connected-Layer nimmt alle Feature-Vektoren aus dem letzten Max-Pooling-Layer entgegen. Darauf folgt ein optionaler Dropout-Layer für Regularisierung, auf den einem Output-Layer folgt, der für jede Zielklasse einen Ausgangsknoten besitzt. Die Gewichte der Kanten zwischen den Fully-Connected-Layern beschreiben somit die Gewichtung der Feature-Vektoren in Bezug auf die Zielklassen (vgl. [Nag+11]).

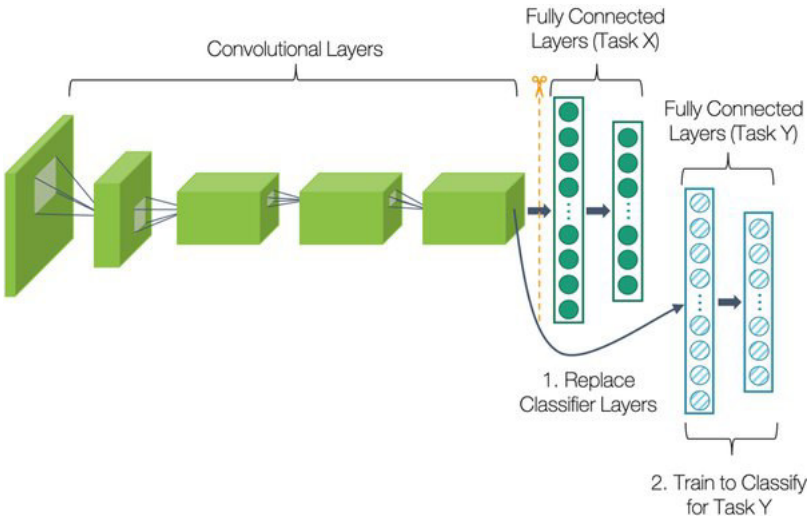

Abbildung 2.2: Transfer-Learning (aus Koul et al. [KGK20]). Convolutional-Layer werden auf einem großen Datensatz - z.B. ImageNet - als Feature-Extractor trainiert. Die ursprünglichen Kopf-Layer, die eine Klassifikation mithilfe der extrahierten visuellen Features auf der Zieldomäne X durchführen, werden verworfen und neu auf der Zieldomäne (Y) trainiert. Bei vielen Architekturen bezieht sich X auf den ImageNet-Datensatz ([Sta20]).

Je nach Aufgabenstellung müssen Bildklassifikatoren, als Erweiterung des binären Klassifikationsproblem mit zwei Zielklassen k, mehr Klassen $k > 2$ unterstützen ("multi-class classification"). Zudem liegen Aufgabenstellungen vor, in denen für einen Datenpunkt mehrere Labels $L_{min} \geq 0$ gesetzt sind ("multi-label classification"). Im Bereich Biodiversität wäre eine typische Aufgabe, auf Fotos einer Fotofalle aus $k > 2$ Klassen je nach Bild mindestens $L_{min} \geq 0$ und höchstens $L_{max} \geq 1$ Labels zu setzen.[6]

Schichten und Gewichte in neuronalen Netzen, die Abstraktionen von Informationen leisten, die domänenübergreifend gültig sind, können wiederverwendet werden, um Zeit und Ressourcen zu sparen und auf ein sicheres validiertes Fundament aufzubauen (vgl. [KGK20], [TS10]). Der Transfer von Strukturen aus bereits trainierten neuronalen Netzen in neue Domänen nennt sich Transfer-Learning. In CNNs sind die Abschnitte und Gewichte, die Features aus Bildern extrahieren, häufig wiederverwendbar, sofern sie auf einer repräsentativen Datengrundlage optimiert wurden. Da die Zuordnung von Feature-Vektoren zu ihren Klassen, also beispielsweise die Farbe und Struktur eines tierischen Fells, eine domänenspezifische Aufgabe ist, müssen die Fully-Connected-Layer mit den Fachdaten trainiert werden (s. Abb. 2.2).

Prinzipiell kann der Feature-Extractor eines CNNs zusätzlich auf spezifischen Fachdaten trainiert werden. Dies sollte jedoch nur mit Hyperparametern geschehen, die dafür sorgen, dass Gewichte der wiederverwendeten Layer nur geringfügig angepasst werden. Oft wird der Feature-Extractor beim Training eines Klassifikators hingegen eingefroren und nur der Fully-Connected-Layer trainiert. Diese Entscheidung kann auf Basis von Hyperparameter-Tuning getroffen werden.

[6]Beispiel: Wildschwein, Jaguar, Capybara $\rightarrow K = 3$. 0 bis 5 Labels pro Bild. Bild1: Jaguar, Bild2: Wildschwein, Capybara, Bild3: Leer. $\rightarrow L_1 = 1, L_2 = 2, L_3 = 0$. $L_{1,2,3} \geq L_{min} = 0$. $L_{1,2,3} \leq L_{max} = 5$.

2.3.4 Object-Detection

> The problem definition of object detection is to determine where objects are located in a given image (object localization) and which category each object belongs to (object classification). So the pipeline of traditional object detection models can be mainly divided into three stages: informative region selection, feature extraction and classification.
>
> - Jiao et al. 2019 ([Jia+19], S. 1)

Gemäß dieser Definition müssten die drei Teilaufgaben der Region-Selection, der Feature-Extraction und der Klassifikation mit jeweils passenden Werkzeugen gelöst werden. Für die Feature-Extraction kann entweder auf klassische Algorithmen, wie SIFT und Histogram of oriented gradients (HOG) oder CNNs zurückgegriffen werden. Die Klassifikation könnte mit traditionellen Ansätzen, wie Support-Vector-Machines, oder mit modernen neuronalen Netzen vorgenommen werden (vgl. [Jia+19]).[7]

In den 2010er Jahren haben sich zwei Lösungsansätze herausgebildet, die einerseits dem oben genannten traditionellen Ansatz folgen und auf Basis von informativen Bildregionen Features extrahieren und klassifizieren und andererseits das Problem als eine Regression der korrekten Bounding-Box gegenüber dem Bild auffassen (vgl. [Jia+19], S. 3). In den folgenden Abschnitten werden die drei Konzepte vorgestellt.

Region-Based-Convolutional-Neural-Networks

Ein Region-Based-Convolutional-Neuronal-Network (R-CNN) ermittelt auf einem Eingabebild bis zu tausende "Region-Proposals", um aus jedem davon Features zu extrahieren und diese zu klassifizieren. Im Rahmen von R-CNN von Girshick et al. (2013) waren diese Abschnitte noch in "Selective Search" (vgl. [Uij+13]) für Bildbereiche, CNNs für Feature-Extraction und Support-Vector-Machines für Klassifikation separiert (s. Abb. 2.3) und nicht komplett trainierbar (vgl. [Gir+13], S.1). Eine verbesserte Variante des R-CNN in Form von Fast-R-CNN integrierte alle Module in ein durchgängig trainierbares Modell (vgl. [Gir15]) und Faster R-CNN (vgl. [Ren+15]) optimiert darauf die Integration der CNN-Architektur durch Attention-Mechanismen und mit einem sogenannten Region-Proposal-Network (s. Abb. 2.3).

Bounding-Box-Regression

Das Object-Detection-Verfahren YOLO nach Redmon et al. ([Red+15]) wählt entgegen der R-CNNs den Ansatz, Bounding-Boxes und dazugehörige Klassen direkt auf dem Bild vorherzusagen. Dieser One-Stage-Object-Detection-Ansatz ermittelt mithilfe eines eher kleinen an GoogLeNet (vgl. [Sze+15]) angelehnten CNNs auf einem über dem Bild liegenden Raster Vektoren für klassifizierte Bounding-Boxen (s. Abbildung 2.4). Ein Netz wird darauf trainiert, diese Ermittlung zu optimieren und für jede Zelle mehrere klassifizierte Bounding-Boxes mit je X, Y, Breite, Höhe und Klasse zu kodieren.

[7] Zusätzlich zu den genannten Ansätzen sind auch in der Object-Detection die Transformer-basierten Ansätze, wie Detection Transformer (DETR), im Rahmen der gängigen Benchmarks in den letzten Jahren dominierend und zogen mit CNN-basierten Ansätzen gleich (vgl. [Zou+23], S. 4).

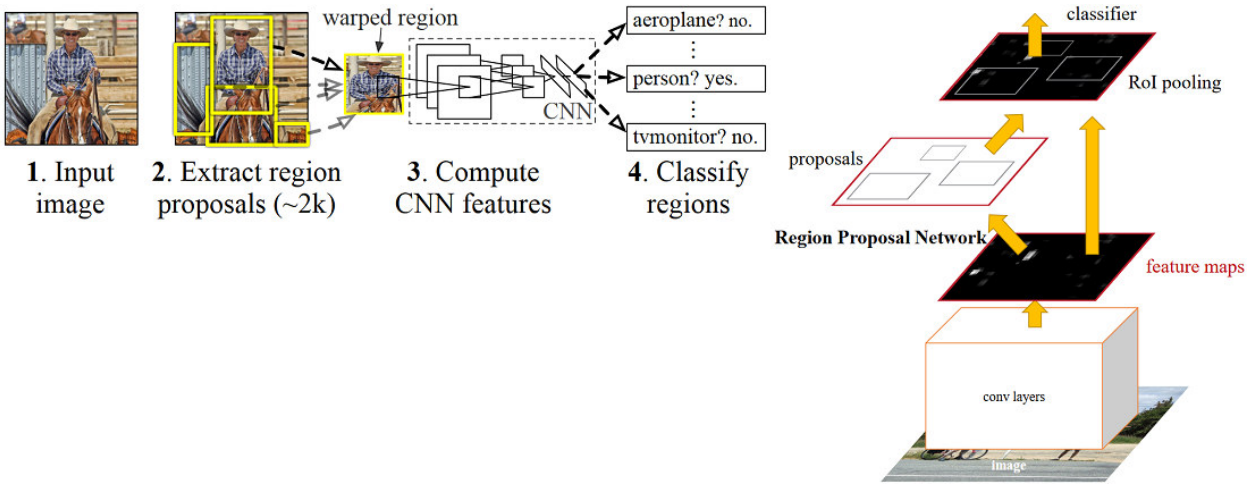

Abbildung 2.3: R-CNN nach Girshick et al. (links, [Gir+13]) und Faster R-CNN nach Ren et al. (rechts, [Ren+15]). Bei R-CNN werden aus einem Bild Bildbereiche selektiert, in Features vektorisiert und klassifiziert. Bei Faster R-CNN werden die drei Schritte mit Convolutional-Layers, einem Region-Proposal-Network und einem Klassifikator in ein neuronales Netz integriert.

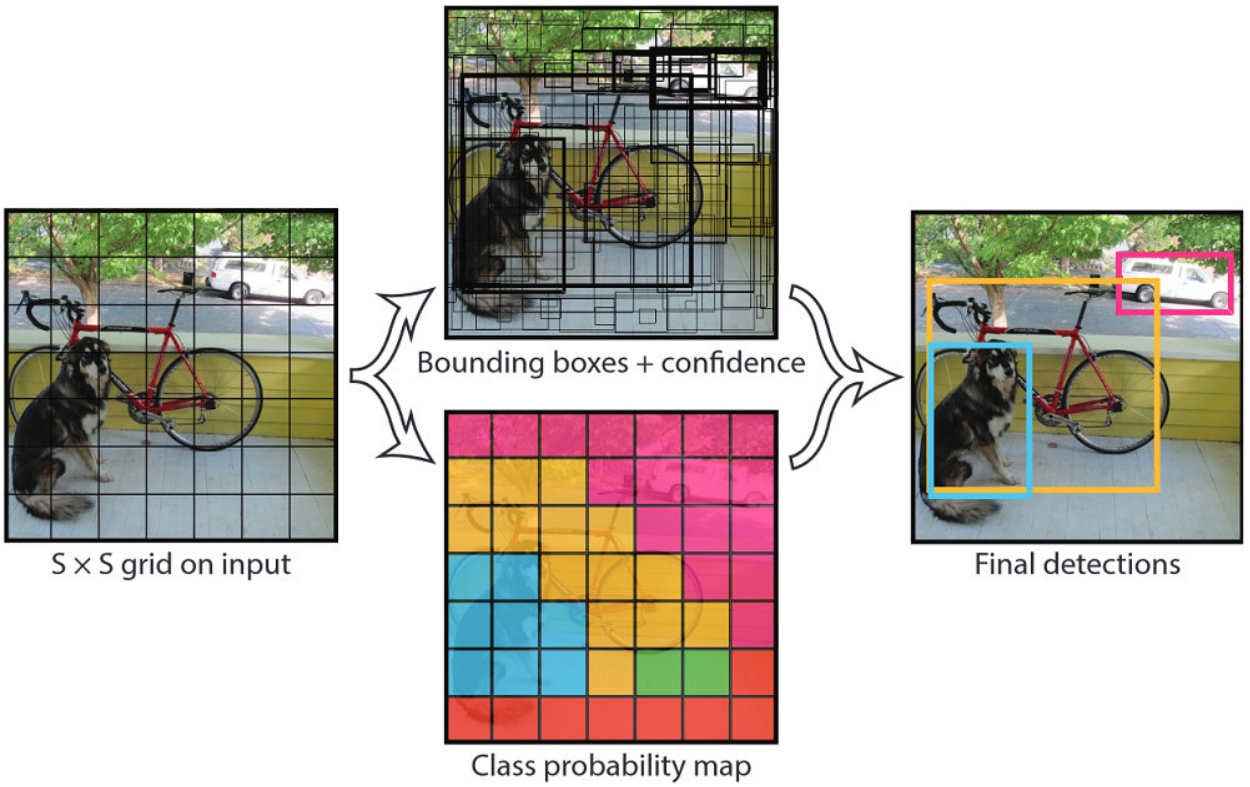

Abbildung 2.4: Ablauf von YOLO nach Redmon et al. ([Red+15]). Auf einem $S \times S$-Raster werden Bounding-Boxen mit Konfidenzen errechnet und für jede Zelle auch eine Wahrscheinlichkeit über die Klassenzugehörigkeit berechnet. Diese Aspekte werden zusammengeführt.

Mithilfe von Non-Max-Suppression werden zudem sich überdeckende gleich annotierte Boxen entfernt. In den darauffolgenden Verbesserungsschritten werden die Input-Auflösungen, die Netzstruktur und die technische Basis angepasst, um Geschwindigkeit und Performanz zu verbessern.

MegaDetector

MegaDetector ist ein auf YOLO (s. Abschnitt 2.3.4) basierendes Objekterkennungsmodell von Microsoft (vgl. [Mic24]), das für Fotofallenfotos optimiert ist (vgl. [BMY19]). Auf Basis von einigen Dutzend Datensätzen, die mithilfe von Fotofallen zusammengestellt wurden, steht Stand März 2024 eine Version MegaDetectorV5 zur Verfügung, die Objekterkennung und Klassifizierung out-of-the-box unterstützt. Im Kern steht die Objekterkennung, die mit ungesehenen Daten auf mindestens Experten-Niveau abschneidet (vgl. [BMY19]). Darauf aufbauend wäre prinzipiell auch möglich, eigene mit der Objekterkennung kombinierte Bildklassifikatoren zu trainieren.

Es existieren zwei Varianten von MegaDetectorV5. MDv5a wurde auf den Fotofallendaten, dem Datensatz Common Objects in Context (COCO) (vgl. [Lin+14]) sowie dem iNaturalist 2017-Datensatz ([van+20]) trainiert. MDv5b wurde nur auf den Fotofallendaten trainiert. Beide Versionen lassen sich mithilfe von Python in Software-Projekte integrieren und liefern für ein Foto eine Liste von Bounding-Boxen mit erkannten Klassen und einer dazugehörigen Konfidenz (Score). Ohne einen nachtrainierten Klassifikator, werden in Version 5 nur drei Klassen an den Bounding-Boxen benannt: 1: Tier, 2: Mensch, 3: Fahrzeug. Zudem wird die Klasse 0 für leere Bilder verwendet. MegaDetector kann mithilfe eines hohen oder niedrigen Schwellenwerts mehr oder weniger empfindlich eingestellt werden. Bei MDv4 waren noch Werte von T in $[0.80, 0.90]$ für eine gute Objekterkennung üblich, wohingegen bei MDv5 eine gute Objekterkennung mit Schwellenwerten bei $T = 0.20$ möglich ist.

2.4 Datenvorbereitung und Training

Das Training von Bildklassifikatoren und Objekt-Detektoren ("Modell") ist sehr auf eine große Menge von sauberen und korrekten Daten angewiesen. Die Größe der Trainingsdatensätze, die für eine erfolgreiche Klassifikation und Detektion von ungesehenen Daten nötig ist, hängt von verschiedenen Faktoren ab. Dazu zählt die angeforderte Qualität des Modells in Bezug auf eine Zielmetrik, wie Genauigkeit, Präzision oder Fehlerrate und die Variabilität der Bilder und Motive. Je höher die Performanz des Modells sein soll und je variabler die Daten sind, desto mehr Daten sind für das Training nötig.[8]

Gemäß der verschiedenen Klassifikations- oder Objekterkennungsverfahren mit "multi-class"- und "multi-label"-Ansatz können auch in Datensätzen mehrere Klassen vorkommen "multi-class dataset" oder pro Datenpunkt mehrere Klassen beschrieben sein "multi-label dataset". In der Datenvorbereitung muss somit berücksichtigt werden, dass Datensätze zur Aufgabenstellung zugeschnitten werden.

[8] Dem entgegen steht der Ansatz von iNaturalist, nur noch je maximal 1000 diverse Fotos pro Klasse für das Training ihres Klassifikators einzusetzen. Offenbar reicht diese Anzahl aus, die optischen Merkmale der Art zu extrahieren (vgl. [iNa23], 4. Grafik "Taxon Accumulation Curve").

2.4.1 Splitting

Für ein datengetriebenes Vorgehen werden nicht alle Daten direkt für das Optimieren der Parameter verwendet. Um die Performanz des Modells auch auf ungesehenen Daten zu beurteilen, wird ein kleiner Teil des Datensatzes beiseitegelegt ("Testdaten") und nach dem Training für die Evaluation herangezogen. Die Metriken, die gegenüber den ungesehenen Daten errechnet werden, gelten dann als die Performanz des Klassifikators. Weiterhin wird ein weiterer Datensatz beiseitegelegt, der während des Trainings als Referenzpunkt eingesetzt wird ("Validierungsdaten").[9]

Mit dem Validierungsdatensatz, mithilfe dessen nach jedem Trainingsschritt ("Epoche") eine Performanzmetrik berechnet werden kann, kann ein Overfitting des Modells vermieden werden. Overfitting würde sich zeigen, wenn die Metriken gegenüber den Trainingsdaten gut sind, jedoch die Metriken gegenüber Validierungs- und Testdaten deutlich schlechter sind.

Häufige Aufteilungen von Daten-Splits in Trainings-, Validierungs- und Testdaten sind 70%/10%/20%, 70%/15%/15% oder auch 60%/20%/20% (vgl. [CN08]). Je größer der Trainingsdatensatz ist, desto mehr Bildmerkmale können gefunden und gelernt werden. Je größer der Validierungsdatensatz ist, desto verlässlicher kann ein Modell gefunden werden, das mithilfe der Trainingsdaten optimale und übertragbare Ergebnisse liefert. Je größer der Testdatensatz ist, desto genauer ist die Aussage darüber, wie die Metriken des Modells mit neuen ungesehen Daten sein werden.[10]

Gerade im Bereich von Biodiversitätsdaten liegen häufig Datensätze vor, die eine ungleichmäßige Klassenverteilung besitzen. Bei Bio-Monitoring im Bereich der Biodiversitätsforschung ist naheliegend, dass gewisse Tierarten häufig, seltener und sehr selten gefunden werden - sogenannte "Long-Tail"-Datensätze. Zumeist ist jedoch wichtig, auch die visuellen Merkmale von Tierarten in die Modelle zu integrieren, die selten sind. Dazu ist wichtig, dass randomisierte Daten-Splits die Klassenverteilung der Datensätze beibehalten ("Stratified" Split). Auf diese Weise wäre bspw. die Häufigkeit von Leoparden in Trainings-, Validierungs- und Testdatensätzen in etwa gleich hoch. Dies ist die Grundlage, um Training, Validierung und Evaluation fehlerfrei durchzuführen.

Je nach Anwendungsfall ist häufig auch wichtig, dass alle Genauigkeiten, mit welchen die jeweiligen Tierarten - also Klassen des Modells - erkannt werden, wenig variieren. Es ist beispielsweise wünschenswert, dass die Sicherheit, mit der ein Leopard auf einem Foto erkannt wird, ähnlich hoch ist wie die, ein Wildschwein auf einem Foto zu erkennen.

Bei Vorliegen von unbalancierten Datensätzen wird im Bereich des Trainings auf Sampling-Methoden zurückgegriffen (vgl. [MRA20], [JZ19]). Im Bereich der Evaluation werden Metriken eingesetzt, welche die Balance der Datensätze berücksichtigen (s. Abschnitt 2.6).

[9] In der Literatur werden die Begriffe Testdaten und Validierungsdaten z.T. auch umgekehrt verwendet.

[10] Wenn sich Gesamtdatensätze im Bereich von sechs- bis siebenstelliger Anzahl von Datenpunkten befinden, können Validierungs- und Testdatensätze prozentual reduziert werden, da absolut noch genügend Vergleiche für eine verlässliche Modellselektion und Evaluation vorliegen (vgl. [CN08]).

2.4.2 Overfitting

Ein wesentlicher Aspekt des Trainings und der Optimierung von Machine-Learning-Modellen ist, Overfitting zu vermeiden. Overfitting bezeichnet das Problem, dass ein Modell zwar auf den bereitgestellten Daten optimiert ist, jedoch bei ungesehenen Daten derselben Domäne schlechte Güte erzielt - also der Transfer der Güte schlecht ist. Modelle mit Overfitting haben - wie ein Schüler, der auswendig lernt - eher die einzelnen Inhalte der Trainingsdaten verinnerlicht, statt das eigentliche Ziel zu erreichen, die abstrakten Strukturen der Klassen zu extrahieren und zu internalisieren. Die Reduzierung von Overfitting kann auch als Vorgehen erklärt werden, den Fehler in den Validierungs- und Testdaten parallel zum Fehler in Trainingsdaten zu reduzieren.

2.4.3 Regularisierung

Regularisierung beschreibt im Bereich Machine-Learning, Overfitting in Trainings-Routinen zu reduzieren, indem arithmetisch in die Struktur der neuronalen Netze eingegriffen wird oder die Ablaufsteuerung modifiziert wird. Zu den arithmetischen Verfahren gehören Verfahren, wie L_1- und L_2-Regularisierung, Label-Smoothing und Dropout-Regularisierung. Zur Ablaufsteuerung lässt sich Early-Stopping zählen. Alle Verfahren haben das Ziel, zu vermeiden, das Netz auf lokale Maxima zu optimieren, die nur den Trainings-Daten gemein sind und bei der Inferenz von ungesehenen Daten schädlich sind.

L_1- und L_2-Regularisierung führen einen Regularisierungsterm $R(\theta)$ ein, der die Höhe der Gewichte in Grenzen hält. Mithilfe eines Faktors α kann parametrisiert die Anpassung des Modells auf der einen und die Regularisierung auf der anderen Seite eingestellt werden (vgl. [Ng04], S. 2). Zusammengefasst liegt der folgende Optimierungsterm für die Parameter des neuronalen Netzes θ mit Datenpunkten M, den Zielmerkmalen y_m, den Input-Features x_m und mit $R(\theta) = ||\theta||_1 = \sum_{i=1}^{n} |\theta_i|$ für L_1-Regularisierung und $R(\theta) = ||\theta||_2^2 = \sum_{i=1}^{n} |\theta_i^2|$ für L_2-Regularisierung vor (vgl. [Ng04], (2)):

$$arg \max_{\theta} \sum_{m=1}^{M} \log p(y_m|x_m;\ \theta) - \alpha R(\theta) \ .$$

Dropout-Regularisierung setzt an, einzelne Knoten und ihre dazugehörigen Kanten in neuronalen Netzen randomisiert fallen zu lassen und dadurch die Komplexität des Netzes zu reduzieren und automatisch viele Strukturen des Netzes randomisiert zu verproben (vgl. [Sri+14], S. 1 f.).
Early-Stopping ist ein Regularisierungsansatz, der verhindert, dass ein Netz zu sehr auf die vorliegenden Trainingsdaten optimiert. Dies wird erreicht, indem geprüft wird, dass eine gewählte Performanzmetrik gegenüber Validierungsdaten steigt und dann stoppt, wenn die Metrik abfällt (vgl. [Pre02], S. 2). Ohne den Einsatz von Early-Stopping würde ein Netz bis zum konfigurierten Ende des Trainings immer weiter auf die Trainingsdaten optimieren. Ursprünglich ist das Early-Stopping für die Fehlermetrik des neuronalen Netzes vorgesehen, kann jedoch prinzipiell an jeder Metrik angewandt werden.

2.4.4 Augmentierung

Um die Datenmenge zu erhöhen, werden bei Machine-Learning-Verfahren die Datenpunkte durch Transformationen variiert und den Trainings- und Evaluations-Routinen mehrfach in

abgewandelter Form zugeführt.[11] Eine häufige Augmentierung im Bereich von Bildklassifikatoren ist der Flip, womit ein Foto sowohl in der Rohfassung als auch in einer gespiegelten Form für das Training herangezogen wird (vgl. [SK19], S. 7). Dadurch kann die Feature-Extraction unterstützt werden. Vorausgesetzt, dass der Feature-Extractor Features unabhängig ihrer Orientierung im Raum extrahieren kann, befinden sich jedoch in augmentierten Daten prinzipiell keine neuen Informationen.[12] Augmentierung kann zusätzlich zu dem Einsatz beim Training auch in der Validierung oder im Test eingesetzt werden (vgl. [Tae+11]). In jedem Fall muss gesichert werden, dass die Augmentierung nur Schritte umfasst, die das Bild so weit verfälschen, dass das ursprüngliche Label noch auf das Bild zutrifft.[13]

Weitere Augmentierungen betreffen Farbton, Sättigung, Kontrast, Gamma-Wert, Auflösung, Bildqualität, Schärfe, Rotation oder Verwendung zentraler oder zufälliger Bildausschnitte. Alle Augmentierungen sollten so eingesetzt werden, dass sie hypothetische Vorkommnisse in den Daten abbilden. Wenn angenommen werden kann, dass Bilder im Datenbestand sich in einer Eigenschaft stark unterscheiden, sollte diese Eigenschaft in der Augmentierung berücksichtigt werden.

2.5 Modellmetriken

Klassifikatoren und Objekt-Detektoren müssen, insbesondere weil sie auf variierenden Daten arbeiten, auf ihre Güte geprüft werden. Für die Evaluation von Machine-Learning-Modellen kommen verschiedene Metriken zum Einsatz, die in diesem Abschnitt vorgestellt werden. Einige Metriken sind sowohl auf Klassifikation als auch auf Objekt-Detektion anwendbar.

Die Auswahl der richtigen Metrik hängt von den Anforderungen des Forschungsfelds ab und ist entscheidend für die Einschätzung der Güte eines Machine-Learning-Modells. Je nach Priorisierung kann es beispielsweise wichtiger sein, möglichst wenige falsche Vorhersagen zu treffen oder gewisse Klassen möglichst vollständig zu finden (vgl. [DMG17], S. 8).

2.5.1 Konfusionsmatrix

Für binäre Klassifikation und Klassifikation mehrerer Ausprägungen der Zielklasse wird die Bewertung des Modells anhand der erfolgreichen und fehlerhaften Klassifikationen/Vorhersagen bemessen. Die Erfolge und Fehler teilen sich im binären Fall mit $k = 2$ Klassen in vier Kategorien, die in einer Konfusionsmatrix mit den Achsen “wahr”/Annotation und “vorhergesagt”/Klassifikation besteht (s. Tabelle 2.1). True-Positive beschreibt die Datenpunkte, die positiv sind und positiv vorhergesagt wurden. True-Negative beschreibt die Datenpunkte, die negativ sind und negativ vorhergesagt wurden. False-Positive beschreibt die negativen Datenpunkte, die fälschlicherweise als positiv vorhergesagt wurden. False-Negative beschreibt die positiven Datenpunkte, die fälschlicherweise als negativ vorhergesagt wurden.

[11] Augmentierung kann auch als eine Form von Regularisierung betrachtet werden.

[12] Beispielsweise könnte kein gutes Modell nur auf verschiedenen augmentierten Varianten eines einzigen Rohbilds trainiert werden.

[13] Beispielsweise wäre ein horizontaler Flip bei einem Klassifikator äußerst kontraproduktiv, der herausfinden soll, ob die linke oder rechte Seite eines Tieres in einem Bild zu sehen ist. Vertikale Flips sind beispielsweise bei Fotos von Landschaften problematisch, auf welchen sich unten der Boden und oben der Himmel befindet.

	Positiv vorhergesagt	Negativ vorhergesagt
Wirklich Positiv	True-Positive	False-Negative
Wirklich Negativ	False-Positive	True-Negative

Tabelle 2.1: Konfusionsmatrix einer binären Klassifikation ($k = 2$)

Auch eine Object-Detection kann anhand einer Konfusionsmatrix bewertet werden. Das Problem lässt sich als binäre Klassifikation für das Setzen einer Bounding-Box betrachten. Die Zeilen der Konfusionsmatrix beschreiben die Existenz einer wahren Bounding-Box auf einem Bild. Die Spalten beschreiben, ob die Box korrekt gesetzt wurde. Eine Box wird als korrekt betrachtet, wenn ihre Koordinaten sich genügend mit der wahren Box überlappen (s. Abschnitt 2.5.3). True Positive (TP) beschreibt somit, dass eine Bounding-Box an der richtigen Stelle gesetzt wurde. False Negative (FN) beschreibt, dass keine Bounding-Box gesetzt wurde, obwohl in den Annotationen eine Bounding-Box vorlag. False Positive (FP) beschreibt, dass eine falsche Bounding-Box gesetzt wurde.[14] True Negative (TN) beschreibt, dass wahrheitsgemäß keine Bounding-Box gefunden wurde.

Aus der Anzahl der TP, FP, FN und TN können Metriken abgeleitet werden, um Klassifikatoren und Objektdetektoren zu vergleichen. Wichtige Metriken hierfür sind:

- **Accuracy** $\frac{TP+TN}{TP+FP+FN+TN}$ Der Anteil an korrekt klassifizierten Datenpunkten gegenüber aller Datenpunkte.
- **Precision**: $\frac{TP}{TP+FP}$ Der Anteil an korrekt positiv klassifizierten Datenpunkten gegenüber aller positiv klassifizierter Datenpunkte.
- **Recall**: $\frac{TP}{TP+FN}$ Der Anteil an korrekt positiv klassifizierten Datenpunkten gegenüber aller wirklich positiver Datenpunkte.
- **Area under curve (AUC)**: Die Fläche unter der Receiver operation statistic (ROC)-Kurve, die die True-Positive-Rate als Funktion der False-Positive-Rate ($\frac{FP}{FP+TN}$) darstellt.
- F_1**-Score**[15]: $\frac{2 \cdot Precision \cdot Recall}{Precision+Recall} = \frac{2\ TP}{2\ TP+FP+FN}$ Harmonisches Mittel aus Prec. und Rec.

Je nach Anwendungsfall sind ein oder mehrere Metriken besonders wichtig. Dementsprechend wird die Modellselektion anhand der wichtigen Metriken durchgeführt. F-Metriken und AUC sind in der Lage, mehrere Metriken zusammenzufassen und vermeiden, ein Modell nur aus der einseitigen Perspektive von Accuracy und Precision zu betrachten und zu optimieren.

Im Fall von $k > 2$ Klassen ($\mathrm{K} = \{c_0, c_1, \ldots c_{k-1}\}$) und einem Label $L = 1$ pro Datenpunkt ("multi-class"-Klassifikation) wird eine Konfusionsmatrix meist als $k \times k$-Matrix aufgespannt, da jeder Datenpunkt korrekt als seine wahre Klasse oder fälschlicherweise als jede andere Klasse vorhergesagt werden kann (s. Abbildung 2.2). Für weitere Berechnungen können die Negativen einer Zeile als False-Positive und die Negativen einer Spalte als False-Negative

[14] Für gewisse geometrische Bewertungen (s.u.) kann es Sinn ergeben, die FP in zwei Metriken zu erweitern: False-Positives wegen mangelnder Überlappung und FP wegen falschem Label.

[15] vgl. zudem Varianten von gemittelten F_1-Scores nach Opitz et al. [OB-8] in Abschnitt 2.5.2

zusammengefasst werden. True-Negatives werden nicht deklariert, sondern als ein Positive einer anderen Klasse interpretiert.

Wahre Klasse ↓ | Vorhergesagte Klasse →	c_0	c_1	...	c_{k-1}
c_0	Positive	Negative	...	N
c_1	Negative	P	...	N
...	...	...	...	...
c_{k-1}	N	N	...	P

Tabelle 2.2: Konfusionsmatrix einer Klassifikation mit k Klassen

Für die Object-Detection kann die Konfusionsmatrix auf $(k+1) \times (k+1)$ erweitert werden. Die Positives P beschreiben, dass eine Bounding-Box mit der korrekten Klasse ausgezeichnet wurde. Die Negatives N beschreiben damit, dass eine Bounding-Box mit einer falschen Klasse ausgezeichnet wurde. In der Zeile c_k befinden sich die Metriken für alle Datenpunkte, welche keine wahren Bounding-Boxen besitzen. In der Spalte c_k befinden sich die Metriken für die Fälle in denen keine Bounding-Box vorhergesagt wurde. Die Zelle $c_k|c_k$ enthält die True-Negatives.

Die Metriken Accuracy, Precision, Recall, AUC und F_1-Score können prinzipiell auch für Klassifikation von $k > 2$ Klassen und die Object-Detection berechnet werden. Hierbei ist zu beachten, dass die Metriken pro Klasse berechnet und gemittelt werden können ("macro-average"), pro Klasse berechnet und gewichtet gemittelt werden können ("weighted-average") und global berechnet werden können ("micro-average"). Eine Mittelung über die Klassen stellt sicher, dass jede Klasse gleichmäßig berücksichtigt wird. Eine Gewichtung kann eingesetzt werden, um die Performanz eines Modells in Bezug auf eine Klasse höher zu priorisieren, als in Bezug auf eine andere Klasse.[16] Eine Berechnung mit "micro-average" dagegen gewichtet jeden Datenpunkt gleichmäßig, statt die Klassen gleichmäßig zu gewichten.

2.5.2 F_1-Score

F_1-Score ist ein gleich gewichteter Spezialfall aus der Gruppe der F_β-Metriken

$$F_\beta = \frac{(\beta^2 + 1)\ P\ R}{\beta^2 P + R}, \quad (0 \leq \beta \leq +\infty), \tag{2.5.1}$$

die Precision und Recall gemäß β gewichten und harmonisch mitteln. Precision und Recall bekommen bei $\beta = 1$ somit gleiches Gewicht.[17]

[16] Wenn für eine Bilderkennung wichtig ist, dass auf jeden Fall alle abgebildeten Jaguare wirklich erkannt werden, sollte der Recall für die Klasse Jaguar höher gewichtet werden: weighted-average Recall.

[17] Die F_β-Metriken basieren auf den Effectiveness-Metriken nach van Rijsbergen (vgl. [Van79], Kap. Evaluation).

Der F_1-Score lässt sich ohne Klassengewichtung nach Opitz et al. ([OB-8]) auf mindestens drei Weisen berechnen:

- **"Micro-average F_1-Score"**, $F_{1,micro}$: Die Konfusionsmatrix wird über alle Klassen aufsummiert und aus den globalen Werten für TP, FP, FN und TN werden Precision, Recall und dann F_1 berechnet. Da mit diesem Vorgehen $FP = FN = N$ gilt[18], folgt unter Berücksichtigung von $TN = 0$, $Precision = \frac{TP}{TP+FP} = \frac{TP}{TP+N} = \frac{TP}{TP+FN} = Recall = F_{1,micro}$
- **"Averaged F_1",** F_1: Das arithmetische Mittel über die harmonischen Mittel. F_1-Scores werden für jede Klasse berechnet und dann arithmetisch gemittelt. Daraus ergibt sich für k Klassen, die Precision P_k und der Recall R_k des Klassifikators in Bezug auf Klasse k

$$F_1 = \frac{1}{k}\sum_{i}^{k} F_{1,i} = \frac{1}{k}\sum_{i}^{k} \frac{2P_iR_i}{P_i + R_i} \tag{2.5.2}$$

  - [OB-8], (2)

- **"F_1 of averages",** $\mathcal{F}_1$: Das harmonische Mittel über arithmetische Mittel. Precision und Recall werden für jede Klasse in der Menge der Klassen K berechnet, dann gemittelt und aus diesen Werten der $\mathcal{F}_1$-Score berechnet. Daraus ergibt sich

$$\mathcal{F}_1 = \frac{2\bar{P}\bar{R}}{\bar{P} + \bar{R}} = 2\frac{(\frac{1}{k}\sum_{i\in K} P_i)(\frac{1}{k}\sum_{i\in K} R_i)}{\frac{1}{k}\sum_{i\in K} P_i + \frac{1}{k}\sum_{i\in K} R_i} \tag{2.5.3}$$

  - [OB-8], (3)

Gemäß Opitz et al. ist die Nutzung des F_1-Scores als arithmetisches Mittel der harmonischen Mittel robuster bei Vorliegen von in Bezug auf die Klassenhäufigkeiten unbalancierten Datensätzen. Dieser Score wird im weiteren als mittlerer F_1-Score bezeichnet.

2.5.3 IoU

Um Bounding-Boxen auf Bildern quantitativ zu vergleichen, wird das Maß Intersection over Union (IoU) eingesetzt.[19] Diese Metrik aus der Mengenlehre bildet das Verhältnis aus Überlappungs- und Vereinigungsmenge von zwei Mengen oder Bounding-Boxen A und B: (s. Abbildung 2.5, vgl. [Rez+19])

$$IoU = \frac{|A \cap B|}{|A \cup B|}.$$

In der Object-Detection wird eine abgewandelte geometrische Form der IoU verwendet, die über Eckpunkte der beiden Boxen und ihrer Überlappungs- und Vereinigungsfläche berechnet wird. Die IoU kann auf Bildern sowohl mit absoluten Pixelkoordinaten als auch mit Koordinaten relativ zur Bildgröße berechnet werden.

Für die Bewertung von Objektdetektoren spielt IoU eine große Rolle. Bei Benchmarks von Objektdetektoren wird eine Bounding-Box als korrekt eingeschätzt, wenn sie sich zu einem

[18] Bei Betrachtung der Konfusionsmatrix kann eine Fehlklassifikation als ein False-Positive in einer Zeile oder False-Negative in einer Spalte gelesen werden.

[19] Außerhalb des Bereichs Machine-Learning findet sich auch der Name Jaccard-Index.

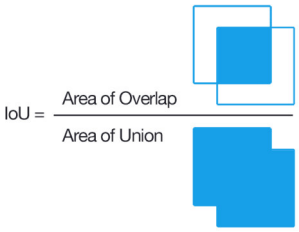

Abbildung 2.5: IoU - vgl. [Ros16]

gewissen Anteil mit der wahren Bounding-Box überlappt. Die Grade der Überlappung werden dann als Average Precision (AP) gemittelt und zuletzt über alle Klassen als mean Average Precision (mAP) gemittelt (s. Abschnitt 2.5.4).

2.5.4 Average-Precision

Für den Vergleich von verschiedenen Objektdetektoren in Bezug auf ihre Güte wird häufig die sogenannte AP, die mAP und der Average Recall (AR) verwendet. Die AP soll Aussagen darüber liefern, wie korrekt und geometrisch genau die Bounding-Boxen vom Objektdetektor vorhergesagt werden. Die mAP mittelt die AP über alle Klassen. Der AR kann Aussagen darüber treffen, wie viele der tatsächlichen Objekte im Bild vom Objektdetektor gefunden wurden (vgl. [PND20]). In dieser Arbeit sind die Recall-Maße nicht relevant, da nur Fotos verwendet werden, auf welchen ein Objekt abgebildet ist.

Das hier vorgestellte Verfahren basiert auf der Spezifikation des COCO-Datensatzes, der ein Benchmark für Objektdetektoren darstellt. Die Basismetriken TP, FP, FN und TN sind der binären Konfusionsmatrix entnommen. Das Multi-Class-Problem wird bei der Evaluation somit vorerst als k-weise binär aufgefasst. Die Metriken jeder Klasse werden später gemittelt und zusammengeführt. Im Folgenden sind erstmal TP, FP, FN, TN, P, R, AP und mAP als Metriken einer einzelnen Klasse aufzufassen.

Die AP geht auf die Fläche unter der Precision-Recall-Kurve zurück. Für eine Approximation der Fläche wird häufig ein Wert interpoliert. Für die Berechnung von Evaluationsergebnissen des COCO-Datensatzes wird eine Interpolation mit 101 Schritten verwendet (vgl. [COC11]).

Die Average-Precision wird im Rahmen der COCO-Evaluation dann für verschiedene IoUs berechnet. Dies entspringt der Idee, die Präzision einer Bounding-Box anhand unterschiedlich strenger IoU-Werte, also Überlappungsgraden zwischen der wahren und der vorhergesagten, zu berechnen. Evaluationen, die eine Überlappung der wahren mit der vorhergesagten Bounding-Box von 95% akzeptieren, sind deutlich strenger als solche, die 50% Überlappung akzeptieren. Die IoUs, die für die Evaluation herangezogen werden, kann man für 10 IoUs zwischen 0.5 und 0.95 mit Schrittweite 0.05 wie folgt kodieren: [IoU=.5:.05:.95].

Die Precision und der Recall für den IoU-Schwellenwert 50% [IoU=.5] berechne sich als

$$\begin{aligned} P^{[IoU=.5]} &= \frac{TP^{[IoU=.5]}}{TP^{[IoU=.5]} + FP^{[IoU=.5]}} \\ R^{[IoU=.5]} &= \frac{TP^{[IoU=.5]}}{TP^{[IoU=.5]} + FN^{[IoU=.5]}} \end{aligned} \tag{2.5.4}$$

mit $TP^{[IoU=.5]}$ als True Positives, die sich mit $IoU \geq 0.5$ überlappen und $FP^{[IoU=.5]}$ als False Positives, die sich mit $IoU < 0.5$ mit dem wahren Bild überlappen.

Die Average-Precision mit 101 Schritten für [IoU=.5] ergibt sich aus

$$\begin{aligned} AP^{[IoU=.5]} = \frac{1}{101} \sum_{R \in \{0.0, 0.01, \ldots, 0.99, 1.0\}} P_{interp}(R), \\ P_{interp}(R) = \max_{\tilde{R}:\tilde{R} \geq R} P(\tilde{R}), \end{aligned} \tag{2.5.5}$$

mit R dem Recall bei [IoU=.5] und $\tilde{R}$ dem stufenweisen rechtsseitigen Maximum der Precision bei [IoU=.5] an der Stelle $\tilde{R}$ (vgl. [PND20], (4) und (5)). Für die über 10 IoUs gemittelte Average-Precision ergibt sich

$$AP^{[IoU=.5:.05:.95]} = \frac{1}{10} \sum_{i=0}^{9} AP^{[IoU=0.5+i/20]} \ . \tag{2.5.6}$$

Die mean Average-Precision mAP bei [IoU=.5:.05:.95] über alle Klassen lässt sich dann als

$$mAP^{[IoU=.5:.05:.95]} = \frac{1}{k} \sum_{i=1}^{K} AP_i^{[IoU=.5:.05:.95]} \tag{2.5.7}$$

beschreiben (vgl. [PND20], (8)).

In dieser Arbeit werden die vorhergesagten Klassen des Objektdetektoren nicht weiterverwendet oder evaluiert. Alle gefundenen Boxen werden nur anhand ihrer Position und Passgenauigkeit beurteilt.

2.6 Datenqualitätsdimensionen

Im Bereich der Verarbeitung von Daten, bestehen spezifische Kriterien, die für die Einschätzung der Datensätze wichtig sind. Im Folgenden werden aus den Datenqualitätsdimensionen nach Pipino et al. [PLW02] "Appropriate amount of data", "Free-of-Error", "Objectivity" und "Relevancy" erläutert, die für die Bearbeitung der These mit den vorliegenden Daten wichtig sind.

2.6.1 Angemessene Datenmenge

Ausgehend von Pipino et al. beschreibt "Appropriate amount of data":

> the extent to which the volume of data is appropriate for the task at hand
>
> [PLW02], S. 212

Für ein datengetriebenes Vorgehen muss sichergestellt werden, dass ein Modell möglichst viele reale Varianten der Daten für das Training zur Verfügung hat. Je mehr Variabilität in der Materie vorliegt, desto mehr Daten werden für gute Vorhersagen benötigt. Je mehr Daten vorliegen, desto asymptotisch besser wird ein Modell die darinliegenden Muster verinnerlichen. Je mehr Daten vorliegen, desto bessere Aussagen können über den praktischen Einsatz getroffen werden. Die tatsächliche Anzahl an Datenpunkten, die für eine erfolgreiche Durchführung eines Machine-Learning-Vorhabens nötig ist, die zudem noch durch Augmentierung künstlich erhöht werden kann, ist jedoch nicht generisch nennbar (vgl. [DMG17], S. 4). Typischerweise bestehen Datensätze für Bildklassifikatoren mit niedrigen k aus mindestens vier- bis fünfstelligen Datenpunkten.

Zudem ist zu berücksichtigen, dass nicht nur der Gesamtdatensatz groß genug sein muss, sondern auch jede Klasse hinreichend häufig vertreten sein muss (s.a. Abschnitt 2.4.1), um die der Klasse zugrundeliegenden Muster internalisieren zu können. Bei machine-learning-basierten Computer-Vision-Verfahren wird eine Untergrenze häufig bei zwei- bis dreistelliger Anzahl an gut gestreuten Datenpunkten gezogen (vgl. [iNa23]). Um keine Klasse zu sehr zu priorisieren - auch mit Hinblick auf Undersampling -, wäre möglicherweise eine Obergrenze denkbar.

Darüber hinaus muss eingeschätzt werden, wie hoch die Variabilität in den Daten ist. Häufig korrelieren visuelle Merkmale mit anderen Merkmalen der Datensätze. Bei Fotofallendaten ist der Bildhintergrund an einer Station identisch. Im Bereich des Bio-Monitorings kann es zudem zu saisonalen Variationen durch Schnee, Regen, Sonne und Landwirtschaft kommen. Machine-Learning-Modelle können häufig nur gute Vorhersagen für Bilder treffen, die als Bildvariante im Training vorlagen.

2.6.2 Fehlerfreiheit

Ausgehend von Pipino et al. beschreibt Free-of-Error:

> the extent to which data is correct and reliable
>
> [PLW02], S. 212

Wie bei allen Datensätzen muss damit gerechnet werden, dass Labels nicht korrekt sind. Es lassen sich bei Biodiversitätsdaten für Taxonerkennung folgende Fehler annehmen:

F.1 **Falsch klassifiziertes Taxon**
Beispiel: In einem Datensatz, der als Zielmerkmal binäre Namen auf Art-Ebene enthält: Sichtung eines Jaguars (*Panthera onca*) wird als Leopard (*Panthera pardus*) ausgezeichnet. Entspricht einem False-Negative-Fehler.

F.2 **Taxon korrekt, aber zu unspezifisch**
Beispiel: In einem Datensatz mit gemischten Taxa verschiedener Ebenen: Sichtung eines Jaguars (*Panthera onca*) wird als Katze (Pantherinae sp.) ausgezeichnet.

F.3 **Taxon korrekt, aber zu spezifisch**
Beispiel: In einem Datensatz mit gemischten Taxa verschiedener Ebenen: Sichtung eines Jaguars (*Panthera onca*) wird als Jaguar (*Panthera onca*) ausgezeichnet, statt als Katze (Pantherinae sp.).

Es sind mindestens folgende Begründungen für o.g. Fehler denkbar:[20]

R.1 **Bestimmungsfehler**
Im Datenerhebungsprozess war das Taxon zwar bekannt, aber nicht korrekt, präzise genug oder grob genug ausgezeichnet.[21] Dies kann durch erneute Begutachtung reduziert werden.

R.2 **Fehlende Berücksichtigung**
Im Datenerhebungsprozess wurde das Taxon nicht berücksichtigt. Dies kann durch Einbezug von mehr Taxa reduziert werden.

R.3 **Unbekannte Art**
Das Taxon wurde generell noch nicht beschrieben (vgl. [GO04], S. 663). Hier besteht auf der Ebene der Daten keine Möglichkeit der Optimierung. Der Fehler kann trotz Schätzungen, dass noch etwa 86% (vgl. [Ame11]) der landlebenden Arten weltweit unbeschrieben sind, in vielen Forschungsprojekten mit Großtieren vernachlässigt werden.

Die Fehlerfreiheit von Datensätzen kann ohne weitere Zuhilfenahme anderer Datensätze nicht gemessen oder evaluiert werden. Nur mithilfe einer Einschätzung des Expertenwissens der Personen, auf welche die Datensätze zurückgehen, lassen eine Einschätzung zu, wie korrekt ein Datensatz sein dürfte. Je mehr Expertenwissen unabhängiger Experten in die Zusammenstellung der Daten fließt, desto weniger Fehler lassen sich annehmen.

Fehler in Datensätzen führen zwangsläufig zu Fehlern in auf ihnen trainierten Machine-Learning-Modellen. Sie können jedoch durch eine überwiegende Menge an korrekten Datenpunkten ausgeglichen werden.

[20] Alle genannten Begründungen übertragen sich auch auf Fehler, die aus Taxon-Klassifikatoren hervorgehen.
[21] Dies entspricht in der Konfusionsmatrix den Negatives, bzw. False-Positives/True-Negatives.

2.6.3 Objektivität

Ausgehend von Pipino et al. beschreibt Objectivity:

> the extent to which data is unbiased, unprejudiced, and impartial
>
> [PLW02], S. 212

Biologische Datensätze, die Beobachtungen von Menschen gegenüber ihrer Umgebung entspringen, sind einigen Biases unterlegen. Einige Arten sind häufiger, häufiger anzutreffen, scheuer oder neugieriger als andere Arten. Bei Fotos, die von Citizen-Scientists geschossen und eingereicht werden, besteht zudem ein Bias in Bezug darauf, welche Individuen überhaupt entdeckt, fotografiert und erkannt werden, bevor überhaupt eine Veröffentlichung stattfindet. Bei Fotofallenfotos besteht dieser Veröffentlichungs-Bias nicht, jedoch sind nur Tiere zu sehen, die eine passende Größe und Geschwindigkeit besitzen und sich auf Höhe der Kamera bewegen. Datensätze im Bio-Monitoring mit Fotofallen sind häufig in Bezug auf das Zielmerkmal unbalanciert (vgl. [Bee+21], S. 2; [Rig+23], S. 6; [DMG17], S. 7).

Als Datenquelle für das Trainieren von Klassifikationsmodellen können unbalancierte Datensätze ein Problem darstellen. Sollte der Validierungs- oder Evaluations-Datensatz viele Punkte weniger Klassen enthalten, sind die auf diesen Datensätzen berechneten simplen Metriken, wie Accuracy und Kreuzentropie nur für diese wenigen häufig vertretenen Klassen aussagekräftig. Im Rahmen dieser Arbeit werden für die Balance eines Datensatzes der Shannon-Gleichheits-Index und der Imbalance-Factor verwendet.

Shannon-Gleichheits-Index

Gegeben eines Vektors von Häufigkeiten (Anzahl) f, k Anzahl Klassen und n Gesamtanzahl an Datenpunkten ist der Shannon-Diversitäts-Index

$$H(f) = n \log n - \sum_{i=1}^{k} f_i \log f_i \ .^{22} \tag{2.6.1}$$

Der Shannon-Gleichheits-Index E_H ergibt sich aus einer Normierung des Shannon-Diversitäts-Indexes H auf $[0, 1]$ in Form von

$$E_H(f) = \frac{H(f)}{\log k} \ . \tag{2.6.2}$$

vgl. [NIS24]

Daraus folgt $E_H = 1$ für exakt gleichmäßig verteilte Datensätze.

Imbalance-Ratio

Die Balance zwischen größter und kleinster Klasse in einem Datensatz kann, gegeben der häufigsten Anzahl f_{max} und der seltensten Anzahl f_{min} als Imbalance-Ratio oder -Factor

$$\begin{aligned} IR(f) &= \frac{f_{min}}{f_{max}} \\ IF(f) &= \frac{f_{max}}{f_{min}} \end{aligned} \tag{2.6.3}$$

gemessen werden (vgl. [ZGX20] S. 3, [OIL17] S. 34, [PFK23] S. 4160, [Zha+21] S. 2).

2.6.4 Relevanz

Ausgehend von Pipino et al. beschreibt Relevancy:

> the extent to which data is applicable and helpful for the task at hand
>
> [PLW02], S. 212

Im Zusammenhang der Relevanz lässt sich diskutieren, ob Daten in einer Form vorliegen, die für die Aufgabenstellung hilfreich ist. Ein Aspekt sei dabei die Auflösung der Fotos.

Die Auflösung hat primär eine Auswirkung auf die Geschwindigkeit des Trainings und der Inference der Klassifikations- und Object-Detection-Modelle. Größere Bilder sind - je nach Speicher-Hardware - aufwändiger zu laden. Größere Bilder haben bei der Object-Detection zudem eine längere Laufzeit, da mehr Bereiche für Bounding-Boxes infrage kommen. Wie eingangs beschrieben, besteht ein Informationsverlust, wenn große Bilder in ein Netz eingespeist werden, dass eine deutlich kleinere Input-Auflösung zulässt. Die Auflösungsreduktion von Bildern hat aber auch den Vorteil, dass die Menge Features insgesamt reduziert wird. Dies bezieht auch die irrelevanten Features im Hintergrund mit ein. Sind Bilder zu klein kann, je nach zu klassifizierenden Objekt, keine ausreichend gute Klassifizierung vorgenommen werden.

3 Related Work

Seit 2011 wird in der Disziplin FGVC jährlich ein Workshop im Rahmen der Conference on Computer Vision and Pattern Recognition (CVPR) mit den neuesten Entwicklungen zum Thema organisiert. Object-Detection und Image-Classification auf Naturbildern im Kontext von Biodiversität sind schon seit einiger Zeit zentrale Aspekte. Wo vorher zum Teil noch auf Klassifikatoren ohne neuronale Netze gesetzt wurde (vgl. [Hao+06], [HL06], [NZ08]), sind ab den 2010er Jahren viele verschiedene Ansätze diskutiert worden, die wesentlichen visuellen Features aus Bildern zu extrahieren und zu priorisieren. Die Verfahren werden in diesem Bereich häufig an den Datensätzen ImageNet ([Den+09]), der aus 14 Millionen gemischten annotierten Fotos besteht, Microsoft COCO ([Lin+14]), der aus 328.000 zum Teil mehrfach annotierten Bildern besteht, oder CUB-200-2011 ([Wel+10]), der aus etwa 12.000 annotierten Fotos von 200 nordamerikanischen Vogelarten besteht, bemessen.

Unter anderen Gaston et al. hatten 2004 generell skizziert, wie eine automatische optische biologische Bestimmung mit Training und Inferenz aussehen könnte, und welche Schwierigkeiten damals noch vorlagen (vgl. [GO04], S. 658). Skalierbare Lösungen lagen damals mangels Rechenleistung und Datensätzen noch nicht vor. Der Bereich Machine-Learning bekam ab Mitte der 2010er Jahre größeren Auftrieb, nachdem der große Datensatz "Snapshot Serengeti" (vgl. [Swa+15]) veröffentlicht wurde, der aus hunderten Fotofallen etwa 1.2 Millionen Fotos zusammenstellte und Chen et al. 2014 erstmalig ein Modell für die Erkennung von Tieren auf Fotofallen mit CNNs erarbeitete (vgl. [Che+14]). In den folgenden Jahren, unter anderem aufbauend auf "Snapshot Serengeti", wurden die Möglichkeiten von Objekterkennung, Arterkennung und Tieridentifikation von verschiedenen Seiten in vielfältigen Habitaten beleuchtet (vgl. [Bee+21], [Che+19], [Car+20] [Bee21], [Lar21], [Mie+21], [Süß21], [Tui+22], [LB22], [Cla+23], [DMG17], [Rig+23]). Aus den Forschungen entstand auch das MegaDetector-Projekt (vgl. [BMY19]). Eine große Herausforderung stellt weiterhin die mangelhafte Fähigkeit von Machine-Learning-Modellen dar, Ergebnisse aus einem Forschungsprojekt auf andere Orte zu übertragen (vgl. [BvP18]).

Vergleichende Analysen der Forschungsprojekte sind jedoch schwierig, da unterschiedliche Zielausrichtungen, Vorgehensweisen, Datenvorbereitungen und Metriken vorliegen. Desprez et al. fasst die wesentlichen Aspekte für ein erfolgreiches machine-learning-getriebenes Artenschutzprojekt zusammen. Der Fokus muss auf gutem Daten-Management liegen. Datenselektion, Splitting und die Berücksichtigung von unbalancierten Daten sind zentral. Weiterhin ist die Wahl des Machine-Learning-Verfahrens und der Metriken sowie die Erklärbarkeit ("explainable AI") der Algorithmen wichtig.

Die Fortschritte im Bereich Image-Classification gehen auf verschiedene Aspekte zurück: Neben immer besserer Extraktion und Selektion von Features durch CNNs (vgl. [Sze+15], [Cho10]), ist auch die Fokussierung von wichtigen Bildausschnitten relevant (vgl. [Kra+15], [Rec+], [Tae+11], [Wan+], [Süß21]). Für die Fokussierung wird einerseits versucht, die

Aufmerksamkeit (Attention) des Netzwerkes auf Bereiche zu lenken, die eine hohe Auffälligkeit (Saliency) besitzen und diese Fokussierung differenzierbar in die neuronalen Netze zu integrieren (vgl. [Rec+], [BMK14], [Mni+14], [Li+17], [Yan+18]). Andererseits werden Ausschnitte aus Bildern herausgetrennt und losgelöst weiterverarbeitet (vgl [Par+], [BMY19]). Die Fokussierung ist wichtig, um Verwechslungen mit anderen Objekten im Bild zu vermeiden und die Schärfe der Features beizubehalten.[1] Dies ist insbesondere dann notwendig, wenn hochauflösende Fotos ohne Fokussierung im Rohbestand vorliegen. Der Datensatz Vogel-CUB-200-2011 ist jedoch beispielsweise schon sehr fokussiert und zeigt Vögel sehr zentriert und klar.

Viele Ansätze für Object-Detection und Image-Classification basieren auf Transfer-Learning und machen sich auf ImageNet (vgl. [Den+09]) vortrainierte Feature-Extractor zunutze (vgl. [Kra+15], [BvP18]). Die Fully-Connected-Layer - und zum Teil auch die Feature-Extractors - werden in einigen Arbeiten dann spezifisch für den jeweiligen Bildbestand trainiert. In einigen Fällen wird mit Bounding-Boxen-Daten von Experten gearbeitet, wohingegen mit Supervised- und Unsupervised-Learning versucht wird, auf diese zu verzichten, da ihre Erstellung aufwändig ist (vgl. [Kra+15], [Wan+]).

In einigen Arbeiten wurde die MegaDetector-Object-Detection bereits eingesetzt und gegen annotierte Daten verglichen. Beery et al. erreicht mit MegaDetector eine mAP von etwa 71-90% auf Fotofallenfotos. Bei Norman et al. konnte MegaDetectorV3 erfolgreich für Bildausschnitte aus Fotofallenfotos herangezogen werden, um eine CNN-basierte Arterkennung zu implementieren (vgl. [Nor+23]). Bei Fennell et al. konnte auf einem Fotofallen-Datensatz aus dem Westen von Kanada mit MegaDetector[2] mit Schwellenwert (Threshold) 0.90 eine Accuracy von 97,2% für Menschen (F-Score = 0.97) und 96,6% für Tiere (F-Score = 0.87) erreicht werden (vgl. [FBB22]).

In einer Arbeit von Beery et al. ([BvP18]) konnte die Fokussierung auf Bildausschnitte von Fotofallenfotos mithilfe einer Faster R-CNN-Object-Detection in der Datenvorbereitung die Accuracy eines Klassifikators um zwölf Prozentpunkte[3] heben (vgl. [BvP18], S. 480). Zudem zeigt Wang (vgl. [Wan23]) die Möglichkeit, Bildausschnitte nicht per Bounding-Boxes, sondern mit Segmentierungsverfahren zu ermitteln.

Seit neuestem sind auch ViT und Hybride aus Transformern und CNNs präsent, die in Benchmarks zu Object-Detection und Image-Classification gleichwertige Ergebnisse liefern, als CNNs (vgl. [Dos+22], [Dia+03], [Pap24], [PKS20]).

[1] Es werden neben maßstabsgetreuer Fokussierung in Form eines Ausschnitts zum Teil auch Bereiche mithilfe von Verzerrung fokussiert. (vgl. [Zhe+19])

[2] Vermutlich kam MegaDetectorV4 zum Einsatz

[3] Accuracy: 80% auf Rohbildern. 92% auf Bildausschnitten.

4 Methodik

Die Anforderung für citizen-science-gestützte Biodiversitätsforschung besteht darin, Wildtiere, Nutztiere, Menschen und Fahrzeuge auf Fotofallenfotos zu finden und zu klassifizieren. Andere Objekte sind in der Regel nicht von Interesse und sollen ignoriert werden. In einigen Arbeiten im Bereich FGVC wurde bereits dargelegt, dass das Fokussieren auf einen wichtigen Bildbereich die visuelle Klassifikation verbessern kann. Auch im Bereich Biomonitoring liegen bereits Belege für diese These vor. Es ist anzunehmen, dass die Fokussierung die Quote an relevanten Features gegenüber irrelevanten Features im Schritt der Feature-Extraction erhöht und dadurch die Klassifikation verbessert (vgl. [Par+]). Zudem besteht die Vermutung, dass die Verwendung von Bildausschnitten die Modelle übertragbarer bzw. generalisierbarer macht. Das Nutzen von object-detection-basierten Bildausschnitten kann als eine machine-learning-basierte Augmentierung aufgefasst werden.

Im ersten Versuch wird der Frage nachgegangen, ob das vorherige Ausschneiden von Bildausschnitten aus Rohbildern mithilfe von Object-Detection-Modellen bessere Klassifikationsergebnisse liefert, als direkt auf den Rohbildern zu arbeiten. Im zweiten Experiment soll die Performanz von Klassifikationsmodellen, die auf unterschiedlichen Bildausschnitten basieren, verglichen werden. Im "WildLIVE"-Forschungsprojekt lag ein annotierter Datensatz mit Fotos und Bounding-Boxes vor, der für den Vergleich herangezogen wird. Es stellt sich die Frage, ob das händische Setzen von Bounding-Boxes durch den Einsatz von Machine-Learning-Modellen ergänzt oder ersetzt werden kann. Sollten bestehende CNN-Architekturen in der Lage sein, auch auf herunterskalierten Rohbildern noch genug Features zu erkennen, wäre ein Einsatz von Object-Detection nicht nötig. Zuletzt findet in Anbetracht der Vorgehensweisen von Beery et al. 2017 (vgl. [BvP18]) eine Prüfung statt, ob die Klassifikationsergebnisse auf Fotomaterial übertragbar sind, die von Fotofallen stammen, die nicht im Training inbegriffen waren. Dies ist wichtig, da Fotos von Fotofallen in der Wildnis häufig viel Hintergrundrauschen besitzen und sich die Gegebenheiten an unterschiedlichen Stationen stark unterscheiden können.

Diese Arbeit teilt sich somit in drei aufeinander aufbauenden Versuche, die im Folgenden vorgestellt werden:

6.1 Vergleich von Klassifikatoren auf Basis von Rohbildern gegenüber Klassifikatoren auf Basis von Object-Detection-Bildausschnitten

6.2 Vergleich von Klassifikatoren auf Basis von Expertenausschnitten gegenüber Klassifikatoren auf Basis von Object-Detection-Bildausschnitten

6.3 Prüfung der Klassifikatoren auf Übertragbarkeit

Alle Versuche werden in Kapitel 7 anhand des mittleren F_1-Score bewertet. Im zweiten Versuch wird zudem die Objekterkennung mit IoU und mAP evaluiert.

5 Datengrundlage

Das Senckenberg Forschungsinstitut stellt für die Durchführung dieser Arbeit einen Datensatz aus dem “WildLIVE!”-Projekt (vgl. [WJ02], [Jan+24]) zur Verfügung, der im Folgenden exploriert wird.

5.1 Datenexploration

Aus dem Datenbestand des Forschungsprojekts Chiquitano aus Bolivien wird ein Abzug der von den Citizen-Scientists auf Fotos aus dem Zeitraum Oktober 2017 bis September 2023 gesetzten Bounding-Boxes verwendet (s. Abbildung 5.1). Die Daten liegen im CSV-Format vor. In den etwa 550 000 Zeilen liegen Foto-ID, Download-Link, Spezies-Kategorie, Trivialname der Spezies, Koordinaten der Bounding-Box (Left, Top, Width, Height), Reviewer-Kennung, Review-Fortschritt, Reviewer-Einschätzung zur Bildqualität (“blurry”, “clear”, “outstanding_picture”, ...), Kamera-Standort/Station, Nutzerkennung und Erstellzeitpunkt vor.

Jedes vorliegende Foto wird von ein oder mehreren Citizen-Scientists bestimmt. Dabei werden für jedes Foto eine oder mehrere Bounding-Boxen gesetzt und jede Bounding-Box mit einem Label bestehend aus Kategorie (Aves, Rodentia, Marsupialia, Others, ...) und Trivialname (Jaguar, Human, Vehicle, ...) versehen. Die Kategorien und Trivialnamen waren den Citizen-Scientists vorgegeben. Ein Teil der Labels und Boxen wurde von Fachpersonal einem Review unterzogen. Alle vorliegenden Download-Links waren gültig und mit ihnen konnte ein Foto bezogen werden. Datenlücken oder offensichtlich ungültige Werte lagen nicht vor.

Abbildung 5.1: Kollage von “WildLIVE!”-Fotos. Zu sehen ist eine Kuh, ein Fahrzeug, ein verwackelt abgebildetes Kleinsäugetier, *Dicotyles tajacu*, ein Mensch und *Panthera pardus* (von links oben nach rechts unten).

Die Auswertung der Daten ergibt, dass

- etwa 50 000 unterschiedliche Fotos mit Bounding-Boxen versehen wurden, wobei auf jedem Foto mindestens ein Objekt markiert wurde;
- etwa 200 000 der 550 000 Bounding-Boxen auf diesen Fotos einem Review einer Fachperson unterzogen wurden;
- davon etwa 37 600 unterschiedliche Fotos vorliegen, die Bounding-Boxen besitzen, die einem Review unterzogen wurden;
- davon etwa 22 700 unterschiedliche Fotos vorliegen, die (review-abgesichert) von allen Citizen-Scientists mit nur genau einer Bounding-Box markiert wurden;
  - einerseits davon etwa 21 500 Fotos von allen Citizen-Scientists (review-abgesichert) mit der selben Kategorie markiert wurden,
    und etwa 1100 Fotos von den Citizen-Scientists (review-abgesichert) mit unterschiedlichen Kategorien markiert wurden;
  - andererseits davon etwa 21 200 Fotos von allen Citizen-Scientists (review-abgesichert) mit dem selben Trivialnamen markiert wurden,
    und etwa 1500 Fotos von den Citizen-Scientists (review-abgesichert) mit unterschiedlichen Trivialnamen markiert wurden;
  - zuletzt davon 20 362 Fotos von allen Citizen-Scientists (review-abgesichert) mit dem selben Trivialnamen und der selben Kategorie markiert wurden,
    und etwa 2300 Fotos von den Citizen-Scientists (review-abgesichert) mit unterschiedlichem Trivialnamen oder unterschiedlicher Kategorie markiert wurden.[1][2]

Es werden nur Fotos ausgewählt, die einem Review unterzogen wurden, da auf Fotos ohne Review eher falsche Bounding-Boxes oder Labels zu erwarten sind. Diese hätten die Versuchsergebnisse verfälscht. Für die weitere Durchführung der Experimente (s. Abschnitte 6.1 und 6.2) war es nötig, nur Fotos zu extrahieren, auf welchen nur ein Objekt zu sehen ist. Dies war wichtig, um jedem Foto eine einzige Klasse in Form eines Taxons und einen einzigartigen Bildausschnitt zuordnen zu können.

Die Trivialnamen und Kategorien wurden an Bounding-Boxen auf den Fotos markiert, die per X, Y, Breite und Höhe spezifiziert sind. Auch mehrere auf dem gleichen Bild gesetzte Bounding-Boxen, unterscheiden sich marginal in ihren Koordinaten. Es handelt sich somit um einen Multi-Label- und Multi-Class-Datensatz, da für jedes Foto mehrere Klassen in Form von Bounding-Boxes gesetzt sind. Für die weitere Nutzung der Bounding-Boxen wird ein Mittelwert der Bounding-Boxen jedes Fotos errechnet (s. Abschnitt 5.2).

[1] Wenn bei Kategorien Einstimmigkeit vorlag, lag Uneinstimmigkeit über Trivialnamen in den Kategorien Artiodactyla (422 von 6726), Aves (134 von 2796), Carnivora (225 von 3436), Marsupialia (36 von 82), Rodentia (45 von 3615) und Xenarthra (304 von 1241) vor.

[2] Wenn bei Trivialnamen Einstimmigkeit vorlag, lag Uneinstimmigkeit nur über Kategorie für “Cattle” und “Human” vor (others bzw. cattle_or_human).

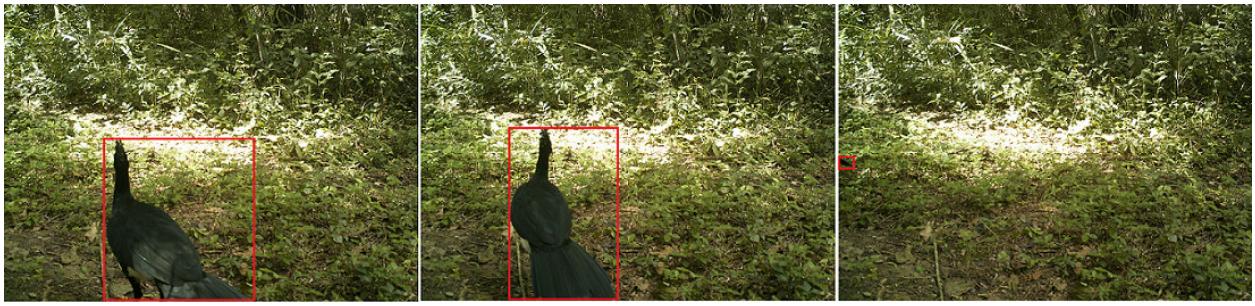

Abbildung 5.2: *Crax fasciolata* läuft durch das Sichtfeld der Kamera. Auf den ersten beiden Fotos ist das Tier noch gut bestimmbar. Auf dem letzten Bild, auf dem am linken Bildrand die Spitze von Schwanzfedern erkennbar ist, erschließt sich die Bestimmung nur auf Basis der vorhergehenden Fotos und des zeitlichen Kontexts.

5.2 Vorverarbeitung & Splitting

Für die Weiterverarbeitung werden die einmaligen 20 362 Fotos verwendet, welche review-abgesichert sind, auf welchen immer nur ein Objekt entdeckt wurde und von allen Beteiligten - in Bezug auf Kategorie und Trivialname - gleich benannt wurde. Aus den Daten werden die Merkmale Bild, Trivialname, Kategorie, X, Y, Breite, Höhe übernommen und alle anderen Werte verworfen. An diesem Punkt lässt sich von einem Multi-Class-Datensatz mit einem Label - bestehend aus Trivialname und Kategorie - sprechen.

Bei der Sichtung der Foto-Daten von "WildLIVE!" fällt auf, dass in einigen Fällen Objekte entdeckt und bestimmt wurden, die nicht ohne weitere Informationen visuell hätten bestimmt werden können. Es ist zu vermuten, dass weitere Informationen oder Kontext genutzt wurden. Bei der Auslösung der Fotofalle entstehen mehrere Fotos des selben Objekts. Wenn diese Fotos nacheinander zur Bestimmung bearbeitet werden, ergibt sich eine Fotoserie (s. Abbildung 5.2), in der auf manchen Fotos das Objekt gut bestimmt werden kann und auf manchen Fotos das Objekt zwar noch sichtbar aber nicht ohne dieses Vorwissen bestimmbar wäre und sich nur noch mit einer Bounding-Box lokalisieren lässt. Letztere Fotos stellen für Machine-Learning-Klassifikatoren eine enorme Herausforderung dar oder sind womöglich unlösbar.[3]

Im Datensatz finden sich elf unterschiedliche Werte von Kategorien (artiodactyla[4], aves[5], carnivora[6], cattle_or_human, marsupialia[7], others, perissodactyla[8], primates[9], reptilia[10], rodentia[11], xenarthra[12]) und 67 unterschiedliche Werte von Trivialnamen (amazonian_motmot,

[3] In einigen weiterführenden Arbeiten wurden die Informationen über Fotoserien genutzt, um einzelne Randbilder zu entfernen oder die Serie als Ganzes zu klassifizieren (vgl. [BvP18]).
[4] Paarhufer, Ordnung der Mammalia
[5] Vögel, Klasse der Wirbeltiere
[6] Raubtiere, Ordnung der Mammalia
[7] Beuteltiere, Unterklasse der Mammalia
[8] Unpaarhufer, Ordnung der Mammalia
[9] Affen, Ordnung der Säugetiere
[10] Reptilien, Klasse der Wirbeltiere
[11] Nagetiere, Ordnung der Säugetiere
[12] Nebengelenktiere, Ordnung der Säugetiere

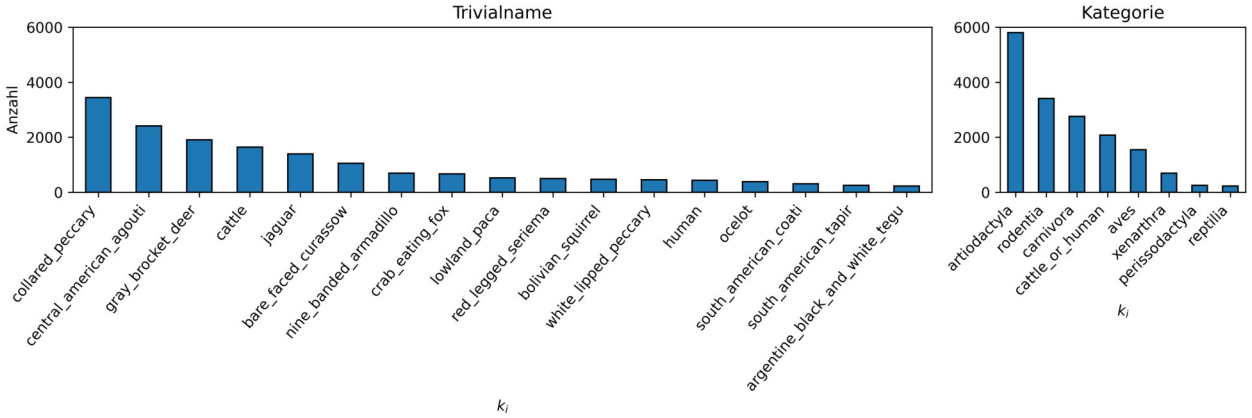

Abbildung 5.3: Die Histogramme der Klassen Trivialname und Kategorie im gefilterten Datensatz SWL-2023. Insbesondere bei Trivialname sind etwa 80% der Klassen seltener als 500-mal im Datensatz vorhanden. Im Histogramm der Kategorien sieht man, dass vier Klassen nur selten vorkommen.

amphibian_sp, argentine_black_and_white_tegu, armadillo_sp, bare_faced_curassow, bat_sp, bird_sp, black_vulture, boat_billed_heron, ...). Es fällt auf, dass ein eher kleiner Bruchteil der Trivialnamen wiederum einen kategorieartigen Namen (z.B. “bird_sp” für “bird species”) darstellt. Diese zehn Taxa (biologische Familien, Ordnungen und Unterordnungen) machen etwa 10% (1200 Datenpunkte) der Daten aus und sind zumeist auf Fotos gesetzt worden, auf denen ein Tier nicht genau bestimmt werden konnte. Weitere schwer bestimmbare etwa 200 Fotos mit Trivialnamen “something_unidentifiable” oder “some_rat_like_rodent” befinden sich im Datensatz. Es fällt zudem auf, dass die Daten durch die Verwendung von “cattle”, “human”, “vehicle” und “equipment” in Wildtiere und zivilisatorische Objekte unterscheidet. Dies ist sowohl für eine biologische Einordnung (bei Vieh und Mensch nicht von Interesse) sowie für Datenschutz/Privatsphäre-Bedenken relevant (vgl. [FBB22], Kap. 3.5). Abbildung 5.3 zeigt die Häufigkeiten der Trivialnamen und Kategorien im Datensatz.[13][14]

Für die Weiterverarbeitung unter Nutzung des Zielmerkmals Trivialname wird der bisherige Datensatz von den Fotos mit den Trivialnamen “*_sp”, “something_unidentifiable” und “some_rat_like_rodent” bereinigt, da aus diesen Fotos nur wenige nützliche Informationen zu ziehen sind und sie die Evaluationsergebnisse verfälschen könnten (vgl. [DMG17], S. 10). Zudem werden nur Arten betrachtet, die im Datensatz mindestens 25-mal vorkommen.[15]

[13] Häufige Kategorien sind: Artiodactyla (30%), Rodentia (17%), Carnivora (15%), Aves (12,5%), Cattle/Human (9%). Seltenste Klasse ist: Marsupialia (0.2%).

[14] Häufige Arten/Trivialnamen sind: collared_peccary (*Dicotyles tajacu*, 17%), central_american_agouti (*Dasyprocta punctata*, 12%), cattle (*Bos taurus*, 8%), gray_brocket_deer (*Mazama gouazoubira*, 9%), bare_faced_curassow (*Crax fasciolata*, 5%), jaguar (*Panthera pardus*, 7%) und nine_banded_armadillo (*Dasypus novemcictus*, 3%). Seltenste Klassen sind: red_footed_tortoise (*Chelonoidis carbonarius*, 2 Fotos, 0.01%), tataupa_tinamou (*Crypturellus tataupa*, 3 Fotos, 0.015%), small_billed_tinamou (*Crypturellus parvirostris*, 5 Fotos, 0.025%), black_vulture (*Coragyps atratus*, 6 Fotos, 0.029%).

[15] Folgende zehn Klassen kommen seltener als 25-mal vor: black_vulture (*Coragyps atratus*), dog (*Canis familiaris*), lesser_yellow_headed_vulture (*Cathartes burrovianus*), limpkin (*Aramus guarauna*), neotropical_otter (*Lontra longicaudis*), plush_crested_jay (*Cyanocorax chrysops*), red_footed_tortoise (*Chelonoidis carbonarius*), red_necked_woodpecker (*Campephilus rubricollis*), small_billed_tinamou (*Crypturellus parvirostris*), tataupa_tinamou (*Cerdocyon thous*).

Der Datensatz - im weiteren Verlauf SWL-2023 und D genannt - reduziert sich damit auf 18 850 Fotos mit Bounding-Boxen in 11 Kategorien und 43 Trivialnamen.

Zielmerkmal	Shannon-Gleichheits-Index $E_{H,x}$	Imbalance-Factor IF_x
Trivialname	0.775	137.72
Kategorie	0.776	527.55

Tabelle 5.1: Balance-Maße des gefilterten Datensatzes SWL-2023 bzw. D in Bezug auf die Zielmerkmale

SWL-2023 ist in Bezug auf beide Zielmerkmale gemäß des Shannon-Gleichheits-Maßes und des Imbalance-Factors (s. Tabelle 5.1 und Abschnitt 2.6.3) eher unbalanciert.[16] Die Daten werden im letzten Schritt randomisiert in einen Trainings- (70%), Validierungs- (10%) und Testdatensatz (20%) gesplittet. Die Verteilung der Klassen im Ursprungsdatensatz wird in den Teildatensätzen beibehalten ("Stratified Sample").

In vielen vergleichbaren Arbeiten werden Splits anhand von fachlichen Kriterien durchgeführt, um Overfitting auf ungewollte Bildmerkmale zu vermeiden. Dabei spielt "Data Leakage" eine Rolle, das von Desprez et al. (vgl. [DMG17]) detailliert für die Domäne der Fotofallendaten beschrieben wurde. Insbesondere fast identische Fotos, die durch Serienaufnahme entstehen, sind ein Problem. Um diesem Problem zu begegnen, ist ein vorgelagertes Aufteilen der Datensätze nach Station, Fotoserie oder Zeit nötig. Dies wird im dritten Versuch dieser Arbeit betrachtet (s. Abschnitt 6.3). In Abbildung 5.4 findet sich ein Histogramm zu der Verteilung der verschiedenen Arten nach Station. Auf der Basis der Art/Station-Verteilung könnte ein fachlicher Split durchgeführt werden.

[16] Der Imbalance-Factor von ImageNet-LT, Places-LT und iNaturalist beträgt gemessen von Zhang et al. 256, 996 and 500. CIFAR100-LT besitzt drei Varianten mit unterschiedlichen Faktoren von 10, 50, 100 (vgl. [Zha+09] S. 2).

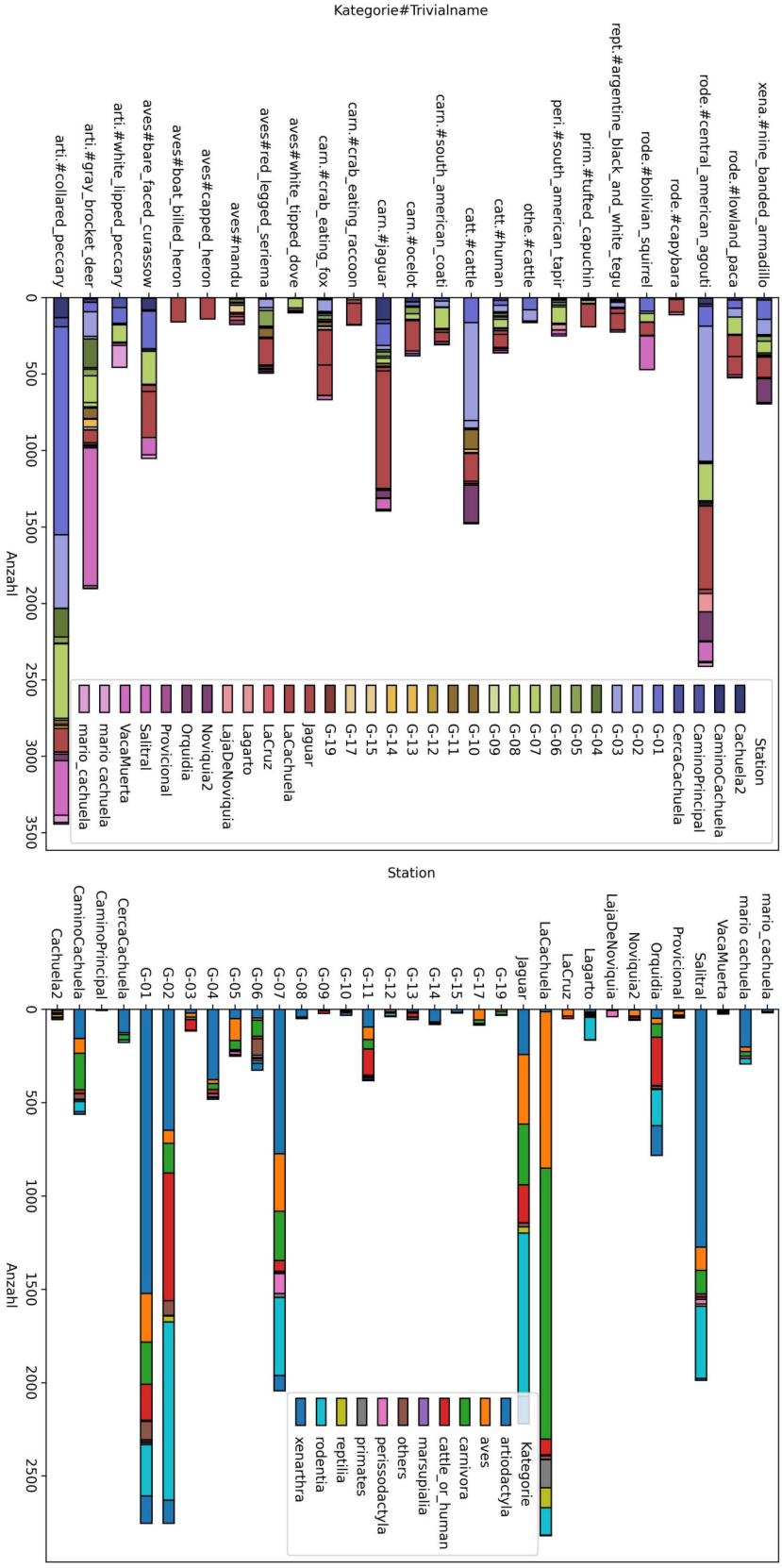

Abbildung 5.4: Histogramme der Vorkommnisse von Arten anhand ihrer Trivialnamen an Fotofallenstationen und die Verteilungen der Kategorien. Nur Arten, die mindestens 100-mal vorkommen.

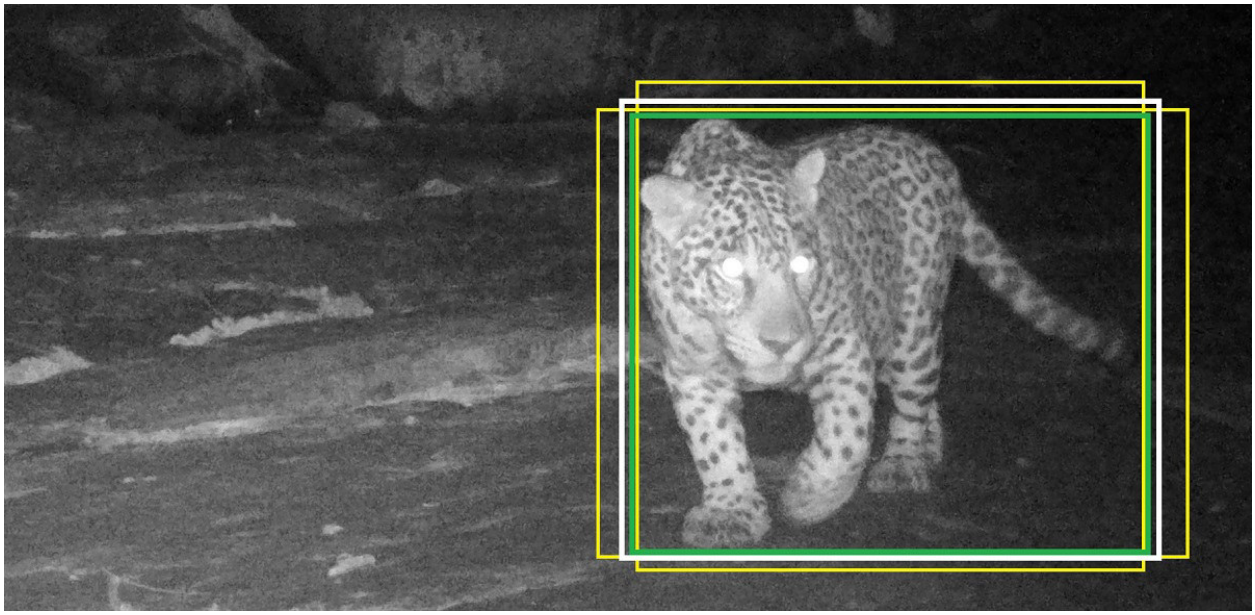

Abbildung 5.5: Auf einem Foto von *Panthera Pardus* sind vier Bounding-Boxen eingezeichnet. Zwei manuell gesetzte Bounding-Boxen (gelb) und die berechnete Average-Bounding-Box (weiß) mit je viel Platz um das Objekt und die beste manuell gesetzte mittlere Bounding-Box (grün) mit der engsten Kontur. Die grüne Box hat im Vergleich zur weißen Box die größte IoU. Die beiden gelben Boxen haben im Vergleich zur weißen Box kleinere IoU und überlappen schlechter.

Bounding-Box-Daten

Für die Klassifikation von Rohbildern nach Kategorie und Trivialname sind nur die Werte Foto als Input und Kategorie bzw. Trivialname als Zielmerkmal nötig. Für die Arbeit mit Bildausschnitten werden auch die Bounding-Box-Koordinaten X, Y, Breite und Höhe benötigt. Jede nach dem obigen Vorgehen gefilterte Beobachtung im Datensatz besitzt mehrere - jedoch immer mindestens eine - Bounding-Box mit demselben Label.

Bounding-Boxen wurden mithilfe einer Labeling-Software gesetzt. Alle Unterstützer wurden angehalten, die Bounding-Boxen mit der Labeling-Software möglichst eng um das zu bestimmende Objekt zu setzen. Es ist aber bei Fotos mit Auflösungen im Bereich 2000-3000px anzunehmen, dass die Bounding-Boxen des selben Objekts auf dem selben Foto nicht exakt die selben Maße besitzen. Um aus Bounding-Boxen gleicher Konfidenz die mittlere Box zu ermitteln, wird im Rahmen dieser Arbeit folgendes Verfahren eingesetzt: Die mittlere Bounding-Box, $b_{Mitte} = (X_1, Y_1, X_2, Y_2)$ aus einer Menge von Bounding-Boxen $B = \{b_1, b_2, \dots b_n\}$ mit $b_i = (X_{1,i}, Y_{1,i}, X_{2,i}, Y_{2,i})$ ist die, welche zur Average-Box $\overline{b} = (\overline{X_1}, \overline{Y_1}, \overline{X_2}, \overline{Y_2})$ die höchste IoU (s. Abschnitt 2.5.3) besitzt. b_{Mitte} aus B auf einem Bild ist die Bounding-Box, so dass

$$IoU(b_{Mitte}, \overline{b}) \geq IoU(b_i, \overline{b}) \, für \, alle \, b_i \in B. \tag{5.2.1}$$

Abbildung 5.5 zeigt das Beispiel mit drei manuell gesetzten Bounding-Boxen. Für die Verarbeitung wird für jedes Bild die mittlere Bounding-Box b_{Mitte} verwendet und andere verworfen.

In Abbildung 5.6 sind die Größen der Bounding-Boxen in D abgebildet.

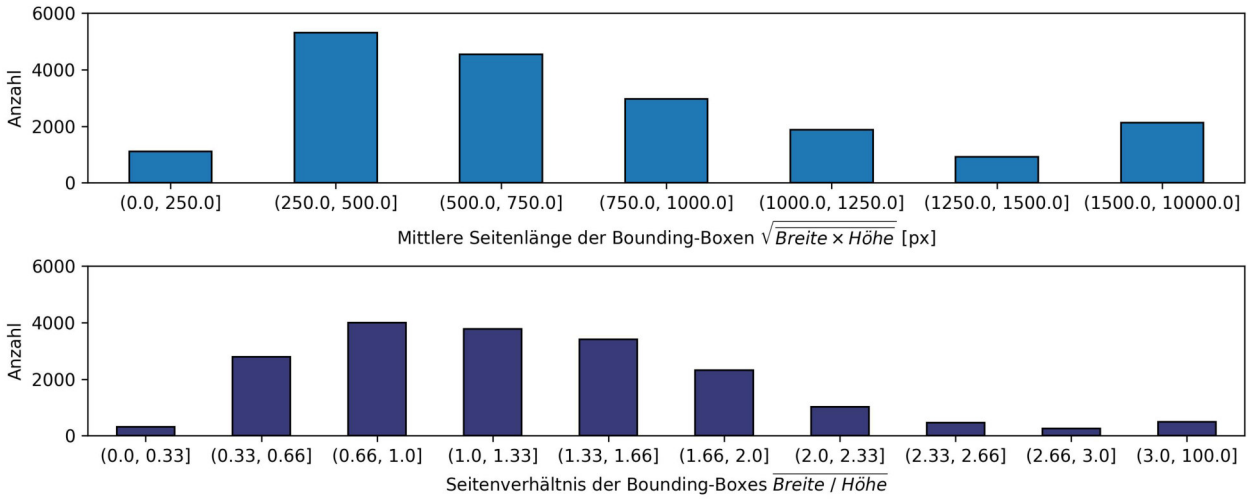

Abbildung 5.6: Größenverteilung als Histogramm der Bounding-Boxen in D nach mittlerer Seitenlänge $\sqrt{\overline{Breite \times Höhe}}$ und mittlerem Seitenverhältnis $\overline{Breite\ /\ Höhe}$. Median der Seitenlängen: 654px, Median der Seitenverhältnisse: 1.20.

5.3 Datenqualitätsdimensionen

Anhand der in Abschnitt 2.6 vorgestellten Datenqualitätsdimensionen nach Pipino et al. lässt sich SWL-2023 wie folgt zusammenfassen:

- **Angemessene Datenmenge**
  Es liegen etwa 18 850 Datenpunkte vor. Einige Klassen sind sehr selten und müssen aus dem Datensatz ausgeschlossen werden. Die restlichen Klassen sind mit mindestens 25 Datenpunkten zum Teil weiterhin selten. Die Daten sind zudem an verschiedenen Stationen mit verschiedenen Kameras über einen längeren Zeitraum gesammelt worden. Es lassen sich in den Daten wenig saisonale Verschiebungen vermuten. Ein Split nach Zeiträumen oder Fotoserien ist nicht ohne weiteres möglich. Datenmengen der einzelnen Stationen sind sehr ungleichmäßig verteilt (s. Abbildung 5.4). Alle angegebenen Fotos des “WildLIVE!”-Projekts waren verfügbar. Für weitere Versuche wären leere Bilder nötig gewesen, um die True Negative der Object-Detection zu evaluieren.

- **Fehlerfreiheit**
  Mithilfe von Reviews und der Anwendung eines einstimmigen Konsens werden nur feh-lergeprüfte Daten verwendet. Durch Filterung sind theoretisch nur Fehler der Kategorie F.1 möglich (s. Abschnitt 2.6.2). Es kann somit vorkommen, dass die Daten Bestimmungen enthalten, die einer falschen Art zugeordnet sind.

- **Objektivität**
  Der Datensatz ist in Bezug auf Klassen unbalanciert, was bei Training und Evaluation berücksichtigt werden muss. Zudem werden alle Verfahren randomisiert und keine weiteren händischen Selektionen durchgeführt.

- **Relevanz**
  Daten liegen in Rohfassung mit ausreichend großer Auflösung vor. Tiere sind prinzipiell gut bestimmbar. Eine Eingrenzung auf klar erkennbare Objekte wäre, wenn nötig, mithilfe des Merkmals “Bildqualität” möglich.

6 Versuchsaufbau

In diesem Abschnitt werden die Details zur Durchführung der drei in Abschnitt 4 vorgestellten Versuche erläutert. Zuerst wird der Versuchsaufbau der Vergleiche zwischen Rohbildklassifikatoren und Ausschnittklassifikatoren (6.1) und danach der Versuchsaufbau der Vergleiche zwischen verschiedenen Ausschnittklassifikatoren (6.2) beschrieben. Dann wird der dritte Versuch zur Prüfung der Übertragbarkeit der Modelle beschrieben (6.3). In Abschnitt 6.4 findet sich eine Erläuterung der Implementierungsdetails und einiger Hindernisse im Projektverlauf.

6.1 Klassifikatoren mit Rohbildern & Bildausschnitten

In diesem ersten Versuch soll die These, dass Ausschnittklassifikatoren besser als Rohbildklassifikatoren abschneiden, anhand des “WildLIVE!” -Datensatzes SWL-2023 aus Abschnitt 5 validiert werden. Dazu wird der in Abbildung 6.1 skizzierte Versuchsaufbau implementiert (von oben links nach unten rechts zu lesen): Aus Daten des “WildLIVE!” -Projekts werden annotierte Beobachtungen von Tieren entnommen und gefiltert (s. Abschnitt 6.1.1). Unter Einsatz der Download-Links werden die Bilder heruntergeladen und abgelegt. Mithilfe von MegaDetectorv5a werden aus den Rohbildern Ausschnitte extrahiert und abgelegt. Das Label von “WildLIVE!” wird für die Ausschnitte übernommen und das Label von MegaDetector (“Tier”, “Mensch”, “Fahrzeug”) ohne weitere Prüfung verworfen (s. Abschnitt 6.1.2). Der Datensatz mit Bildausschnitten ist damit, je nach Erfolg der Object-Detection, kleiner als der Datensatz mit den Rohbildern. Mithilfe des Trainings- und Validierungsdatensatzes werden mit Tensorflow/Keras zwei Klassifikationsmodelle trainiert (s. Abschnitt 6.1.3). Diese beiden Bildklassifikatoren können mithilfe des jeweiligen Testdatensatzes anhand von Modellmetriken evaluiert und verglichen werden (s. Abschnitt 6.1.4).

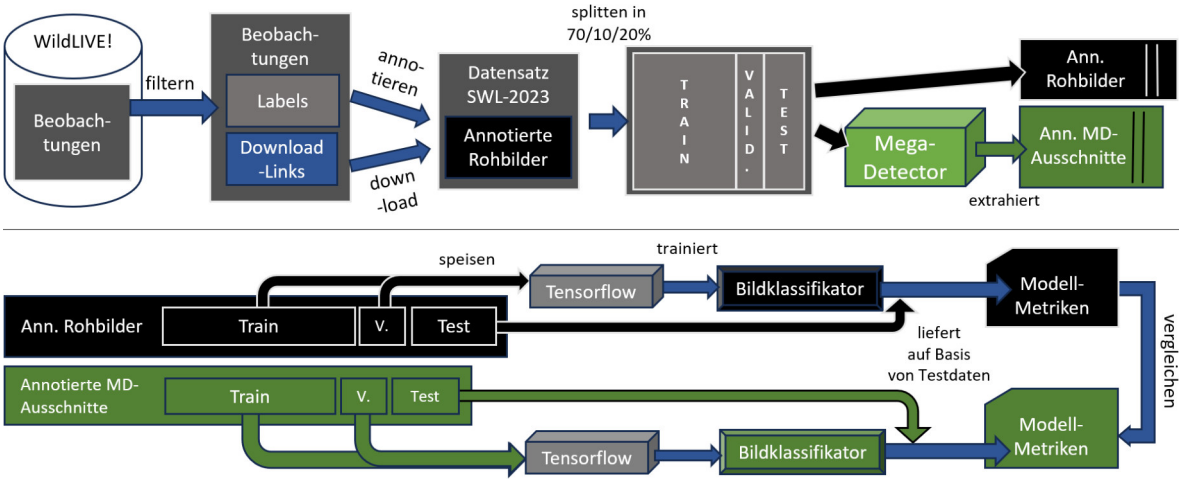

Abbildung 6.1: Versuchsaufbau “Klassifikatoren mit Rohbildern & Bildausschnitten”

Das Experiment wird sowohl gegen das Zielmerkmal “Trivialname” als auch “Kategorie” durchgeführt.

6.1.1 Datenvorbereitung

Die Daten werden gemäß Abschnitt 5.2 aus dem Rohbestand selektiert. Es liegen somit in SWL-2023 für beide Zielmerkmale jeweils etwa 18 850 annotierte Fotofallenfotos vor, auf denen laut Experten exakt ein Objekt zu sehen ist und die Bestimmung laut Experten eindeutig ist. Aus dem Datensatz werden die Pfade, von denen die Bilder heruntergeladen werden können, und die Labels “Trivialname” und “Kategorie” für jedes Foto entnommen. Dieser Datensatz D wird in die Teile für Training D^T, Validierung D^V und Test/Evaluation D^E geteilt.

6.1.2 Object-Detection

Neben dem Datensatz D_{Roh}, der aus Rohbildern besteht, wird ein neuer Datensatz D_{MD} mithilfe von Object-Detection auf den Rohbildern erzeugt. Dieser Datensatz enthält Ausschnitte der Rohbilder, auf welchem sich Tiere, Menschen, Fahrzeuge und Gegenstände befinden. Für die Object-Detection wird MegaDetectorv5a verwendet.

MegaDetector wird in diesem Fall ohne Batch-Verarbeitung[1] pro Bild aufgerufen und liefert dafür eine Liste von Detektionen zurück, die aus der Bounding-Box (X, Y, Breite, Höhe)[2], der Klassifikation (1 = Tier, 2 = Mensch oder Vieh, 3 = Fahrzeug) und einer Konfidenz $[0-1]$ besteht. Alle Detektionen oberhalb eines Konfidenzwerts von $T = 0.20$ (vgl. [Wil24]) werden als erfolgreich anerkannt und übernommen. Die Labels der Ausschnitte werden aus dem Label des Rohbilds entnommen. Dies ist von der Multiplizität her möglich, da in der Vorverarbeitung nur Beobachtungen bzw. Fotos verwendet wurden, die auf Basis der Reviewer-Meinung nur ein Tier enthalten. Es kann vorkommen, dass die Object-Detection mehrere Objekte findet. Es wird jedoch immer nur das Objekt mit der höchsten Konfidenz verwendet. MegaDetector ist durch das YOLO-Verfahren darauf trainiert, Bounding-Boxen möglichst eng um das gefundene Objekt zu setzen.

Da nicht auf allen genutzten Fotos von der Object-Detection ein Tier gefunden wird, reduziert sich der Datensatz D_{MD}. Mit etwa 70% dieser Fotos wird dann das Modell trainiert und mit etwa 10% dieser Fotos validiert. Etwa 20% liegen noch als Testdaten bereit. Die Anteile von 70%, 10% und 20% könnten nach der Object-Detection prinzipiell leicht abweichen, da der Split vorher geschieht. Somit wäre es möglich, dass in einem Teildatensatz weniger oder mehr Objekte gefunden werden. Dementsprechend sind auch Abweichungen der Klassenverteilungen in jedem Teildatensatz möglich.

[1] Eine Batch-Verarbeitung ist prinzipiell zur Beschleunigung des Verfahrens möglich, wurde aber in dieser Arbeit nicht betrachtet. Durch eine Batch-Verarbeitung könnte die Verarbeitungsgeschwindigkeit erhöht werden.

[2] Wertebereich für alle Koordinaten ist $[0-1]$, also relativ zur Bildauflösung.

Sollte das Object-Detection-Modell einen falschen Ausschnitt liefern, würde dieser in der nachfolgenden Trainings- und Evaluationsroutine als Ground-Truth herangezogen werden.[3] Im zweiten Experiment kann eine quantitative Abschätzung dieses Ausschnittfehlers vorgenommen werden.

Die Reduzierung des Datensatzes nach der Object-Detection wirft das Problem auf, dass bei Training, Validierung und Evaluation von anderen Datenbeständen ausgegangen wird. Es wäre auch möglich, den Split erst nach der Object-Detection durchzuführen. Dies kann aber zur Vermischung von Trainings- und Test-Daten führen, was auf alle Fälle vermieden werden muss.

6.1.3 Tuning & Training

Das Training des Klassifikators wird mithilfe von Tensorflow und Keras durchgeführt. Um die Trainingszeiten zu reduzieren und die Vorteile eines universellen Feature-Extraktors zu nutzen, wird Transfer-Learning auf einem mit ImageNet vortrainierten Netz mit einer verlässlichen Basisarchitektur/Backbone, wie InceptionV3, Xception, ConvNeXt oder EfficientnetV2, angewendet. Auf dieser Basis setzt ein letzter Fully-Connected-Kopf-Layer auf (s. Abschnitt 2.3), der die extrahierten Features in die vom Datensatz vorgegebenen Zielklassen klassifiziert. Beim Tuning besteht die Möglichkeit, den Feature-Extractor/das Backbone zu trainieren ("Training: Ganzes Netz"), oder nur die Fully-Connected-Layer zu trainieren.

Als Loss-Funktion kommt die durch Label-Smoothing erweiterte kategorielle Kreuzentropie zum Einsatz.[4][5] Als Optimierer dient RMSProp. Die initiale Lernrate und ein Decay der Lernrate ist konfigurierbar.[6]

Einige Basisarchitekturen haben L_1- oder L_2-Regularisierung in ihren Layern vorgesehen. Zusätzlich ist im Training für Regularisierung Early-Stopping auf Basis des mittleren F_1-Scores[7] implementiert.[8] Auch das Label-Smoothing des Loss trägt zur Regularisierung bei. Die initiale Lernrate kann zudem konfiguriert werden.

In der Trainingspipeline liegen einige Hyperparameter vor, für die aus der Literatur kein optimaler Wert ermittelt werden kann. Die Hyperparameter sind in Tabelle 6.1 aufgeführt und in die drei Bereiche Netz (Struktur des Neuronalen Netzes), Augmentierung (Vorverarbeitung der Trainingsbilder) und Training (Ablaufparameter des Trainings) eingeteilt. Alle weiteren

[3] Beispiel: Auf einem Rohbild ist ein Jaguar neben einem Baum zu sehen. Der Rohbildklassifikator trainiert auf diesem Gesamtbild das Label "Jaguar". MegaDetector erkennt fälschlicherweise statt des Jaguars den Baum als Objekt von Interesse und der Bildausschnitt mit einem Baum wird aus dem Rohbild extrahiert. Dieser Ausschnitt wird dann im Training oder der Evaluation mit dem Label "Jaguar" geführt und schleust auf diese Weise Fehler in Training - "Features eines Baums beschreiben einen Jaguar" - oder Evaluation - "einen Baum als Jaguar zu klassifizieren ist korrekt" - ein.

[4] Der Hinge-Loss wird auch verprobt, liefert jedoch deutlich schlechtere Ergebnisse.

[5] Mit einer Anzahl von 43 Klassen führt dies zu den Zielwerten $\frac{0.1}{43}$ und $0.9 + \frac{0.1}{43}$, statt 0 und 1.

[6] Ein Learning-Rate-Decay alle 3 Epochen um 5% lieferte keine besseren oder stabileren Ergebnisse als ohne Learning-Rate-Decay.

[7] Heißt bei Tensorflow "Macro F_1-Score"

[8] Das Early-Stopping könnte, insbesondere unter Berücksichtigung der Zielmetrik des Tunings, auch auf dem F1-Score durchgeführt werden. Die Experimente haben diesbezüglich nur minimale Abweichungen der Klassifikationsgüte gezeigt.

Parameter sind entweder abhängig von der Laufzeitumgebung (Epochen, Batch-Size), oder der Wertebereich besitzt nur einen Wert (Vortrainigsdatensatz ImageNet).[9]

Hyperparameter	Wertebereich	Bereich
Netzarchitektur	{Xception, InceptionV3, ...}	Netz
Dropout-Rate	[0% - 20%]	Netz
Bildauflösung	[71^2px^2 - 500^2px^2]	Netz
Horizontaler Flip	{Ja, Nein}	Augmentierung
Farbaugmentierung	[0% - 100%]	Augmentierung
Zentralausschnitt	[75% - 100%]	Augmentierung
Label-Smooth-Rate	[0% - 20%]	Training
RMSProp: ρ	[0.9 - 0.999]	Training
RMSProp: Momentum β	[0.9 - 0.999]	Training
RMSProp: ϵ	[0.1 - 1.0]	Training
Initiale Lernrate η_0	[0.0001 - 0.1]	Training
Training: Ganzes Netz	{Ja, Nein}	Training

Tabelle 6.1: Hyperparameter für das Tuning

Das Tuning wird mithilfe von KerasTuner (vgl. [OMa+19]) für beide Klassifikatoren (Rohbildklassifikator und Bildausschnittklassifikator) in je zwei Schritten angegangen und gegen die Metrik F_1-Score des Validierungsdatensatzes optimiert. Der Testdatensatz wird nicht in das Tuning einbezogen. Es ist zu vermuten, dass aufgrund der Unterschiede zwischen Rohbildern und Ausschnitten ein separates Tuning von zwei Modellen der beiden Bildtypen nötig ist. Zwischen Ausschnitten, die von Menschen angefertigt wurden, und Ausschnitten, die aus einer Object-Detection hervorgehen, wird hier nicht unterschieden.

Zuerst wird mit allen Tuning-Parameter-Werten mithilfe des Hyperband-Verfahrens (vgl. [Lis+18]) eine Vorselektion der Hyperparameter-Werte vorgenommen. Mit Hyperband geht aber einher, dass nicht alle Permutationen in der vollen Ausprägung bis zur letzten Epoche durchlaufen werden. Dies führt dazu, dass Hyperparameter-Konstellationen, die langsamer konvergieren, benachteiligt werden und womöglich schlechterdings aussortiert werden. Im zweiten Schritt wird die engere Auswahl der Parameter mit einer Grid-Search, also einer permutativen Kombination aller Hyperparameter miteinander, abgeschlossen. Alle Hyperparameter-Permutationen auf den beiden Datensätzen - Rohbilder und Bildausschnitte - aber durch alle Epochen (inkl. Early-Stopping gegen F_1-Score des Validierungsdurchlaufs) durchzuführen ist rechenaufwändig.

In jeder Trainingsepoche wird der gesamte Trainingsdatensatz D^T_{Roh} oder D^T_{MD} - 70% des Gesamtdatensatzes - in Batches geteilt und dem Modell zum Training vorgelegt.[10] Nach jeder Epoche werden mithilfe des Validierungsdatensatzes D^V_{Roh} oder D^V_{MD} - 10% des Gesamtdatensatzes - die Metriken Loss, Accuracy, F_1-Score und Weitere berechnet. Die restlichen 20%

[9] Verwendete Netzarchitekturen/Backbones: Xception, InceptionV3, DenseNet201, EfficientNetV2B3, InceptionResNetV2, ResNet152V2, MobileNetV2, RegNetX320, RegNetY320, VGG19

[10] Für die Beschleunigung des Tunings kann auch ein Bruchteil der Daten verwendet werden. Dann lässt sich aber von schlechteren Ergebnissen ausgehen.

des Datensatzes und werden ausschließlich später für die Evaluation/den Test verwendet (s. Abschnitt 6.1.4).[11] Aus dem Training resultieren die vier Klassifikatoren für Kategorie und Trivialname sowie Rohbilder und Bildausschnitte K_{Roh}^{Kat}, K_{MD}^{Kat}, K_{Roh}^{Triv}, K_{MD}^{Triv}.

Die Augmentierung ist randomisiert und wird bei jeder Verwendung eines Bilds neu ausgeführt. Bei Nutzung des horizontalen Flips werden manche Bilder an der mittleren vertikalen Achse gespiegelt. Bei der Farbaugmentierung werden Farbton (Hue), Sättigung, Helligkeit und Kontrast verzerrt.[12] Bei Nutzung des "Zentralausschnitts" würde ohne Berücksichtigung des Inhalts nur ein fixer zentraler Ausschnitt - angegeben in Prozent der Breite und Höhe - des Bilds für das Training verwendet. Dies kann, ganz im Sinne der zentralen These dieser Thesis, in einigen Fällen noch einmal die Fokussierung auf das Objekt von Interesse erhöhen. Bei Bildmaterial, das schon optimal auf das Objekt zugeschnitten ist, oder auf dem häufig wertvolle Features an den Bildrändern vorliegen, ist das Vorgehen schädlich. Bei Rohbildern aus Fotofallen kann auch damit gerechnet werden, dass sich Objekte überall auf dem Bild befinden und ein central-crop eher schädlich ist.

Viele Datensätze im Bereich Biodiversität sind unbalanciert und haben eine "Long-Tail"-Struktur - so auch bei den Datensätzen dieser Arbeit (s. Abschnitt 5.2). Es stellt sich die Frage, welche Auswirkungen die Unbalance in den Trainings-, Validierungs- und Test-Datensätzen hat. Um eine möglichst gute Performanz des Modells auch auf Basis von Daten zu erlangen, die unbalanciert sind, wird als Tuning-Metrik der gemittelte F_1-Score jeder Klassenausprägung verwendet. Weitere Maßnahmen zur Optimierung des Modells im Angesicht unbalancierter Daten, wie Oversampling oder Undersampling, werden nicht vorgenommen.

6.1.4 Evaluation

Mithilfe der Testdatensätze D_{Roh}^{E} oder D_{MD}^{E}, die 20% des jeweiligen Gesamtdatensatzes ausmachen, werden Evaluationen der trainierten Modelle M vorgenommen. Als Metriken wird insbesondere der F_1-Score, aber auch Precision, Recall und Top-1-, Top-3-, Top-10-Accuracy betrachtet. Die Evaluation fand immer gegen gleichgeartete Fotos statt. M_{Roh} werden gegen D_{Roh} evaluiert und M_{MD} werden gegen D_{MD} evaluiert. In jedem Fall werden Modelle nur gegen Testdaten evaluiert, die dem Modell vorher nicht beim Training zur Verfügung standen, da der Split zu Beginn für alle Trainings und Evaluationen ausgeführt wurde. Für Fehlersuche und bessere Verständlichkeit der Evaluationsergebnisse stehen Routinen bereit, um fehlklassifizierte Fotos in tatsächlicher Auflösung, also "aus Sicht" des Modells, zu inspizieren.

Es werden anhand einer 11×11-Konfusionsmatrix für Kategorien sowie einer 43×43-Konfusionsmatrix für Trivialnamen auch klassenweise Metriken errechnet, die anzeigen, welche Klassen gut oder schlecht klassifiziert werden. Zudem lässt sich ablesen, welche Klassen fälschlicherweise als welche andere Klasse vorhergesagt werden. Es besteht die Annahme, dass optisch gut separierbare Klassen und häufige Klassen besser klassifiziert werden als seltene ähnliche Klassen und optisch ähnliche Klassen häufiger verwechselt werden.

[11] Eine Kreuzvalidierung - bspw. K-Fold-Cross-Validation - ist nicht vorgesehen, wäre aber integrierbar, um die Selektion des besten Modells auf Basis der Trainings- und Validierungsdaten zu verbessern.

[12] Anpassungsumfang der Farbaugmentierung für Farbton: $\pm 8\%$, Sättigung: $-40\% / +60\%$, Helligkeit: $\pm 5\%$, Kontrast: $\pm 30\%$

Aus der Evaluation der Modelle folgt, welcher Klassifikator für den Einsatz von Bildklassifikation am besten geeignet ist. Eine Evaluation der von MegaDetector gefundenen Ausschnitte selbst kann in diesem Versuch mangels Referenzdaten nicht vorgenommen werden. Die Güte des eingesetzten Object-Detection-Modells überträgt sich in die Qualität der extrahierten Ausschnitte. Fehler, die durch den Einsatz des Object-Detection-Modells in einen Teildatensatzes übergehen, werden somit nur indirekt über die Performanz des Klassifikators quantifiziert.

6.2 Klassifikatoren mit manuellen und automatischen Bildausschnitten

Im zweiten Experiment wird der in Abbildung 6.2 skizzierte Versuchsaufbau implementiert. Aus dem "WildLIVE!"-Datensatz SWL-2023 werden Beobachtungen entnommen und gefiltert. Zusätzlich zu den Labels und den Download-Links der Fotos werden auch die Bounding-Box-Koordinaten verwendet. Er wird, wie im ersten Experiment, in Trainings-, Validierungs- und Testdaten (D^T, D^V, D^E) geteilt.

Mithilfe der Bounding-Boxes und der annotierten Rohbilder werden Bildausschnitte ("Expertenausschnitte") ausgeschnitten und abgelegt. Die annotierten Rohbilder werden ohne Weiterverarbeitung abgelegt. Aus den Rohbildern werden mithilfe von MegaDetector und anderen Object-Detection-Modellen Ausschnitte gesucht und ausgeschnitten. Das Label des Rohbilds wird ohne weitere Prüfung für den Ausschnitt übernommen. Optimalerweise findet die Object-Detection exakt den Ausschnitt, den auch die Experten gefunden haben. Die Datensätze D_{Roh}, D_{Exp}, D_{MD}, ... bestehen jeweils aus einem Trainings-, Validierungs- und Testdatensatz D^T_{Roh}, D^V_{Roh}, D^E_{Roh}, D^T_{Exp}, ... und werden separat abgelegt. D_{Exp} und D_{Roh} sind gleich groß und D_{OD}, je nach Erfolg der Object-Detection, kleiner. Alle Trainings- und Validierungsdatensätze werden separat in Tensorflow/Keras gespeist und jeweils ein Bildklassifikator trainiert. Mithilfe der Validierungs- und Testdaten wird jedes Modell evaluiert und diese Modellmetriken verglichen. In diesem Versuch können die von der Object-Detection

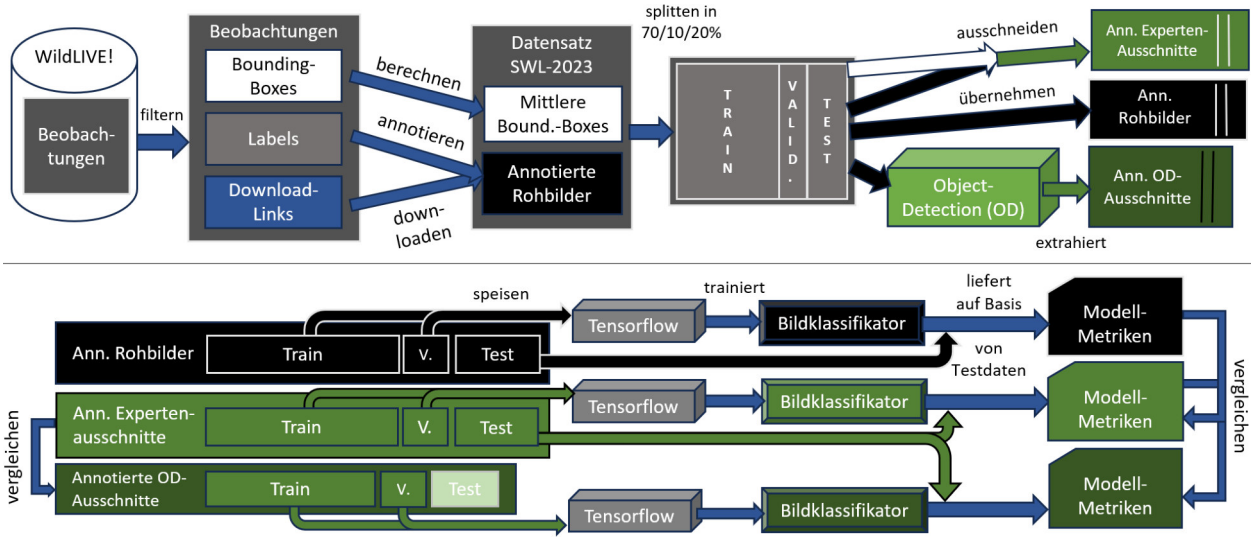

Abbildung 6.2: Versuchsaufbau "Klassifikatoren mit manuellen und automatischen Bildausschnitten"

gelieferten Bildausschnitte bzw. Bounding-Boxes direkt verglichen werden. Dazu werden Expertenausschnitte mit den Object-Detection-Ausschnitten verglichen (s. Abschnitt 6.2.4).

6.2.1 Datenvorbereitung

Die Daten werden simultan zu Abschnitt 5.2 aus dem Rohbestand selektiert. Es liegen somit etwa 18 850 annotierte Fotofallenfotos vor, auf denen laut Experten exakt ein Objekt zu sehen ist und die Bestimmung laut Experten in Bezug auf Trivialname und Kategorie eindeutig ist. Das Splitting der Datensätze findet vor der Object-Detection bzw. vor dem Ausschneiden statt. Aus dem Datensatz werden die Pfade zu den Fotos entnommen, bei denen die Bilder heruntergeladen werden können, die Labels "Trivialname" und "Kategorie" und die Koordinaten der Bounding-Box, welche die Citizen-Scientists und Reviewer gesetzt haben. Den Bounding-Boxen wird gemäß Abschnitt 6.1.1 die mittlere Bounding-Box b_{Mitte} für die Weiterverarbeitung entnommen. Alle weiteren Vorverarbeitungen geschehen synonym zum ersten Experiment (s. Abschnitt 6.1.1). Der Daten-split aus dem ersten Versuch wird wiederverwendet.

6.2.2 Expertenausschnitte & Object-Detection

Die Expertenausschnitte werden anhand der Bounding-Box-Daten der Citizen-Scientists und Reviewer erzeugt. Zu jedem Rohbild liegt eine Bounding-Box mit den Parametern X, Y, Breite und Höhe[13] vor, der aus der mittleren Bounding-Box b_{Mitte} errechnet ist. Zusätzlich zu dem Vorgehen im ersten Experiment, wird hier MegaDetectorv5a und MegaDetectorv5b mit verschiedenen Konfidenz-Schwellenwerten T ausgeführt. Es werden die Schwellenwerte $T = 0.20, 0.50, 0.80$ verwendet (MDx_T).

Darüber hinaus wird mit Faster R-CNN noch eine weitere Object-Detection eingesetzt ($FRCNN_{55}$). Das verwendete Modell[14] ist auf dem Open Images V4-Datensatz trainiert und besitzt einen Inception Resnet V2 Feature-Extractor. Das Modell ist im Gegensatz zu MegaDetector nicht auf Fotofallendaten optimiert und basiert zudem nicht auf der YOLO-Architektur. Als Konfidenz-Schwellenwert T kommt 0.55 zum Einsatz. Zudem wird eine Liste von akzeptierten Klassen[15] hinterlegt, die auf das Setting des "WildLIVE!"-Projektes passt. In Summe werden somit sechs Object-Detection-Prozesse durchgeführt: MegaDetectorv5a (MDa_{20}, $T = 0.20$), MegaDetectorv5a (MDa_{50}, $T = 0.50$), MegaDetectorv5a (MDa_{80}, $T = 0.80$), MegaDetectorv5b (MDb_{20}, $T = 0.20$), MegaDetectorv5b (MDb_{50}, $T = 0.50$), Faster R-CNN ($FRCNN_{55}$, $T = 0.55$)

[13] In Pixeln

[14] `https://tfhub.dev/google/faster_rcnn/openimages_v4/inception_resnet_v2/1`

[15] Animal, Pig, Deer, Bird, Cat, Tiger, Kangaroo, Dog, Squirrel, Duck, Leopard, Cattle, Cheetah, Jaguar, ...

6.2.3 Tuning & Training

Das Tuning und Training werden wie im ersten Experiment durchgeführt (s. Abschnitt 6.1.3). Aus dem Training resultieren für jedes Zielmerkmal $z \in Z = \{$ Trivialname, Kategorie $\}$ acht Bildklassifikatoren:

Typ	Name	Trainingsdaten	Validierungsdaten
Rohbildklassifikator	K^z_{Roh}	D^T_{Roh}	D^V_{Roh}
Bildausschnittklassifikator	K^z_{Roh}	D^T_{Exp}	D^V_{Roh}
Bildausschnittklassifikator	$K^z_{MDa_{20}}$	$D^T_{MDa_{20}}$	$D^V_{MDa_{20}}$
Bildausschnittklassifikator	$K^z_{MDa_{50}}$	$D^T_{MDa_{50}}$	$D^V_{MDa_{50}}$
Bildausschnittklassifikator	$K^z_{MDa_{80}}$	$D^T_{MDa_{80}}$	$D^V_{MDa_{80}}$
Bildausschnittklassifikator	$K^z_{MDb_{20}}$	$D^T_{MDb_{20}}$	$D^V_{MDb_{20}}$
Bildausschnittklassifikator	$K^z_{MDb_{50}}$	$D^T_{MDb_{50}}$	$D^V_{MDb_{50}}$
Bildausschnittklassifikator	$K^z_{FRCNN_{55}}$	$D^T_{FRCNN_{55}}$	$D^V_{FRCNN_{55}}$

Tabelle 6.2: Übersicht der Bildklassifikatoren für Versuch 2. Ein Klassifikator K^z wird somit auf D^T trainiert und mit Hyperparameter-Tuning mithilfe von D^V selektiert.

Das Tuning der Modelle wird, wie im ersten Experiment, mit identischen Parametern gegen den mittleren F_1-Score der Validierungsdaten durchgeführt. Für die Klassifikatoren, die auf Ausschnitten trainiert werden, die verschiedenen Object-Detection-Modellen entspringen, werden nicht separat Hyperparameter gesucht.

6.2.4 Evaluation

Evaluation der Bildklassifikatoren

Die Bewertung der acht Modelle K^z wird anhand des mittleren F_1-Scores durchgeführt. Die Rohbildklassifikatoren werden gegen Rohbilder evaluiert. Die Ausschnittklassifikatoren werden gegen die Expertenausschnitte von D_{Exp} evaluiert, da diese von Menschen, die über das größte Fachwissen und die größte Sorgfalt verfügen, erstellt wurden. Bei einer Evaluation mit Bildausschnitten, die aus einer Object-Detection hervorgehen, wäre eine Verfälschung der Evaluationsergebnisse zu erwarten, die von der Güte der Object-Detection abhängt.

Evaluation der Object-Detection

Da somit für Ausschnitte ein Referenzdatensatz ("ground-truth") besteht, lassen sich die Object-Detection-Verfahren auch direkt bewerten. Anders als im ersten Experiment, in dem die Güte der Object-Detection nur indirekt über die Güte des Bildklassifikators evaluiert werden konnte, kann in diesem Experiment die Güte der Ausschnitte direkt berechnet werden. Es könnte möglich sein, dass bei treffenden und genaueren Ausschnitten die Güte des Klassifikators auch besser ist. Als Basis dienen die manuellen Expertenausschnitte von "WildLIVE!"-Citizen-Scientists und -Reviewern.

Bei der Bewertung der Güte eines Objektdetektors auf einem Datensatz können ähnliche Metriken wie bei einer Klassifikation verwendet werden. Es liegt auch hier eine Konfusions-

matrix vor.

TP, FN und FP können regulär evaluiert werden (s. Abschnitt 2.5), aber TN wird in dieser Arbeit nicht verwendet, da der Grunddatensatz nur Bilder mit einer Bounding-Box enthält. Somit $TN = 0$. Die in dieser Arbeit eingesetzten Objektdetektoren bestimmen nicht automatisch die Klasse des gefundenen Objekts. Bounding-Boxen werden somit nur anhand ihrer Position im Bild bewertet.

TP, FN und FP sind im Rahmen von Objektdetektoren relativ zum IoU-Schwellenwert $[IoU = T]$. Vorhergesagte Bounding-Boxen b mit $IoU(b, b_{Referenz}) > T$ gelten als korrekt vorhergesagt. Aus den Bausteinen der Klassifikationsmetriken für Objekterkennung können dann die bekannten Metriken, wie Accuracy, Precision, Recall und F_1-Score, unter Berücksichtigung von $TN = 0$ berechnet werden:[16]

Vorhersagerate

$$PPR = \frac{TP + FP}{TP + FP + FN} \tag{6.2.1}$$

Wie häufig Bounding-Boxes - korrekt oder falsch - vom Objektdetektor OD auf allen Bildern gefunden wurden, bestimmt, wie groß der für Training und Validierung vorliegende Datensatz D^T_{OD} bzw. D^V_{OD} ist.

Accuracy

$$A = \frac{TP}{TP + FP + FN} \tag{6.2.2}$$

Wie häufig eine korrekte Bounding-Box gegenüber aller Bilder gefunden wurde.

Precision

$$P = \frac{TP}{TP + FP} \tag{6.2.3}$$

Wie häufig gefundene Bounding-Boxen wirklich korrekt waren.

Recall

$$R = \frac{TP}{TP + FN} \tag{6.2.4}$$

Wie häufig auf Bildern mit wahren Bounding-Boxes die korrekte Bounding-Box gefunden wurde.

[16] Alle Metriken verstehen sich mit Zusatz $[IoU=T]$

Mittlere IoU der TP

$$\overline{IoU_{TP}} = \frac{1}{TP} \sum_{b \in B_{TP}} IoU(b_{Referenz}, b) \tag{6.2.5}$$

Die gemittelte IoU auf Basis aller True-Positive-Boxen B_{TP} gegenüber ihren Referenz-Bounding-Boxen $b_{Referenz}$. Wie präzise die Object-Detection die Bounding-Boxen gegenüber der Referenz-Bounding-Box $b_{Referenz}$ überdeckt, in den Fällen wo die Bounding-Box b ein Treffer war. Sie gibt die geometrische Präzision in Form der Passgenauigkeit des Object-Detection-Modells gegenüber der Referenz an.[17]

COCO AP
Die mittlere Precision aus der Fläche unter dem Precision-Recall-Graph (vgl. [COC11] und Abschnitt 2.5.4).

Die zentrale Metrik für die Bewertung der Object-Detection wird die COCO $AP^{[IoU=.5:.05:.95]}$ sein. Der mAP kommt nicht zum Einsatz, da die Object-Detection in diesen Versuchen nicht direkt eine Klassifizierung vornimmt.

6.3 Prüfung auf Übertragbarkeit

Nach Abschluss der Versuche mit Daten mit einem randomisierten Split in SWL-2023 in den Abschnitten 6.1 und 6.2, besteht die Möglichkeit, die Übertragbarkeit/Generalisierbarkeit der Modelle zu überprüfen. Auf Basis der Erkenntnisse von Beery et al. [BvP18] besteht der Verdacht, dass Bilder von bekannten Kamerapositionen gut, jedoch Bilder anderer Kameraposition schlecht erkannt werden. Zu dieser Frage eröffnet ein weiterer Versuch die Möglichkeit, zu untersuchen, ob Serienbilder und spezielle Bildkonstellationen eine überproportional gute Klassifikation von Rohbildern oder Bildausschnitten begründen. Es wird statt eines randomisierten Splits ein fachlicher Split nach Stationen im Datensatz SWL-2023 vorgenommen.

Die Kamerastationen sind im Datensatz SWL-2023 sehr ungleichmäßig verteilt. Die Arten, die an jeder Station vorkommen, sind auch ungleichmäßig verteilt. Sie bieten jedoch die Möglichkeit, eine Verteilung zwischen Trainings-, Validierungs- und Testdaten nach 70%/10%/20% vorzunehmen (s. Abbildungen 6.3, 6.4).

Um einen Split zu ermöglichen, bei dem jeder Teildatensatz genügend Beobachtungen jeder Klasse besitzt, wird die Klassenmenge auf die häufigsten Arten reduziert, die mindestens 200-mal im Gesamtdatensatz vorkommen. Der neue Datensatz S reduziert sich auf 7.400 Bilder mit 17 Klassen. Alle weiteren Hyperparameter, Bildausschnitte, Vorgehensweisen und Metriken werden beibehalten.

[17] In anderen Kontexten wären auch IoU_P oder IoU_{FP} hilfreich, die auf allen gefundenen Bounding-Boxen gegenüber der Referenz-Bounding-Box - inklusive aller Bounding-Boxen $b \in B$ mit $IoU(b_{Referenz}, b) <= T$ oder aller Bounding-Boxen ohne Beschränkung der IoU basieren. Es gelte $IoU_P \leq IoU_{TP}$ und hätte durch die Ausreißer, die ohnehin kein Treffer waren, weniger Aussagekraft. IoU_{FP} beschriebe, wie sehr sich die False-Positive-Bounding-Boxen, die wegen mangelnder Überlappung Negativ sind, mit den korrekten Bounding-Boxen überlappen.

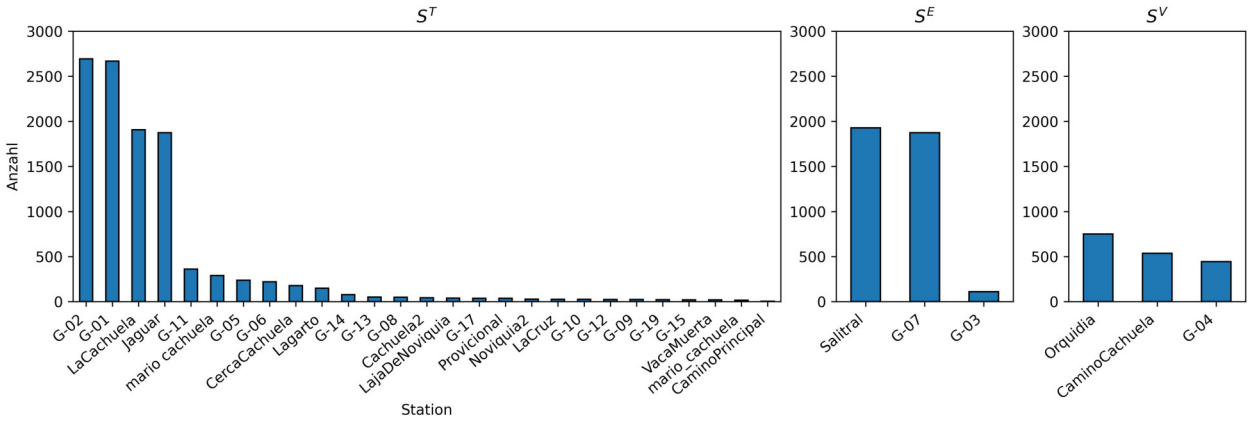

Abbildung 6.3: Verteilung der Daten S auf die Kamerastationen und Split-Datensätze für Training S^T, Test S^E und Evaluation S^V

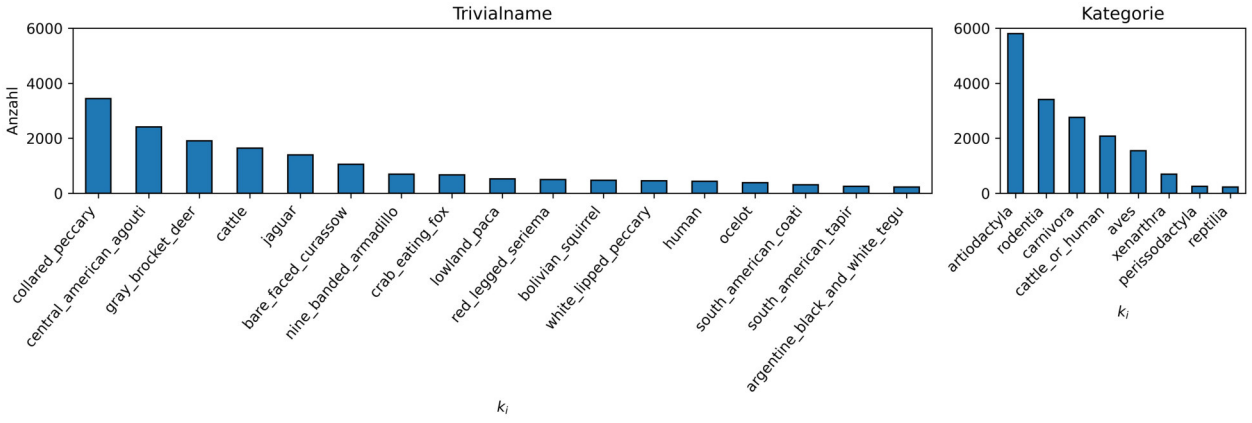

Abbildung 6.4: Verteilung der Daten S auf die Zielklassen

6.4 Implementierung

Bei der Implementierung wird als Ausgangspunkt der Computer-Vision-Code von iNaturalist verwendet (vgl. [iNa24]), die auf Python und Tensorflow/Keras basiert. Routinen für Datenvorbereitung, Augmentierung, Training und Evaluation lagen im iNaturalist-Projekt bereits vor und konnten für die Zwecke dieser Thesis erweitert und wiederverwendet werden.

6.4.1 Hardware

Die Umgebung wird mit Python 3.9, Conda und Docker aufgebaut. Das Training und Tuning wird mit Tensorflow/Keras in Version 2.10.1 und die Objekterkennung mit MegaDetectorV5 mit GPU-Unterstützung abgewickelt. Zum einen auf zwei parallelisierten NVIDIA GeForce RTX 2080 Ti (11 GB) und zudem auf einer NVIDIA RTX A2000 (8 GB) werden die GPU-intensiven Routinen durchgeführt. In beiden Fällen kamen zum Teil SSD- und HDD-Festplatten zum Einsatz.

Mit dem RTX-A2000-GPU war der Einsatz der CUDA-Version 11.2.2 (cudnn 8.1.0.77) für die GPU-Beschleunigung möglich. Mit den RTX-2080-Ti-GPUs kam die CUDA-Version 12.1.3 (cudnn 8.9.2.26) zum Einsatz.[18] Für die GPU-Parallelisierung konnte die Mirrored-Strategy[19] und für die Rechenbeschleunigung und Speicheroptimierung der Mixed-Precision-Modus[20] von Tensorflow verwendet werden. Die Python-Umgebung wird mit Conda und Docker ausgeführt.

6.4.2 Hindernisse

Zu Beginn des Projekts zeigten sich Probleme mit den Koordinaten der Bounding-Boxes. Weil offenbar aus dem eingesetzten System für das annotieren der Bounding-Boxes falsche Koordinaten exportiert wurden, waren häufig die Werte für Breite und Höhe vertauscht. Dies führte dazu, dass falsche Bildausschnitte verwendet wurden, die für das Training offensichtlich schädlich sind.

Der Einsatz von Hyperband-Tuning führte einige Zeit in die Irre, da einige letztendlich gute Modelle häufig in den ersten Epochen noch schlechte Ergebnisse lieferten. Hyperband sortiert jedoch Hyperparameterkonfigurationen mit schlechten Ergebnissen frühzeitig aus und gibt ihnen nicht die Chance bis zu einer späten Epoche durchzurechnen. Dies war insbesondere bei Parametrisierung ohne Einfrieren der Basisarchitektur (Ganzes Netz trainieren: “Ja”) der Fall. Es stellte sich somit erst relativ spät heraus, dass ein Training des gesamten Netzes vorteilhaft ist. Gerade dieses Training ist jedoch sehr rechenintensiv und musste darüber hinaus in der vollen Epochenlänge durchgeführt werden, was die Experimentlaufzeiten radikal in die Höhe trieb.

Bei unterschiedlichen Basisarchitekturen und Bildauflösungen stellt sich eine unterschiedliche Speicherverwendung der GPU ein, die mit der Anzahl an Parametern des neuronalen Netzes und der Bildauflösung wächst. Bei einer konstanten Menge an Speicher muss die Batch-Size reduziert werden, wenn die Anzahl an Netzparametern steigt. Somit wäre bei einem Tuning

[18] `https://developer.nvidia.com/rdp/cudnn-archive`
[19] `https://www.tensorflow.org/api_docs/python/tf/distribute/MirroredStrategy`
[20] `https://www.tensorflow.org/guide/mixed_precision`

eine Kopplung von Netzarchitektur und Bildauflösung an Batch-Size nötig, um ohne Speicherfehler trainieren zu können. Eine Kalkulation einer maximalen Batch-Size, die für eine gegebene Netzarchitektur und Bildauflösung auf einer spezifischen GPU wäre wünschenswert. Womöglich ist eine Näherung nur durch Ausprobieren möglich.

Die Selektion von guten Modellen mithilfe von Tensorboard ist prinzipiell praktisch und optisch gut aufbereitet. Im Laufe der Durchführung stellte sich jedoch heraus, dass die Modellmetriken nicht korrekt gemäß des Early-Stoppings angezeigt werden. Tensorboard zeigt immer die Metriken der letzten Epoche an, wobei Early-Stopping in diesem Projekt natürlich konfiguriert war, die beste Epoche/Metrik innerhalb der letzten Epochen zu wählen - gemäß Early-Stopping-Patience.

7 Ergebnisse

Gemäß der Reihenfolge der Versuchsaufbauten von Versuch 1 und 2 werden zunächst die Ergebnisse der Object-Detection, dann die Tuning- und Trainings-Ergebnisse und zuletzt die Evaluationsergebnisse mit Testdaten besprochen. Da die beiden Experimente aufeinander aufbauen, werden die Ergebnisse hier zusammengefasst diskutiert.

7.1 Object-Detection

Aus den Rohbildern können bei unterschiedlicher Parametrisierung mit MegaDetectorv5 mit einer COCO AP$^{[IoU=.5:.05:.95]}$ von bis zu 80%, einem Recall$^{[IoU=0.75]}$ von bis zu 96% und einer Vorhersagerate von bis zu 96%, Bounding-Boxen ausgeschnitten werden. Die Faster-R-CNN-Object-Detection schneidet deutlich schlechter ab, was bei der fehlenden Optimierung auf Fotofallendaten nicht überraschend ist (s. Abbildung 7.1). Beide Modelle benötigen auf einer RTX-A2000 GPU etwa 1000-1500ms/Bild für die Object-Detection im Non-Batch-Modus.

Die besten Ergebnisse liefert gemäß der COCO AP$^{[IoU=.5:.05:.75]}$ das MegaDetector-Modell v5a mit Threshold $T = 0.50$ (MDa_{50}) mit einem Wert von 0.801. Dieses Modell liefert mit Vorhersagerate 94.6% eine Bounding-Box, womit sich der Trainingsdatensatz nur geringfügig verkleinert. Die Precision$^{[IoU=.75]}$ dieser Bounding-Boxen beträgt für MDa_{50} 96,4% (s. Abbildung 7.2). Die mittlere IoU der Bounding-Boxen in TP$^{[IoU=.75]}$ beträgt 0.928, die mittlere Konfidenz in TP$^{[IoU=.75]}$ beträgt 0.926 und die mittlere Seitenlänge der Predicted Positive beträgt 1007px (gegenüber 1002px im Referenzdatensatz).

Die b-Variante von MegaDetector MDb_{20} schneidet in der AP-Metrik bei gleichem Schwellenwert schlechter ab als sein Pendant der a-Variante. Konfigurationen mit Schwellenwert $T = 0.20$ erreichen erwartungsgemäß eine niedrigere Precision und einen höheren Recall.

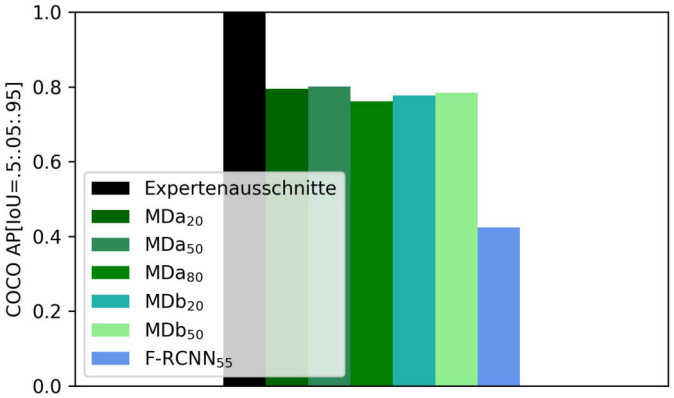

Abbildung 7.1: COCO AP$^{[IoU=.5:.05:.95]}$ der verschiedenen Object-Detection-Verfahren auf D_{Roh} von SWL-2023 gegenüber der Bounding-Box-Referenz D_{Exp}.

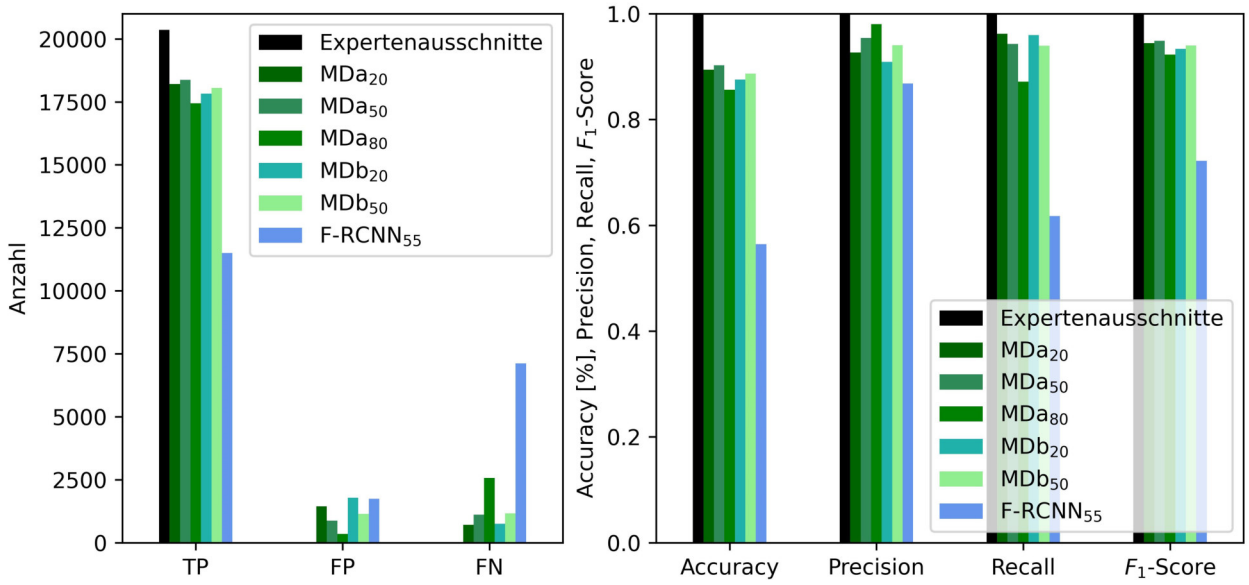

Abbildung 7.2: Metriken$^{[IoU=.75]}$ der verschiedenen Objektdetektoren gegenüber D_{Exp} - aus der binären Konfusionsmatrix für Object-Detection berechnet. Alle Fotos besitzen exakt eine wahre Bounding-Box $\Rightarrow TN = 0$.

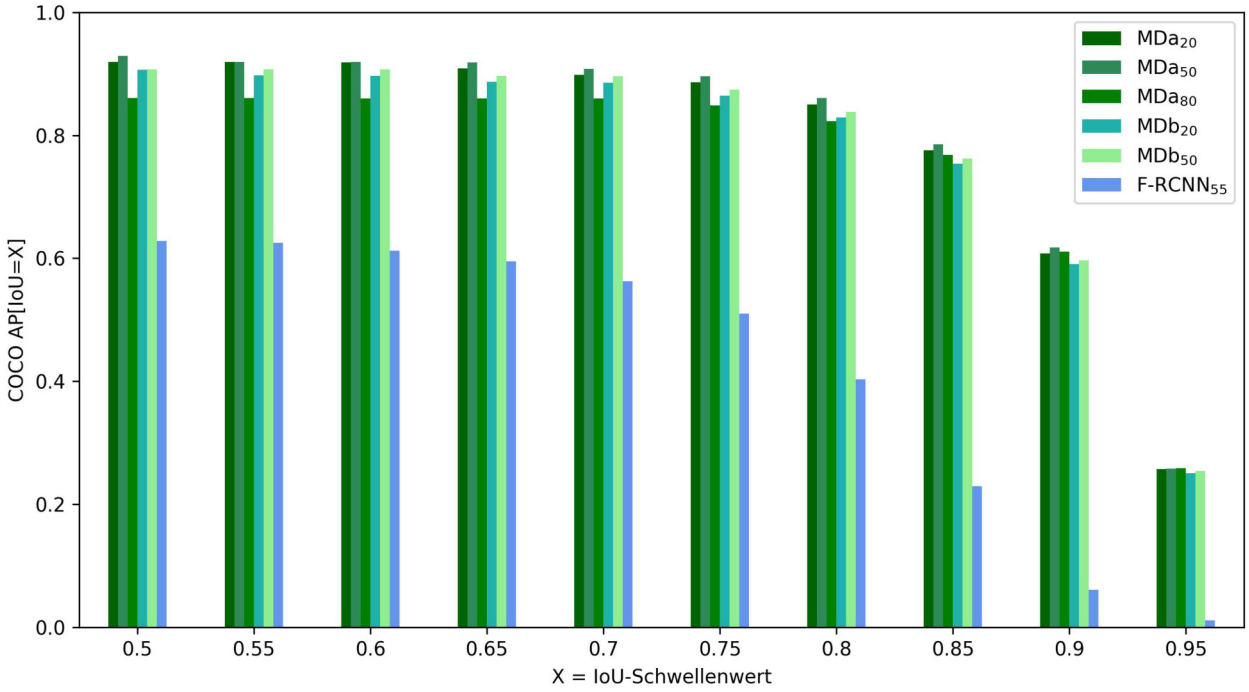

Abbildung 7.3: COCO AP[IoU=X] der verschiedenen Objektdetektoren gegenüber D_{Exp}

Konfigurationen mit $T = 0.80$ liefern eine höhere Precision und einen deutlich niedrigeren Recall. Alle MegaDetector-Modelle haben einen ähnlichen Verlauf gegenüber strengeren IoU-Schwellenwerten (s. Abbildung 7.3). Faster R-CNN ist bei allen Metriken klar abgeschlagen.

Die von MDa_{50} nicht erkannten Bounding-Boxen betreffen überwiegend verwackelte, verschwommene oder sehr kleine Objekte (s. Abbildung 7.4).

7.2 Hyperparameter-Tuning

Das Hyperparameter-Tuning, das in einer groben und einer feinen Suche durchgeführt wird, liefert beste Hyperparameter-Konfigurationen für die Rohbildklassifikatoren sowie die Ausschnittklassifikatoren. Die Ergebnisse zeigen, dass ein Tuning auf dem Zielmerkmal Trivialname ausreicht. Dies könnte darauf zurückzuführen sein, dass die beiden Merkmale sich semantisch überlappen. Für ein auf Trivialnamen optimiertes Modell wäre es einfach, die Kategorie vorherzusagen, da diese direkt mit einer simplen 1-zu-1-Zuordnung aus dem Trivialnamen ableitbar ist.[1]

Die folgenden Parameter werden nicht einem Hyperparameter-Tuning unterzogen: Batch-Size (variabel je nach Umgebung)[2], Max-Epochen (70 Epochen), Early-Stopping-Patience (10 Epochen), Early-Stopping-Metrik (mittlerer F_1-Score) und Transfer-Learning-Gewichte (ImageNet).

7.2.1 Hyperband-Tuning

Im ersten Schritt werden grobe Richtungen für Hyperparameter sondiert und schwache Basisarchitekturen aussortiert. Wertebereiche der einzelnen Hyperparameter werden nur kategorisch aufgespannt. Es liegen keine Hyperparameter vor, für die in einem kontinuierlichen Bereich gesucht wird. Alle Float- und Integer-Werte werden als kategorielle Werte hinterlegt. Da bei zwei Bildtypen und 13 Hyperparametern mit jeweils zwischen zwei und zehn kategoriellen Werten leider eine kombinatorische Explosion vorliegt, kann nicht jede Kombination zur vollen Ausprägung verprobt werden.

[1] Eine Optimierung auf Kategorie könnte umgekehrt womöglich zu unspezifisch sein.

[2] Auf zwei NVIDIA GeForce RTX 2080 Ti für Xception mit 299x299px-Auflösung und Training des gesamten Netzes: 48 Bilder pro Batch

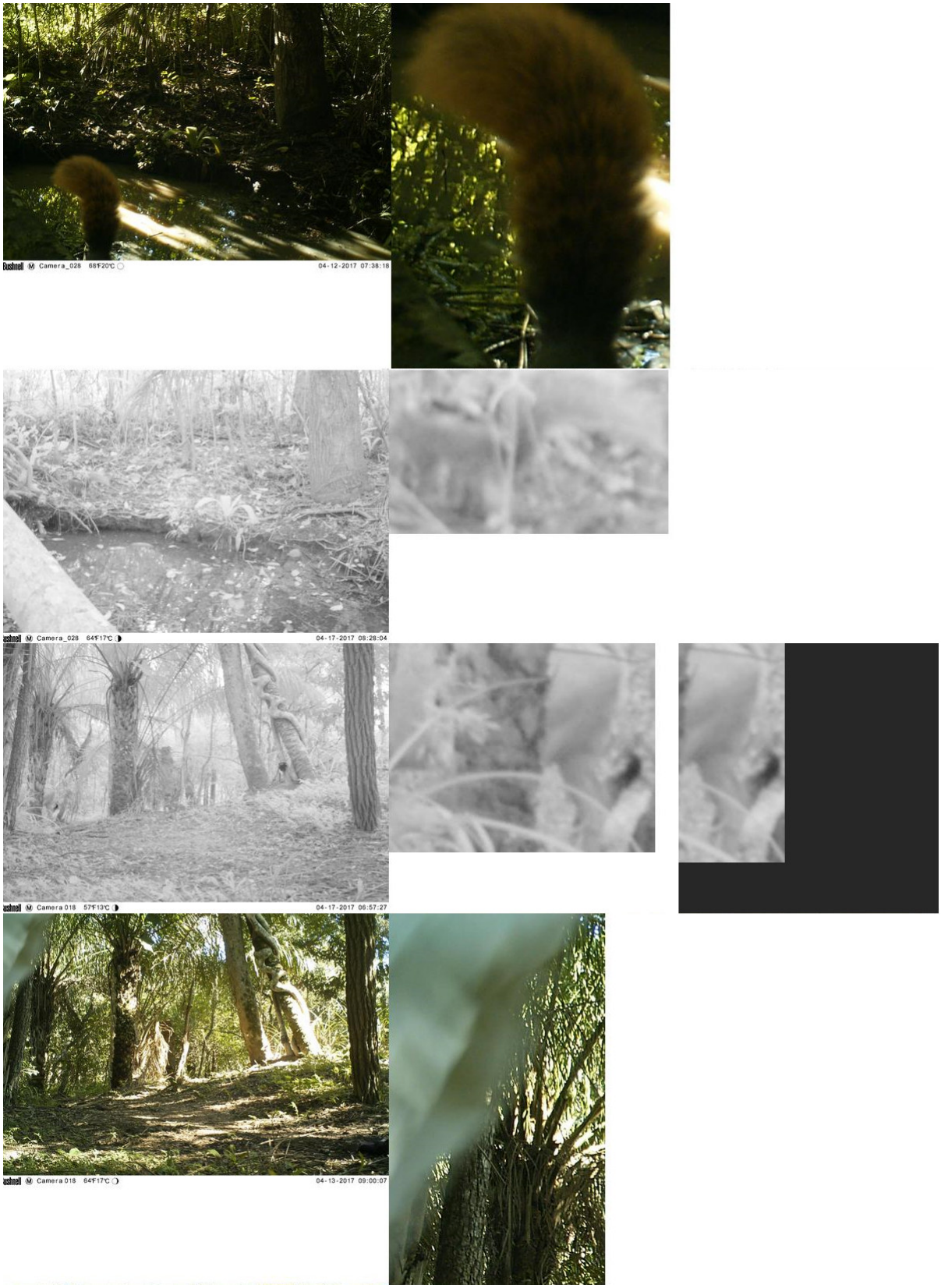

Abbildung 7.4: Rohbilder (rechts), optimale Ausschnitte (Mitte), von MDa_{50} gefundene Ausschnitte (rechts) für ausgewählte Fotos. Zu sehen wäre zweimal ein Eichhörnchen, ein Hinterteil eines Huftiers und ein Stück einer Kuh (von oben nach unten).

Aus dem ersten Tuning-Schritt gehen folgende Erkenntnisse für Rohbilder und Bildausschnitte hervor:

- Basisarchitekturen mit mehr Parametern sind besser, jedoch ressourcenintensiver. Der Zusammenhang zwischen Parameterzahl und Trainingslaufzeit ist mindestens linear. Xception liefert im Rahmen der vorliegenden Hardware-Ausstattung die besten Ergebnisse.
- Bildaugmentierung durch Flip und Farbe ist förderlich.
- Augmentierung mit zentralen Bildausschnitten ($< 100\%$) führt zu schlechteren Ergebnissen.
- Höher aufgelöste Bilder sind förderlich, jedoch ressourcenintensiver. Der Zusammenhang zwischen Größe und Trainingslaufzeit ist quadratisch.
- Kategorielle Kreuzentropie liefert gute Ergebnisse.
- Training mit Dropout-Rate $> 0\%$ führt zu schlechteren Ergebnissen.
- Label-Smoothing-Raten von 10% sind förderlich.
- RMSProp-Hyperparameter ρ und Momentum sind koabhängig.
- ϵ hat ein Optimum bei 0.1.
- Die Initiale Lernrate η_0 mit Werten 0.001, 0.0001 ist koabhängig zu den RMSProp-Hyperparametern.
- Das ganze Netz, inklusive vortrainiertem Feature-Extractor zu trainieren, ist förderlich.

7.2.2 Grid-Search-Tuning

Im zweiten Schritt wird auf Basis der im ersten Schritt ermittelten Tendenzen eine Grid-Search auf folgenden Wertebereichen durchgeführt:

Die daraus resultierenden 24 Kombinationen für jeden Bildtyp werden in insgesamt 48 Durchläufen mit den Trainings- und Validierungsdatensätzen D^T_{Exp}/D^E_{Exp} sowie D^T_{Roh}/D^E_{Roh} vollständig durchgeführt. Die Grid-Search führt zu den in Tabelle 7.1 abgebildeten Parametern. Es finden sich entgegen der Annahme wenig Unterschiede für Hyperparameter des Trainings auf Rohbildern oder Bildausschnitten. Die Hyperparameter-Kombination aus $[\rho = 0.99, \beta = 0.99, \eta_0 = 0.001)]$ schneidet bei Xception bei allen Auflösungen am besten ab. Eine höhere Bildauflösung beim Rohbildklassifikator ist vorteilhaft (Auflösung$|F_1$: $299^2\text{px}^2|0.843$, $400^2\text{px}^2|0.844$, $500^2\text{px}^2|0.863$). Beim Training von Bildausschnitten ist der Unterschied gegenüber der Auflösung marginal (Auflösung$|F_1$: $299^2\text{px}^2|0.904$, $400^2\text{px}^2|0.916$, $500^2\text{px}^2|0.906$).

Hyperparameter	Wertemenge	Bereich
Netzarchitektur	{Xception}	Netz
Dropout-Rate	{ 0% }	Netz
Bildauflösung	{ 299^2px^2, 400^2px^2, 500^2px^2 }	Netz
Horizontaler Flip	{ Ja }	Augmentierung
Farbaugmentierung	{ 100% }	Augmentierung
Zentralausschnitt	{ 100% }	Augmentierung
Label-Smooth-Rate	{ 10% }	Training
RMSProp: ρ	{ 0.99, 0.999 }	Training
RMSProp: Momentum β	{ 0.99, 0.999 }	Training
RMSProp: ϵ	{ 0.1 }	Training
Initiale Lernrate η_0	{ 0.0001, 0.001 }	Training
Training: Ganzes Netz	{ Ja }	Training

Tabelle 7.1: Eingegrenzte Hyperparameter für das Tuning der Rohbild- und Bildausschnittklassifikatoren

Hyperparameter	Optimum für **Rohbilder**	Optimum für **Bildausschnitte**
Netzarchitektur	Xception	
Dropout-Rate	0%	
Bildauflösung	400^2px^2	500^2px^2
Horizontaler Flip	Ja	
Farbaugmentierung	100%	
Zentralausschnitt	100%	
Loss-Funktion	Kateg. Kreuzentropie	
Label-Smooth-Rate	10%	
RMSProp: ρ	0.99	
RMSProp: Momentum	0.99	
RMSProp: ϵ	0.1	
Initiale Lernrate η_0	0.001	
Training: Ganzes Netz	Ja	

Tabelle 7.2: Optimale Hyperparameter der Rohbild- und Bildausschnittklassifikatoren

Auf Basis von Xception und der optimalen Werte für ρ, β, η_0 und Bildauflösung werden noch ein letztes Mal separat leicht variierte Hyperparameter von Dropout-Rate, Label-Smooth-Rate, Flip-Augmentierung, Farbaugmentierung, Zentralausschnitt und Loss-Funktion getestet, um ein globales Hyperparameter-Optimum zu bestätigen. Keine variierte Hyperparameterkonfiguration zeigt bessere Ergebnisse. Optimale Hyperparameter finden sich in Tabelle 7.2.

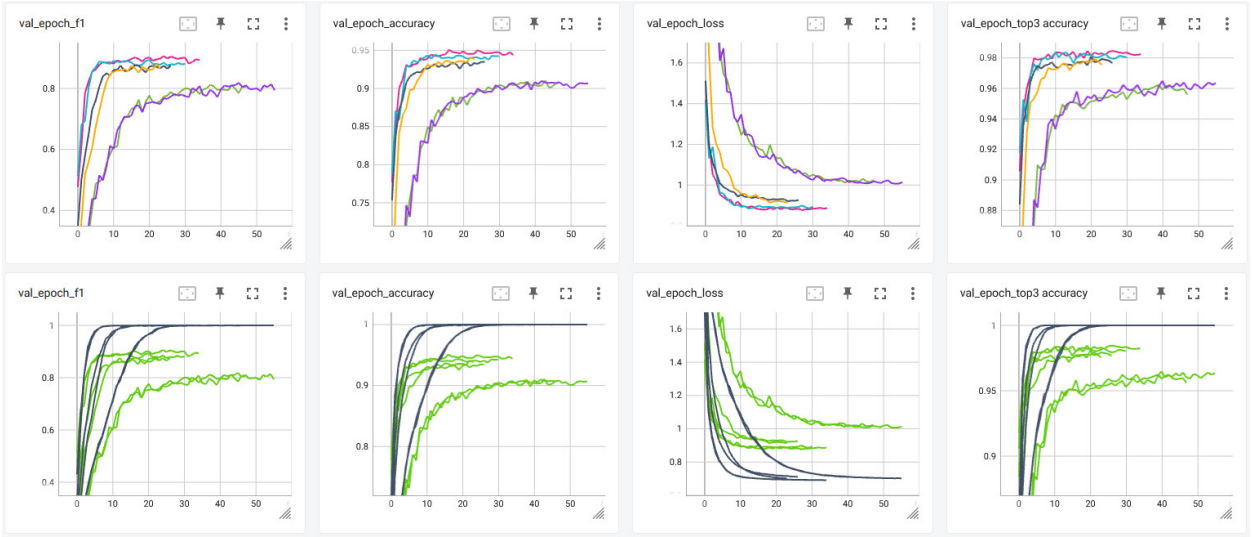

Abbildung 7.5: Konvergenz von verschiedenen Tuning-Durchläufen von Bildausschnittklassifikatoren bei F_1-Score, Accuracy, Loss und Top-3-Accuracy gegenüber Validierungsdaten (oben) und Konvergenz von Metriken gegenüber Trainingsdaten (schwarz) und Validierungsdaten (grün) in Tensorboard

Das Training inklusive Validierung benötigt für Bildausschnitte etwa 3:15 Minuten und für Rohbilder etwa 5:25 Minuten pro Epoche und erreicht einen optimalen F_1-Score nach 22-45 respektive 15-45 Epochen (s. Abbildung 7.5). Die Modelle für Ausschnitte sind durchweg in der Lage, innerhalb von 5-20 Epochen auf ihre Trainingsdaten zu optimieren und Accuracy und F_1-Score gegenüber den Trainingsdaten von 100% bzw. 1.0 zu erreichen. Gegenüber den Validierungsdaten erreichen Loss, Accuracy und F_1-Score im selben Zeitraum Plateaus, die dem letztendlichen Optimum dieses Trainingsdurchlaufs entspricht (s. Abbildung 7.5). Die Modelle für Rohbilder erreichen nach 10-25 Epochen ihre Plateaus auf den Trainings- und Validierungsdaten. Die Early-Stopping-Patience könnte somit womöglich noch leicht reduziert werden.

Es zeigt sich zudem, dass ein Optimum des F_1-Scores im Tuning mit den Optima von Loss und Accuracy zusammenfällt (s. Abbildung 7.5). Eine Modellselektion könnte somit womöglich auch auf diesen Metriken durchgeführt werden. Stichprobenartig kann bestätigt werden, dass ein Tuning auf der Kategorie nicht nötig ist. Die Ergebnisse zeigen, dass gute Hyperparameterkombinationen für Trivialname auch für Kategorie gut sind.

7.3 Evaluation der Bildklassifikation

Mithilfe der im Tuning für Bildausschnitte und Rohbilder ermittelten Hyperparameter werden die Modelle auf den Rohbildern, Expertenausschnitten und Ausschnitten aus der Object-Detection gegen die Zielmerkmale Trivialname sowie Kategorie trainiert. Die Klassifikatoren werden dann gegenüber einigen vorliegenden Testdatensätzen evaluiert (s. Abbildung 7.6), die aus der Verarbeitung der Rohbilder mit Object-Detection entstanden sind. Die Evaluationsergebnisse im F_1-Score schwanken über alle Klassifikatoren hinweg um etwa 0.009 und 0.008 am F_1-Score respektive Accuracy. Die zentrale Metrik stellt bei der Auswertung der mittlere F_1-Score dar (s. Abschnitt 2.5.2). Der Rohbildtestdatensatz D_{Roh} und der Expertenausschnitt-

testdatensatz D_{Exp} sind jedoch die wichtige Referenz. Die Rohbildklassifikatoren K_{Roh} werden anhand ihrer Metriken bei D_{Roh} evaluiert und alle Ausschnittklassifikatoren K_{Exp}, $K_{MDa_{20}}$, $K_{MDa_{50}}$, $K_{MDa_{80}}$, $K_{MDb_{20}}$, $K_{MDb_{50}}$, $K_{FRCNN_{55}}$ anhand ihrer Metriken bei D_{Exp}.

Es lassen sich folgende Tendenzen erkennen: Klassifikatoren schneiden bei andersgearteten Testdaten schlecht ab (z.B. $F_1(K^{Triv}_{Roh}|D^E_{Exp})$ oder $F_1(K^{Triv}_{Exp}|D^E_{Roh})$, also F_1-Score von K^{Triv}_{Roh} evaluiert am Datensatz D^E_{Exp} und F_1-Score von K^{Triv}_{Exp} evaluiert am Datensatz D^E_{Roh}). Klassifikation der Kategorien ist ähnlich erfolgreich, wie die Klassifikation der Trivialnamen ($F_1(K^{Kat}_{Exp}|D^E_{Exp}) = 0.907$, $F_1(K^{Triv}_{Exp}|D^E_{Exp}) = 0.909$). Klassifikatoren, die auf Expertenausschnitten trainiert sind, schneiden insgesamt am besten ab. Die Klassifikatoren, die auf Faster-R-CNN-Ausschnitten trainiert wurden, schneiden am schlechtesten ab. Klassifikatoren, die auf MegaDetector-Ausschnitten trainiert wurden, schneiden in etwa gleich gut ab.

Unter den Object-Detection-Ausschnittklassifikatoren für Trivialname K^{Triv}_{OD} erreicht gegen Testdaten MegaDetectorv5b mit Threshold $T = 0.50$ den höchsten F_1-Score. Der beste solche Klassifikator gegen Testdaten für Kategorie K^{Kat}_{OD} erreicht bei MegaDetectorv5a mit Threshold $T = 0.20$ den höchsten F_1-Score (s. Tabelle 7.3).

Klassifikator	Berechnungsgrundlage	mittlerer F_1-Score
Trivialname		
K^{Triv}_{Roh}	D^V_{Roh}	0.855
K^{Triv}_{Roh}	D^E_{Roh}	0.8734
K^{Triv}_{Exp}	D^V_{Exp}	**0.904**
K^{Triv}_{Exp}	D^E_{Exp}	**0.909**
$K^{Triv}_{MDb_{50}}$	$D^V_{MDb_{50}}$	0.862
$K^{Triv}_{MDb_{50}}$	D^E_{Exp}	0.8889
Kategorie		
K^{Kat}_{Roh}	D^V_{Roh}	0.925
K^{Kat}_{Roh}	D^E_{Roh}	0.8855
K^{Kat}_{Exp}	D^V_{Exp}	0.921
K^{Kat}_{Exp}	D^E_{Exp}	0.9066
$K^{Kat}_{MDa_{20}}$	$D^V_{MDa_{20}}$	**0.941**
$K^{Kat}_{MDa_{20}}$	D^E_{Exp}	**0.9084**

Tabelle 7.3: Beste Evaluationsergebnisse der Klassifikatoren für Trivialname und Kategorie gegen Validierungs- und Testdaten

Zielmerkmal Trivialname

Bei der Beurteilung der Klassifikatoren ist auch die Genauigkeit in Bezug auf einzelne Klassen zu berücksichtigen. Anhand der Konfusionsmatrizen kann beurteilt werden, welche Klassen besser oder schlechter erkannt werden (s. Anhang 9.1, 9.2). Es lässt sich schließen, dass sich bis auf den Bereich der Artiodactyla (hier: Wildschweine und Rehe) keine gehäuften Fehler erkennen lassen. Jedoch schneiden Klassen im F_1-Score schlecht ab, die selten auftraten. Dazu lassen sich die seltenen Klassen “giant_anteater”, “horse” und “equipment” zählen.

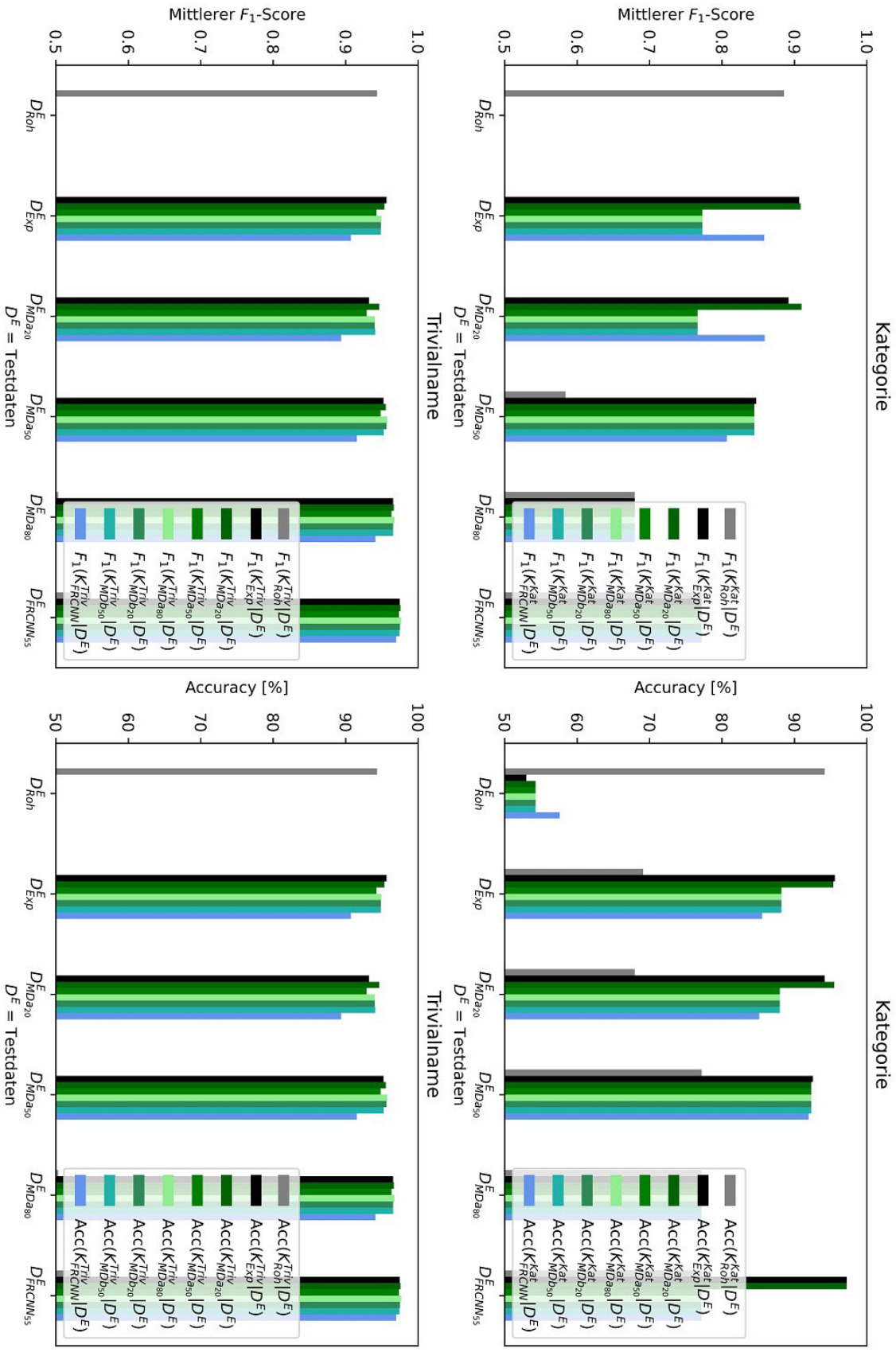

Abbildung 7.6: Mittlerer F_1-Score und Accuracy der verschiedenen Klassifikatoren für Trivialnamen und Kategorien K^{Triv} und K^{Kat} gegenüber verschiedenen Testdatensätzen D^E

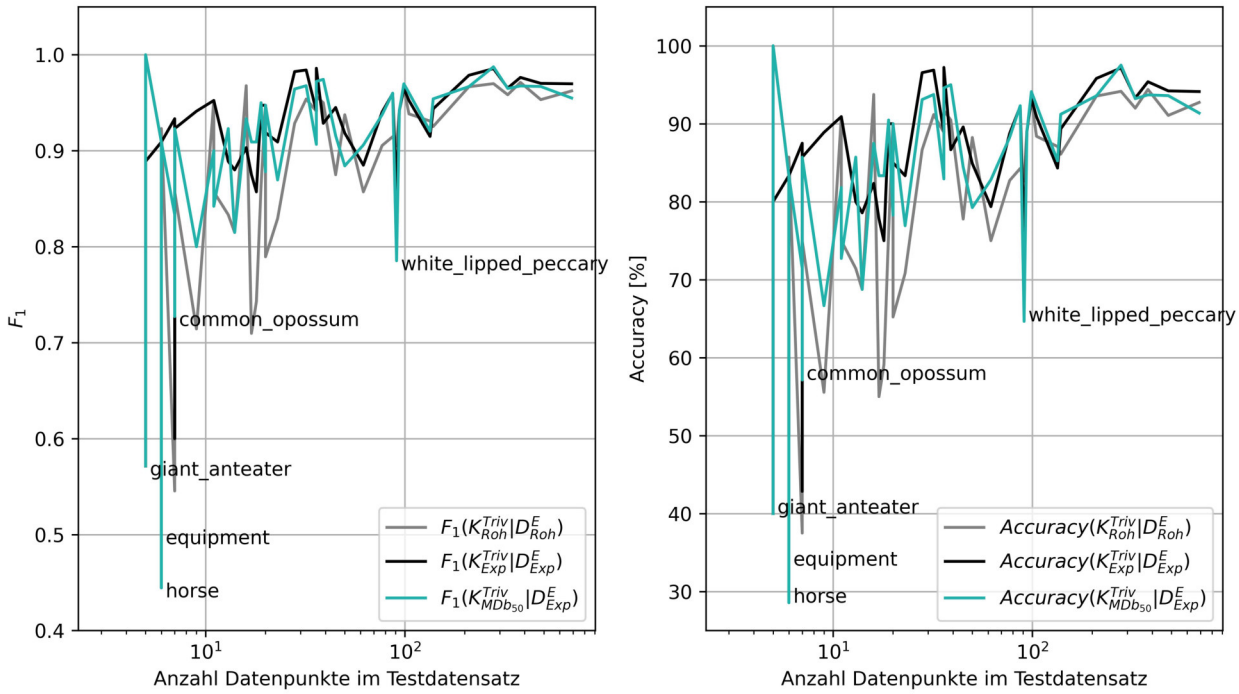

Abbildung 7.7: F_1-Score und Accuracy der drei besten Klassifikatoren bei Trivialnamen. Klassen sind anhand der Anzahl Datenpunkte im Testdatensatz aufgereiht. Die Anzahl an Datenpunkten im Testdatensatz ist auch proportional zur Anzahl Datenpunkten in Trainings- und Validierungsdaten.

Klassen, die häufiger im Datensatz vorkommen, werden tendenziell besser erkannt (s. Abbildung 7.7). Keine Klasse wird jedoch perfekt wiedererkannt. K^{Triv}_{Exp} schneidet überwiegend besser als $K^{Triv}_{MDb_{50}}$ und dieser besser als K^{Triv}_{Roh} ab. Die Kurven verlaufen an einigen Stellen in etwa parallel zueinander, was auf strukturelle Fehler oder Unkenntlichkeiten im Datensatz vermuten lässt.

Ausschnittklassifikatoren K^{Triv}_{OD} nähern sich der Güte von K^{Triv}_{Exp} an, wenn die selteneren Klassen bei der Beurteilung außer Acht gelassen werden, während sie weiterhin gemeinsam besser abschneiden als Rohbildklassifikatoren (s. Tabelle 7.4).

Es lässt sich zusammenfassen, dass die Abweichungen zwischen den Metriken der verschiedenen MegaDetector-Ausschnittklassifikatoren offenbar nicht signifikant sind. Die Auswahl von MegaDetectorv5b mit Threshold $T = 0.50$ scheint aber akzeptable Ergebnisse zu liefern. Die minimalen Abweichungen lassen sich vermutlich mit den sehr geringen Unterschieden in den Datensätzen $D^T_{MDx_T}$ erklären. Größere Mengen von eher unscharfen oder womöglich falsch ausgeschnittenen Bildern bei niedrigen T auf der einen Seite und geringere Mengen wichtiger Ausschnitte bei hohen T auf der anderen Seite können die Qualität des Trainingsdatensatzes mindern.

Klassifikator	Klassenmenge	mittlerer F_1-Score
K^{Triv}_{Roh}	Alle Klassen	0.8734
K^{Triv}_{Exp}	Alle Klassen	**0.9085**
$K^{Triv}_{MDb_{50}}$	Alle Klassen	0.8889
K^{Triv}_{Roh}	≥ 10 Datenpunkten in D^E	0.9065
K^{Triv}_{Exp}	≥ 10 Datenpunkten in D^E	**0.9371**
$K^{Triv}_{MDb_{50}}$	≥ 10 Datenpunkten in D^E	0.9274
K^{Triv}_{Roh}	≥ 20 Datenpunkten in D^E	0.9270
K^{Triv}_{Exp}	≥ 20 Datenpunkten in D^E	**0.9467**
$K^{Triv}_{MDb_{50}}$	≥ 20 Datenpunkten in D^E	0.9369

Tabelle 7.4: Klassifikationsergebnisse der drei besten Bildklassifikatoren ohne Berücksichtigung seltener Klassen

Ist auch der Ausschnittklassifikator auf Expertenausschnitten insgesamt am besten, ist jedoch insbesondere der Vergleich zwischen den Metriken von K_{Roh} und K_{MDx} relevant. In neuen Forschungsprojekten liegen keine Bounding-Boxen von Experten vor, womit auf die automatisch gefundenen Ausschnitte zurückgegriffen werden muss. Erst bei einer signifikanten Verbesserung durch das Schneiden von Ausschnitten gegenüber der Verwendung von Rohbildern ist der Aufwand dieser Vorverarbeitung zu rechtfertigen. Diese Verbesserung beträgt hier nur 0.0155 F_1-Punkte (+1.77%).

Zielmerkmal Kategorie

Die Konfusionsmatrix des Kategorie-Klassifikators $K^{Kat}_{MDa_{20}}$ zeigt gute Ergebnisse. Nur wenige Tiere außerhalb der Kategorien “cattle_or_human” und “others” werden schlecht erkannt (s. Abbildung 7.8). Sogar die seltene Kategorie Marsupialia wird gut, fast fehlerfrei erkannt. Der Klassifikator ist zudem in der Diagonale der Konfusionsmatrix besser als der Rohbildklassifikator für Kategorien. Klassen, die häufiger im Datensatz vorkommen, werden tendenziell besser erkannt und bei den meisten Kategorien liefert der Bildausschnittklassifikator leicht bessere Ergebnisse als der Rohbildklassifikator (s. Abbildung 7.9).

Es lässt sich zusammenfassen, dass die Abweichungen zwischen den Metriken der verschiedenen MegaDetector-Ausschnittklassifikatoren offenbar auch hier nicht signifikant sind. Die Auswahl von MegaDetectorv5a mit Threshold $T = 0.20$ scheint mit leichtem Abstand die besten Ergebnisse zu liefern. Die Verbesserung zwischen Rohbildklassikator K^{Kat}_{Roh} und MegaDetector-Ausschnittklassifikator $K^{Kat}_{MDa_{20}}$ beträgt hier 0.0229 F_1-Punkte (+2.59%). Erstaunlicherweise schneidet hier der Ausschnittklassifikator mit MegaDetector-Ausschnitten entgegen der Modelle gegen Trivialname sogar leicht besser ab als der Ausschnittklassifikator auf Expertenausschnitten K^{Kat}_{Exp} (s. Tabelle 7.3).

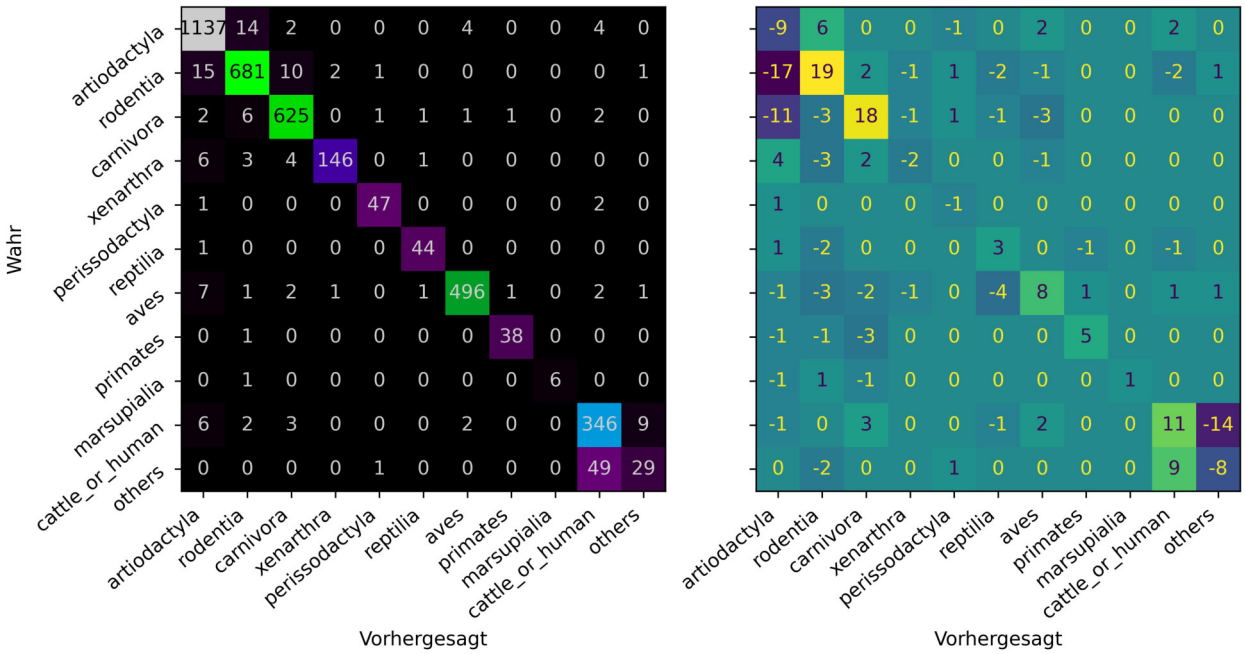

Abbildung 7.8: Links Konfusionsmatrix des Klassifikators MegaDetector-Bildausschnittklassifikators $K^{Kat}_{MDa_{20}}$ gegen Experten-Trainingsdaten D^{E}_{Exp} und rechts Verbesserung anhand subtrahierter Konfusionsmatrix $K^{Kat}_{MDa_{20}} - K^{Kat}_{Roh}$. Gehäufte Verwechslungen sind bei artiodactyla, rodentia und carnivora sowie zwischen cattle_or_human and others zu beobachten.

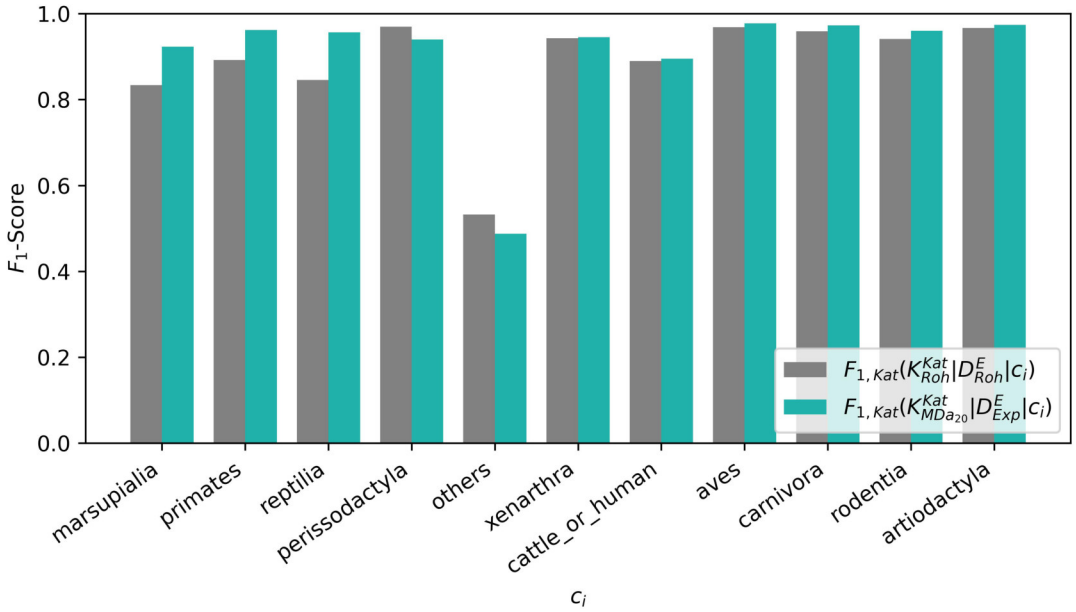

Abbildung 7.9: F_1-Scores der Kategorien c_i des Rohbildklassifikators K^{Kat}_{Roh} und des MegaDetector-Bildausschnittklassifikators $K^{Kat}_{MDa_{20}}$
aufsteigend sortiert nach Datenmenge der Kategorie c_i

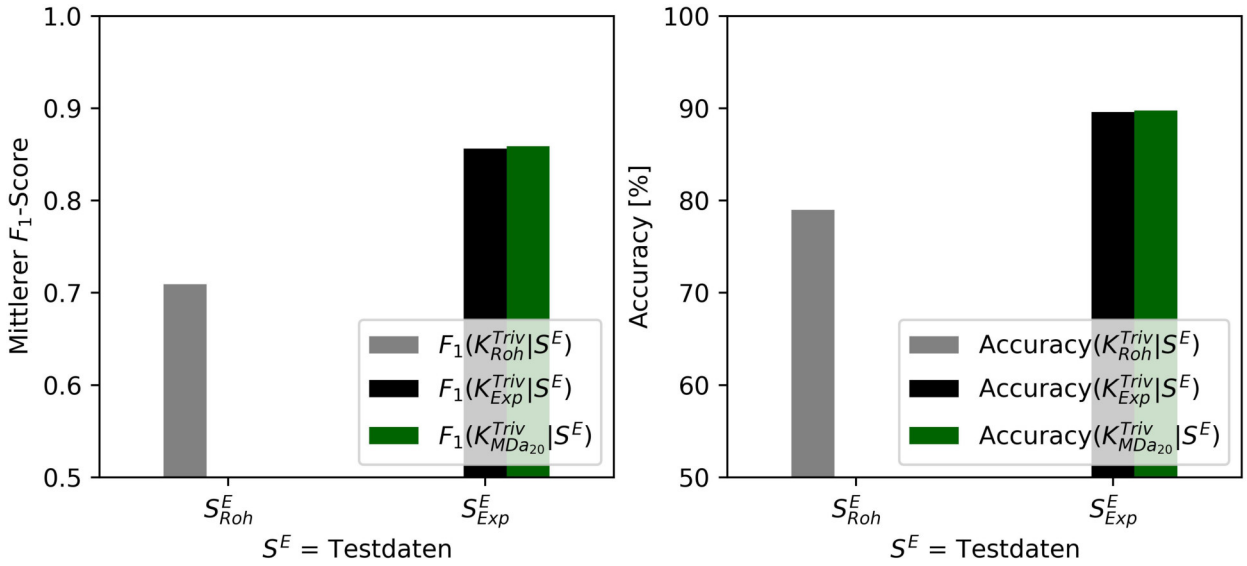

Abbildung 7.10: F_1-Scores und Accuracies des Rohbildklassifikators K^{Triv}_{Roh}, Bildausschnittklassifikators aus Basis von Expertendaten K^{Triv}_{Exp} und Bildausschnittklassifikators auf Basis von MegaDetector-Ausschnitten $K^{Triv}_{MDa_{20}}$ bei stationenweise gesplitteten Daten

7.4 Übertragbarkeit der Klassifikationsergebnisse auf andere Standorte

Entgegen der Ergebnisse bei den vorherigen Versuchen auf Basis von randomisierten Splits, zeigt sich beim Evaluieren anhand von Fotomaterial neuer Stationen ein sehr klarer Trend, dass Klassifikation auf Bildausschnitten vorteilhaft ist (s. Abbildung 7.10). Der Rohbildklassifikator K^{Triv}_{Roh} ($F_1 = 0.709$) schneidet deutlich schlechter ab als die beiden ähnlich erfolgreichen Ausschnittklassifikatoren K^{Triv}_{Exp} ($F_1 = 0.856$) und $K^{Triv}_{MDa_{20}}$ ($F_1 = 0.858$). Andere Ausschnittklassifikatoren, die auf anderen MegaDetector-Konfigurationen trainiert wurden, liefern sehr ähnlich gute Ergebnisse. Die Konfiguration v5a/$T = 0.20$ schneidet mit einem hauchdünnen Vorsprung am besten ab. Auch die Top-1-Accuracy zeigt einen deutlichen Unterschied zugunsten des Bildausschnittklassifikators.

Es ist kein eindeutiger Trend erkennbar, dass Klassen mit mehr Datenmaterial deutlich besser erkannt werden (s. Abbildung 7.11). Es zeigt sich, dass schwierig zu unterscheidende Arten, wie Jaguar, Ozelot, Halsband- und Weißbartpekari insbesondere durch Ausschnittklassifikatoren besser auseinandergehalten werden können.

Anhand der subtrahierten Konfusionsmatrix (s. Abbildung 7.12) zeigt sich, dass an fast allen Stellen auf der Diagonale der Ausschnittklassifikator $K^{Triv}_{MDa_{20}}$ besser abschneidet. An einigen Stellen liegen jedoch auch deutliche Verschlechterungen vor: gray_brocket_deer/collared_peccary, gray_brocket_deer/argentine_black_and_white_tegu, bolivian_squirrel/cattle, south _american_tapir/gray_brocket_deer.

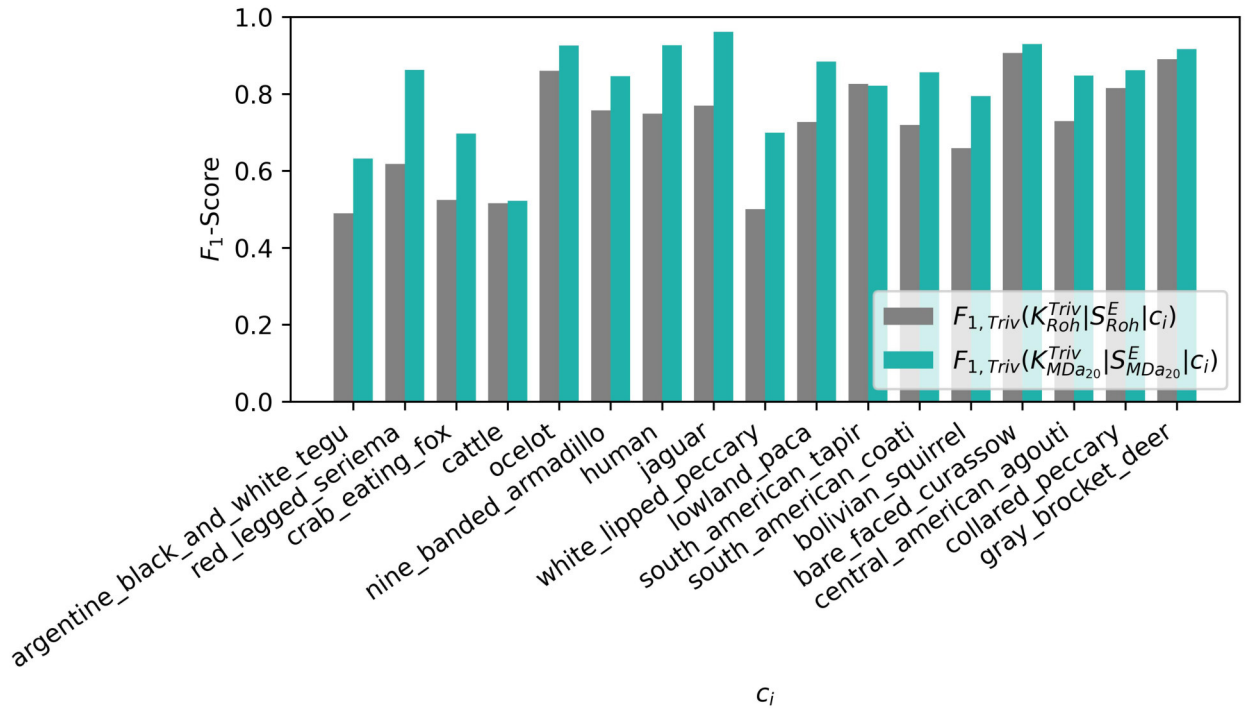

Abbildung 7.11: F_1-Scores der Trivialnamen c_i der Klassifikatoren K^{Triv}_{Roh} und $K^{Triv}_{MDa_{20}}$ gegenüber S aufsteigend sortiert nach Datenmenge der Klasse c_i

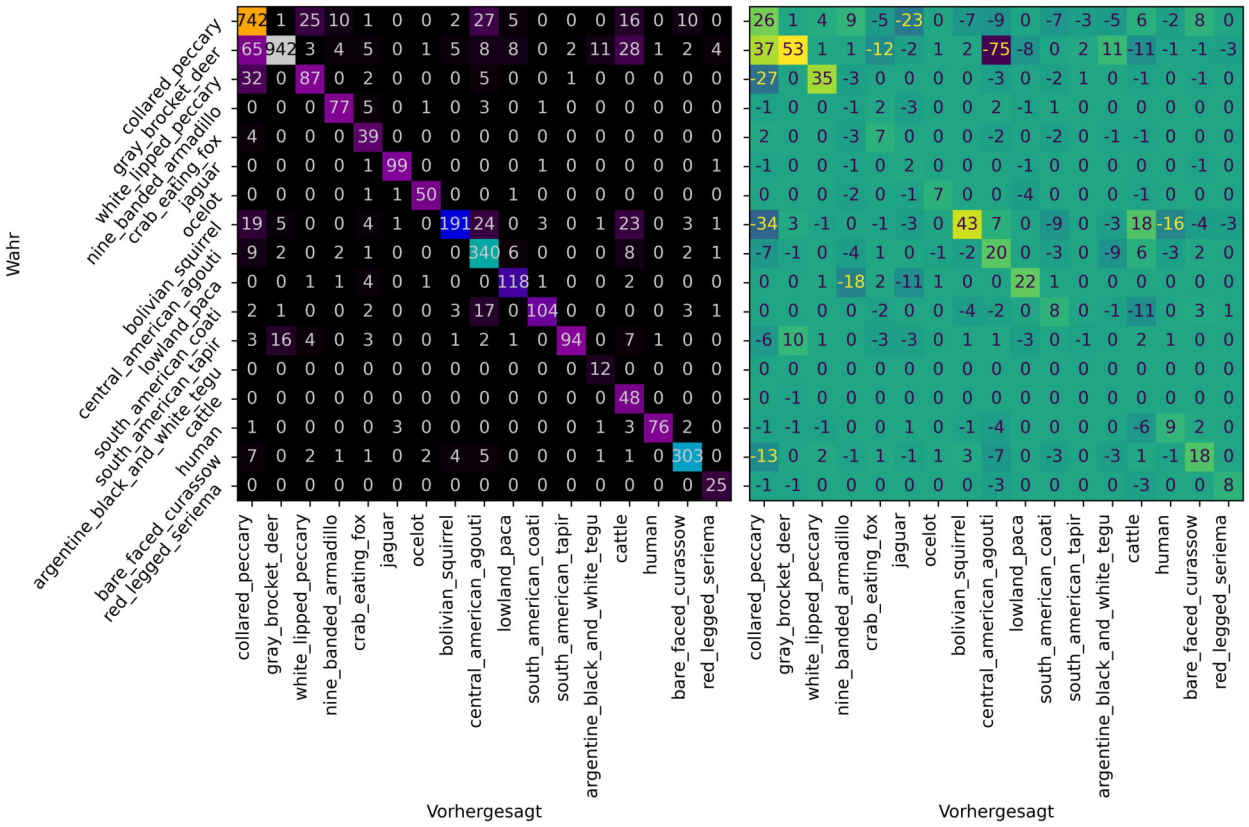

Abbildung 7.12: Links Konfusionsmatrix des Klassifikators $K^{Triv}_{MDa_{20}}$ gegen D^E_{Exp} und rechts Verbesserung anhand subtrahierter Konfusionsmatrix $K^{Kat}_{MDa_{20}} - K^{Triv}_{Roh}$. Alle Daten auf Basis der stationenweise gesplitteten Daten S.

8 Diskussion

Die Auswertungen der Object-Detection und Bildklassifikation auf SWL-2023 von "Wild-LIVE!" zeigen gemischte Resultate. Die Klassifikationsergebnisse mit randomisiertem Split sind besser, wenn Bildausschnitte verwendet werden. Dies ist aber nur dann eindeutig der Fall, wenn die Bildausschnitte von Experten verwendet werden. Modelle, die auf Ausschnitten trainiert sind, die per Object-Detection gewonnen wurden, schneiden trotz einer hohen Überdeckung mit den Expertenausschnitten durchweg etwas schlechter ab. Es besteht zwar immer noch ein Vorsprung dieser Modelle gegenüber Rohbildklassifikatoren, der aber womöglich nicht aussagekräftig ist (Trivialname: $F_1 = 0.888$ ggü. 0.873, Kategorie: $F_1 = 0.908$ ggü. 0.886). Bei der Klassifikation von Bildern von anderen ungesehenen Stationen schneiden jegliche Ausschnittklassifikatoren jedoch sehr gut ab.

Beim Vergleich der bewerteten Object-Detection-Modelle und der Bildklassifikatoren fällt auf, dass der Datensatz mit der besten COCO AP nicht den besten Klassifikator trainiert. MegaDetector v5a mit Threshold $T = 0.50$ hat die beste COCO AP, während der beste Klassifikator auf MegaDetector v5b mit Threshold $T = 0.50$ trainiert wurde. Die Unterschiede liegen jedoch in einem Bereich, der als statistisches Rauschen interpretiert werden kann.

Auffällig ist beim besten Bildklassifikator auf Object-Detection-Bildausschnitten die schlechte Genauigkeit bei Weißbartpekari (100 Datenpunkte, 0.785 F_1, 64.6% Accuracy), die den Halsbandpekari (708 Datenpunkte, 0.955 F_1, 91.4% Accuracy) sehr ähnlich sind. Hier zeigt die Analyse der Fotos, dass viele Fotos zwar eindeutig mit einer der beiden Arten markiert sind, dieser Schluss aber auf Basis des Fotomaterials nur sehr schwer oder gar nicht möglich wäre (s. Abbildung 8.1). Hier offenbart sich vermutlich das Problem mit Bestimmungen auf Basis von Fotoserien (s.a. Abschnitt 5.2) oder womöglich sogar Fehlbestimmungen im Referenzdatensatz.

Dem Modell $K^{Triv}_{MDb_{50}}$ unterlaufen jedoch auch in der Breite noch einige Fehler, die mit einer genaueren Analyse ergründet werden sollten (s. Abbildung 8.2).

Klassifikatoren für das Zielmerkmal Kategorie liefern leicht bessere Ergebnisse. Der beste gemittelte F_1-Score ist zwar nicht deutlich höher als der des Klassifikators für Trivialnamen, jedoch lässt sich dies primär auf das schlechte Abschneiden der Klasse "others" zurückführen, die sich in den Daten stark mit der Klasse "cattle_or_human" überlagert. Eine Zusammenführung dieser Klassen ist empfehlenswert, wenn die Trennung aus fachlicher Perspektive vertretbar ist. In der Tendenz ist die Bestimmung anhand von Ausschnitten für alle Klassen von Vorteil und liefert bei allen Tiergruppen F_1-Scores von mindestens 0.94.

Abbildung 8.1: Von $K^{Triv}_{MDb_{50}}$ fehlklassifizierte Bildausschnitte mit abgebildeten Pekari-Wildschweinen (Tayassuidae). Von links nach rechts zwei Weißbartpekari (*Tayassu pecari*), die als Halsbandpekari (*Dicotyles tajacu*) eingestuft wurden, wo die Merkmale der Köpfe nicht klar erkennbar sind. Ein Halsbandpekari, dessen "Halsband" erkennbar ist, aber fehlklassifiziert wurde und ein Pekari, das scheinbar ein Halsband hat, aber von Experten als Weißbartpekari bestimmt wurde.

Abbildung 8.2: Von $K^{Triv}_{MDb_{50}}$ fehlklassifizierte Ausschnitte. Von links nach rechts eine Kuh, die womöglich wegen des Nachtfotos als ein (nachtaktiver) Puma (*Puma concolor*) klassifiziert wird; ein Margay (*Leopardus wiedii*), der als ein optisch sehr ähnlicher Ozelot (*Leopardus pardalis*) klassifiziert wird; ein Schwanzbüschel eines Nasenbärs (*Nasua nasua*), der auch zu einem Eichhörnchen gepasst hätte; ein Schwarzweißer Teju, dessen Muster im Entferntesten an einen Ozelot erinnert.

Für einen Einsatz der hier vorgestellten Klassifikationsmodelle lässt sich festhalten, dass Ausschnittklassifikatoren den Rohbildklassifikatoren mindestens gleichwertig gegenüberstehen. Diese Erkenntnis ist auch deshalb relevant, da ein Einsatz von Object-Detection bei Vorkommen von Fotos mit mehreren Tieren unverzichtbar ist. Mit den hier vorliegenden Resultaten kann festgehalten werden, dass durch das Ausschneiden keine Einbußen bei der Klassifikationsgüte zu erwarten ist - im Gegenteil.

Übertragbarkeit auf neue Stationen

Durch den dritten Versuch kann eindeutig gezeigt werden, dass die vorgestellten Ergebnisse auch auf vollständig neuem Bildmaterial im selben Ökosystem akzeptable Ergebnisse liefern. Mit nur 200 oder mehr Fotos pro Klasse können Klassifikatoren entwickelt werden, die sowohl hohe Precision als auch Recall erzielen. Eher schwierig zu unterscheidende Tierarten, wie Jaguar und Ozelot ($F_1 = 0.961$ und 0.907), können sehr gut separiert werden.

Die Ergebnisse bei Evaluation mit Fotos von ungesehenen Stationen zeigen zudem, dass die Genauigkeiten insgesamt schlechter sind als bei der Evaluation mit Fotos von bekannten Stationen ($F_1 = 0.888$ ggü. 0.858). Die Verschlechterung ist jedoch nicht groß. Sie lässt sich womöglich auf die Problematik beim randomisierten Split zurückführen, dass durch das Vorkommen von Fotoserien Testbilder vorliegen, die Trainingsbildern sehr ähnlich sind.

9 Ausblick

Die Ergebnisse zeigen, dass Training auf Bildausschnitten für die Artbestimmung auf Fotofallenfotos hilfreich ist. Darauf aufbauend können weitere Aspekte der Machine-Learning-Verfahren betrachtet werden und auch die Einbettung in eine Biomonitoring-Datenplattform untersucht werden.

Bei den eingesetzten Object-Detection-Modellen wurde im Rahmen dieser Arbeit darauf verzichtet, ein eigenes Training vorzunehmen. Es gilt zu erörtern, ob die Güte von MegaDetector durch das Speisen von Daten aus dem WildLIVE!-Projekt verbessert werden kann. Eine Veröffentlichung des annotierten WildLIVE!-Datensatzes für die Weiterentwicklung von offenen Object-Detection- und Bildklassifikationsmodellen wie MegaDetector könnte zudem ein Ziel darstellen (s. [Mic03], Abschnitt "Existing Collaborators").

Statt dem Einsatz von viereckigen Bounding-Boxes wäre eine Bildsegmentierung mit Verfahren, wie Segment Anything (vgl. [Kir+23]) von Vorteil, um jegliches Umgebungsrauschen aus dem Bild zu entfernen und die Klassifikation nur mit dem Objekt selbst zu speisen. Segmentierung könnte auch gut mit Object-Tracking kombiniert werden (vgl. [Wan23]).

Mit dem übergreifenden Ziel, eine Datenplattform für machine-learning-gestützte Bestimmung von Fotofallenfotos zu etablieren, liefert diese Arbeit einige Argumente, die Object-Detection von MegaDetector zu integrieren. Insbesondere, wenn mehrere Tiere im Bild zu sehen sind, ist jedoch der Einsatz von MegaDetector unverzichtbar. Eine Klassifikation kann dann auf den einzelnen Bildausschnitten vorgenommen werden. Durch das Wegfallen von visuellem Kontext lässt sich auf Basis dieser Arbeit kein gravierender Informationsverlust erkennen.

Eine Bestimmungsplattform für Fotofallendaten könnte aus folgenden Modulen bestehen:

- Daten-Ingest-Modul für eine Einspeisung von Fotos in das System
- Katalog für die Pflege der Metadaten und biologischen Taxonomien
- User-Interface-Module für manuelle Lokalisierung und Bestimmung von Objekten, Tieren und Menschen mithilfe von Computern und womöglich portablen Geräten
- Modul für die machine-learning-gestützte Vorhersage von leeren Bildern, Bounding-Boxes und Tierarten. Auswertungsmodule für die Zusammenfassung der Ergebnisse über Raum, Zeit, Arten und Bildmerkmale
- Konsens-Algorithmus, der Bestimmungen von Experten, Citizen-Scientists und Machine-Learning-Modul zusammenbringt
- Exportmodule für die Bereitstellung von Daten an zentrale Biodiversitätsdatenbanken, wie gbif (vgl. [Tel11], `https://www.gbif.org/`)

Abbildung 9.1: Kollage einiger MegaDetector-Bildausschnitte aus “WildLIVE!”

Für verschiedene Anwendungsfälle ist es hilfreich, die Bildsequenzen geordnet abzulegen und nicht als komplett isolierte Fotos zu betrachten. Die Fotos können manuell in der Sequenz besser bestimmt werden. Herausragende Bilder einer Sequenz könnten zudem markiert werden. Klassifikation kann dann auch auf die Klassifikation von Sequenzen statt einzelnen Bildern ausgeweitet werden, sobald ein Object-Tracking im Bild etabliert ist.

Mit Unterstützung von ökologischem Fachwissen wäre es möglich, Bildklassifikatoren bei neuen Forschungsvorhaben wiederzuverwenden. Eine Untermenge von Tierarten, die ein bestehender Bildklassifikator kennt, würde sich daran festmachen, welche Tierarten an dem Standort des Forschungsprojekts bekanntermaßen vorkommen. Tierarten, die am Ort keinesfalls vorkommen, sollten ausgeschlossen werden. Daraus resultieren mehrere Klassifikationsmodelle für verschiedene Habitate oder Forschungsprojekte. Object-Detection-Modelle können womöglich sehr großflächig wiederverwendet werden und müssten nur bei extremen Abweichungen des Fotomaterials, wie Unterwasserfotos, Nachtfotos oder Makrofotos angepasst werden (vgl. [Lop+20], [Mar+17]).

Um den besten Klassifikator für eine Biomonitoring-Plattform unter gegebenen Hardware-Limitationen zu konstruieren, ist noch weiteres Fine-Tuning zu empfehlen. Die hier vorgestellten Modelle basieren auf CNNs, wohingegen Vision Transformer aktuell in einigen Benchmarks im Bereich Naturfotos führend sind (vgl. [Pap24]). Zudem wurden in dieser Arbeit nicht alle Möglichkeiten von modernen CNNs, wie ConvNeXt oder EfficientNetV2 verprobt (vgl. [Liu+22], [MQ21]). Es besteht die Hoffnung, dass die Güte der Klassifikatoren, statt mit Xception, durch den Einsatz anderer Basisarchitekturen um einige Prozent verbessert werden kann.

Im Datensatz SWL-2023 liegen einige ungenaue Bestimmungen vor, die wegen unscharfem Bildmaterial nicht auf eine Tierart eingegrenzt werden können. Ein gutes Beispiel dafür sind die Pekari-Wildschweine. Im Datensatz wurden diese Bestimmungen auf einer höheren Ebene in der biologischen Taxonomie, zum Beispiel Gattung, Familie oder Ordnung, gesetzt (“pekari_sp”, “bat_sp”, . . .). Die hier vorgestellten Modelle für die Vorhersage von Arten schlossen die unspezifischen Bestimmungen aus methodischen Gründen aus. Bestimmungsmodelle müssen jedoch idealerweise diese Unschärfe in ihre Routinen integrieren und bei Unsicherheit in Bezug auf eine taxonomische Ebene, eine Bestimmung auf einer höheren Ebene vorschlagen. Hilfreich wäre zudem, in die Bestimmung örtliche und zeitliche Aspekte einzubeziehen, die aus historischen Daten abgeleitet werden könnten, da einige Arten nur an gewissen Orten oder zu gewissen Tages- oder Jahreszeiten zu beobachten sind (vgl. [Col+23]).

In Bezug auf ein Machine-Learning-Modul sind zudem folgende Aspekte zu betrachten: Es muss sichergestellt werden, dass die Einschätzung eines leeren Bilds korrekt ist und keine sichtbaren Individuen übersehen werden. Mehrere unterschiedliche Object-Detection-Module und auch unterschiedliche Klassifikationsmodule können für Ensemble-Ansätze kombiniert werden. Dies kann zum Beispiel durch unterschiedliche Parametrisierung der Modelle oder durch die Kombination von unterschiedlichen Architekturen (vgl. [Men+23]) erreicht werden. Die Bewertung der Bildklassifikation sollte auch mit Bestimmungen von Experten kombiniert werden. Die Bestimmungen der menschlichen und maschinellen Teilnehmer dieses Ensembles fließen dann mit jeweiligen Gewichtungen in die Gesamtbewertung ein. Auch dieses Zusammenspiel kann somit als ein Ensemble-Ansatz zwischen Machine-Learning-Modellen und Experten betrachtet werden, welches das Ziel hat, eine Beobachtung mit einer möglichst hohen Genauigkeit einzuordnen.

Anhang

Architektur	β	ρ	η	Auflösung	$F^V_{1,Roh}$	$F^V_{1,Exp}$
Xception	0.99	0.99	10^{-3}	299x299	0.84345	0.90438
Xception	0.999	0.99	10^{-3}	299x299	0.80399	0.84507
Xception	0.99	0.999	10^{-3}	299x299	0.85364	0.89364
Xception	0.999	0.999	10^{-3}	299x299	0.80346	0.83841
Xception	0.99	0.99	10^{-4}	299x299	0.80441	0.88121
Xception	0.999	0.99	10^{-4}	299x299	0.8353[2]	0.89553
Xception	0.99	0.999	10^{-4}	299x299	0.83528	0.89044
Xception	0.999	0.999	10^{-4}	299x299	0.82554	0.8913[2]
Xception	0.99	0.99	10^{-3}	400x400	0.82999	**0.91578**
Xception	0.999	0.99	10^{-3}	400x400	0.7936[2]	0.84040
Xception	0.99	0.999	10^{-3}	400x400	0.84409	0.90087
Xception	0.999	0.999	10^{-3}	400x400	0.79789	0.8311[2]
Xception	0.99	0.99	10^{-4}	400x400	0.81496	0.88368
Xception	0.999	0.99	10^{-4}	400x400	0.83550	0.89810
Xception	0.99	0.999	10^{-4}	400x400	0.8120[2]	0.88914
Xception	0.999	0.999	10^{-4}	400x400	0.82931	0.90248
Xception	0.99	0.99	10^{-3}	500x500	**0.86306**	0.90606
Xception	0.999	0.99	10^{-3}	500x500	0.80064	0.8119[2]
Xception	0.99	0.999	10^{-3}	500x500	0.85564	0.89319
Xception	0.999	0.999	10^{-3}	500x500	0.79267	0.81859
Xception	0.99	0.99	10^{-4}	500x500	0.83888	0.87654
Xception	0.999	0.99	10^{-4}	500x500	0.83029	0.88860
Xception	0.99	0.999	10^{-4}	500x500	0.82641	0.88433
Xception	0.999	0.999	10^{-4}	500x500	0.84029	0.89198

Tabelle 9.1: Tuning-Ergebnisse des Fine-Tunings im letzten Schritt mit Basisnetzarchitektur/Backbone, RMSprop Momentum β, RMSprop ρ, Initialer Lernrate η, Bildauflösung und dem F_1-Score gegen den Validierungsdatensatz D^V_{Roh} bzw. D^V_{Exp}

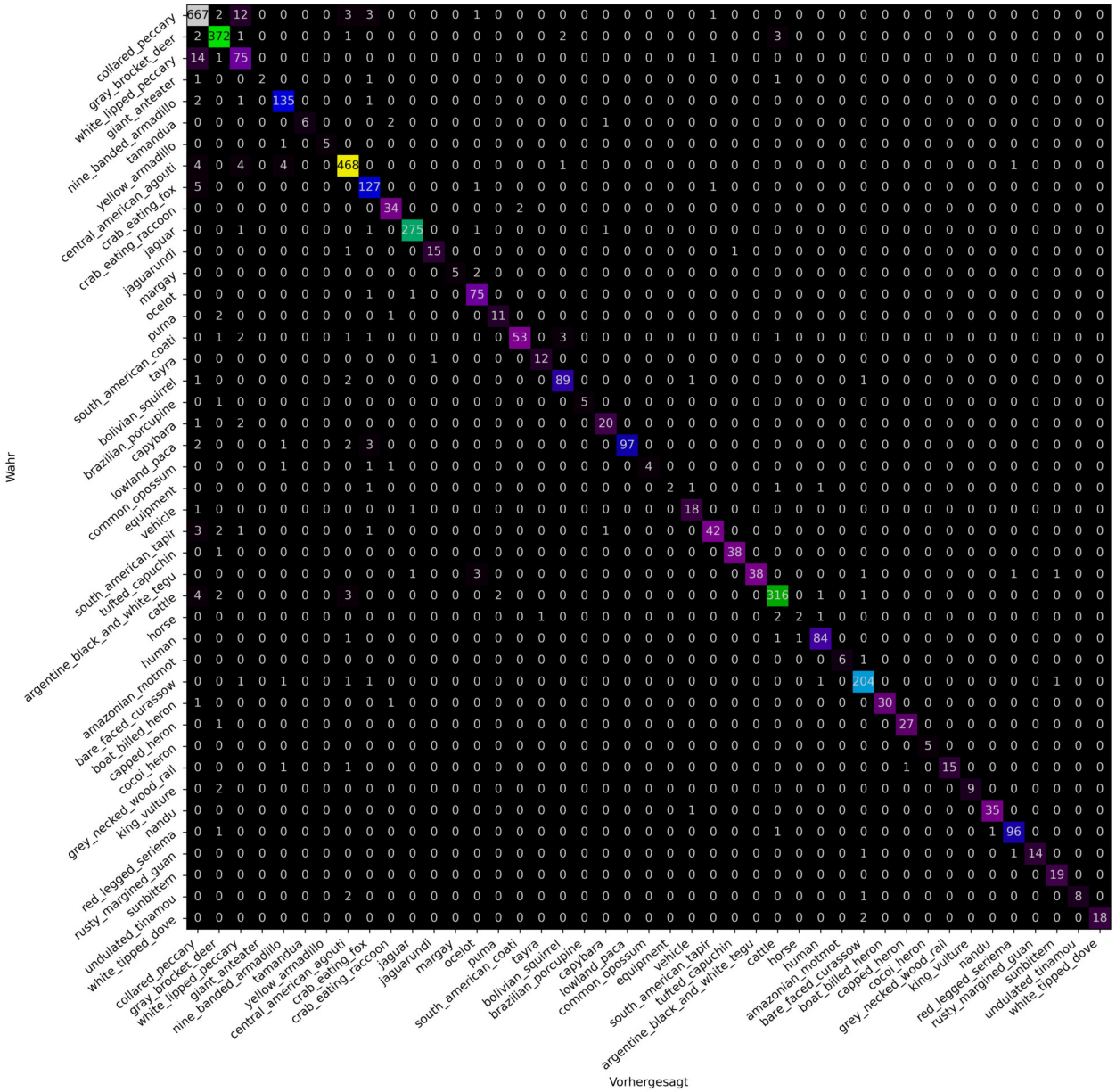

Abbildung 9.1: Konfusionsmatrix des Klassifikators $K^{Triv}_{MDb_{50}}$ gegen D^{E}_{Exp} bei randomisiertem Split. Gehäufte Verwechslungen sind bei collared_peccary, white_lipped_peccary und gray_brocket_deer zu beobachten.

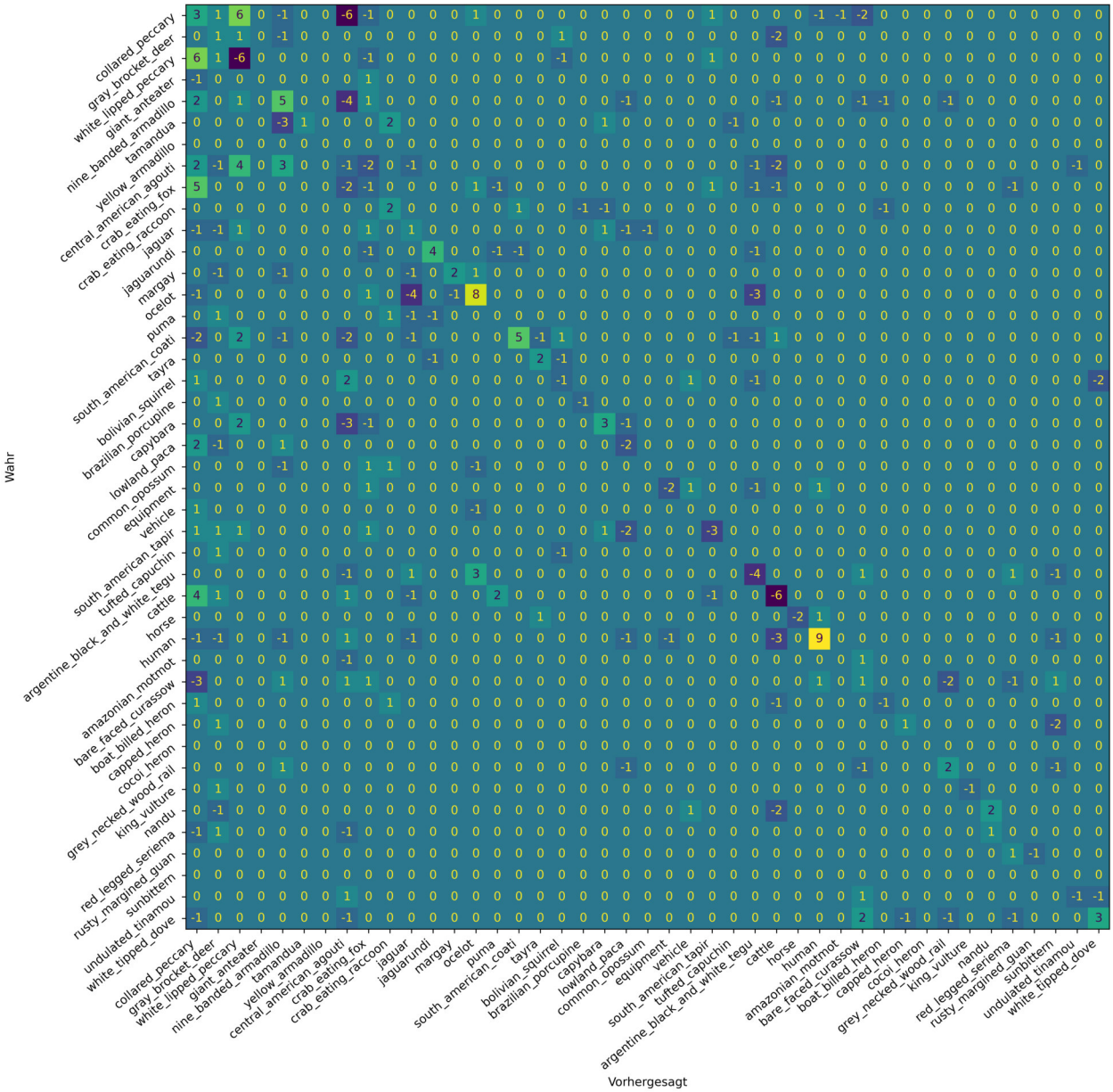

Abbildung 9.2: Subtrahierte Konfusionsmatrix: $K^{Triv}_{MDb_{50}} - K^{Triv}_{Roh}$ gegen D_{Exp} respektive D_{Roh}. $K^{Triv}_{MDb_{50}}$ schneidet bei einigen Klassen besser, bei einigen Klassen schlechter als K^{Triv}_{Roh} ab (randomisierter Split).

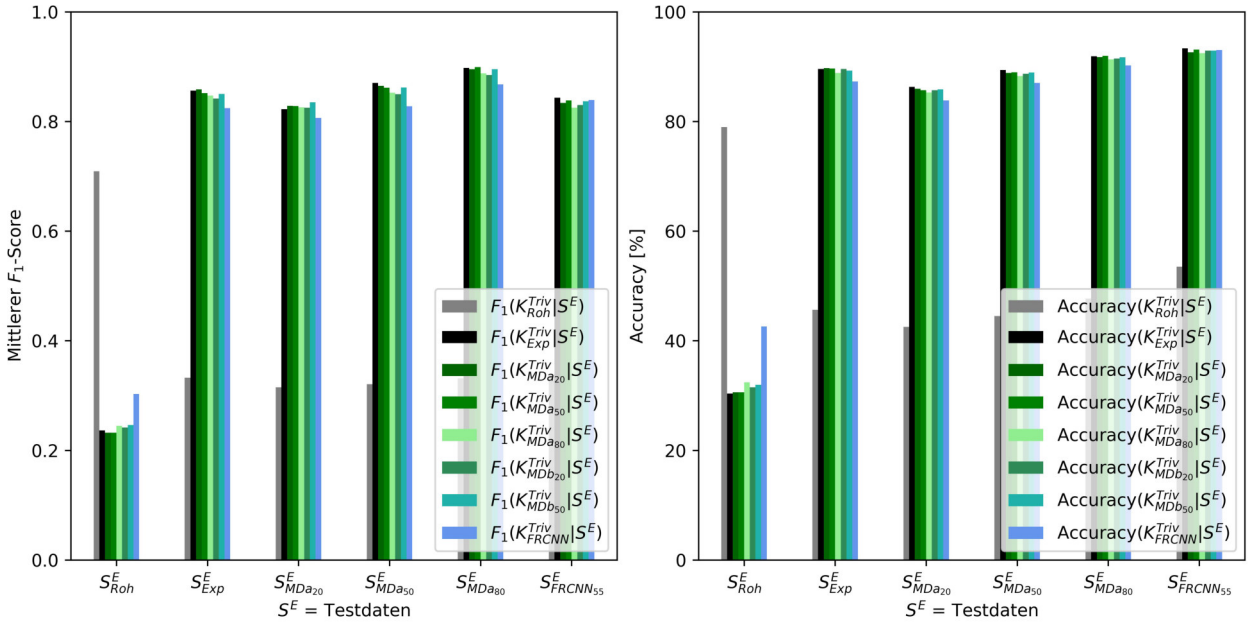

Abbildung 9.3: F_1-Score und Accuracy für Klassifikatoren K^{Triv} mit 17 Klassen mit Split nach Stationen gegenüber verschiedenen Testdatensätzen S^E

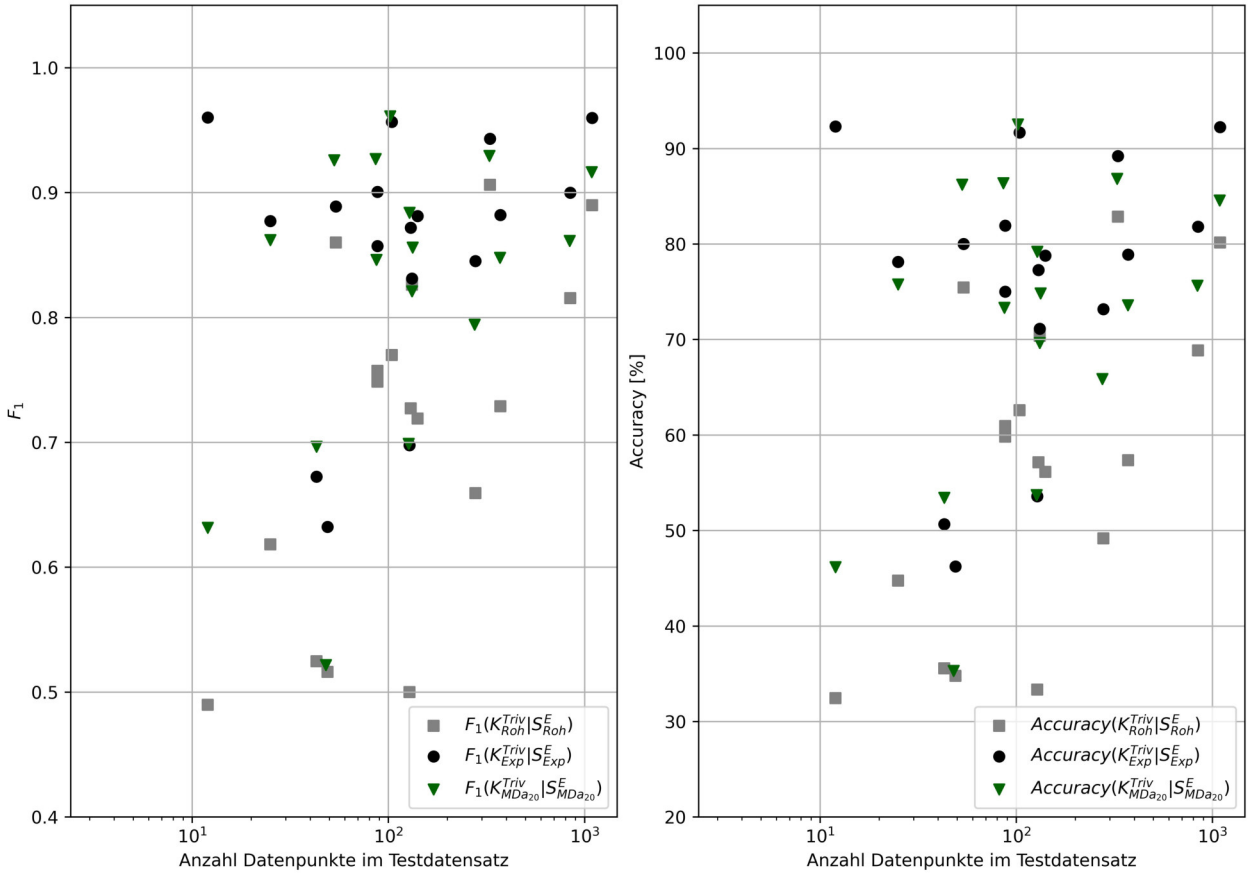

Abbildung 9.4: F_1-Scores und Accuracies der drei besten Klassifikatoren K^{Triv} für 17 Klassen anhand der Klassengrößen

Literatur

[Ame11] American Association for the Advancement of Science. *Staggering numbers of undiscovered species*. 2024-03-11. URL: https://www.aaas.org/staggering-numbers-undiscovered-species (besucht am 11. 03. 2024).

[Bál+18] Miklós Bálint u. a. „Accuracy, limitations and cost efficiency of eDNA-based community survey in tropical frogs“. In: *Molecular Ecology Resources* 18.6 (2018), S. 1415–1426. ISSN: 1755-0998. DOI: 10.1111/1755-0998.12934.

[Bee+21] Sara Beery u. a. *The iWildCam 2021 Competition Dataset*. FGVC8 Workshop at CVPR 2021. 2021. URL: http://arxiv.org/pdf/2105.03494.

[Bee21] Sara Beery. „Scaling Biodiversity Monitoring for the Data Age“. In: *XRDS: Crossroads, The ACM Magazine for Students* 27.4 (2021), S. 14–18. ISSN: 1528-4972. DOI: 10.1145/3466857.

[BLB05] Dominique Brossard, Bruce Lewenstein und Rick Bonney. „Scientific knowledge and attitude change: The impact of a citizen science project“. In: *International Journal of Science Education* 27.9 (2005), S. 1099–1121. ISSN: 0950-0693. DOI: 10.1080/09500690500069483.

[BMK14] Jimmy Ba, Volodymyr Mnih und Koray Kavukcuoglu. *Multiple Object Recognition with Visual Attention*. 2014. URL: http://arxiv.org/pdf/1412.7755.pdf.

[BMY19] Sara Beery, Dan Morris und Siyu Yang. *Efficient Pipeline for Camera Trap Image Review*. From the Data Mining and AI for Conservation Workshop at KDD19. 2019. URL: http://arxiv.org/pdf/1907.06772.pdf.

[Bro+19] Eduardo Sonnewend Brondízio u. a., Hrsg. *The global assessment report of the intergovernmental science-policy platform on biodiversity and ecosystem services*. Bonn: Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services (IPBES), 2019. 1082 S. ISBN: 9783947851201.

[BvP18] Sara Beery, Grant van Horn und Pietro Perona. „Recognition in terra incognita“. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, S. 456–473.

[Car+20] Christin Carl u. a. „Automated detection of European wild mammal species in camera trap images with an existing and pre-trained computer vision model“. En;en. In: *European Journal of Wildlife Research* 66.4 (2020). PII: 1404, S. 1–7. ISSN: 1612-4642. DOI: 10.1007/s10344-020-01404-y. URL: https://link.springer.com/article/10.1007/s10344-020-01404-y.

[Che+] Wenjie Chen u. a. „Learning How to Zoom In: Weakly Supervised ROI-Based-DAM for Fine-Grained Visual Classification“. In: Bd. 12892, S. 118–130. DOI: 10.1007/978-3-030-86340-1_10.

[Che+14] Guobin Chen u. a. „Deep convolutional neural network based species recognition for wild animal monitoring“. In: *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014. DOI: 10.1109/icip.2014.7025172.

[Che+19] Ruilong Chen u. a. „Wildlife surveillance using deep learning methods“. In: *Ecology and evolution* 9.17 (2019), S. 9453–9466. ISSN: 2045-7758. DOI: 10.1002/ece3.5410. eprint: 31534668.

[Cho10] François Chollet. *Xception: Deep Learning with Depthwise Separable Convolutions*. 2016-07-10. URL: https://arxiv.org/pdf/1610.02357.

[Cla+23] Laurence A. Clarfeld u. a. „Evaluating a tandem human-machine approach to labelling of wildlife in remote camera monitoring“. In: *Ecological Informatics* 77 (2023). PII: S1574954123002868, S. 102257. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2023.102257.

[Cle11] Elsa E. Cleland. *Biodiversity and Ecosystem Stability*. Version Nature Education. Nature. 2011. URL: https://www.nature.com/scitable/knowledge/library/biodiversity-and-ecosystem-stability-17059965/ (besucht am 20. 02. 2024).

[CN08] Coursera und Andrew Y. Ng. *Train / Dev / Test sets - Practical Aspects of Deep Learning | Coursera*. 2024-03-08. URL: https://www.coursera.org/lecture/deep-neural-network/train-dev-test-sets-cxG1s (besucht am 08. 03. 2024).

[COC11] COCO. *COCO - Common Objects in Context*. 2023-09-11. URL: https://cocodataset.org/#detection-eval (besucht am 03. 12. 2024).

[Col+23] Elijah Cole u. a. „Spatial Implicit Neural Representations for Global-Scale Species Mapping“. In: *International Conference on Machine Learning* (2023), S. 6320–6342. URL: https://proceedings.mlr.press/v202/cole23a.html.

[Cos20] Mark John Costello. „Taxonomy as the key to life“. In: *Megataxa* 1.2 (2020). ISSN: 2703-3082. DOI: 10.11646/megataxa.1.2.1.

[Cul+03] Phil F. Culverhouse u. a. „Do experts make mistakes? A comparison of human and machine indentification of dinoflagellates“. In: *Marine ecology progress series* 247 (2003), S. 17–25.

[Dau23] Sonja N. K. Daum. „Eine Praxis der (technischen) Fürsorge. Ästhetik und Biodiversitätsschutz“. In: *Die Ästhetik der Technowissenschaften des 21. Jahrhunderts*. Hrsg. von Marco Tamborini. 2023, S. 229–245. ISBN: 978-3-534-40790-3.

[Den+09] J. Deng u. a. „ImageNet: A Large-Scale Hierarchical Image Database“. In: *CVPR09*. 2009.

[Dia+03] Qishuai Diao u. a. *MetaFormer: A Unified Meta Framework for Fine-Grained Recognition*. 2022-05-03. URL: http://arxiv.org/pdf/2203.02751.pdf.

[DMG17] Marine Desprez, Vincent Miele und Olivier Gimenez. *Nine tips for ecologists using machine learning*. 2023-05-17. URL: http://arxiv.org/pdf/2305.10472.

[Dos+22] Alexey Dosovitskiy u. a. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2020-10-22. URL: http://arxiv.org/pdf/2010.11929.pdf.

[FBB22] Mitchell Fennell, Christopher Beirne und A. Cole Burton. „Use of object detection in camera trap image identification: Assessing a method to rapidly and accurately classify human and animal detections for research and application in recreation ecology“. In: *Global Ecology and Conservation* 35 (2022), e02104. ISSN: 23519894. DOI: 10.1016/j.gecco.2022.e02104. URL: https://www.sciencedirect.com/science/article/pii/S2351989422001068.

[FGV31] FGVC. *FGVC.ORG*. 2019-5-31. URL: https://www.fgvc.org/papers/ (besucht am 20. 02. 2024).

[Gir+13] Ross Girshick u. a. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 11/11/2013. URL: http://arxiv.org/pdf/1311.2524.

[Gir15] Ross Girshick. *Fast R-CNN*. To appear in ICCV 2015. 4/30/2015. URL: http://arxiv.org/pdf/1504.08083.pdf.

[GO04] Kevin J. Gaston und Mark A. O'Neill. „Automated species identification: why not?“ In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 359.1444 (2004), S. 655–667.

[Gom+21] Francisco Gomez-Donoso u. a. „Accurate Multilevel Classification for Wildlife Images“. In: *Computational intelligence and neuroscience* 2021 (2021), S. 6690590. DOI: 10.1155/2021/6690590. eprint: 33868399. URL: https://www.hindawi.com/journals/cin/2021/6690590/.

[GSW92] Volker Grimm, Eric Schmidt und Christian Wissel. „On the application of stability concepts in ecology“. In: *Ecological Modelling* 63.1-4 (1992). PII: 030438009290067O, S. 143–161. ISSN: 03043800. DOI: 10.1016/0304-3800(92)90067-O. URL: https://www.sciencedirect.com/science/article/pii/030438009290067O.

[Han+20] Oskar L. P. Hansen u. a. „Species-level image classification with convolutional neural network enables insect identification from habitus images“. In: *Ecology and evolution* 10.2 (2020), S. 737–747. ISSN: 2045-7758. DOI: 10.1002/ece3.5921. eprint: 32015839.

[Hao+06] Hao Zhang u. a. „SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition“. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*. IEEE, 2006. DOI: 10.1109/cvpr.2006.301.

[He+10] Kaiming He u. a. *Deep Residual Learning for Image Recognition*. Tech report. 2015-12-10. URL: http://arxiv.org/pdf/1512.03385.pdf.

[He+21] Ju He u. a. *TransFG: A Transformer Architecture for Fine-grained Recognition*. 3/14/2021. URL: http://arxiv.org/pdf/2103.07976.pdf.

[HGB18] S. Hecker, L. Garbe und A. Bonn. *The European citizen science landscape – a snapshot*. eng. Hrsg. von S. Hecker u. a. Unter Mitarb. von S. Hecker u. a. Citizen Science - Innovation in Open Science, Society and Policy. London: UCL Press, 2018. 190-200. URL: https://discovery.ucl.ac.uk/id/eprint/10066022/.

[HL06] Fu Jie Huang und Yann LeCun. „Large-scale learning with svm and convolutional for generic object categorization“. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Bd. 1. IEEE. 2006, S. 284–291.

[iNa06] iNaturalist. *A larger experiment to learn about the accuracy of iNaturalist observations · iNaturalist*. 2024-04-06. URL: https://www.inaturalist.org/blog/91400-a-larger-experiment-to-learn-about-the-accuracy-of-inaturalist-observations (besucht am 06. 04. 2024).

[iNa23] iNaturalist. *A New Vision Model! · iNaturalist*. 7/21/2023. URL: https://www.inaturalist.org/blog/31806-a-new-vision-model (besucht am 21. 07. 2023).

[iNa24] iNaturalist. *inaturalist/inatVisionTraining*. 3/12/2024. URL: https://github.com/inaturalist/inatVisionTraining (besucht am 12. 03. 2024).

[Jan+11] Martin Jansen u. a. „Integrative inventory of Bolivia's lowland anurans reveals hidden diversity". en. In: *Zoologica Scripta* 40.6 (2011), S. 567–583. ISSN: 1463-6409. DOI: 10.1111/j.1463-6409.2011.00498.x.

[Jan+20] Martin Jansen u. a. „A camera trapping survey of mammals in the mixed landscape of Bolivia's Chiquitano region with a special focus on the Jaguar". In: *Check List* 16.2 (2020), S. 323–335.

[Jan+24] Martin Jansen u. a. „Engaging Citizen Scientists in Biodiversity Monitoring: Insights from the WildLIVE! Project". In: *Citizen Science: Theory and Practice* 9.1 (2024), S. 6. ISSN: 2057-4991. DOI: 10.5334/cstp.665. URL: https://theoryandpractice.citizenscienceassociation.org/articles/10.5334/cstp.665.

[JGK09] Martin Jansen, Lucindo Gonzales Álvarez und Gunther Köhler. „Article Description of a new species of Xenopholis (Serpentes: Colubridae) from the Cerrado of Bolivia, with comments on Xenopholis scalaris in Bolivia". In: *Zootaxa* 2222.2222 (2009), S. 31. ISSN: 1175-5334. DOI: 10.5281/zenodo.190108. URL: https://www.researchgate.net/profile/martin-jansen-2/publication/232710579_article_description_of_a_new_species_of_xenopholis_serpentes_colubridae_from_the_cerrado_of_bolivia_with_comments_on_xenopholis_scalaris_in_bolivia.

[Jia+19] Licheng Jiao u. a. „A Survey of Deep Learning-Based Object Detection". In: *IEEE Access* 7 (2019), S. 128837–128868. ISSN: 2169-3536. DOI: 10.1109/access.2019.2939201.

[JM06] Joseph L. Mundy und Joseph L. Mundy. „Object Recognition in the Geometric Era: A Retrospective". In: *Toward Category-Level Object Recognition*. Hrsg. von Jean Ponce u. a. Bd. 4170. Lecture Notes in Computer Science. Springer, 2006, S. 3–28. DOI: 10.1007/11957959_1.

[JZ19] Jonathon Byrd und Zachary Lipton. „What is the Effect of Importance Weighting in Deep Learning?" In: *International Conference on Machine Learning* (2019), S. 872–881. URL: https://proceedings.mlr.press/v97/byrd19a.html.

[KGK20] Anirudh Krishen Koul, Siddha Ganju und Meher Kasam. *Practical deep learning for cloud, mobile, and edge. Real-world AI and computer-vision projects using Python, Keras, and TensorFlow*. First edition. Beijing u. a.: O'Reilly, 2020. ISBN: 9781492034865.

[Kir+23] Alexander Kirillov u. a. „Segment Anything“. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2023. DOI: 10.1109/iccv51070.2023.00371.

[Kra+15] Jonathan Krause u. a. „Fine-Grained Recognition Without Part Annotations“. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June. 2015.

[KSH12] Alex Krizhevsky, Ilya Sutskever und Geoffrey Hinton. „ImageNet Classification with Deep Convolutional Neural Networks“. In: *Neural Information Processing Systems* 25 (2012). 01. DOI: 10.1145/3065386.

[Kun18] Werner Kunz. „Wohin steuert die Taxonomie?“ In: *Biologie in unserer Zeit* 48.3 (2018), S. 170–178. ISSN: 0045-205X. DOI: 10.1002/biuz.201810647.

[Lar21] Julia Larson. *Assessing Convolutional Neural Network Animal Classification Models for Practical Applications in Wildlife Conservation*. San Jose State University Library, 2021. DOI: 10.31979/etd.ysr5-th9v.

[LB22] Scott Leorna und Todd Brinkman. „Human vs. machine: Detecting wildlife in camera trap images“. In: *Ecological Informatics* 72 (2022), S. 101876. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2022.101876. URL: https://www.sciencedirect.com/science/article/pii/S1574954122003260.

[Lec+98] Y. Lecun u. a. „Gradient-based learning applied to document recognition“. In: *Proceedings of the IEEE* 86.11 (1998), S. 2278–2324. DOI: 10.1109/5.726791.

[Lee+23] Hoesung Lee u. a. *IPCC, 2023: Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]. IPCC, Geneva, Switzerland*. 2023. DOI: 10.59327/IPCC/AR6-9789291691647.

[Li+17] Zhichao Li u. a. *Dynamic Computational Time for Visual Attention*. 2017. URL: http://arxiv.org/pdf/1703.10332.pdf.

[Lin+14] Tsung-Yi Lin u. a. *Microsoft COCO: Common Objects in Context*. 5/1/2014. URL: http://arxiv.org/pdf/1405.0312.pdf.

[Lis+18] Lisha Li u. a. „Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization“. In: *Journal of Machine Learning Research* 18.185 (2018), S. 1–52. ISSN: 1533-7928. URL: http://jmlr.org/papers/v18/16-558.html.

[Liu+22] Zhuang Liu u. a. „A ConvNet for the 2020s“. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022. DOI: 10.1109/cvpr52688.2022.01167.

[LKF10] Yann LeCun, Koray Kavukcuoglu und Clement Farabet. „Convolutional networks and applications in vision“. In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. 2010, S. 253–256. DOI: 10.1109/ISCAS.2010.5537907.

[Lop+20] Vanesa Lopez-Vazquez u. a. „Video Image Enhancement and Machine Learning Pipeline for Underwater Animal Detection and Classification at Cabled Observatories“. In: *Sensors* 20.3 (2020), S. 726. ISSN: 1424-8220. DOI: 10.3390/s20030726. URL: https://www.mdpi.com/1424-8220/20/3/726.

[LW07] D. Lu und Q. Weng. „A survey of image classification methods and techniques for improving classification performance“. In: *International Journal of Remote Sensing* 28.5 (2007), S. 823–870. ISSN: 0143-1161. DOI: 10.1080/01431160600746456.

[Mar+17] Chloé Martineau u. a. „A survey on image-based insect classification“. en. In: *Pattern Recognition* 65 (2017), S. 273–284. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2016.12.020.

[Men+23] De-Yao Meng u. a. „A method for automatic identification and separation of wildlife images using ensemble learning“. In: *Ecological Informatics* 77 (2023), S. 102262. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2023.102262. URL: https://www.sciencedirect.com/science/article/pii/S1574954123002911.

[Mic03] Microsoft. *CameraTraps/megadetector.md at main · microsoft/CameraTraps*. 2024-04-03. URL: https://github.com/microsoft/CameraTraps/blob/main/megadetector.md (besucht am 03. 04. 2024).

[Mic24] Microsoft. *microsoft/CameraTraps: PyTorch Wildlife: a Collaborative Deep Learning Framework for Conservation*. 3/11/2024. URL: https://github.com/microsoft/cameratraps (besucht am 11. 03. 2024).

[Mie+21] Vincent Miele u. a. „Revisiting animal photo-identification using deep metric learning and network analysis“. In: *Methods in Ecology and Evolution* 12.5 (2021), S. 863–873. DOI: 10.1111/2041-210X.13577.

[Mni+14] Volodymyr Mnih u. a. *Recurrent Models of Visual Attention*. 2014. URL: http://arxiv.org/pdf/1406.6247.pdf.

[MQ21] Mingxing Tan und Quoc Le. „EfficientNetV2: Smaller Models and Faster Training“. In: *International Conference on Machine Learning* (2021), S. 10096–10106. URL: http://proceedings.mlr.press/v139/tan21a.html.

[MRA20] Roweida Mohammed, Jumanah Rawashdeh und Malak Abdullah. „Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results“. In: *2020 11th International Conference on Information and Communication Systems (ICICS)*. 2020, S. 243–248. DOI: 10.1109/ICICS49469.2020.239556.

[Nag+11] Jawad Nagi u. a. „Max-pooling convolutional neural networks for vision-based hand gesture recognition“. In: *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. 2011, S. 342–347. DOI: 10.1109/ICSIPA.2011.6144164.

[Ng04] Andrew Ng. „Feature selection, L 1 vs. L 2 regularization, and rotational invariance“. In: *Proceedings of the Twenty-First International Conference on Machine Learning* (2004). 09. DOI: 10.1145/1015330.1015435.

[NIS24] NIST. *Shannon Diversity Index*. 2/5/2024. URL: https://www.itl.nist.gov/div898/software/dataplot/refman2/auxillar/shannon.htm (besucht am 25. 02. 2024).

[Nor+18] Mohammad Sadegh Norouzzadeh u. a. „Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning“. In: *Proceedings of the National Academy of Sciences of the United States of America* 115.25 (2018), E5716–E5725. DOI: 10.1073/pnas.1719367115. eprint: 29871948.

[Nor+23] Danielle L. Norman u. a. „Can CNN–based species classification generalise across variation in habitat within a camera trap survey?“ In: *Methods in Ecology and Evolution* 14.1 (2023), S. 242–251. ISSN: 2041-210X. DOI: 10.1111/2041-210X.14031. URL: https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/2041-210X.14031.

[NZ08] Maria-Elena Nilsback und Andrew Zisserman. „Automated Flower Classification over a Large Number of Classes“. In: *Indian Conference on Computer Vision, Graphics and Image Processing*. 2008.

[OB-8] Juri Opitz und Sebastian Burst. *Macro F1 and Macro F1*. 2019-11-8. URL: http://arxiv.org/pdf/1911.03347.

[OIL17] Jonathan Ortigosa-Hernández, Iñaki Inza und Jose A. Lozano. „Measuring the class-imbalance extent of multi-class problems“. In: *Pattern Recognition Letters* 98 (2017), S. 32–38. ISSN: 01678655. DOI: 10.1016/j.patrec.2017.08.002. URL: https://www.sciencedirect.com/science/article/pii/S016786551730257X.

[OMa+] Niall O'Mahony u. a. „Deep Learning vs. Traditional Computer Vision“. en. In: S. 128–144. DOI: 10.1007/978-3-030-17795-9_10. URL: https://link.springer.com/chapter/10.1007/978-3-030-17795-9_10.

[OMa+19] Tom O'Malley u. a. *KerasTuner*. https://github.com/keras-team/keras-tuner. 2019.

[One21] One Earth. *Chiquitano Dry Forests | One Earth*. 2024-3-21. URL: https://www.oneearth.org/ecoregions/chiquitano-dry-forests/ (besucht am 25. 03. 2024).

[Pap24] Papers with Code. *Papers with Code - CUB-200-2011 Benchmark (Fine-Grained Image Classification)*. 2024. URL: https://paperswithcode.com/sota/fine-grained-image-classification-on-cub-200 (besucht am 04. 03. 2024).

[Par+] Jason Parham u. a. „An Animal Detection Pipeline for Identification“. In: S. 1075–1083. DOI: 10.1109/WACV.2018.00123.

[PFK23] Mohsen Pirizadeh, Hadi Farahani und Saeed Reza Kheradpisheh. „Imbalance factor: a simple new scale for measuring inter-class imbalance extent in classification problems“. En;en. In: *Knowledge and Information Systems* 65.10 (2023). PII: 1881, S. 4157–4183. ISSN: 0219-1377. DOI: 10.1007/s10115-023-01881-y. URL: https://link.springer.com/article/10.1007/s10115-023-01881-y.

[PKS20] Rita Pucci, Vincent J. Kalkman und Dan Stowell. *Comparison between transformers and convolutional models for fine-grained classification of insects*. 2023-07-20. URL: http://arxiv.org/pdf/2307.11112v1.

[PLW02] Leo L. Pipino, Yang W. Lee und Richard Y. Wang. „Data quality assessment“. In: *Communications of the ACM* 45.4 (2002), S. 211–218.

[PND20] Rafael Padilla, Sergio L. Netto und Eduardo A. B. Da Silva. „A survey on performance metrics for object-detection algorithms“. In: *2020 international conference on systems, signals and image processing (IWSSIP)*. IEEE. 2020, S. 237–242.

[Pre02] Lutz Prechelt. „Early stopping-but when?“ In: *Neural Networks: Tricks of the trade*. Springer, 2002, S. 55–69.

[Rec+] Adria Recasens u. a. „Learning to Zoom: a Saliency-Based Sampling Layer for Neural Networks“. In: S. 51–66. URL: https://openaccess.thecvf.com/content_ECCV_2018/html/Adria_Recasens_Learning_to_Zoom_ECCV_2018_paper.html.

[Red+15] Joseph Redmon u. a. *You Only Look Once: Unified, Real-Time Object Detection*. 2015. URL: http://arxiv.org/pdf/1506.02640.pdf.

[Ren+15] Shaoqing Ren u. a. „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks“. In: *Advances in Neural Information Processing Systems* 28 (2015).

[Rez+19] Hamid Rezatofighi u. a. „Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression“. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, S. 658–666. URL: https://openaccess.thecvf.com/content_CVPR_2019/html/Rezatofighi_Generalized_Intersection_Over_Union_A_Metric_and_a_Loss_for_CVPR_2019_paper.html.

[Rig+23] Noa Rigoudy u. a. „The DeepFaune initiative: a collaborative effort towards the automatic identification of European fauna in camera trap images“. In: *European Journal of Wildlife Research* 69.6 (2023), S. 1–12. ISSN: 1612-4642. DOI: 10.1007/s10344-023-01742-7. URL: https://link.springer.com/article/10.1007/s10344-023-01742-7.

[Rob16] Martin J Robbins. „Does an AI need to make love to Rembrandt's girlfriend to make art?“ In: *The Guardian* (6. Mai 2016). URL: https://www.theguardian.com/science/2016/may/06/does-an-ai-need-to-make-love-to-rembrandts-girlfriend-to-make-art (besucht am 04. 03. 2024).

[Rom+19] Alfredo Romero-Muñoz u. a. „Fires scorching Bolivia's Chiquitano forest“. EN. In: *Science* (2019). DOI: 10.1126/science.aaz7264.

[Ros16] Adrian Rosebrock. *Intersection over Union (IoU) for object detection - PyImageSearch*. 2016. URL: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/ (besucht am 25. 02. 2024).

[RP14] Hauke Riesch und Clive Potter. „Citizen science as seen by scientists: Methodological, epistemological and ethical dimensions“. In: *Public understanding of science (Bristol, England)* 23.1 (2014). Journal Article Research Support, Non-U.S. Gov't, S. 107–120. DOI: 10.1177/0963662513497324. eprint: 23982281.

[SB11] Stephen O'Hara und Bruce A. Draper. *Introduction to the Bag of Features Paradigm for Image Classificationand Retrieval*. 2011. URL: https://www.researchgate.net/publication/48190777_Introduction_to_the_Bag_of_Features_Paradigm_for_Image_Classificationand_Retrieval.

[Sil09] Jonathan Silvertown. „A new dawn for citizen science“. In: *Trends in Ecology & Evolution* 24.9 (2009), S. 467–471. ISSN: 0169-5347. DOI: 10.1016/j.tree.2009.03.017.

[SJM18] Neha Sharma, Vibhor Jain und Anju Mishra. „An Analysis Of Convolutional Neural Networks For Image Classification“. In: *Procedia Computer Science* 132 (2018). PII: S1877050918309335, S. 377–384. ISSN: 18770509. DOI: 10.1016/j.procs.2018.05.198. URL: https://www.sciencedirect.com/science/article/pii/S1877050918309335.

[SK19] Connor Shorten und Taghi M. Khoshgoftaar. „A survey on Image Data Augmentation for Deep Learning“. In: *Journal of Big Data* 6.1 (2019), S. 1–48. DOI: 10.1186/s40537-019-0197-0. URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0.

[Sri+14] Nitish Srivastava u. a. „Dropout: a simple way to prevent neural networks from overfitting“. In: *The journal of machine learning research* 15.1 (2014), S. 1929–1958.

[Sta20] Stanford Vision Lab. *ImageNet*. 2020. URL: https://image-net.org/challenges/LSVRC/ (besucht am 04. 03. 2024).

[Süß21] Vanessa Süßle. *Individual identification of patterned solitary species based on unlabeled video data*. 2021.

[Swa+15] Alexandra Swanson u. a. „Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna“. En;en. In: *Scientific Data* 2.1 (2015), S. 1–14. ISSN: 2052-4463. DOI: 10.1038/sdata.2015.26. URL: https://www.nature.com/articles/sdata201526.

[SZ04] Karen Simonyan und Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014-09-04. URL: http://arxiv.org/pdf/1409.1556.pdf.

[Sze+15] Christian Szegedy u. a. „Going deeper with convolutions“. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015. DOI: 10.1109/cvpr.2015.7298594.

[Sze+16] Christian Szegedy u. a. „Rethinking the Inception Architecture for Computer Vision“. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. DOI: 10.1109/cvpr.2016.308.

[Sze22] Richard Szeliski. *Computer Vision. Algorithms and Applications*. en. Springer Nature, 2022. ISBN: 9783030343729.

[Tae+11] Mohammad Reza Taesiri u. a. *ImageNet-Hard: The Hardest Images Remaining from a Study of the Power of Zoom and Spatial Biases in Image Classification*. NeurIPS 2023 Track on Datasets and Benchmarks. 2023-4-11. URL: https://arxiv.org/pdf/2304.05538.pdf.

[Tel11] Anders Telenius. „Biodiversity information goes public: GBIF at your service“. In: *Nordic Journal of Botany* 29.3 (2011), S. 378–381. ISSN: 0107-055X. DOI: 10.1111/j.1756-1051.2011.01167.x.

[TS10] Lisa Torrey und Jude Shavlik. „Transfer Learning“. In: *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. IGI Global, 2010, S. 242–264. DOI: 10.4018/978-1-60566-766-9.ch011. URL: https://www.igi-global.com/chapter/transfer-learning/36988.

[Tui+22] Devis Tuia u. a. „Perspectives in machine learning for wildlife conservation“. In: *Nature communications* 13.1 (2022), S. 792. DOI: 10.1038/s41467-022-27980-y. eprint: 35140206. URL: https://www.nature.com/articles/s41467-022-27980-y.

[Uij+13] J. R. R. Uijlings u. a. „Selective Search for Object Recognition“. In: *International Journal of Computer Vision* 104.2 (2013), S. 154–171. ISSN: 1573-1405. DOI: 10.1007/s11263-013-0620-5. URL: https://link.springer.com/article/10.1007/s11263-013-0620-5.

[van+20] Grant van Horn u. a. *The iNaturalist Species Classification and Detection Dataset*. CVPR 2018. 2017-7-20. URL: https://arxiv.org/pdf/1707.06642.

[Van79] CJ Van Rijsbergen. *Information retrieval*. Butterworth-Heinemann, 1979.

[Wan+] Dequan Wang u. a. „Multiple Granularity Descriptors for Fine-Grained Categorization“. In: S. 2399–2406. DOI: 10.1109/ICCV.2015.276.

[Wan23] Huiyi Wang. *When Segment and Track Anything Meets Wildlife Videos*. 2023.

[Wei+16] Yunchao Wei u. a. „HCP: A Flexible CNN Framework for Multi-label Image Classification“. In: *IEEE transactions on pattern analysis and machine intelligence* 38.9 (2016), S. 1901–1907. DOI: 10.1109/TPAMI.2015.2491929. eprint: 26513778. URL: https://pubmed.ncbi.nlm.nih.gov/26513778/.

[Wel+10] Peter Welinder u. a. „Caltech-UCSD Birds 200“. In: (2010). 09.

[Wil14] WildLIVE! *Zusammengefasst | WildLIVE!* 2023-08-14. URL: https://wildlive.sgn.one/de/ueber-das-projekt-2/ (besucht am 06. 02. 2024).

[Wil24] WildEye. *MegaDetector Version 5 | WildEye*. 3/25/2024. URL: https://wildeyeconservation.org/megadetector-version-5/ (besucht am 25. 03. 2024).

[WJ02] WildLIVE! Und Martin Jansen. *WildLIVE! Entdecke die wilden Tiere Boliviens*. 2023-08-02. URL: https://wildlive.sgn.one/de/ (besucht am 02. 08. 2023).

[WN14] Rüdiger Wittig und Manfred Niekisch. *Biodiversität: Grundlagen, Gefährdung, Schutz*. ger. Wittig, Rüdiger (VerfasserIn) Niekisch, Manfred (VerfasserIn). Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. 585 S. ISBN: 9783642546945. URL: http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1545411.

[Yam+90] Kouichi Yamaguchi u. a. „A neural network for speaker-independent isolated word recognition“. In: *First International Conference on Spoken Language Processing (ICSLP 1990)* (ISCA). ISCA: ISCA, 1990. DOI: 10.21437/icslp.1990-282.

[Yan+18] Ze Yang u. a. *Learning to Navigate for Fine-grained Classification*. Accepted by ECCV 2018. 9/2/2018. URL: http://arxiv.org/pdf/1809.00287v1.

[ZGX20] Rui Zhu, Yiwen Guo und Jing-Hao Xue. „Adjusting the imbalance ratio by the dimensionality of imbalanced data“. In: *Pattern Recognition Letters* 133 (2020). PII: S0167865520300829, S. 217–223. ISSN: 01678655. DOI: 10.1016/j.patrec.2020.03.004. URL: https://www.sciencedirect.com/science/article/pii/S0167865520300829.

[Zha+09] Yifan Zhang u. a. *Deep Long-Tailed Learning: A Survey*. Published in IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021-10-09. URL: http://arxiv.org/pdf/2110.04596.pdf.

[Zha+21] Chang-Bin Zhang u. a. „Delving Deep Into Label Smoothing“. In: *IEEE Transactions on Image Processing* 30 (2021), S. 5984–5996. ISSN: 1057-7149. DOI: 10.1109/TIP.2021.3089942. URL: http://arxiv.org/pdf/2011.12562.

[Zhe+19] Heliang Zheng u. a. *Looking for the Devil in the Details: Learning Trilinear Attention Sampling Network for Fine-grained Image Recognition*. 3/14/2019. URL: https://arxiv.org/pdf/1903.06150.pdf.

[ZL17] Barret Zoph und Quoc V. Le. *Neural Architecture Search with Reinforcement Learning*. 2017.

[Zop+21] Barret Zoph u. a. *Learning Transferable Architectures for Scalable Image Recognition*. 2017-07-21. URL: http://arxiv.org/pdf/1707.07012.

[Zou+23] Zhengxia Zou u. a. „Object Detection in 20 Years: A Survey“. In: *Proceedings of the IEEE* 111.3 (2023), S. 257–276. ISSN: 00189219. DOI: 10.1109/jproc.2023.3238524.

Inhalt der DVD

Datei	**Inhalt**
/code/*	Quellcode der Experimente
/evaluation/*	Evaluationsergebnisse von Versuch 2 und Versuch 3
/models/*	Keras-Modelle von Versuch 2 & 3
/poster/*	Poster für Ankündigung des Kolloquiums
/thesis/Thesis-Koehler-Optimierung-von-Bildklassifikatoren-Biodiversitaet-Citizen-Science.pdf	Diese Arbeit im PDF-Format
/thesis/abbildungen/*	Abbildungen aus der Thesis
/tuning/*	Tuning-Ergebnisse des Fine-Tunings in Versuch 2 (KerasTuner)