

Bachelorarbeit

AI-Recycling: Objektklassifizierung im Recyclinghof Hagenholz

Institution: Zurich University of Applied Sciences (ZHAW)
School of Management and Law

Betreuerin: Elena Gavagnin

Eingereicht am: 09.06.2021

Verfasser: Lukas Roth



Management Summary

Durch den kontinuierlich wachsenden Rohstoffverbrauch weltweit, ist die effiziente Nutzung und Wiederverwendung von Werkstoffen von zentraler Bedeutung. Im Recyclinghof Hagenholz (RH) werden in 25 Containern Materialien getrennt gesammelt. Die Zuordnung ist, durch die Vielfalt an Sammelkategorien, für viele Kunden schwierig und immer wieder landen Objekte in den falschen Containern.

Diese Arbeit zeigt eine Lösung auf, wie mittels eines Machine Learning Modells ein Objektklassifikator erstellt werden kann, welcher anhand eines Fotos des Recyclingobjektes, dessen Klassifizierung der Sammelkategorie liefern kann. Das Modell kommt in einer Mobile-App zum Einsatz und soll dem Kunden des RH die Unterstützung bei der korrekten Zuordnung des mitgebrachten Materials bieten.

Das methodische Vorgehen besteht aus drei Teilen: der Datenerhebung für das Image Dataset, der Modellentwicklung und der Entwicklung des Prototyps. Die Datenerhebung umfasst das Sammeln eigener Fotos aus dem RH, die Evaluierung öffentlicher Datasets sowie die Methode des Image-Scraping. Bei der Modellentwicklung wurde ein Modell von Grund auf, sowie zwei Modelle mittels der Retraining-Methode, auf Basis der vortrainierten Modelle mit MobileNet bzw. EfficientNet entwickelt. Validiert wurden die Modelle mittels Out-Of-Sample (OOS) und K-Fold Cross Validation (KFCV). Das Retrained MobileNetV3 Model (RMM) wurde in die entwickelte Android App integriert und abschliessend im RH einem Feldtest unterzogen.

Das Dataset besteht aus 24 Klassen, aus jeweils ca. 600 Bildern. Die darauf trainierten Modelle haben wie folgt abgeschnitten: Das einfache Modell konnte im OOS 46.8, bzw. 46.2 Prozent Genauigkeit in der KFCV erreichen. Das RMM konnte eine Genauigkeit von 85.9 (OOS), bzw. 88.34 Prozent in der KFCV und mit dem Retrained EfficientNet-B0 Model (REM) liegt die Genauigkeit bei 86.7 in OOS, bzw. 87.6 Prozent in der KFCV. Die Confusion Matrix (CM) zeigte Defizite des RMM in den Klassen Elektro, Mischmaterialien, Metall, sowie Buntmetall. Im Feldtest konnte der Prototyp in 32 von 40 Tests den richtigen Container erkennen, wobei sichtbar wurde, dass in der Klasse Sperrgut die Erkennung sehr schlecht ist und die Kategorien Elektro, Mischmaterialien und Metall, im Vergleich zur CM gut abschnitten.

Diese Arbeit zeigt, dass mittels vortrainierter Modelle und eines verhältnismässig kleinen Datasets, aus vielen Klassen, bereits eine gute Erkennung erreicht werden kann. Es zeigt sich jedoch, dass es schwierig ist, Klassen vollständig abzubilden, welche aus unendlicher Anzahl Objekte bestehen, wie Sperrgut oder Metall. Objekte, welche sich nur durch ihr Material unterscheiden, nicht aber durch Form und Farbe, können kaum durch Image-Classification unterschieden werden.

Durch die zeitliche Limitierung der Arbeit konnte nur ein begrenzter Aufwand in die Entwicklung des Modells, sowie des Datasets gesteckt werden. Durch eine umfangreichere Datenerhebung im RH, könnte eine praxisnähere Datengrundlage geschaffen werden, auf welcher durch weitere Optimierung der Hyperparameter und eines besseren Fine-Tunings im Modell, durchaus noch bessere Resultate möglich wären.

Inhaltsverzeichnis

I.	Abbildungsverzeichnis	V
II.	Tabellenverzeichnis	VI
III.	Abkürzungen	VII
1	Einleitung	1
1.1	Forschungslücke	2
1.2	Zielsetzung	2
1.3	Beitrag der Arbeit	3
1.4	Relevanz	3
1.5	Aufbau der Arbeit	4
2	Related Work	4
3	Theoretische Grundlagen	6
3.1	Supervised Learning	7
3.2	Convolutional Neural Network	7
3.2.1	Convolutional Layer	8
3.2.2	Pooling Layer	8
3.2.3	Fully Connected Layer	9
3.3	CNN-Architekturen	9
3.3.1	MobileNet	10
3.3.2	EfficientNet	10
3.4	Transfer Learning & Retraining	11
3.5	Over- und Underfitting	11
4	Systemanalyse: Recyclinghof Hagenholz	11
4.1	Entsorgungsprozess	12
4.2	Weitere Entsorgung	12
4.3	Lageplan	13
4.4	Containerorganisation	14
4.5	Fehlplatzierung von Objekten	15
4.6	Recyclingklassen	16
5	Methode	17
5.1	Datenerhebung	17
5.1.1	Bilder erstellen	17
5.1.2	Vorhandene Datasets	19
5.1.3	Image Scraping	20
5.2	Datenaufbereitung	21
5.2.1	Duplikate	21
5.2.2	Qualität	21

5.2.3	Vertical Flip.....	22
5.2.4	Labeling.....	22
5.2.5	Zusammenführen der Daten	22
5.3	Model Evaluation & Architekturwahl	22
5.3.1	Model from Scratch	22
5.3.2	Retrained Models.....	23
5.4	Model Training.....	25
5.4.1	Augmentation	26
5.4.2	Fine-Tuning	27
5.4.3	Hyperparameter	27
5.5	Validierung.....	28
5.5.1	Out-Of-Sample Test.....	29
5.5.2	Confusion Matrix.....	29
5.5.3	K-Fold Cross Validation.....	29
5.6	Mobile App Entwicklung	30
5.7	Feldtest.....	31
5.8	Technologie-Stack.....	32
6	Resultate.....	33
6.1	Image-Dataset.....	33
6.2	ML-Modell	35
6.2.1	Einfaches Modell (From Scratch).....	35
6.2.2	Retrained Model MobileNetV3	36
6.2.3	Retrained Model EfficientNet-B0.....	41
6.3	Mobile App.....	43
6.4	Feldtest.....	44
7	Diskussion.....	45
8	Konklusion	46
IV.	Literaturverzeichnis	48
V.	Anhang	54
A.	Container in Recyclinghof Hagenholz.....	54
B.	Verwendete Suchbegriffe im Image-Scraping.....	60
C.	Details K-Fold Cross Validation mit «Model from Scratch»	63
D.	Vortest	64
E.	Feldtest: Testprotokoll.....	65
F.	Feldtest: Resultate als Screenshots	67

I. Abbildungsverzeichnis

Abbildung 1: Einfache CNN Architektur.....	8
Abbildung 2: Lageplan Recyclinghof Hagenholz.....	13
Abbildung 3: Containerorganisation.....	14
Abbildung 4: Schwerpunkte der Arbeit.....	17
Abbildung 5: Stichprobe der angefertigten Bilder.....	18
Abbildung 6: Architektur des einfachen Modells.....	23
Abbildung 7: Retrained Model Architektur auf Basis von EfficientNet-B0.....	24
Abbildung 8: Retrained Model Architektur auf Basis von MobileNetV3.....	25
Abbildung 9: Vorgehen in Model Training.....	25
Abbildung 10: Augmentation angewandt auf ein Beispielbild vom Trainingsdataset.....	26
Abbildung 11: Training & Validation Datenzusammensetzung mit K-Fold Cross Validation.....	30
Abbildung 12: Verzeichnisstruktur mit den fünf Datasets.....	30
Abbildung 13: Übersicht der gesammelten Bilder.....	34
Abbildung 14: Herkunft der Bilder im Dataset.....	34
Abbildung 15: Accuracy in Out-Of-Sample Test.....	35
Abbildung 16: Loss in Out-Of-Sample Test mit dem einfachen Modell.....	35
Abbildung 17: Accuracy in Out-Of-Sample Test mit dem Retrained MobileNetV3 Modell.....	37
Abbildung 18: Loss in Out-Of-Sample Test mit dem Retrained MobileNetV3 Modell.....	37
Abbildung 19: Confusion Matrix anhand des Testdatasets im Out-Of-Sample Test.....	39
Abbildung 20: Accuracy in K-Fold Cross Validation mit dem Retrained MobileNetV3.....	40
Abbildung 21: Loss in K-Fold Cross Validation mit dem Retrained MobileNetV3 Modell.....	40
Abbildung 22: Accuracy in Out-Of-Sample Test mit dem Retrained EfficientNet-B0 Model...	41
Abbildung 23: Loss in Out-Of-Sample Test mit dem Retrained EfficientNet-B0 Model.....	41
Abbildung 24: Accuracy in K-Fold Cross Validation mit dem Retrained EfficientNet-B0.....	42
Abbildung 25: Loss in K-Fold Cross Validation mit dem Retrained EfficientNet-B0 Model...	42
Abbildung 26: Prototyp Main-View.....	43
Abbildung 27: Prototyp Result-View.....	43

II. Tabellenverzeichnis

Tabelle 1: Recyclingklassen im Recyclinghof Hagenholz.....	16
Tabelle 2: Accuracy und Loss Resultate im Out-Of-Sample Test mit dem Einfachen Modell .	36
Tabelle 3: K-Fold Cross Validation Resultate im einfachen Modell	36
Tabelle 4: Dropout Evaluation	36
Tabelle 5: L2-Regularization Evaluation	37
Tabelle 6:Accuracy und Loss Resultate im Out-Of-Sample	38
Tabelle 7: Resultate in K-Fold Cross Validation mit dem Retrained MobileNetV3 Modell	41
Tabelle 8:Resultate in Out-Of-Sample Test (Retrained EfficientNet-B0 Model).....	42
Tabelle 9:Resultate der K-Fold Cross Validation mit dem Retrained EfficientNet-B0 Model..	42

III. Abkürzungen

CM Confusion Matrix

CNN Convolutional Neural Network

IDE Integrated Development Environment

ILSVRC ImageNet Large Scale Visual Recognition Challenge

ML Machine Learning

RH Recyclinghof Hagenholz

SSD Single Shot Detector

SVM Support Vector Machine

VGG Visual Geometry Group

1 Einleitung

Gemäss Bundesamt für Umwelt könnte rund 20 Prozent des Inhaltes der Abfallsäcke in Schweizer Haushalten eingespart werden, wenn Recyclingmaterialien vollständig in den unterschiedlichen Sammelkategorien abgegeben würden (Bundesamt für Umwelt [BAFU], 2014, S. 39). So rechnete das Bundesamt für Umwelt im Jahr 2014 mit einer potenziellen jährlichen Menge von 340'000 Tonnen Abfall, welcher nicht in der Kehrichtverbrennung landen müsste (BAFU, 2014, S. 39).

Die Summe aller Emissionen aus der Abfallverbrennung ergeben rund 6.4 Prozent der Treibhausgasemissionen in der Schweiz (BAFU, 2020, S. 18). Die Möglichkeiten, um die CO₂ Emissionen in der Abfallverbrennung weiter zu senken sind sehr limitiert. Die effektivste Möglichkeit besteht aber darin die Gesamtabfallmenge zu reduzieren (SATW, 2014 zitiert in BAFU, 2016, S. 25). Die Trennung von Wertstoffen und Abfällen und die Wiederverwertung von Rohstoffen sind daher von zentraler Bedeutung für die Umwelt. Die Zunahme der Gesamtbevölkerung und das wirtschaftliche Wachstum werden durch den daraus resultierenden gesteigerten Konsum, als Treiber der kontinuierlich wachsenden Rohstoffverbrauches angesehen (Schweizerischer Bundesrat, 2018, S. 17). Entwicklungsländer belasten daher die Umwelt im Vergleich weniger als wohlhabendere Länder mit hohem Konsumverhalten (Schweizerischer Bundesrat, 2018, S. 20).

Die Schweiz ist eines der führenden Länder in Sache Recycling (BAFU, 2016, S. 29). Trotzdem ist es von Wichtigkeit die Recycling-Prozesse weiter zu optimieren und eine bestmögliche Wiederverwertung von Rohstoffen zu gewährleisten. Das Bundesamt für Umwelt (2014, S. 51) erläutert den Handlungsbedarf folgendermassen:

«Um dieses Reduktionspotenzial zu realisieren, müssen die Logistik und die Infrastruktur der Separatsammlungen kontinuierlich überprüft und verbessert werden».

Doch wo bietet der Prozess des Recyclings und der Separierung von Materialien konkret Verbesserungspotential?

Im Recyclinghof Hagenholz kommen täglich zahlreiche Kunden mit unterschiedlichsten Abfällen und Recyclingmaterialien vorbei, um diese umweltgerecht zu entsorgen. Für den Laien kann die Vielfalt an Recyclingkategorien überwältigend wirken und schnell kommt für viele Kunden die Frage auf, in welchen der zahlreichen Container sie ihre Gegenstände abgeben können. Die gängige Unterscheidung, in die Recyclingkategorien Glas, PET, Karton oder Alu sind den meisten bekannt. Wo aber, sollen die alten Skis oder die ausgediente Mikrowelle abgegeben werden?

Die Mitarbeiter des Recyclinghofs Hagenholz werden mit solchen Fragen tagtäglich konfrontiert und helfen jedem Kunden weiter, damit die abgegebenen Objekte ihren Platz im korrekten Container finden. Schlussendlich kommt es aber trotzdem häufig vor, dass Kunden ihre Materialien

in den falschen Container platzieren. Gerade bei Elektromaterialien ist es kompliziert zu wissen, dass Geräte mit Lithiumbatterien nicht zur Kategorie «Elektro» gehören oder dass ein PC-Monitor einer separaten Kategorie «Bildschirme & TV» angehört, und wiederum alte Röhrenbildschirme in den Sonderabfall müssen. Diese Fehlzweisungen werden heute mit mühsamer Handarbeit nachsortiert.

Die Komplexität der Klassifizierung von Recyclingmaterialien stellt somit sowohl für die Bevölkerung der Stadt Zürich, sowie auch für den Recyclinghof Hagenholz und deren Mitarbeiter ein Hindernis im Recyclingprozess dar.

1.1 Forschungslücke

Objektklassifizierung kommt in unterschiedlichsten Anwendungsbereichen zum Einsatz. Basis für eine gute Erkennungsrate bildet dabei eine umfangreiche Datengrundlage, um visuelle Varianzen der Objekte in Klassen abbilden zu können (Lin et al., 2011, S. 1689). Dies ist gerade auch der Knackpunkt. Meist fehlt die Datengrundlage, um ein Modell auf einen Anwendungsfall trainieren zu können. Ein Image Dataset, welches alle Recyclingkategorien des Recyclinghofs Hagenholz abdeckt, existiert momentan nicht. Open Source Datasets TACO (Proença & Simões, 2020) oder TrashNet (Yang & Thung, 2016) bilden immer nur eine Teilmenge der benötigten Datengrundlage. Sie sind ausserdem nicht auf den Anwendungsfall im Recyclinghof zugeschnitten. Das fängt bereits bei einfachen Kategorien wie Plastik an. Was für bisherige Forschungsprojekte als Plastik klassifiziert wird, wird beispielsweise im Recyclinghof Hagenholz ganz anders deklariert und kann je nach Objekt einer anderen Klasse zugewiesen werden. Zudem sind die vorhandenen Datasets in einem anderen Kontext entstanden. Bilder von Objekten auf der Strasse oder in der Natur wie im TACO-Dataset, entsprechen nicht dem optimalen Environment. Ein Dataset mit Bildern direkt aus dem Recyclinghof Hagenholz, geschossen aus dem Blickwinkel des Benutzers einer Applikation, ist daher eine Neuheit. Ebenso die Problemlösung der Recyclingobjektklassifizierung in Form einer Mobile App für den Anwendungsbereich direkt im Recyclinghof, existiert zum Zeitpunkt dieser Arbeit nicht.

1.2 Zielsetzung

Das Ziel dieses Forschungsprojekts ist, einen Objektklassifikator mittels Machine Learning zu entwickeln, welcher verschiedene Recycling-Objekte anhand von Bildern, den richtigen Sammelkategorien zuweisen kann. Das daraus resultierende Machine Learning Model soll danach in Form eines Prototyps einer Mobile App angewandt werden. Zur Erfüllung der Zielsetzung muss zudem eine Datengrundlage geschaffen werden, um den Objektklassifikator erst trainieren zu können. Basis für das Modell bildet daher eine eigene Bildersammlung der Recyclingobjekte aller zu klassifizierenden Kategorien. Dieses Dataset wird mit Bildern erstellt, welche einerseits aus den

eigens gemachten Fotos aus dem Recyclinghof Hagenholz stammen und andererseits aus dem im Internet gesammelten Bildmaterial.

Mit der App sollte dann in einem ersten Schritt ein Foto eines zu recycelnden Gegenstandes gemacht werden können. Dieses sollte danach durch den Objekt-Klassifikator analysiert werden und in einem nächsten Schritt dem Benutzer eine Rückmeldung geben, welche die richtige Sammelkategorie, bzw. der richtige Sammelcontainer im Recyclinghof Hagenholz ist. Die Anwendung soll darauf ausgelegt sein, innerhalb des Recyclinghofs Hagenholz zum Einsatz zu kommen und soll somit für diesen Kontext optimiert werden.

Im Rahmen der Umsetzung dieser Arbeit stellen sich mehrere Arbeitsfragen, welche behandelt werden sollen:

- Was für eine Erkennungsrate kann erreicht werden?
- Wie viele Bilder werden für eine gute Erkennungsrate benötigt?
- Für welche Kategorien wird es schwierig bzw. einfach Bilder zu bekommen?
- Was ist eine gute Erkennungsrate?
- Welche Abfallkategorien gibt es zu unterscheiden?
- Welche Abfallkategorien sind schwierig, bzw. einfach zu erkennen?
- Kann ein entwickeltes Modell auch im Feldtest gute Resultate erzielen?

Mit dieser App soll die Komplexität der Abfalldifferenzierung vereinfacht und mögliche Fehlplatzierungen von Objekten im Recyclinghof Hagenholz vermieden werden.

1.3 Beitrag der Arbeit

Die in dieser Arbeit entwickelten Artefakte setzen sich folgendermassen zusammen: Einerseits wurde das Recycling-Dataset, bestehend aus den gesammelten Fotos aus dem Recyclinghof Hagenholz, erstellt. Weiter stellt das trainierte Image Classification Model, ein weiteres Artefakt dar. Als drittes Artefakt wurde das Model in eine Mobile App integriert als Prototyp bereitgestellt. Das letzte Artefakt stellt die Auswertung des Praxistests des Prototyps dar.

1.4 Relevanz

Wer sich selbst über die Abfallkategorien informieren möchte, um herauszufinden, zu welcher Kategorie sein Objekt gehört, kann sich einerseits auf der Website der Stadt Zürich (Stadt Zürich o. J. -c) informieren, oder sich eine Übersicht über die Entsorgungskategorien beschaffen, indem man sich die App «Sauberes Zürich» herunterlädt (Stadt Zürich, o. J. -a). Die Informationen sind sehr umfangreich mit einigen Beispielen von Gegenständen und Materialien veranschaulicht. Eine schnelle Möglichkeit, die Sammelkategorie und deren Standort im Recyclinghof Hagenholz eines Gegenstandes herauszufinden, gibt es aber nicht.

Durch den in dieser Arbeit entwickelte Prototyp soll geprüft werden, ob es möglich ist, die Abfalltrennung mittels Objektklassifizierung von Recyclingobjekten im Recyclinghof Hagenholz zu unterstützen, um den Prozess der Abfalltrennung für den Kunden im Recyclinghof Hagenholz zu vereinfachen. Für den Kunden bietet dieser Prototyp dann eine Lösung für die Frage, in welchen Container er sein zu entsorgendes Objekt abgeben muss. Damit könnten einerseits Wartezeiten im Recyclinghof verkürzt und andererseits das Bewusstsein der Bevölkerung für richtiges Abfalltrennen gestärkt werden.

Durch die entwickelte Lösung könnte zudem das Personal des Recyclinghofes Hagenholz entlastet werden. Die Anzahl Hilfeleistungen bei der Zuordnung von Recyclingobjekten durch Mitarbeiter des Recyclinghofes Hagenholz könnten mit Hilfe der App reduziert werden. Von einer verbesserten Klassifizierung von Recyclingmaterialien könnte weiter der Verarbeitungsprozess im Recyclinghof profitieren, da der heutige Aufwand in der Nachsortierung minimiert werden kann, wenn die Kundschaft durch die Hilfe der Mobile App die Objekte korrekt entsorgt.

Diese Arbeit soll einen Beitrag leisten, die Trennung von Werkstoffen und die damit hergehende Rohstoffverwertung in der Stadt Zürich weiter zu optimieren, um so Ressourcen zu schonen und die Umweltbelastung zu reduzieren.

1.5 Aufbau der Arbeit

Die Arbeit ist wie folgt aufgebaut: In den ersten beiden Kapiteln wird der Forschungsstand anhand verwandter Arbeiten aufgezeigt und auf die theoretischen Grundlagen eingegangen. Nachfolgend wird in einer Systemanalyse den Kontext des Recyclinghof Hagenholz aufgezeigt. Im darauffolgenden Kapitel wird das methodische Vorgehen dargelegt. Darin wird einerseits die Datenerhebung, sowie das Erstellen des Datasets und das Training des Machine Learning Modells beschrieben. Ausserdem wird aufgezeigt, wie die Mobile-App umgesetzt wurde. Anhand der Methode wird erläutert, wie vorgegangen wurde, um zu den Resultaten zu gelangen, welche dann im Kapitel Resultate vorgestellt werden. Das Kapitel Resultate stellt die erlangten Ergebnisse der Arbeit vor. Dies beinhaltet das Dataset, die Modelle, sowie der entstandene Prototyp (Mobile-App). Abschliessen werden die Resultate im Kapitel Diskussion nochmals aufgegriffen und im abschliessenden Kapitel Konklusion bewertet.

2 Related Work

Das Konzept, zu erkennen, zu welcher Klasse ein Objekt, beziehungsweise das Bild eines Objektes gehört, ist im Kontext von Machine Learning eine bekannte Aufgabe und nennt sich Image-Classification.

Objekterkennung und Klassifizierung wird in zahlreichen Branchen und Anwendungsgebieten eingesetzt. Gerade im Forschungsgebiet des Autonomous Driving kommt die Objekterkennung

und semantische Segmentierung bei der Ermittlung von Hindernissen und der Analyse der Umgebung zum Einsatz (Fujiyoshi et al., 2019, S. 250). Ein weiteres Anwendungsgebiet von Image Classification bietet die Medizin. Mittels Bildanalyse-Systemen können Pathologen und Pathologinnen in der Diagnose von Brustkrebs unterstützt werden (Spanhol et al., 2016, S. 2560). Dabei unterscheidet das Modell anhand von Aufnahmen von Gewebeproben, ob diese krebsartig oder nicht krebsartig sind (Spanhol et al., 2016, S. 2560). Ein anderes Beispiel stellt die Fehlererkennung in der Fertigung von Solarzellen dar. Su, Chen, Zhu, Liu und Liu (2019) beschreiben wie mit Bilderkennung strukturelle Fehler in der Herstellung von Solarzellen frühzeitig erkannt und somit die Qualitätssicherung garantiert werden kann.

Auch im Zusammenspiel von Abfall und Bildererkennung gibt es bereits einige Projekte. Beispielsweise haben Ruiz, Sánchez, Vélez und Raducanu (2019) verschiedene CNN¹-Architekturen für die automatische Klassifizierung von Müll evaluiert. Basis war das TrashNet Dataset von Yang und Thung (2016). Die besten Resultate konnten Ruiz et al. (2019, S. 430) dabei in ihrem Experiment mit der ResNet-Architektur erzielen, mit einer durchschnittlichen Genauigkeit von 88.66 Prozent.

Das erwähnte TrashNet Dataset entstand in einem weiteren Projekt, in welchem Yang und Thung (2016) versuchten, eine Recyclingkategorie für ein Objekt auf einem Bild zu ermitteln. Die Resultate des dabei entstandenen CNN auf Basis der ResNet-Architektur sind mit einer Genauigkeit von nur 22 Prozent nicht sehr aussagekräftig. Das bei diesem Projekt entstandene Dataset deckt sechs Recyclingklassen ab: glass, paper, cardboard, plastic, metal und trash (Yang & Thung, 2016, S. 2). Insgesamt umfasst es 2400 Bilder, wobei jeweils 400-500 Bilder pro Klasse entstanden sind.

Auf Basis des TrashNet Datasets wurden verschiedene weitere Klassifikatoren erstellt, wie das mit ResNet-50 und SVM² entwickelte Modell von Olugboja und Zenghui (2019) mit einer Erkennungsrate von 87 Prozent. Melinte, Travediu und Dumitriu (2020) befassen sich mit der Optimierung der Leistung von CNN-Objektdetektoren zur Erkennung von Siedlungsabfällen. Sie beschreiben ihre Erkennungsrate als die bisher beste eines Modelles auf Basis des TrashNet Datasets, mit einer Genauigkeit von 97,63 Prozent mit SSD³ bzw. 95,76 Prozent Genauigkeit mit Faster R-CNN⁴.

Kennedy, Kennedy und Knowles (2018) beschreiben das Problem der kleinen Anzahl verfügbarer Daten für das Trainieren eines neuronalen Netzes. Sie sehen das Potential in vortrainierten

¹ Convolutional Neural Network (siehe Kapitel 3.2)

² Support Vector Machine ist ein weiterer Algorithmus zur Objektklassifizierung (Noble, 2006, S. 1565)

³ Single Shot Detector stellt eine Methode zur Erkennung mehrerer Objekte in einem Bild dar (Liu et al., 2016, S. 21)

⁴ Gehört zur Modellfamilie der R-CNN und wird speziell im Bereich der Object Detection eingesetzt (Ren et al., 2016)

Modellen, wobei sie als Basis VGG19⁵, gewichtet auf ImageNet, nutzen und mittels Transfer Learning ihr Modell auf das TrashNet Dataset trainieren (Kennedy et al., 2018, S. 3). Die Erkennungsrate ergab dabei 88.42% (Kennedy et al., 2018, S. 3).

Salimi, Bayu und Wibowo (2018) untersuchen die Möglichkeit, mittels SVM zu identifizieren, ob Objekte entweder organischer, nicht organischer oder kein Abfall sind. Dabei erreichten sie eine Genauigkeit von 82.7 Prozent in der Cross Validation und 73.49 Prozent Genauigkeit im Konfusionsmatrix-Validierungstest (Salimi et al., 2018, S. 383). Genutzt wurde dabei ein eigens erstelltes Dataset.

Proença und Simões (2020) untersuchen die Möglichkeit Abfall in verschiedenen natürlichen Umgebungen wie Wasser, Sand Strassen und weiteren Szenarien zu erkennen und zu klassifizieren. Dafür haben sie ein eigenes Dataset namens TACO erstellt und experimentieren mit R-CNN zur Segmentierung von Abfallobjekten. Sie beschreiben die Schwierigkeit kleine Objekte wie Zigarettenstummel zu erkennen und nennen die Bildauflösung als mögliche Ursache (Proença & Simões, 2020).

Auch Wang, Cai, Liang und Ye (2020) befassen sich mit der Lokalisierung von Abfallobjekten. Einerseits verwenden sie dazu das von TACO Dataset von Proença und Simões (2020), haben aber weiter auch ein eigenes öffentliches Dataset namens WJU-Waste erstellt. Sie haben dabei mit verschiedenen Methoden und CNN-Architekturen experimentiert und konnten mit dem ResNet-101, angewandt auf das WJU-Waste Dataset einen IoU⁶ von 87.84 erreichen. Ebenfalls mit ResNet-101, diesmal aber auf das TACO-Dataset angewandt, konnte ein IoU von 86.58 erreicht werden.

Im nachfolgenden Kapitel werden die Grundlagen der Theorie von Bilderkennung aufgezeigt.

3 Theoretische Grundlagen

Wie die im vorherigen Kapitel beschrieben Arbeiten, gibt es verschiedene Aufgaben in der Bilderkennung: die Klassifizierung, die Objekterkennung und die Segmentierung. Klassifizierung beschreibt die Methode, in welcher ein Bild einer vordefinierten Klasse zugeordnet werden soll (Druzhkov & Kustikova, 2016, S. 9). Dabei nimmt laut Druzhkov und Kustikova (2016, S. 9) die Komplexität mit der Anzahl Klassen zu. Bei der Objekterkennung sollen ein oder mehrere Objekte und deren Lokalisierung in einem Bild erkannt werden (Druzhkov & Kustikova, 2016, S. 11) und in der Segmentierung soll ein Bild nach verschiedenen Regionen unterteilt werden (Raut et al., 2009, S. 420) um beispielsweise, wie Wang et al. (2020) und Proença und Simões (2020),

⁵ Visual Geometry Group (siehe Kapitel 3.3)

⁶ Intersection over Union, gilt als die beliebteste Metrik zur Bewertung der Objekterkennungsbenchmark (Rezatofighi et al., 2019, S. 658)

Abfall-Objekte vom Hintergrund zu trennen. Im Rahmen dieser Arbeit wird auf Grund knapper Ressourcen, die verwendete Methode auf die Klassifizierung begrenzt.

Um zu verstehen wie auf Basis eines Bilderdatensets, ein Machine-Learning-Modell trainiert werden kann, um Objekte klassifizieren zu können, werden in diesem Kapitel die theoretischen Grundlagen der Thematik dargelegt.

3.1 Supervised Learning

Supervised Learning, oder auch «Überwachtes Lernen» ist die meistverwendete Form des Machine Learning (LeCun et al., 2015, S. 436). Supervised Learning zeichnet sich durch die Verwendung bereits verfügbarer, gelabelter Trainingsdaten aus (Cunningham et al., 2008, S. 21). Dabei weiss das Lernsystem, welche Labels mit welchen Trainingsdaten verknüpft sein sollen. In Klassifizierungsaufgaben stellen diese Label die Klassenbezeichnungen dar, nach welchen im Modell unterschieden werden soll (Cunningham et al., 2008, S. 21). Algorithmen des Supervised Learning leiten aus diesen Trainingsdaten Modelle ab, welche dann verwendet werden können, um nicht gelabelte Daten zu klassifizieren (Cunningham et al., 2008, S. 21).

Während des Trainings wird dem Modell ein Bild als Input gegeben, welches analysiert wird und einen Output in der Form eines Vektors mit den Scores der Wahrscheinlichkeiten jeder einzelnen Kategorie generiert (LeCun et al., 2015, S. 436). Das Ziel ist es, dass der Score der korrekten Kategorie dementsprechend die Höchste ist. Eine Fehlerfunktion zeigt dann wie stark die gemessenen Werte von den Zielwerten abweichen. Daraus resultiert eine Umgewichtung interner Gewichte (weights) für das weitere Training (LeCun et al., 2015, S. 436).

Die Supervised-Learning-Methode kann verschiedenermassen implementiert werden. Eine bewährte Lösung der Image-Classification-Aufgabe, bietet die Nutzung von Convolutional Neural Networks.

3.2 Convolutional Neural Network

Convolutional Neural Network (CNN) haben sich Albawi et al. (2017, S. 1) zufolge als sehr leistungsfähig im Lösen von Machine Learning Problemen herausgestellt. CNNs sind seit Anfang der 2000er Jahre sowohl in verschiedenen Klassifizierungs- wie auch Segmentierungs- und Objekterkennungsaufgaben erfolgreich umgesetzt (LeCun et al., 2015, S. 439). Sie zeigen, dass gerade im Bereich von Image Classification mit grossen Datenmengen, aber auch bei der Spracherkennung und Verarbeitung erstaunliche Resultate erzielt werden konnten (Albawi et al., 2017, S. 1).

Genauer ist CNN dafür entworfen, Daten in Format von mehreren Arrays zu verarbeiten (LeCun et al., 2015, S. 439). Im Falle der Bildverarbeitung beispielsweise, besteht ein Farbbild aus drei 2D Arrays, welche jeweils die Pixelintensitäten der Farben Rot, Grün und Blau (RGB) enthalten

(LeCun et al., 2015, S. 439). Die typische CNN-Architektur, welche ein solches Bild durchläuft, besteht aus einer Reihe von Schichten wie in Abbildung 1 dargestellt.

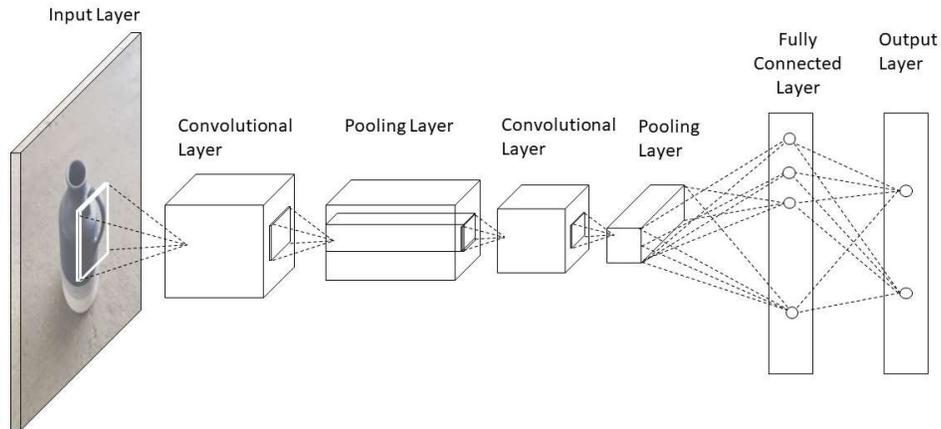


Abbildung 1: Einfache CNN Architektur (eigene Darstellung nach Noman, 2020, S. 42)

Die ersten Schichten bestehen aus zwei Arten von Layern, den Pooling Layers und den Convolutional Layers (LeCun et al., 2015, S. 439). Klassischerweise werden dabei mehrere Convolutional Layer übereinandergestapelt, gefolgt von einem Pooling Layer. Danach folgen wieder mehrere Convolutional Layers, wieder gefolgt von einem Pooling Layer und so weiter, bis am Schluss einige vollständig verbundene Schichten folgen, ergänzt von der abschliessenden Schicht, welche die Voraussage über die Wahrscheinlichkeiten für die Klassen liefert (Noman, 2020, S. 42).

Nachfolgend werden die einzelnen Arten von Schichten in einem CNN näher betrachtet.

3.2.1 Convolutional Layer

Die Aufgabe der Convolutional Layer besteht darin, lokale Verknüpfungen von Merkmalen aus dem vorherigen Layer zu erkennen (LeCun et al., 2015, S. 439). LeCun et al. (2015) beschreiben, dass die Units in einem Convolutional Layer in Feature-Maps organisiert sind, wobei die Units der lokalen Schicht mit den Feature-Maps der vorherigen Schichten, durch eine sogenannte Filter-Bank durch eine Menge von Gewichten verbunden sind. Gemäss LeCun et al (2015) verwenden alle Units in derselben Feature-Map dabei dieselbe Filter-Bank und somit dieselben weights. Der Hintergrund dabei, liegt darin, dass ein erkanntes Merkmal in einem Bildausschnitt auch an weiteren Positionen im Bild auftreten kann Banks (LeCun et al., 2015, S. 439). Verschiedene Feature-Maps in einem Convolutional Layer nutzen jedoch unterschiedliche Filter-Banks (LeCun et al., 2015, S. 439).

3.2.2 Pooling Layer

Pooling Layer sind dafür verantwortlich, mehrere ähnliche Merkmale zu einem zusammenzuführen (LeCun et al., 2015, S. 439). Die Idee hinter dem Pooling Layer liegt darin, die Komplexität für darauffolgende Layer mittels Down-Sampling zu reduzieren (Albawi et al., 2017, S. 5).

Vereinfacht gesagt, kann die Reduzierung der Bildauflösung genannt werden, welche ähnlich funktioniert. Im Pooling Layer werden überflüssige Informationen entfernt und die Datenmenge verringert. Laut Ablawi et al. (2017, S. 5) hat Pooling dabei aber keinen Einfluss auf die Anzahl der Filter. Dieser Layer führt somit zu erhöhter Berechnungsgeschwindigkeit, wobei die Leistungsfähigkeit nicht verringert wird.

3.2.3 Fully Connected Layer

Der Fully Connected Layer befindet sich am Anschluss der Folge von mehreren Pooling und Convolutional Layer. Die Schicht ist sowohl mit den vorherigen, sowie mit den darauffolgenden nächsten Schichten verknüpft, wie in Abbildung 1 **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt. Dabei werden alle Merkmale der vorhergehenden Schichten mit den Ausgabemerkmale des Output Layers verknüpft (Albawi et al., 2017, S. 5). Nach Albawi et al. (2017, S. 5) ist der negative Aspekt des Fully Connected Layers, dass er viele Parameter enthält, welche im Training berechnet werden müssen und diese Schicht somit viel Zeit in Anspruch nimmt. Deshalb wird versucht die Anzahl dieser Verbindungen gering zu halten.

3.3 CNN-Architekturen

CNN-Architekturen können aus verschiedensten Konstellationen von Layern bestehen. In der Vergangenheit haben unterschiedliche CNN-Architekturen Bekanntheit erlangt. Die nachfolgenden sechs Architekturen gehören zu den populärsten und werden als Grundlage für viele Bildverarbeitungsanwendungen genutzt.

- **LeNet** wurde von Yann LeCun entwickelt, um Ziffern zu erkennen (LeCun et al., 1990). Er hat mit seiner Arbeit einen wesentlichen Grundstein in der Entwicklung von CNN gelegt (Albawi et al., 2017, S. 4).
- **AlexNet**, entwickelt von Alex Krizhevsky, Ilya Sutskever und Geoffrey Hinton (2012) hat zum ersten Mal Effektivität in der Nutzung von Deep Learning im Kontext von Computer Vision gezeigt (Albawi et al., 2017, S. 4).
- **GoogLeNet**, erschaffen von Christian Szegedy und weiterer Mitarbeitende von Google (Szegedy et al., 2015), gewann den ILSVRC⁷-Wettbewerb 2014. Weiter bildet es gemäss Aloysius und Geetha (2017, S. 4) die Grundlage von Inception V4.
- **VGGNET**, entwickelt von Karen Simonyan und Andrew Zisserman (2014) aus dem Visual Geometry Group (VGG) Research Lab (Oxford University, gewann den zweiten Platz im ILSVRC-Wettbewerb 2014 und hat zur Entwicklung einer genaueren CNN-Architektur geführt (Aloysius & Geetha, 2017, S. 4).

⁷ ImageNet Large Scale Visual Recognition Challenge

- **ResNET**, entwickelt von Kaiming He und Mitarbeitenden von Microsoft (2016), gewann den ILSVRC Wettbewerb 2015. Das ResNet-Netzwerk ist bekannt für seine sehr tiefe Architektur mit 152 Layer (Noman, 2020, S. 45).

In dieser Arbeit wird nicht näher auf diese Architekturen eingegangen, da diese nicht direkt verwendet werden. Da das Ziel dieser Arbeit daraus besteht, das entwickelte Image Classification Model in eine Mobile App zu integrieren, sind jedoch zwei weitere CNN-Architekturen interessant, nämlich MobileNet und EfficientNet, welche in den nächsten zwei Kapitel vorgestellt werden.

3.3.1 MobileNet

Howard et al. (2017, S. 1) stellen fest, dass bisherige Fortschritte in CNN-Architekturen bezüglich besserer Genauigkeit, die Modelle in Bezug auf Geschwindigkeit und Grösse eher ineffizienter werden lassen. In Anwendungsbereichen wie Autonomous Driving, Robotik oder Augmented Reality müssen jedoch die Erkennungsergebnisse zeitnah auf Computern mit geringer Rechenleistung erfolgen. MobileNet ist eine Architektur, welche darauf ausgelegt ist, sehr kleine Modelle mit geringer Reaktionszeit zu erstellen. Howard et al. (2017, S. 1) heben hervor, dass sie zudem leicht in mobile Bildverarbeitungsanwendungen zu integrieren sind und einfach an dessen Anforderungen angepasst werden können. Zudem bietet MobileNet mehrere Möglichkeiten, die damit trainierten Modelle weiter zu komprimieren. Dies kann gemäss Howard et al. (2017, S. 4) über den Parameter des «Width Multiplier» oder des «Resolution Multiplier» Parameter geschehen. Bei dem ersten Parameter geht es ihnen zufolge darum, das Netzwerk über alle Layer gleichmässig auszudünnen. Der zweite Parameter stellt die Anfangsauflösung des Bildes dar, welche zur Reduzierung der Rechenkosten eines neuronalen Netzes beitragen kann (Howard et al., 2017, S. 4).

3.3.2 EfficientNet

Die von Tan und Le (2019) entwickelten EfficientNet B0 bis B7 bieten eine Modellfamilie, welche eine viel bessere Effizienz und Genauigkeit als bisherige Modelle bieten. Sie stellen fest, dass eine sorgfältige Anpassung von Netztiefe, -breite und -auflösung zu einer besseren Performance führen kann und erreichen mit ihrem EfficientNet-B7 eine Top-1-Genauigkeit von 84.4% angewendet auf das ImageNet-Dataset (Tan & Le, 2019, S. 6105). Im Vergleich zu bisheriger Praktiken wie MobileNet und ResNet, werden diese Grössen von Netztiefe, -breite und -auflösung in EfficientNet nicht willkürlich angepasst, sondern in der von Tan und Le (2019) entwickelten Methode einheitlich anhand von festen Variablen skaliert.

Wie mit EfficientNet oder MobileNet trainierte Modelle genutzt werden können, um eigene Modelle zu trainieren, wird im nächsten Kapitel beschrieben.

3.4 Transfer Learning & Retraining

Um eine Objektklassifizierungsaufgabe zu lösen, muss nicht zwingend eine CNN von Grund auf neu trainiert werden. Es besteht die weitere Möglichkeit, Teile eines bereits trainierten Modells zu nutzen, um damit für eine andere Aufgabe, ein Modell neu zu trainieren. Der Transfer von Model-Features ist dabei ein gängiger Weg des Transfer Learning und erlaubt eine verbesserte Leistung des neuen Modells durch diesen Transfer (Shao et al., 2015, S. 1022). Der Vorteil dieser Technik liegt darin, dass für das Training des neuen Modells erheblich weniger Daten benötigt werden und somit ein Zeitersparnis im Training erreicht werden kann (Géron, 2019, S. 300).

3.5 Over- und Underfitting

Im Model Training ist mit zwei Herausforderungen umzugehen, um gute Resultate zu erzielen. Die erste nennt sich Overfitting (Überanpassung). Dies tritt auf, wenn das Modell zu gut auf die einzelnen Trainingsdaten angepasst ist und keine allgemeinen Muster daraus abgeleitet werden konnten (Ruizendaal, 2017). Das Modell tut sich dann schwer daran, weitere Daten ausserhalb des Trainingsdatensatzes zu erkennen. Erkannt wird ein Overfitting daran, dass im Training die Training Accuracy höher ist, als diejenige des Validierungs- bzw. Testdatasets (Ruizendaal, 2017). Um dem entgegenzuwirken, können verschiedene Massnahmen getroffen werden. Die beste Möglichkeit damit umzugehen, ist das Dataset zu erweitern. Denn vielfach ist eine unzureichende Datenmenge die Ursache von Overfitting (Ruizendaal, 2017). Eine weitere Möglichkeit besteht darin, die Modellarchitektur zu vereinfachen, denn eine zu komplexe Architektur kann ebenfalls eine Ursache von Overfitting sein. Zudem kann mit Data Augmentation gearbeitet werden, um die Bilder zu transformieren oder es kann mit Dropout und L1 & L2 Regulatoren gearbeitet werden (Ruizendaal, 2017).

Die zweite Herausforderung beschreibt das Gegenteil von Overfitting, nämlich das Underfitting (Unteranpassung). Dabei ist die Accuracy der Validierung höher als die des Trainingsdatasets (Ruizendaal, 2017). Die Massnahmen, um Underfitting zu verhindern sind entgegengesetzt der Massnahmen im Falle von Overfitting. Eine häufige Ursache ist, dass ein zu hoher Dropout definiert wurde, oder dass die CNN-Architektur zu simpel ist (Ruizendaal, 2017).

4 Systemanalyse: Recyclinghof Hagenholz

Um zu erfahren, wie der Recyclinghof Hagenholz aufgebaut ist und wie ihre Recycling- und Abfallcontainer organisiert sind, wurde mit Hilfe der Stadt Zürich, der Kontakt zum Recyclinghof Hagenholz hergestellt. So konnte eine Führung durch den Recyclinghof Hagenholz arrangiert werden. Dabei wurden alle Container des Recyclinghofes gemustert und wichtige Details zu den Recyclingkategorien, deren Inhalte und Abgrenzung zu anderen Kategorien geklärt. Zudem

konnten die Prozesse, sowie die Organisation der Container und deren Standorte innerhalb des Recyclinghofes geklärt werden, welche nachfolgend beschrieben werden.

4.1 Entsorgungsprozess

Im Recyclinghof Hagenholz kommen täglich hunderte Kunden vorbei, um ihre Materialien abzugeben. Im Jahr 2020 waren es, gemäss T. Nussbaum, Projektleiter und Mediensprecher von ERZ Entsorgung und Recycling Zürich (persönliche Kommunikation, 21. Mai 2021) im Schnitt rund 387 Fahrzeuge pro Tag. An Samstagen, rund um die Feiertage und um den Jahreswechsel herum, sei der Ansturm am höchsten. Dazu kommen die Zügelstage, jeweils am 31.3, 30.6 und 30.09 an denen, laut Herrn Nussbaum, ein erhöhtes Aufkommen entsorgungswilliger Kunden besteht.

Der Entsorgungsprozess läuft dabei für den Kunden im Recyclinghof folgendermassen ab:

1. Am Eingang des Areals im Hagenholz, wird das Auto gewogen.
2. Danach kann der Kunde in den Recyclinghof fahren und sein Auto parkieren.
3. Zu diesem Zeitpunkt ist der Kunde bereit, seine mitgebrachten Materialien in den zahlreichen Containern abzugeben.
4. Der Kunde nimmt dann seine Objekte aus dem Fahrzeug und sucht den jeweiligen Container, für die mitgebrachten Gegenstände und deponiert diese am korrekten Platz.
5. Ist der Kunde fertig, verlässt er mit dem Auto den Recyclinghof und muss am Ausgang des Areals sein Auto nochmals wiegen.
6. Der Kunde bezahlt die Entsorgungsgebühren oder verwendet allenfalls erhaltene Entsorgungsgutscheine und kann dann das Areal verlassen.

Der entwickelte Prototyp soll in dem beschriebenen Entsorgungsprozess den Schritt 4 unterstützen und es dem Kunden einfacher machen, herauszufinden zu welcher Recyclingkategorie sein mitgebrachtes Gut gehört.

4.2 Weitere Entsorgung

Die gesammelten Materialien werden je Kategorie unterschiedlich weiterverarbeitet, verkauft oder verbrannt. Nachfolgend einige Beispiele was mit den Materialien aus dem Recyclinghof Hagenholz geschieht.

Rund 71'351 Tonnen Hauskehricht (inklusive Sperrgut) wurden im Jahr 2019 in der Stadt Zürich gemäss T. Nussbaum (persönliche Kommunikation, 21. Mai 2021) entsorgt. Abfall (Haus- und Betriebskehricht) sowie Sperrgut wird im Kehrichtheizkraftwerk Hagenholz verbrannt und damit Strom und Wärme für rund 170'000 Wohnungen in Zürich produziert (Stadt Zürich, o. J. -b). Eine Einlieferungsmenge von rund 2289 Tonnen Alteisen und Metalle entstand in den Recyclinghöfen in Zürich im Jahr 2019 (T. Nussbaum, persönliche Kommunikation, 21. Mai 2021). Dabei werden Buntmetalle von Hand im Recyclinghof Hagenholz getrennt, sortiert und ab einer gewissen

Menge, an den meistbietenden Abnehmer verkauft (Fachpersonal RH⁸, persönliche Kommunikation, 17. Februar 2021). Sortiert wird dabei nach verschiedenen Kriterien wie Werkstoff und Beschaffenheit. PET Getränkeflaschen werden sortiert, zerkleinert und geschmolzen und können zu 100 Prozent wieder verwertet werden (Stadt Zürich, o. J. -b). Die gesammelten Textilien werden von Hilfswerken entgegengenommen und an Bedürftige in der Schweiz, aber auch im Ausland verteilt (Stadt Zürich, o. J. -b). Der Abnehmer der Kaffeekapseln wiederum, trennt die Alu-Kapsel von den Kaffeeresten und auch für die Kategorie Bücher gibt es einen Abnehmer, welche die Buchseiten vom Buchrücken trennt (Fachpersonal RH, persönliche Kommunikation, 17. Februar 2021).

4.3 Lageplan

Der Recyclinghof Hagenholz wird von den Kunden mit dem eigenen Fahrzeug erreicht. Dort stehen Parkplätze zu Verfügung, um die mitgebrachten Recyclingmaterialien auszuladen und in den jeweiligen Containern abzugeben. Im Recyclinghof werden 25 Kategorien von Recyclingmaterialien unterschieden, welche getrennt gesammelt werden. Wie in Abbildung 2 dargestellt, ist der Recyclinghof in zwei Entsorgungspassagen aufgeteilt, in welchen jeweils dieselben Containerklassen zu Verfügung stehen.

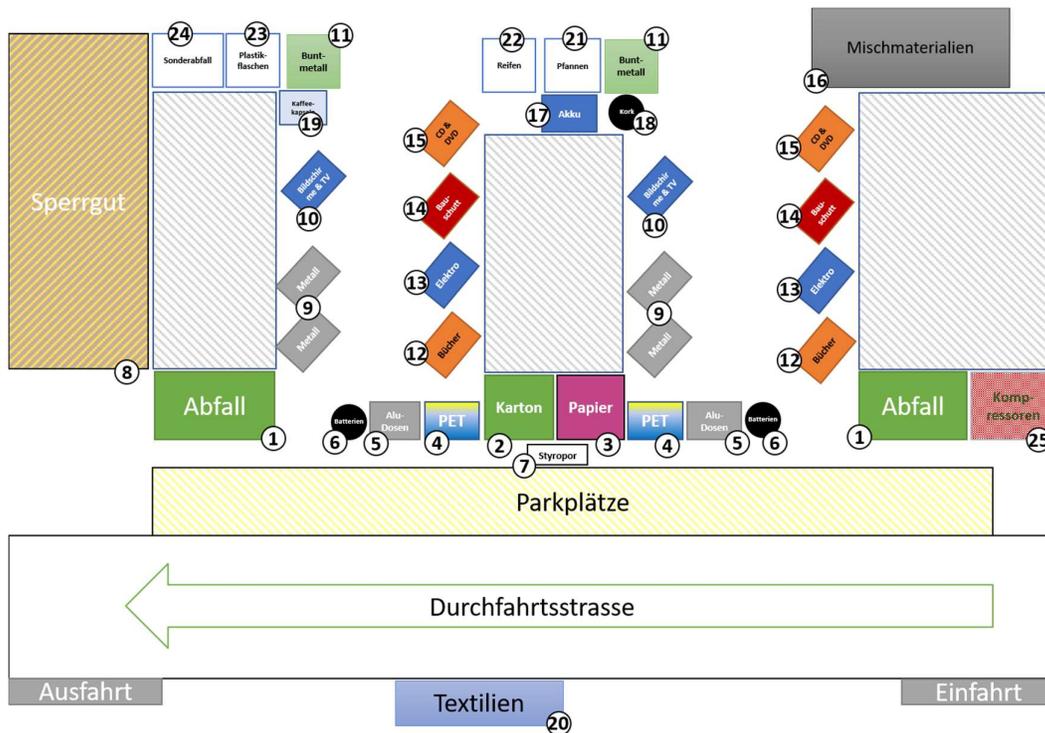


Abbildung 2: Lageplan Recyclinghof Hagenholz

⁸ Recyclinghof Hagenholz

Dabei sind die Container mit einer Kennnummer versehen. Jede Kennnummer entspricht einer Sammelkategorie. Die Kategorien sind in Kapitel 4.6 beschrieben.

4.4 Containerorganisation

Die Container sind, wie in der vorherigen Abbildung 2 zu sehen, in verschiedenen Farben dargestellt. Diese Farben widerspiegeln die tatsächlichen Containerfarben im Recyclinghof. Die Organisation der Containerfarben ist, wie in Abbildung 3 zu sehen, auf fünf Überkategorien organisiert.



Abbildung 3: Containerorganisation

Alles was im weitesten Sinne mit Elektromaterialien zu tun hat, gehört in einen der unterschiedlichen blauen Container. Diese wiederum sind unterteilt in die Kategorien «Lithium-Akku», «TV & Bildschirme» und allgemeiner Elektroschrott «Elektro». Zu «Lithium-Akku» gehören generell alle Geräte mit Akku wie beispielsweise Notebooks, Tablets, Smartphones oder auch elektrische Zahnbürsten. Diese Kategorie gilt als Teil des Sonderabfalls. «TV & Bildschirme» umfasst die Kategorie, in welcher, wie der Name andeutet, alle Computermonitore und Fernseher hingehören. Hierbei ausgeschlossen sind jedoch die Röhrenbildschirme, welche separat als Sonderabfall geführt werden, da diese giftigen Gase enthalten. Die Kategorie «Elektro» deckt die restlichen Elektromaterialien ab, wie Küchengeräte, PCs, Staubsauger oder auch Lampen.

In grauen Containern befindet sich «Altmittel». Zwei mal zwei Container stehen davon in den jeweiligen Recyclingstrassen. Alles aus Eisen ist hier abzugeben. Als Beispiel sind hier Blechdosen, Fahrräder oder Metallbesteck zu nennen. In den grünen Container werden Objekte aus Buntmetallen abgegeben, also alles aus Kupfer, Zink und Aluminium. Häufig werden hier beispielsweise Golfschläger, Skistöcke oder ähnliches entsorgt.

«Bauschutt» befindet sich in den roten Containern. Hier gehören nicht nur Ziegel, Glasscheiben oder Gartenplatten hin, auch Töpfe, Keramikobjekte, sowie Teller und Trinkgläser werden hier entsorgt.

Als letztes gibt es die orangen Container. Dazu gehören die Kategorien «Bücher» und «CD & DVD». Die Kategorie Bücher ist selbsterklärend, bei «CD & DVD» können nicht nur die CDs und DVDs selbst entsorgt werden, auch deren Hüllen, sowie alte Schallplatten und Videokassetten sind hier richtig.

Die weiteren Container sind nicht nach einem Farbschema organisiert, jedoch sind sie wie nachfolgend beschrieben im Recyclinghof platziert. Die gängigen Recyclingmaterialien aus dem

üblichen Haushalt sind direkt angrenzend an den Parkplätzen organisiert. Dazu gehören «Abfall», «Karton», «Papier», «PET», «ALU», «Batterien» und «Styropor». Auf der gegenüberliegenden Seite der Durchfahrtsstrasse befinden sich zudem die Container «Textilien».

Die etwas spezielleren Kategorien «Sonderabfall», «Plastikflaschen», «Reifen», «Pfannen», «Kork» und «Kaffeekapseln» befinden sich am hinteren Ende der beiden Entsorgungspassagen.

Gut erreichbar mit dem Fahrzeug, befindet sich am Ende der Durchfahrtsstrasse die Sammlung des «Sperrgutes». Alles aus Holz, sowie ganze Sofas, Skis oder Matratzen sind hier zu entsorgen. Wichtig dabei ist, dass diese Objekte keine grösseren Metallteile enthalten. Mischmaterialien welche aus Metall und anderem bestehen, gehören nämlich in eine weitere Kategorie, den «Mischmaterialien». Als Beispiel sind Kinderwagen, Stühle und Tische, welche nicht nur Holz, sondern auch Metall enthalten, zu nennen.

4.5 Fehlplatzierung von Objekten

Bei der Abgabe der Materialien werden die Kunden durch die Mitarbeiter des Recyclinghofs betreut und unterstützt. Für viele Kunden ist das Auffinden des richtigen Containers eine Herausforderung. Das zeigt sich, wenn man einen genaueren Blick in die Container wirft. In vielen Containern sind immer wieder Materialien zu finden, welche dort nicht hingehören.

Während der Besuche im Recyclinghof konnte festgestellt werden, dass in den folgenden Recyclingklassen öfters falsche Objekte zu finden sind:

- **Alu-Dosen:** Im Sammelcontainer der Alu-Dosen, befinden sich regelmässig Blechdosen, welche eigentlich in den Container «Metall» gehören würden.
- **Elektro:** In den Elektro-Containern sind teilweise Geräte mit Lithium-Akkus zu finden, welche ebenfalls in einem eigenen Sammelcontainer «Lithium-Akkus» abzugeben sind.
- **Batterien:** Wirft man einen Blick in die Batterie Sammlung, sind ebenfalls häufig «Lithium-Akkus» aufzufinden, welche dort nicht hingehören.
- **Metall:** In den vier Metallcontainern sind häufig auch Objekte aus Kupfer, Zink oder Aluminium vorzufinden. Diese gehören jedoch in den «Buntmetall» Container.
- **Fernseher / Monitore:** In diesen Containern kommt es vor, dass Kunden auch alte Röhrenbildschirme deponieren. Diese gehören jedoch zum Sonderabfall, da sie giftige Gase enthalten, welche bei Beschädigung austreten könnten.

Im Recyclinghof werden die Container regelmässig durch die Mitarbeiter nachkontrolliert und falsch platzierte Gegenstände werden laufend aussortiert und in den korrekten Containern deponiert.

4.6 Recyclingklassen

Im Rahmen der Führung durch den Recyclinghof Hagenholz und der gemachten Beobachtungen während den Foto-Sessions vor Ort, konnten die 25 Recyclingkategorien ermittelt werden, welche in Tabelle 1 aufgeführt sind. Ergänzend dazu wurde eine detaillierte Zusammenstellung der Container, mit der Beschreibung derer genauen Inhalte angefertigt, welche in Anhang A. vorzufinden ist. Hierzu wurden zusätzlich die Informationen der Webseite der Stadt Zürich (o. J. -b) hinzugezogen, um die Angaben zu vervollständigen.

#	Kategorie	Schwierigkeit
1	Abfall	1
2	Sperrgut	3
3	Karton	1
4	Papier	1
5	Styropor	1
6	Alu-Dosen	2
7	PET	1
8	Textilien	1
9	Metall	3
10	Buntmetall	3
11	Elektro	3
12	Bauschutt/ Steingut	1
13	Batterien	2
14	Lithium-Akkus	2
15	Kaffeekapseln	1
16	Bildschirme	2
17	Bücher	1
18	CD-DVD	1
19	Kork	1
20	Pfannen	1
21	Mischmaterialien	3
22	Sonderabfall	3
23	Plastikflaschen (nicht PET)	1
24	Pneu	1
25	Kompressoren	1

Tabelle 1: Recyclingklassen im Recyclinghof Hagenholz

Die Kategorien werden in verschiedene Schwierigkeitsstufen unterteilt, wobei 1 bedeutet, dass die Kategorie für den Kunden einfach verständlich ist. Schwierigkeitsstufe 2 bedeutet, dass es weder besonders einfach noch besonders schwierig ist, die Zuordnung zu dieser Kategorie zu machen und Stufe 3 bedeutet, es ist eher schwer für den Kunden zu verstehen, was zu dieser Kategorie gehört oder nicht. Diese Einstufung basiert auf den gemachten Erfahrungen im Recyclinghof Hagenholz und ist eine grobe Analyse der Situation. Grundlage dafür sind die festgestellten Fehlplatzierungen in den jeweiligen Container und aufgetretene Fragen seitens der Kunden

bezüglich der korrekten Abgabe ihrer Objekte. Anhand dieser Einstufung in Schwierigkeitsgrade wurden die Priorisierung der späteren Datenerhebung der Bilder festgelegt.

5 Methode

Die Schwerpunkte dieser Arbeit sind wie in Abbildung 4 dargestellt, in folgende vier Teilaufgaben unterteilt: Die Datenerhebung und Aufbereitung, ML-Modellentwicklung, die Erstellung des Prototyps und zum Schluss der Feldtest. Basis für diese Arbeit bietet die in Kapitel 4 beschriebene Voranalyse im Recyclinghof Hagenholz.



Abbildung 4: Schwerpunkte der Arbeit

5.1 Datenerhebung

Grundlage für das Machine Learning Model stellt das Image Dataset dar. Daten für das angefertigte Dataset, wurden mit drei unterschiedlichen Methoden gesammelt. Einerseits wurden selbst Fotos von Recyclingobjekten erstellt, weiter wurden vorhandene Datasets auf ihre Brauchbarkeit für diese Arbeit untersucht und als dritte Methode wurde «Scraping» eingesetzt, um das Dataset zu erweitern. Nachfolgend werden die drei Methoden vorgestellt. Wie in Kapitel 4.6 erwähnt, wurden ein Fokus auf die Kategorien mit grossem Schwierigkeitsgrad in der manuellen Zuordnung gelegt.

5.1.1 Bilder erstellen

Für die Erstellung des Recycling-Datasets wurden Fotos von Recyclingobjekten manuell aufgenommen. In insgesamt drei Besuchen im Recyclinghof Hagenholz wurden Bilder unterschiedlicher Objekte, der verschiedenen Recyclingkategorien erstellt und gesammelt. Jedes einzelne Objekt musste dabei aus dem jeweiligen Container herausgenommen, an einem neutralen Platz ausgelegt oder in der Hand gehalten werden. Es wurde dabei speziell beachtet, dass der Hintergrund des Bildes möglichst neutral, also einfarbig erscheint, damit das Modell später nicht durch Objekte im Hintergrund entstellt wird. Ausserdem wurde darauf geachtet, dass ein Teil der Objekte in der Hand gehalten wurden und andere auf den Boden platziert wurden, um beide Möglichkeiten, wie ein Gegenstand für ein Foto platziert werden kann, abzudecken.

Ergänzend wurden Fotos von einzelnen Objekten im Haushalt fotografiert. Dazu gehören Objekte der Kategorien «Alu-Dosen», «Batterien», «PET-Getränkeflaschen» und «Karton». Die Fotos konnten mit kleinem Zeitaufwand erstellt werden und fanden deshalb den Weg in das eigene Dataset.

In der nachfolgenden Abbildung 5 wird eine Stichprobe der erstellten Bilder dargestellt. Darauf zu sehen ist, wie die einzelnen Objekte jeweils entweder auf einer neutralen Oberfläche platziert oder in der Hand gehalten wurden. Die Stichprobe umfasst dabei alle 20 Kategorien, für welche eigene Fotos erstellt und gesammelt wurden.

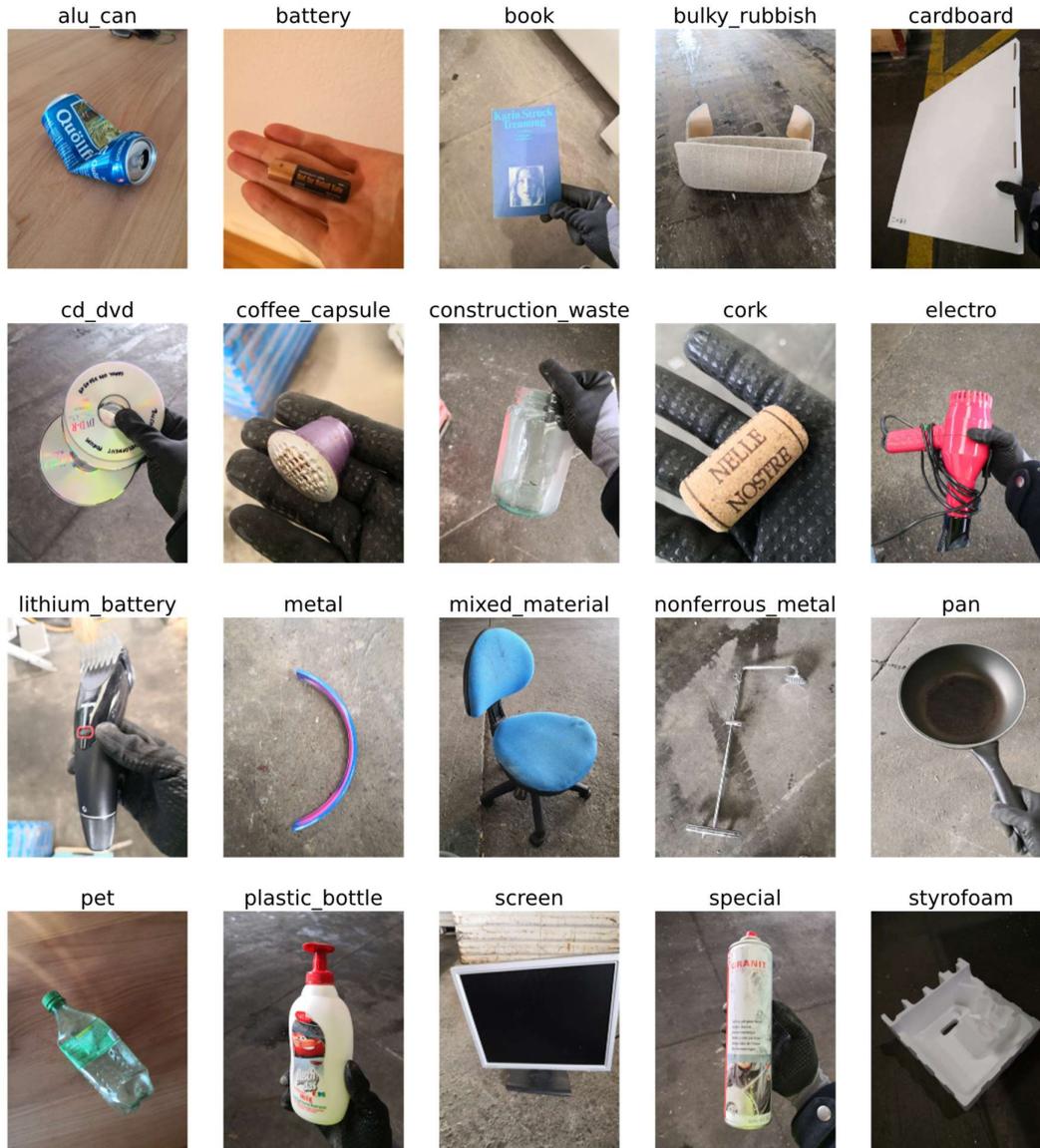


Abbildung 5: Stichprobe der angefertigten Bilder

Das manuelle Sammeln und Erstellen eigener Bilderdaten ist ein sehr zeitaufwändiger Akt. Die Anzahl Objekte im Recyclinghof, ist stark abhängig von der Kundschaft und durch das fortlaufende Abtransportieren der Container, durch die Mitarbeiter des Recyclinghofs limitiert. Daher können pro Besuch im Recyclinghof nur eine beschränkte Anzahl Gegenstände abfotografiert werden. Es konnten in der Projektzeit zu wenige Bilder selbst erstellt werden, um das gesamte

Dataset damit zu erstellen. Daher werden weitere Datenquellen genutzt, welche in den beiden nachfolgenden beiden Kapitel beschrieben sind.

5.1.2 Vorhandene Datasets

Es existieren bereits einige Image Datasets, welche über eine Vielzahl an Bilder verfügen und frei verfügbar genutzt werden können. Solche Open-Source Datasets wurden auf ihre Verwendbarkeit in dieser Arbeit geprüft. Dabei wurden folgende Datasets analysiert:

- **ImageNet** ist ein sehr umfangreiches Dataset, welches rund 1'281'167 Bilder von 1000 Objektklassen umfasst (Russakovsky et al., 2015). Es zeigt sich jedoch als unvorteilhaft, Bilder aus diesem Dataset für die Zwecke dieser Arbeit zu verwenden, da auf Basis dieses Datasets bereits die vortrainierten Modelle mit EfficientNet und MobileNet trainiert wurden, welche in dieser Arbeit eingesetzt wurden. Ausserdem wurden keine Bilder von der in dieser Arbeit verwendeten Kategorien im ImageNet-Dataset vorgefunden.
- Das **MJU-Waste** Dataset besteht aus insgesamt 2475 Bildern von Abfall, gesammelt auf dem Campus der Minjiang University und aufgenommen aus einem Klassenzimmer (Wang et al., 2020). Das Dataset enthält zudem die Annotationen für die Segmentierung, wobei zwischen «waste» und «background» unterschieden wird. Generell sind auf den Bildern die Abfallobjekte sehr klein und der Hintergrund mit dem Klassenzimmer, in dem sich mehrere Studenten befinden, sehr markant. Eine Nutzung der Bilder wäre generell vorstellbar. Dafür müssten die Bilder jedoch zugeschnitten und die Objekte in den Fokus gesetzt werden. Zudem wäre es nötig, die einzelnen Bilder manuell auf die neue Aufgabe umzulabeln. Durch die zeitliche Limitierung dieser Arbeit, wurde davon abgesehen diese Schritte durchzuführen.
- Das Dataset **TACO** von Proença und Simões (2020) besteht aus über 4500 Bilder, gelabelt auf 60 Abfallkategorien, sowie der Klassifizierung nach Bildhintergrund, sowie verschiedenen Kontexten. Zudem enthält das Dataset die Annotationen für die Segmentierung. Die in diesem Dataset vorhandenen Fotos sind aber mehrheitlich im Kontext der Natur entstanden, was bedeutet, dass die Bildhintergründe aus Wiesen, Wald- und Strassenboden bestehen, oder die Aufnahme unter Wasser aufgenommen wurden. Zudem sind die Abfallobjekte nicht im Fokus des Bildes und weiter werden auch mehrere Objekte in einem Bild dargestellt. Aus den genannten Gründen wird das Dataset als eher ungeeignet bewertet für die weitere Nutzung in dieser Arbeit.
- **TrashNet**, das entwickelte Dataset von Yang und Thung (2016) enthält rund 2572 Bilder. Die Bilder sind dabei in sechs Klassen aufgeteilt: «glass» mit 501 Bilder, «paper» mit 594 Bilder, «cardboard» mit 403 Bilder, «plastic» mit 482» Bilder, «metal» mit 410 Bilder und «trash» mit 137 enthaltenen Bilder. Obwohl diese Klassen auf den ersten Blick ähnlich sind, wie die Klassen Bauschutt, Papier, Karton, PET, Metall und Abfall, ist die

Definition der jeweiligen Inhalte im TrashNet Dataset leicht unterschiedlich. Dennoch ist die Qualität und der Standard der Bilder übereinstimmend mit dem Anwendungsfall im Recyclinghof. Ein relabeling der Bilder auf die eigenen Klassen ist jedoch notwendig.

Die Untersuchung der vorhandenen Datasets ergab, dass ausschliesslich Bilder aus dem TrashNet-Dataset für die weitere Verwendung in dieser Arbeit in Frage kamen.

5.1.3 Image Scraping

Bilder aus dem Internet wurden als ergänzende Quelle genutzt, um das Dataset zu erweitern. Dabei wurde eine Technik angewandt, welche sich Web Scraping nennt. Web Scraping ist ein Vorgehen welches Websites automatisch nach vordefinierten Suchgegriffen durchforstet, um automatisiert Daten und Informationen zu erhalten, ohne diese manuell kopieren zu müssen (Ridwang et al., 2020, S. 347). Der Image Scraper wurde in Form eines Python-Skripts erstellt, welches unter dem Dateinamen «web_scraper.py» im GitHub-Repository (siehe Kapitel 6) zu finden ist. In einem ersten Schritt werden URLs zu den gewünschten Inhalten, in diesem Fall den Bildern, gesammelt und als Liste zurückgegeben. In einem weiteren Schritt werden die Bilder hinter den gesammelten Links heruntergeladen und in einem Verzeichnis abgelegt. Das Skript nutzt Selenium, um mittels der Google Chrome Engine Suchabfragen absetzen zu können.

Im Prozess des Scraping haben sich verschiedene Herausforderungen bezüglich Qualität, Grösse und Menge der gefundenen Bilder herausgestellt, welche nachfolgend dargelegt werden.

5.1.3.1 Keywords

Eines der wichtigsten Werkzeuge für die Suche nach brauchbaren Bildern sind die Keywords (Suchbegriffe). Mit der Optimierung passender Suchbegriffe konnten erheblich bessere Resultate erzielt werden. Wird beispielsweise nach «Fernseher» gesucht, besteht der Grossteil der Suchresultate aus Katalogbildern neuer Fernseher, und nicht aus Bildern von gebrauchten TV-Geräten, wie sie im Recyclinghof abgegeben werden. Hilfreich sind dabei Begriffe wie «gebrauchte», «defekte» oder «alt», um Bilder von Objekten zu finden, welche der Qualität von Objekten entsprechen, die im Recyclinghof entsorgt werden. Ein weiteres Keyword, welches genutzt wurde, war «kleinanzeigen». Die Objekte, welche auf Kleinanzeigen angeboten werden, entsprechen dem, was auch entsorgt werden könnte. Auch die Qualität der Bilder kommt dem Anwendungsfall sehr nahe, da in beiden Fällen die Bilder meist mit der Smartphone Kamera geschossen werden. Weiter wurden sowohl deutsche wie auch englische Suchbegriffe verwendet, um möglichst viele Bilderquellen zu erschliessen. Die verwendeten Suchbegriffe sind im Anhang B. zu finden.

Um auf Grundlage der erhobenen Bilderdaten ein brauchbares Image-Dataset zu erstellen, wurden die Daten, wie im nächsten Kapitel beschrieben, weiter aufbereitet.

5.2 Datenaufbereitung

Die Datenaufbereitung ist eine der wichtigsten Aufgabe im Machine Learning. Gemäss Lin et al. (2011, S. 1689) ist es notwendig viele Bildvarianten jeder Klasse zu haben, um visuelle Varianzen der Objekte abzudecken. Output der Datenaufbereitung ist das fertige Image-Dataset, welches für das Training des ML-Models verwendet wird.

5.2.1 Duplikate

Eine Herausforderung, war es mit Bilddoubletten umzugehen. Einige der gesammelten Bilder mittels Scraping wurden nämlich mehrfach in verschiedenen Bildgrössen heruntergeladen. Um die Qualität des Datasets zu gewährleisten, mussten diese aus der Bildersammlung entfernt werden. Dafür wurde eine Python-Skript namens «duplicate_remover.py» erstellt, welches anhand eines Hash die Bilder miteinander vergleicht und immer nur das Bild mit der höchsten Auflösung behält und die restlichen Duplikate löscht. Im Vergleich der Hashwerte wurde versucht ein Optimum zu finden, zwischen Übergenauigkeit, in welcher nicht alle Duplikate gefunden werden und einer Untergenauigkeit, in welcher Duplikate aufgedeckt werden, welche gar keine sind. Dies wurde erreicht durch das Experimentieren verschiedener Hashgrössen, wobei je nach Klasse und Suchbegriff unterschiedliche Hashgrössen verwendet werden, um das Optimum zu erreichen. Mit diesem Verfahren, zur Bereinigung der Duplikate, wurden jeweils ca. 25 bis 50% der gescrapten Bilder je Suchbegriff ausgesondert.

5.2.2 Qualität

Die gescrapte Bildersammlung wurde manuell analysiert, um unpassende Bilder auszusortieren. Zu den ausgesonderten Bildern gehörten beispielsweise Firmenlogos und Zeichnungen, aber auch Bilder von Objekten, welche nicht der gewünschten Kategorie entsprachen. Zudem wurden Bilder aussortiert, welche nicht der Art und Weise entsprachen, wie die Objekte im Recyclinghof Hagenholz entsorgt werden. Dazu zählte beispielsweise ein eingeschalteter Computermonitor, denn dieser wäre während dem Entsorgungsprozess nie am Strom angeschlossen. Je nach Kategorie wurden ca. 10 bis 40% der Bilder entfernt, da diese ungeeignet waren für das Training eines Machine Learning Models.

Ausserdem wurde die Qualität der im Recyclinghof erstellten Fotos nachgeprüft. Die mit dem Smartphone erstellten Bilder waren teilweise verschwommen. Dies wurde erst auf dem Computer ersichtlich und war schwer zu erkennen auf dem kleinen Display des Smartphones. Insgesamt wurden aus den selbst erstellten Bildern etwa 5-10% wegen Unschärfe ausgemustert.

Die Bilder aus dem TrashNet Dataset wurden auf ihre Brauchbarkeit hin analysiert. Da die Klassen nicht mit den in dieser Arbeit erarbeiteten Klassen übereinstimmen, wurden die einzelnen

Bilder manuell durchgegangen. Dabei wurden die relevanten Bilder extrahiert und, für den Anwendungsfall gelabelt, in einem neuen Dataset abgelegt

5.2.3 Vertical Flip

Einige Kategorien stellten eine grosse Herausforderung in der Sammlung dar. Weder mit Scraping, noch in bereits vorhandenen Datasets, konnte die gewünschte Anzahl Bilder mit angemessenem Aufwand bereitgestellt werden. Daher wurde für die Bilder der Kategorien «cork», «pet» und «tire» eine Transformation durchgeführt, um diese vertikal zu spiegeln und so das Dataset künstlich zu erweitern. Dafür wurde das Skript «flip_images.py» erstellt. Diese vertikale Spiegelung wurde bewusst nur auf diesen drei Kategorien durchgeführt, da diese Kategorien für das Modell bereits einfach zu erkennen waren. Das Ziel war, die Gewichtung der Kategorien auszugleichen und nicht etwa die Erkennung dieser drei Kategorien mittels Transformation zu verbessern.

5.2.4 Labeling

Die Bilder aus den drei Quellen «eigene Sammlung», «TrashNet» und «Scraping» wurden in separaten Verzeichnissen als eigene Datasets abgelegt. Dabei stellt die Verzeichnisstruktur in allen drei Datasets das Labeling und somit die Zuweisung, welches Foto zu welcher Kategorie gehört dar. Die Daten wurden separat in den drei Datasets geführt, um die Herkunft der Daten langfristig nachvollziehen zu können. Ausserdem bietet diese Trennung nach Datenherkunft die Möglichkeit mit den unterschiedlichen Datasets Experimente bezüglich verschiedener Datenzusammensetzungen durchzuführen.

5.2.5 Zusammenführen der Daten

Um aus den drei Image-Datasets ein vollumfängliches Dataset für das Training des Modells zu erstellen, wurde das Python-Skript «dataset_merger.py» geschrieben, mit welchem mittels Parameter definiert werden kann, aus welcher Quelle, wie viele Bilder verwendet werden sollen, um am Ende ein Dataset zu bekommen, welches über alle Klassen gleich viele Bilder enthält.

5.3 Model Evaluation & Architekturwahl

Im Rahmen dieser Arbeit wurden drei verschiedene CNN-Modelle entwickelt und validiert, um danach das beste Modell weiter zu optimieren. Ein erstes Modell wurde dabei von Grund auf erstellt. Das zweite auf Basis eines vortrainierten Modells mit EfficientNet-B0 und das dritte Modell, welches entwickelt wurde, basiert auf einem vortrainierten Modell mit MobileNetV3. Nachfolgend werden die drei entwickelten Modelle näher betrachtet.

5.3.1 Model from Scratch

Das erste Modell wurde von Grund auf trainiert. Bei der Erstellung diente ein Tutorial von Tensorflow als Orientierung (TensorFlow, 2021a). Die Architektur des Modells ist, wie in Abbildung

6 dargestellt, folgendermassen aufgebaut: Nach dem Preprocessing im Sequential Layer, in welchem die Bilder, transformiert und auf eine Grösse von 224 mal 224 Pixel zugeschnitten werden, wird in Input in der Rescaling-Schicht auf 1/255 skaliert.

Layer (type)	Output Shape	Param #
sequential_5 (Sequential)	(None, 224, 224, 3)	0
rescaling_1 (Rescaling)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 32)	0
conv2d_2 (Conv2D)	(None, 52, 52, 32)	9248
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 32)	0
dropout_5 (Dropout)	(None, 26, 26, 32)	0
flatten (Flatten)	(None, 21632)	0
dense_5 (Dense)	(None, 128)	2769024
dense_6 (Dense)	(None, 24)	3096
Total params: 2,791,512		
Trainable params: 2,791,512		
Non-trainable params: 0		

Abbildung 6: Architektur des einfachen Modells

Darauf folgen drei Mal abwechselnd, jeweils ein Convolutional und ein Max Pooling Layer. Im Anschluss befindet sich ein Dropout Layer mit einem Dropout von 0.4. Die darauffolgende Flatten Schicht macht aus den mehrdimensionalen ein eindimensionales Array und in der letzten Schicht, dem Fully Connected Layer, basierend auf den 24 Klassen. In der Feinabstimmung des Modells wurden einerseits verschiedene Preprocessing Layer getestet, um das Optimum an Erkennungsgenauigkeit herauszuarbeiten. Zudem wurden verschiedene Dropouts ausprobiert.

5.3.2 Retrained Models

In einem weiteren Schritt wurde mit Transfer Learning versucht, Modelle zu entwickeln, welche besser performen, als das vorhin vorgestellte einfache Modell. Mit Transfer Learning sollen die ersten Schichten eines vorhandenen Netzes in einem neuen Modell wiederverwendet werden (Géron, 2019, S. 299). Die bessere Performance wird dabei erhofft, da mit dieser Technik weniger Trainingsdaten verfügbar sein müssen. Die Ausgabeschicht des ursprünglichen Modells wird dabei üblicherweise durch eine neue, auf die Aufgabe zugeschnittene Ausgabeschicht ersetzt (Géron, 2019, S. 300). Im Optimalfall sollen zudem auch die obersten Schichten des Modells neu trainiert werden.

Die Basis für die folgenden beiden Retrained Models bot ein Tensorflow Tutorial (TensorFlow, 2021b). Das Basismodell wird dabei von Tensorflow Hub bereitgestellt. Der Input Layer wird durch das verwendete Tensorflow Hub Model vorgegeben und ist sowohl bei EfficientNet-B0 (Tan & Le, 2019), wie auch bei MobileNetV3 (Howard et al., 2019) im Format 224 x 224 Pixel. Nach dem Input Layer bildet ein Keras Layer das Pretrained Model ab. Darauf folgt ein Dropout

Layer, gefolgt vom Output Layer. Der Output Layer wird durch die Liste der Klassen definiert wird, welche in der Objektklassifizierungsaufgabe unterschieden werden sollen.

5.3.2.1 EfficientNet-B0

EfficientNet bietet eine ganze Serie an Netzwerken, welche sich in ihrer Genauigkeit, aber auch ihrer Netzwerkgrösse und benötigtem Rechenaufwand unterscheiden (Tan & Le, 2019, S. 6110). Dabei ist B0 das kleinste, mit der schwächsten Performance und B7 das grösste mit der besten Performance der EfficientNet Modelle. In dieser Arbeit wurde die B0 Variante gewählt da diese Ressourcen schonendste Variante der EfficientNet Modellsreihe darstellt. Das EfficientNet-B0 nimmt während des Trainings dennoch, durch seine Grösse, sehr viel Rechenleistung in Anspruch. Auf der verwendeten Entwicklungsumgebung waren die GPU-Ressourcen beschränkt. Ein Fine-Tuning war mit diesem Modell daher nicht möglich.

Layer (type)	Output Shape	Param #
keras_layer_30 (KerasLayer)	(None, 1280)	4049564
dropout_30 (Dropout)	(None, 1280)	0
dense_30 (Dense)	(None, 24)	30744
Total params: 4,080,308		
Trainable params: 30,744		
Non-trainable params: 4,049,564		

Abbildung 7: Retrained Model Architektur auf Basis von EfficientNet-B0

Wie in Abbildung 7 zu erkennen ist, besteht das Modell aus den vortrainierten Schichten aus EfficientNetB0, gefolgt von einer Dropout Schicht und einem Fully Connected Layer basierend auf den 24 Recyclingklassen.

5.3.2.2 MobileNetV3

Das zweite Retrained Model, wurde auf Basis eines vortrainierten MobileNetV3-Small erstellt. MobileNetV3 ist die dritte Generation des Netzwerkes, wobei sich sowohl die Accuracy, wie auch die Geschwindigkeit jeder Generation um einzelne Prozente verbessert hat (Howard et al., 2019, S. 1314). So ist beispielsweise MobileNetV3-Small, trainiert auf ImageNet, um 6.6% genauer, als der Vorgänger MobileNetV2 (Howard et al., 2019). Generell bietet MobileNetV3 eine Small und ein Large Variante, welche jeweils für geringe, beziehungsweise hohe verfügbare Ressourcen ausgerichtet sind. Da das Modell in einer Mobile App zum Einsatz kommt, wurde die V3-Small Variante gewählt. Wie in Abbildung 8 dargestellt, besteht das neue Modell, gleich wie im Modell auf Basis von EfficientNet-B0, aus einer Keras Layer, welche diesmal die Features aus dem vortrainierten MobileNetV3 verwendet, einem nachfolgenden Dropout Layer, und einem Fully Connected Layer, basierend auf den vorhandenen Recyclingklassen. Zudem zusehen ist, dass das Modell um mehr als 2,5-mal weniger Parameter besitzt als das vorhin vorgestellte Modell auf Basis von EfficientNet-B0.

Layer (type)	Output Shape	Param #
keras_layer_26 (KerasLayer)	(None, 1024)	1529968
dropout_26 (Dropout)	(None, 1024)	0
dense_26 (Dense)	(None, 24)	24600
Total params: 1,554,568		
Trainable params: 1,542,456		
Non-trainable params: 12,112		

Abbildung 8: Retrained Model Architektur auf Basis von MobileNetV3

Zudem wurde im Modell mit MobileNetV3, sowohl Augmentation wie auch Fine-Tuning angewandt, was erkennbar ist, an der hohen Anzahl «Trainable Params» in Abbildung 8.

Die Retrained MobileNetV3 Modellvariante bildet die Grundlage für die weitere Entwicklung der Mobile-App, während die EfficientNet-B0 Modellvariante als Benchmark dient, und aufzeigen soll, was mit einem umfangreicheren Modell möglich ist. Die Modelle wurden auf das im Kapitel 5.2 entwickelte Dataset angewandt und miteinander verglichen. Welche Performance die Modelle erzielen, ist im Kapitel 6.2 beschrieben. Nach der ersten Evaluation des Modells wurden im weiteren Training Optimierungen vorgenommen, welche im nachfolgenden Kapitel beschrieben werden.

5.4 Model Training

Bei der Entwicklung des Modells wurde iterativ vorgegangen, wie in Abbildung 9 dargestellt. Dabei wurde generell vom Einfachen ins Komplexe erweitert. Somit sollte garantiert werden, dass zeitnah ein lauffähiges Modell bereitsteht, auf welchem laufend weiter aufgebaut werden konnte. Durch kontinuierliche Anpassung der Parameter im Modell nach Validierung der Resultate, konnte das Modell stetig verbessert werden. Zudem wurde zuerst mit nur wenigen Klassen experimentiert, bevor nacheinander, immer mehr Klassen ins Model Training eingeflossen sind.

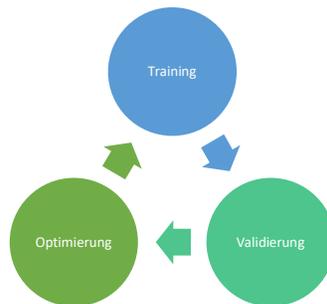


Abbildung 9: Vorgehen in Model Training

In der Aufgabe der Optimierung wurden verschiedene Parameter ausprobiert, dann im Training ausgeführt und danach validiert und mit bisherigen Resultaten verglichen. Durch die Resultate der Validierung wurde das Modell dann wieder optimiert und so weiter. In Abbildung 9 ist dieses iterative Vorgehen dargestellt.

Nachfolgend wird beschrieben mit welchen Optimierungsvarianten versucht wurde, die Performance des Modells zu steigern.

5.4.1 Augmentation

Eine Herausforderung im Model Training ist es mit Overfitting umzugehen. Um dem entgegenzuwirken, wurden verschiedene Image Augmentation Prozesse integriert und angewandt.

Perez und Wang (2017, S. 2) unterscheiden zwischen zwei Methoden der Augmentation. In der einen werden vor dem Training augmentierte Daten generiert, und so das Dataset in seiner Anzahl vervielfacht. In der zweiten Methode werden die Augmentations zur Laufzeit des Modelltrainings ausgeführt. In der vorliegenden Arbeit wird die zweite Methode verwendet, wobei laut Perez und Wang (2017, S. 7) aber zu beachten ist, dass die Laufzeit des Trainings sich in ihren Experimenten verdreifacht hat und nur eine geringe Verbesserung in der Genauigkeit gegenüber der ersten Methode erreicht werden konnte.

Wie sich die einzelnen Arten von Transformationen auf die Testdaten auswirken, wird in Abbildung 10 dargestellt.



Abbildung 10: Augmentation angewandt auf ein Beispielbild vom Trainingsdataset

Die Transformationen werden mit der Keras Komponente «ImageDataGenerator» umgesetzt, aus welcher die nachfolgend beschriebenen Operationen stammen (Keras, o. J. -a): Horizontal Flip beschreibt beispielsweise die Möglichkeit ein Bild horizontal zu spiegeln. Mit Rotation kann ein Bild in einem definierten Wertebereich gedreht werden. Nimmt man beispielsweise den Bereich 45 Grad, werden die Bilder mit zufälligen Werten zwischen -45 und 45 Grad gedreht. Width und Height Shift beschreiben die Transformationen, um ein Bild entweder nach oben und unten beziehungsweise nach links und rechts zu verschieben. Dies geschieht ebenfalls mit zufälliger Ausprägung. Zudem bildet Shear eine weitere Transformationsmöglichkeit. Wie in der Abbildung 10, in der dritt untersten Zeile zu erkennen ist, wird das Bild dadurch verschiedenermassen verzerrt. Des Weiteren bietet Zoom eine Option auf verschiedene Art Bildausschnitte zu vergrößern. Dabei entsteht auch eine Art Verzerrung im Bild. Die letzte verwendete Augmentation-Technik, nennt sich Brightness. Wie der Name andeutet, geht es darum, die Helligkeit des Bildes anzupassen. Über einen definierten Wertebereich kann angegeben werden wie Dunkel, beziehungsweise wie Hell die Transformation maximal sein darf.

Es wurde versucht, optimale Werte für die einzelnen Typen von Augmentation zu finden, in dem jede Augmentation einzeln, automatisiert nach verschiedenen Werten, getestet wurde. Dafür wurde das Skript «augmentation_optimization.py» erstellt. Einzelne Transformationen allein waren aber nicht so entscheidend, wie deren Kombination. Deshalb konnten keine Rückschlüsse aus den automatisierten Tests abgeleitet werden. Aus diesem Grund wurden die Werte in Kombination miteinander manuell geprüft, um eine bestmögliche Model Performance zu erreichen.

5.4.2 Fine-Tuning

Um die Retrained Models zusätzlich zu optimieren, soll mit Fine-Tuning gearbeitet. Dabei werden gewisse Schichten des Modells nochmals neu trainiert, um das Modell auf den neuen Anwendungsfall zu optimieren. Laut Yosinski et al. (2014, S. 8) können Unterschiede festgestellt werden je nachdem, ob obere, mittlere oder untere Schichten des Netzes angesprochen werden. Zudem kommt es auf die Grösse des neuen Datasets, sowie auf die Anzahl Parameter des vortrainierten Modells an, ob das Fine-Tuning einen Nutzen stiftet (Yosinski et al., 2014, S. 2). Aus Zeitgründen wurde in dieser Arbeit jedoch nur mit einem ganzheitlichen Fine-Tuning gearbeitet, in welchem das Fine-Tuning über das gesamte Modell läuft.

5.4.3 Hyperparameter

Die Performance des Modells kann durch Optimierung verschiedener Hyperparameter weiter optimiert werden. Im Rahmen dieser Arbeit wurden nachfolgend beschriebene Hyperparameter verwendet.

5.4.3.1 Dropout

Dropout bietet eine Technik, um dem Overfitting-Problem entgegenzuwirken (Srivastava et al., 2014, S. 1929). Dabei werden im Modelltraining zufällig Units aus dem neuronalen Netz entfernt, um so zu verhindern, dass sich die einzelnen Units zu sehr aneinander anpassen (Srivastava et al., 2014, S. 1929). Um zu evaluieren, welcher Dropout das Modell am besten beeinflusst, wurden Dropout-Werte zwischen 10 und 40 Prozent getestet.

5.4.3.2 L2 Regularisierung

Eine gängige Methode zur Reduzierung von Overfitting besteht, wie bereits erwähnt darin, die Komplexität des Modells zu reduzieren. Dies kann erreicht werden, in dem die Weights im Modell gezwungen werden, nur kleine Values entgegenzunehmen, was dazu führt, dass die Verteilung der Gewichte regelmässiger wird (TensorFlow, 2020). Dieses Konzept wird daher auch als «Weight Regularization» bezeichnet (TensorFlow, 2020). Durch das Hinzufügen von Gewichten zur Verlustfunktion, wird verhindert, dass das Modell während dem Training zu stark an Komplexität zunimmt (Shi & Zhao, 2020, S. 1095). Dabei werden sogenannte «Costs» zur Verlustfunktion hinzugefügt, welche entweder auf L1- und L2 Regularisierung basieren (TensorFlow, 2020).

5.4.3.3 Epochen & Early Stopping

Als weitere Massnahme um dem Overfitting entgegenzuwirken, wurde die Anzahl Epochen im Training beschränkt. Dafür wurden verschiedene Early-Stopping Methoden verwendet. Mit Early Stopping kann definiert werden, dass das Training nur so weit fortgeführt wird, wie sich die Accuracy verbessert, bzw. der Loss sich verkleinert (Keras, o. J. -b). Sobald ein gewisser vordefinierter Grenzwert an Anzahl Epochen (Patience) überschritten wird, in denen keine Verbesserung mehr vorliegt, bricht das Training automatisch ab und der letztbeste Stand des Trainings wird wieder hergestellt (Keras, o. J. -b). Je nach Modell kann die optimale Epochenzahl im Training variieren. Gerade die Schwankungen im Trainingsverlauf mussten bei der Verwendung von Earrly-Stopping beachtet werden.

5.5 Validierung

Bei der Validierung geht es darum, zu erkennen, wie gut ein Modell performt. Das bedeutet, wie gut die Erkennungsrate ist und wo es allenfalls Schwachstellen, beziehungsweise Stärken gibt. Um das Modell auf seine Genauigkeit zu testen, wurden zwei Verfahren angewandt, ein Out-Of-Sample Test und eine K-Fold Cross Validation. Die beiden Verfahren werden nachfolgend erläutert und das eigene Vorgehen beschrieben.

5.5.1 Out-Of-Sample Test

In diesem Testverfahren wird das Modell nach dem Training auf Daten angewandt, welche das Modell bisher noch nie gesehen hat (Out-of-Sample Evaluation, 2017). Das bedeutet, diese Bilder sind weder im Trainings- noch im Validation Dataset enthalten. Diese Daten werden in einem separaten Testdataset geführt.

In der vorliegenden Arbeit wurde die Aufteilung der Daten wie folgt vorgenommen. Es wurden 20 Prozent der Daten aus dem Dataset für Validierung, 20 Prozent für Test und die restlichen 60 Prozent fürs Training verwendet. Der Splitt der Daten in die jeweiligen Datasets wurde zufällig vorgenommen. Dies wurde als wichtig erachtet, da zu Beginn des Projektes jeweils fälschlicherweise nur anhand des Verzeichnisses die Files sortiert ausgelesen wurden. Dabei gab es keine Durchmischung und die Erkennung war dadurch leicht beeinträchtigt.

5.5.2 Confusion Matrix

Weiter wurde das Modell nochmals auf das gesamte Testdataset von Kapitel 5.5.1 angewandt, um die Leistung anhand einer Confusion Matrix zu analysieren. Eine Confusion Matrix stellt die Klassifikationsleistung eines Modells in Bezug auf seine Testdaten dar (Confusion Matrix, 2017, S. 260). Gemäss Ting (Confusion Matrix, 2017, S. 260) handelt es sich dabei um eine zweidimensionale Matrix, in welcher eine Dimension die wahre Klasse und die andere Dimension die vorhergesagte Klasse abbildet. Anhand der Matrix ist somit ersichtlich, wie die Testdaten, mittels des Modells klassifiziert wurden. Somit kann festgestellt werden, mit welchen Klassen das Modell gut, beziehungsweise schlecht performt und allfällige Schwachpunkte können analysiert werden.

5.5.3 K-Fold Cross Validation

Die Menge an Bildern im Dataset spielt eine essenzielle Rolle für die resultierende Genauigkeit des Modells. Da durch den Out-Of-Sample Test 20% der Daten für ein Testdataset verwendet werden, fehlen 20% der Daten im Trainingsdataset, welche im Trainingsprozess verwendet werden könnten. Um dieses Problem lösen zu können und somit die insgesamt 80% der Datenmenge im Training zu verwenden, wurde eine zweite Methode der Evaluation durchgeführt, welche in diesem Kapitel vorgestellt wird.

Laut Wong (2015, S. 2842) ist die K-Fold Cross Validation ein beliebtes Verfahren zur Ermittlung der Leistung eines Klassifizierungsmodells. In der K-Fold Cross Validation werden die vorhandenen Image-Daten in zufällig, k-fach gesplittet, wobei das Verhältnis der Anzahl Bilder je Klasse bestehen bleibt (Wong, 2015, S. 2842). «K» steht hierbei für die Anzahl Splitts, auch «Fold» genannt. Für den Test des Modells wurde $k=5$ verwendet, was bedeutet, dass in fünf Folds geteilt wurde. In einem nächsten Schritt wurden aus den fünf Folds, fünf unterschiedliche Validation- und Trainingsdataset-Kombinationen erstellt.

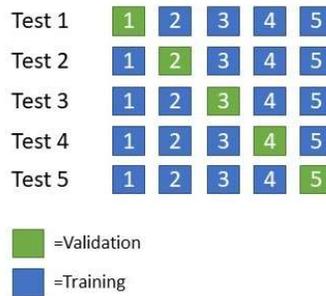


Abbildung 11: Training & Validation Datenzusammensetzung mit K-Fold Cross Validation (eigene Darstellung, nach Scikit Learn (o. J))

Wie in Abbildung 11 dargestellt, beschreibt jeweils ein unterschiedlicher Fold das Validierungsdataset. Die restlichen vier Folds ergeben zusammen das Trainingsdataset. Die eben beschriebenen Datasets wurden wie in Abbildung 12 dargestellt, in einer geschachtelten Verzeichnisstruktur abgelegt.

```

- ds_600_kfold_ds/
├── 0/
│   ├── training/
│   └── validation/
├── 1/
│   ├── training/
│   └── validation/
├── 2/
│   ├── training/
│   └── validation/
├── 3/
│   ├── training/
│   └── validation/
└── 4/
    ├── training/
    └── validation/
  
```

Abbildung 12: Verzeichnisstruktur mit den fünf Datasets

Das Modell wurde danach iterativ fünf Mal mit dem jeweiligen Training und Validation Dataset trainiert und die Ergebnisse der Validierung festgehalten. Die effektive Leistung des Modells wird in der K-Fold Cross Validation durch den Mittelwert der «K»-fachen Ergebnisse berechnet (Wong, 2015, S. 2842). Anhand der fünf Resultate wurde dieser Mittelwert berechnet und die effektive Leistung des entwickelten Modells konnte bestimmt werden.

5.6 Mobile App Entwicklung

Das entwickelte Modell wurde im Anschluss in ein Tensorflow-Lite Model konvertiert und zusammen mit den definierten Labels exportiert. In einer Zwischenprüfung wurde die Genauigkeit des konvertierten Modells nochmals auf Verluste geprüft, um sicherzustellen, dass das Modell immer noch gut performt. Im nächsten Schritt wurde der Prototyp entwickelt, um das trainierte Modell in der Praxis zu testen. Beim Prototyp handelt es sich um eine Mobile-App auf Basis von

Android. Für die Entwicklung wurde der IDE⁹ von Google, das Android-Studio benutzt. Als Programmiersprache wurde Kotlin verwendet, um die folgenden Use-Cases im Prototyp abzubilden:

- Foto aufnehmen
- Foto durch den Objekt-Klassifikator analysieren und korrekten Recyclingcontainer zurück liefern
- Resultat (Recyclingcontainer) in Form eines Textes, zusammen mit dem erstellten Foto darstellen

Eine erste Version des Prototyps konnte unabhängig vom entwickelten ML-Model entwickelt werden. Dabei wurde eine Anleitung von Google Codelabs (Katariya & Lam, o. J.) verwendet, welche ein Beispiel liefert, wie auf Basis eines Flower-Classifiers eine App entwickelt werden kann, um Blumen zur Laufzeit zu erkennen. Ziel war es den Prototypen frühzeitig lauffähig zu haben, um diesen später anzupassen und auf das eigene ML-Model umzustellen. Der Prototyp wurde später nochmals komplett umgeschrieben, da der Prototyp nicht das Kamerabild zur Laufzeit analysieren soll, wie es die Applikation nach der beschriebenen Anleitung tut, sondern ein einzelnes Foto aufgenommen werden sollte, welches dann analysiert werden kann.

Nach Fertigstellung des eigenen ML-Model wurde dieses in die App integriert. Um das Modell in der App verwenden zu können, wurde die Tensorflow Lite Library genutzt. Dabei musste darauf geachtet werden, dass die Input- und Output-Formate, mit derjenigen des Modells übereinstimmen. Hierzu wird das Input Image auf das grösstmögliche Quadrat zugeschnitten und auf 224 mal 224 Pixel herunterskaliert und normalisiert. Der Output in Form eines Arrays, bestehend aus den Wahrscheinlichkeiten je Klasse, wird dann wiederum normalisiert und auf die übersetzten Klassenbeschreibungen gemappt. Die Klasse mit dem höchsten Wahrscheinlichkeitswert bildet schlussendlich das Resultat der Klassifizierungsaufgabe.

Android Studio bietet die Möglichkeit, die App in einem Emulator zu starten, um diese zu debuggen und zu testen. Dabei kann eine am Computer angeschlossene Kamera angesteuert werden. Weiter kann die App auch direkt auf einem, mit dem Computer via USB verbundenen Android Gerät, deployt werden, um die App so auf dem Smartphone zu testen. So konnte in der Testphase des Prototyps, sowohl die funktionalen Anforderungen überprüft, als auch die Performance der App analysiert werden. Im nächsten Kapitel wird der Praxistest der Mobile App näher beschrieben.

5.7 Feldtest

Zum Abschluss wurde der Prototyp im Recyclinghof Hagenholz einem Praxistest unterzogen. Dabei wurden zufällige Objekte aus verschiedenen Containern entnommen und mit der App

⁹ Integrated Development Environment

abfotografiert. Spezielle Beachtung wurde dabei den aus der Validierung erkannten Schwachstellen des Modells geschenkt. Die Resultate der App wurden mit einem internen Mitarbeiter des Recyclinghofs Hagenholz untersucht und auf ihre Richtigkeit hin geprüft. Die Resultate wurden einerseits schriftlich in einem Testprotokoll festgehalten und andererseits wurden Screenshots der jeweiligen Resultate des Prototyps gemacht.

5.8 Technologie-Stack

Für das Vorverarbeiten und Erstellen des Datasets, sowie für die Entwicklung und das Training der Klassifizierungsmodelle wurde jeweils die Programmiersprache Python in der Version 3.8.8 eingesetzt. Als Entwicklungsumgebung wurde Visual Studio Code (v.1.55.2) verwendet. Dabei wurde ein virtuelles Environment für dieses Projekt eingerichtet, in welchem alle benötigten Packages installiert wurde. Zudem wurden die CUDA 11, sowie der NVIDIA GPU Treiber installiert, um die Grafikkartenunterstützung mit Tensorflow zu gewährleisten und damit die Trainingszeiten zu verkürzen. Dabei wurde nach der offiziellen Anleitung von Tensorflow vorgegangen (TensorFlow, o. J).

Im Rahmen dieser Arbeit wurden diverse Python Bibliotheken eingesetzt. Im zugehörigen GitHub Repository befindet sich ein «requirements.txt» mit einer vollständigen Liste der verwendeten Komponenten. Nachfolgend werden die wichtigsten Bibliotheken nochmals aufgeführt.

Für die Entwicklung der Modelle und die Verwendung der vortrainierten Modelle sind die folgenden Libraries zu erwähnen:

- Tensorflow (v. 2.4.1)
- Tensorflow Hub (v. 0.11.0)

Für die Erstellung der Grafiken, Tabellen und der Confusion Matrix wurden nachfolgende Komponenten verwendet:

- Pandas (v. 1.2.3)
- Matplotlib (v. 3.3.4)
- Sklearn (v. 0.24.1)
- Seaborn (v. 0.11.1)

Zudem ist die nachfolgende Bibliothek zu erwähnen, welche im Image Scraper benutzt wurde, um mittels Google Chrome Engine Bilder aus dem Netz zu suchen:

- Selenium WebDriver (v. 3.14.1)

Die Entwicklung der Mobile-App wurde in der Programmiersprache Kotlin und Java umgesetzt. Hierzu wurde Android Studio 4.2 von Google, als IDE verwendet. Die Kamerasteuerung wurde

mit dem «camera2»-Library umgesetzt und für die Verwendung des erstellten Modells wurde die Library «tensorflow-lite» sowie «tensorflow-lite-support» verwendet. Zudem wurde die Applikation auf einer Virtualisierung von einem «Google Pixel 2» mit Android Version 10 getestet, bevor die Applikation dann auf einem «Huawei Mate 10» ebenfalls mit Android Version 10 installiert wurde, welche im Feldtest zum Einsatz kam.

6 Resultate

Das im Rahmen dieser Arbeit erarbeitete Image-Dataset, sowie der Python Code für die Erstellung der Modelle und der Code des Prototyps sind unter folgendem GitHub Repository zu finden:

github.com/zhaw-iwi/AI-Recycling/

Das Repository ist in drei Bereiche unterteilt. Unter «datasets» ist das finale Dataset, sowie die einzelnen Teil- und Validierungs-Datasets enthalten. Unter «ml.models» sind die Skripte zur Erstellung der ML-Modelle, sowie der Datenaufbereitung zu finden und unter «android.app» befindet sich der Code des entwickelten Prototyps.

Nachfolgend werden die einzelnen Resultate vorgestellt. Zuerst wird das erstellte Image-Dataset dargelegt, gefolgt von den Resultaten der drei Modelle und im Anschluss wird der entwickelte Prototyp präsentiert. Zum Schluss wird aufgezeigt, wie der Prototyp im Feldtest abschnitten hat.

6.1 Image-Dataset

Die im Rahmen dieser Arbeit erhobene Gesamtzahl der Bilder, beläuft sich auf insgesamt 18'520 Images, wobei 842 davon selbst angefertigt wurden, 1472 aus dem TrashNet Dataset von Yang und Thung (2016) stammen und 16206 Bilder mittels Scraping via Google bezogen wurden. In der untenstehenden Abbildung 13 wird gezeigt, wie die Zusammensetzung der einzelnen Klassen bezüglich Herkunft der Daten aufgeteilt ist. Die Klassen sind nach dem Total aus eigener Sammlung, Scraped und TrashNet, hier als «kaggle» benannt, sortiert.

Die Unterschiedliche Anzahl Bilder je Kategorie ist durch die unterschiedlichen Schwierigkeitsgrade, um an die Daten heranzukommen, zu erklären. Die Kategorie «Elektro» bildet beispielsweise eine Vielzahl unterschiedlicher Elektrogeräte ab. Somit konnte mit einer Fülle von Suchbegriffen, eine viel grössere Menge qualitativ guter Bilder gesammelt werden als beispielsweise bei der Kategorie «PET-Getränkeflaschen» (in der Abbildung 13 als «pet» aufgeführt), welche aus einer Gattung von Objekten besteht. Aus dieser Gesamtsumme an Bildern wurde das finale Dataset erstellt.

class	own	scraped	kaggle	total
compressor	1.0	0.0	0.0	1.0
plastic_bottle	37.0	494.0	62.0	593.0
pet	29.0	412.0	155.0	596.0
mixed_material	18.0	581.0	0.0	599.0
screen	15.0	585.0	0.0	600.0
trash	0.0	377.0	240.0	617.0
tire	0.0	620.0	0.0	620.0
battery	17.0	607.0	0.0	624.0
alu_can	54.0	508.0	69.0	631.0
styrofoam	8.0	626.0	0.0	634.0
special	6.0	636.0	0.0	642.0
lithium_battery	27.0	636.0	0.0	663.0
pan	12.0	656.0	0.0	668.0
coffee_capsule	17.0	652.0	0.0	669.0
cd_dvd	46.0	629.0	0.0	675.0
cork	19.0	714.0	0.0	733.0
cardboard	40.0	298.0	403.0	741.0
book	29.0	714.0	0.0	743.0
nonferrous_metal	35.0	726.0	0.0	761.0
paper	0.0	253.0	543.0	796.0
metal	157.0	761.0	0.0	918.0
textile	0.0	1022.0	0.0	1022.0
construction_waste	157.0	936.0	0.0	1093.0
bulky_rubbish	25.0	1231.0	0.0	1256.0
electro	93.0	1532.0	0.0	1625.0

Abbildung 13: Übersicht der gesammelten Bilder

Um alle Klassen gleich stark zu Gewichten, wurde das Dataset auf 600 Bilder pro Klasse festgelegt. So sollte verhindert werden, dass beim Training einzelne Kategorien übergewichtet werden. Die Anzahl 600 wurde bestimmt, da das Maximum an Bildern pro Klasse erreicht werden soll und mit 600 Bildern nur drei Kategorien (Plastikflaschen, PET und Mischmaterialien) knapp darunter liegen, mit maximal 1.17 Prozent Abweichung.

In der Zusammensetzung des Datasets wurde darauf geachtet, dass die Bilder aus möglichst realitätsnahem Kontext kommen. Es wurde dafür das Python-Skript «dataset_merger.py» geschrieben, welches die drei Datasets zu einem zusammenführt. Dabei werden die selbst erstellten Bilder aus dem Recyclinghof bevorzugt, danach zufallsmässig weitere Bilder aus dem Scraping-Dataset angereichert und zum Schluss, wird mit den Bildern aus dem TrashNet-Dataset nach Bedarf, auf das Maximum von 600 aufgefüllt. Dabei werden die Bilder ebenfalls zufällig ausgewählt.

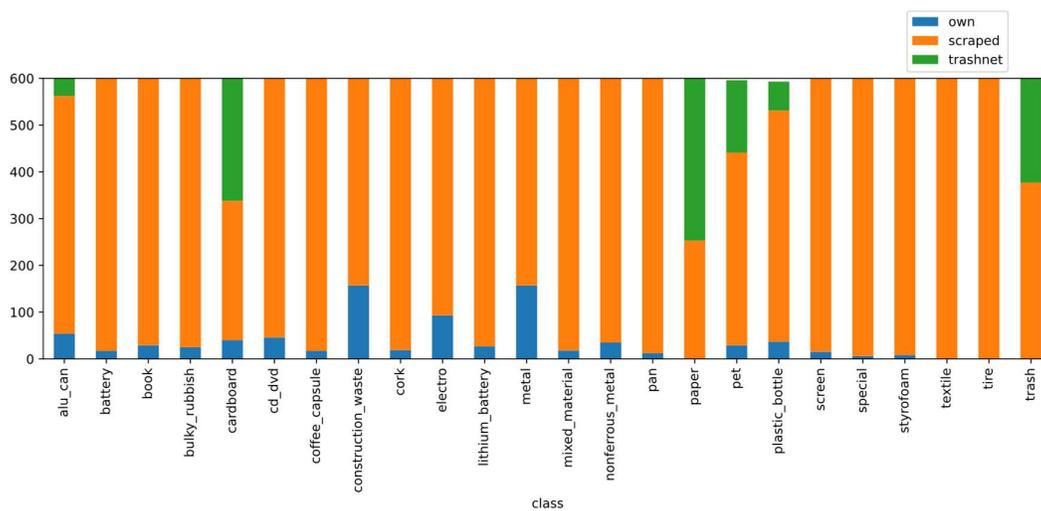


Abbildung 14: Herkunft der Bilder im Dataset

Wie in Abbildung 14 ersichtlich, stellt sich das finale Dataset wie folgt zusammen: Blau dargestellt sind die selbsterstellten Bilder, orange die gescrapten und grün sind die Bilder aus dem TrashNet Dataset dargestellt.

Die Klasse «compressor» wurde bewusst weggelassen, da diese eher eine ungewöhnliche Kategorie darstellt, welche wenig genutzt wird. Zudem wird in dieser Kategorie ohnehin meist fachmännische Hilfe durch das Personal des Recyclinghofs Hagenholz benötigt, da Kompressoren gross und schwer sind.

6.2 ML-Modell

Wie die in Kapitel 5.3 entwickelten Modelle performen, wird nachfolgend beschrieben. Den Anfang macht das einfache Modell, dann werden die Resultate vom Retrained Model auf Basis von EfficientNet-B0 vorgestellt und zum Anschluss wird auf die Resultate vom Retrained Model auf Basis von MobileNetV3 eingegangen. Dabei werden jeweils die Resultate vom Out-Of-Sample Test sowie von der K-Fold Cross Validation separat aufgeführt.

6.2.1 Einfaches Modell (From Scratch)

Im einfachen Modell konnten mit Optimierung in der Augmentation, mit «horizontal flip», «random rotation» = 0.25 und «random zoom» = 0.2, sowie mit dem gewählten Dropout von 0.4 die besten Resultate erzielt werden. Nachfolgend werden die Resultate der beiden Validierungsverfahren dargelegt.

6.2.1.1 Out-Of-Sample Test

In Abbildung 15 ist ersichtlich, wie sich die Genauigkeit im Trainingsverlauf, über insgesamt 24 Epochen stetig verbessert. Anzumerken ist, dass das Modell ab Epoche 14 nur noch kleine Fortschritte macht und immer mehr in ein Overfitting läuft.

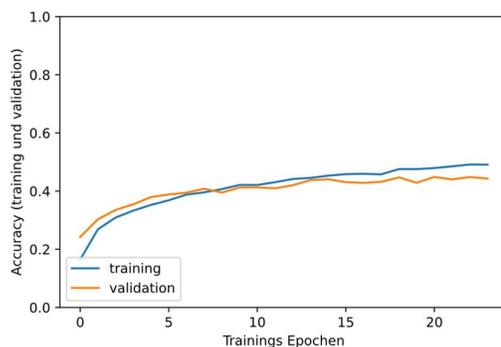


Abbildung 15: Accuracy in Out-Of-Sample Test mit dem einfachen Modell

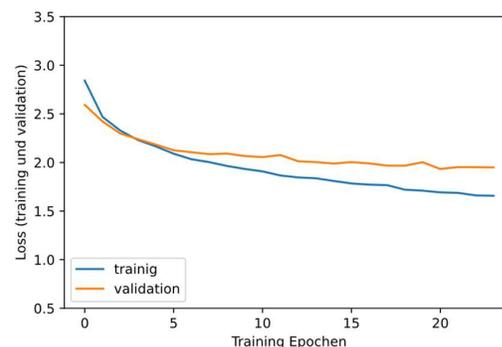


Abbildung 16: Loss in Out-Of-Sample Test mit dem einfachen Modell

Wie in Tabelle 2 ersichtlich, kann eine Validation Accuracy von 44.9 Prozent, mit einem Validation Loss von 1.933 erreicht werden. Angewandt auf das Testdataset, kann sogar 46.8 Prozent

Erkennung, bei einem Loss von 1.872 erreicht werden. Das Modell performt demnach fast 2 Prozent besser mit den Testdaten, wie mit Validierungsdaten, was eher ungewöhnlich ist.

	<i>Accuracy</i>	<i>Loss</i>
Validation	44.873 %	1.933
Test	46.769 %	1.872

Tabelle 2: Accuracy und Loss Resultate im Out-Of-Sample Test mit dem Einfachen Modell

6.2.1.2 K-Fold Cross Validation

Die Resultate der K-Fold Cross Validation sind den Resultaten des Out-Of-Sample Testresultates sehr ähnlich. Wie in Tabelle 3 ersichtlich, wurde Performance-Mittelwert von rund 46.2 Prozent erreicht. Dies widerspiegelt sich auch im Trainingsverlauf.

	<i>Accuracy</i>	<i>Loss</i>
1	44.835 %	1.869
2	48.402 %	1.750
3	45.951 %	1.849
4	45.379 %	1.904
5	46.474 %	1.814
Overall	46.208 %	1.837

Tabelle 3: K-Fold Cross Validation Resultate im einfachen Modell

Wie in Anhang C. abgebildet, weisen die Trainingsverläufe jeweils erhebliche Unterschiede auf. Ein Unterschied von rund 3.6 Prozent zwischen Testlauf 1 und 2 kann dabei festgestellt werden.

6.2.2 Retrained Model MobileNetV3

Bevor die Resultate des umgeschulten Modells mit MobileNetV3 dargelegt werden, soll kurz auf die Selektion der verwendeten Hyperparameter und die Augmentation eingegangen werden. Die Hyperparameter wurden durch das Skript «param_optimization.py» automatisiert evaluiert. Die Evaluation verschiedener Dropout-Größen zwischen 10 bis 40 Prozent hat ergeben, dass das Modell am besten mit einem Dropout von 30 Prozent performt. Während die Accuracy zuerst mit wachsendem Dropout besser wird, fällt diese mit 40 Prozent, wie in Tabelle 4 dargestellt.

Dropout	Accuracy	Loss
10%	82.37 %	1.488
20%	82.02 %	1.478
30%	83.13 %	1.454
40%	82.64 %	1.463

Tabelle 4: Dropout Evaluation

Weiter wird aus der Evaluation verschiedener Werte für L2-Regularization ersichtlich, dass für das Modell das Optimum der Erkennungsrate beim Wert 0.0001 für L2 erreicht werden kann.

L2	Accuracy	Loss
0.1	79.409	1.718
0.01	82.818	1.542
0.001	82.817	1.495
0.0001	83.304	1.458
0.00001	82.643	1.476

Tabelle 5: L2-Regularization Evaluation

Wie in Tabelle 5 ersichtlich, resultiert eine bessere Accuracy, je kleiner der Wert für L2 gewählt wird, bis bei einem L2-Wert von 0.00001 die Performance wieder nachlässt. Durch die Optimierung der Augmentation, sowie durch eine Durchmischung der Trainingsdaten, konnte das Modell iterativ weiter verbessert werden. Im finalen Modell werden Horizontale Spiegelung, Rotationen von plus und minus 45 Grad, Verschiebungen in Höhe und Breite von 0.1, Verzerrungen von 15 Prozent, Vergrößerung von 0-20 Prozent und Helligkeitsanpassungen von 0.3 bis 1.4 vorgenommen.

6.2.2.1 Out-Of-Sample Test

Um zu verhindern, dass das Modell Overfitted, wurden zwei «EarlyStopping»-Callbacks verwendet. Damit bricht das Training ab, sobald der Training-Loss anstatt weiter fällt, mehr als einmal wieder ansteigt (patient=1) oder die Validation-Accuracy sich nicht mehr weiter verbessert, sondern sich mehr als zweimal hintereinander verschlechtert (patient=2). Wie in Abbildung 17 und Abbildung 18 zu sehen, wurde das Modell in 15 Epochen trainiert. Während die Trainings Accuracy sich bis zur letzten Epoche verbessert, kann eine gewisse Stagnierung der Validation Accuracy in Abbildung 17 festgestellt werden. Dasselbe gilt für den Trainings Loss, welcher fortlaufend fällt, während der Validation Loss in immer kleineren Werten fällt, wie in Abbildung 18 dargestellt. Dies bedeutet, dass das Modell, trotz der vorgenommenen Massnahmen gegen Overfitting, immer noch eine Überanpassung aufweist.

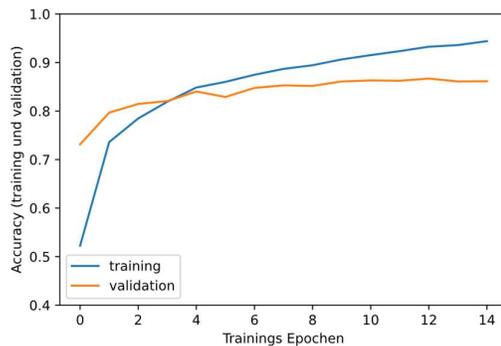


Abbildung 17: Accuracy in Out-Of-Sample Test mit dem Retrained MobileNetV3 Modell

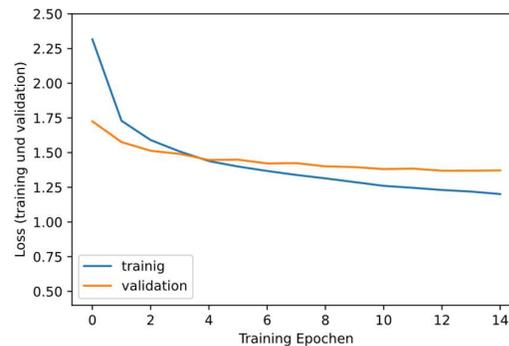


Abbildung 18: Loss in Out-Of-Sample Test mit dem Retrained MobileNetV3 Modell

Das Modell hat wie in Tabelle 6 dargestellt, die folgenden Resultate erzielt: Wird das trainierte Modell auf das Validierungsdataset angewandt, kann eine Genauigkeit von rund 86.6 Prozent im

In-Sample Test festgestellt werden. Im Out-Of-Sample Test, daher angewandt auf das Testdataset, erreicht das Modell immerhin noch eine Genauigkeit von 85.9 Prozent.

	Accuracy	Loss
Validation	86.583 %	1.370
Test	85.928 %	1.402

Tabelle 6: Accuracy und Loss Resultate im Out-Of-Sample Test mit dem Retrained MobileNetV3 Modell

Wirft man einen Blick in die Confusion Matrix (CM) in Abbildung 19, so ist ersichtlich, dass nicht alle Klassen gleich gut performen. In der CM sind vertikal alle wahren Klassen aufgeführt und in der Horizontale, befinden sich die vorhergesagten Klassen. Somit ergibt sich in der Diagonale von oben links nach unten rechts der Bereich, in welchen die Vorhersage mit der tatsächlichen Klasse übereinstimmt. In den umliegenden Feldern, welche nicht auf dieser Diagonale liegen, befinden sich die falsch erkannten Bilder. Zur Verdeutlichung der Resultate bezüglich deren Genauigkeit, sind die jeweiligen Bereiche in Abbildung 19 nach prozentuaalem Anteil je Klasse eingefärbt. Je dunkler das Feld, desto mehr Testbilder einer Klasse wurden dabei in diesem Bereich erkannt. Die Mehrzahl der Klassen wurden demnach sehr gut erkannt, denn in rund 18 der 24 Klassen wurden zwischen 80 bis 97.6 Prozent korrekt vorhergesagt. Es gibt aber auch Schwachstellen, denn die Klasse Buntmetall, hier aufgeführt unter dem Begriff «nonferrous_metal», weist gerade mal eine Genauigkeit von 69.2 % auf. Auffällig ist bei dieser Klasse die Korrelation mit Altmetall «metal» und Bauschutt «construction_waste». Rund 12 Prozent des Buntmetalls wurde als Altmetall klassifiziert und andersherum wurden 3.3 Prozent der Altmetall Objekte als Buntmetall klassifiziert. Die Kategorien sind auch in der Realität sehr nahe beieinander und daher auch für das Modell in gewissen Situationen schwierig zu unterscheiden. Die Korrelation mit Bauschutt beschreibt sich durch die falsch klassifizierten 2.5 Prozent Buntmetall Objekten, welche als Bauschutt und die 5.8 Prozent Bauschutt, welche als Buntmetall erkannt wurden. Der Zusammenhang der Kategorien ist auf den ersten Blick nicht nachvollziehbar, da es sich um generell unterschiedliche Materialien handelt. In Anbetracht der Farben dieser Objekte, kann aber eine gewisse Übereinstimmung festgestellt werden, gerade bei Tontöpfen und ähnlichem, welches Kupfer oder Messing sehr gleichen kann. Die bereits erwähnte Kategorie Altmetall weist neben der Korrelation mit Buntmetall ebenfalls eine Wechselwirkung mit den Kategorien Mischmaterialien «mixed_material» mit 8.3, beziehungsweise 5 Prozent auf. Dies überrascht nicht wirklich, da gerade Mischmaterialien sich durch den Inhalt von Metall von Sperrgut unterscheidet, was bedeutet, dass diese Mischmaterialien auch zu einem gewissen Grad aus Altmetall bestehen. Wenn man sich die Klasse Elektro «electro» anschaut, kann festgestellt werden, dass die Objekte mit rund 7.5 Prozent als Lithiumbatterien «lithium_battery» und 5.8 Prozent als Metall falsch erkannt werden. Auch umgekehrt werden Objekte der Kategorie Lithium Batterie mit 5.8 Prozent und rund 2.5 Prozent der Metallobjekte als Elektro klassifiziert. Gerade die Klasse Elektro ist sehr

umfangreich, und viele der Elektroobjekte bestehen zu einem grossen Teil aus Metall, wie beispielsweise eine Stehlampe. Ebenso bestehen Objekte der Klasse Lithiumbatterie aus Elektrogeräten, welche einen Lithium-Akku enthalten, wie beispielsweise ein Notebook. Daher ist es nicht verwunderlich, dass das Modell hier teilweise Schwierigkeiten hat, diese Klassen sauber zu klassifizieren. Die stärkste Korrelation tritt allerdings zwischen Büchern «book» und CD & DVDs «cd_dvd» auf.

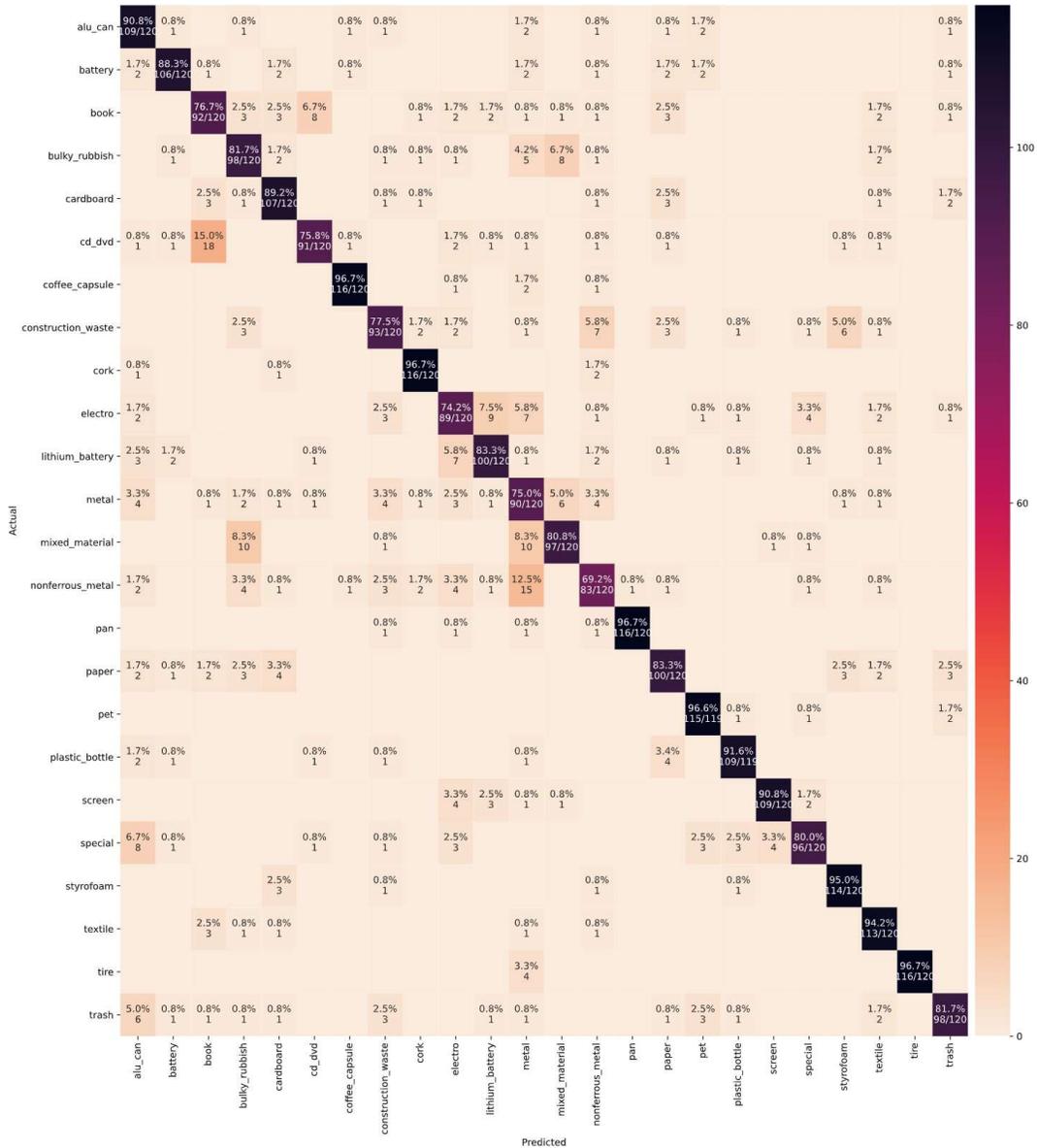


Abbildung 19: Confusion Matrix anhand des Testdatasets im Out-Of-Sample Test des Retrained MobileNetV3 Models

Rund 15 Prozent der Testdaten der Kategorie CD & DVD wurde fälschlicherweise als Bücher, und rund 6.7 Prozent der Bücher wurde als CD & DVDs klassifiziert. Was hier für das menschliche Auge viel einfacher zu unterscheiden ist, scheint dem vorliegenden Modell viel Schwierigkeiten zu bereiten. Der Grund dafür wird in der sehr ähnlichen Aufmachung von Büchern und

CDs bzw. DVD-Hüllen gesehen. Diese ähneln sich sowohl in Grösse und Format wie auch in Farbe und Material.

Weiter sind die Fehlzuzuweisungen von Abfall «trash» mit 5.0 Prozent und Sonderabfall «special» mit 6.7 Prozent, jeweils erkannt als Alu-Dosen «alu_can», zu erwähnen. Beide Kategorien sind im Dataset verhältnismässig schlecht abgebildet, da diese eine Vielzahl an Unterobjektarten umfassen und jeweils eher eine Ausschlusskategorie bilden. Das bedeutet, in Abfall wird beispielsweise alles entsorgt, was zu keiner anderen Kategorie gehört und bei Sonderabfall gehören jegliche Objekte von Sondermüll, welche in der Abteilung Sondermüll wieder in verschiedene Kategorien unterteilt werden. Somit ist es sehr schwierig diese Kategorien mit einem Dataset sauber abbilden zu können.

6.2.2.2 K-Fold Cross Validation

Die Trainingsverläufe in den 5 Einzeltests sind sich sehr ähnlich. Wie in Abbildung 20 dargestellt, verlaufen die jeweiligen Trainings in Bezug auf deren Accuracy ziemlich genau aufeinander. Nur minimale Abweichungen sind erkennbar. Einzig in Test 3 kann eine gröbere Abweichung in Bezug auf die Anzahl trainierten Epochen festgestellt werden. Während die anderen 4 Tests nach 14, beziehungsweise 15 Epochen das Training abgebrochen haben, konnte das Training in Test 3, bis Epoche 19 fortgesetzt werden. Auch der Loss, in Abbildung 21 dargestellt, verläuft bei allen fünf Trainings praktisch auf gleicher Linie.

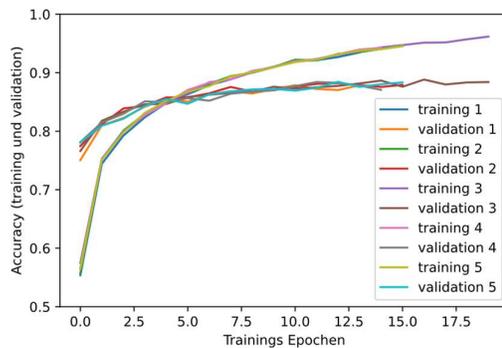


Abbildung 20: Accuracy in K-Fold Cross Validation mit dem Retrained MobileNetV3 Modell

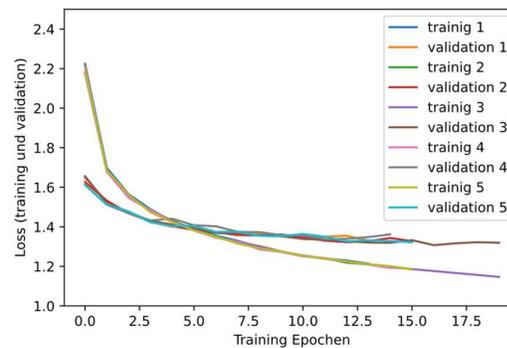


Abbildung 21: Loss in K-Fold Cross Validation mit dem Retrained Mo-bileNetV3 Modell

Das Resultat spricht mit seinen gegenseitig bestätigenden Resultaten für seine Aussagekraft. Zu beachten ist, dass das Modell trotz der strikten Nutzung von Early Stopping, generell overfittet. Von Epoche 7 an, verbessert sich die Validation Accuracy nur noch minimal, während sich die Trainings Accuracy weiter an 100 Prozent annähert. Ein Early Stopping mit einer «Patient» von weniger als 2 führt jedoch dazu, dass es die kleinen Schwankungen im anfänglichen Training nicht übersteht und so die Genauigkeit doch um einige Prozent schlechter wird.

	<i>Accuracy</i>	<i>Loss</i>
1	87.757 %	1.343
2	88.221 %	1.324
3	88.877 %	1.307
4	88.360 %	1.340
5	88.503 %	1.326
Overall	88.344 %	1.328

Tabelle 7: Resultate in K-Fold Cross Validation mit dem Retrained MobileNetV3 Modell

Wie in Tabelle 7 dargestellt, konnte in jedem der fünf Testdurchläufe jeweils um die 88 Prozent erreicht werden. Die daraus resultierende Overall Accuracy liegt dabei bei ca. 88.3 Prozent, bei einem durchschnittlichem Validation Loss von 1.328.

6.2.3 Retrained Model EfficientNet-B0

Im Retrained Model auf Basis von EfficientNet-B0, wurde sowohl auf Fine-Tuning wie auch auf Augmentation verzichtet. Wie bereits erwähnt, bestand keine Möglichkeit Fine-Tuning in diesem Modell anzuwenden, da die vorhandene Entwicklungsumgebung nicht genügend Ressourcen dafür bietet. Da das Modell generell bereits unterfittet, wurde demnach auch auf Augmentation verzichtet, da damit das Modell auch nicht weiter verbessert werden konnte. Wie das Modell performt, wird in den beiden nächsten Unterkapitel dargelegt.

6.2.3.1 Out-Of-Sample Test

Mit einem Blick auf den Trainingsverlauf wird ersichtlich, dass im Out-Of-Sample Test mit dem Retrained Model auf Basis von EfficientNet-B0, die Trainings Accuracy durchwegs schlechter ausfällt als die Accuracy in der Validierung, wie in Abbildung 22 dargestellt. Zudem ist in Abbildung 23, der Verlauf des Loss während des Trainings ersichtlich, bei welchem der Loss der Validierung stets unter dem Loss-Wert des Trainings liegt. Dies sind beides Anzeichen, dass das Modell unterfittet.

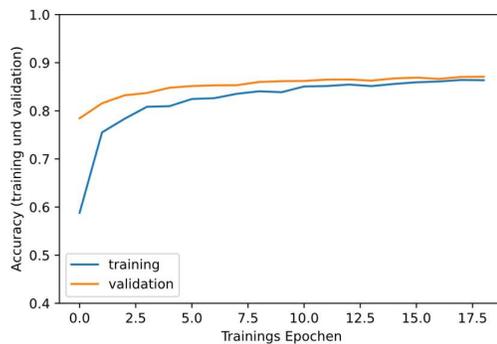


Abbildung 22: Accuracy in Out-Of-Sample Test mit dem Retrained EfficientNet-B0 Model

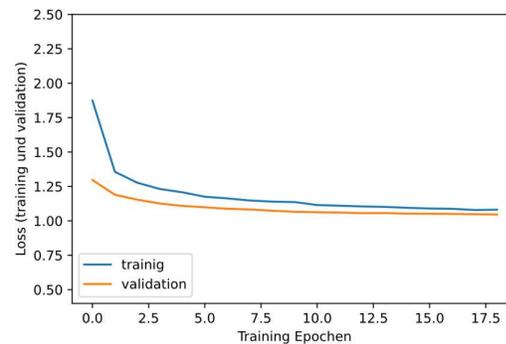


Abbildung 23: Loss in Out-Of-Sample Test mit dem Retrained EfficientNet-B0 Model

Ebenso ist zusehen, dass das Modell nach 19 Epochen, durch den Early-Stoppp abgebrochen wurde, da sich die Genauigkeit nicht mehr weiter verbessern konnte. In Tabelle 8, werden die

Resultate der Validierung dargestellt. Im In-Sample Test, konnte das Modell eine Accuracy von 87.1 Prozent erreichen mit einem Loss von 1.047.

	Accuracy	Loss
Validation	87.070 %	1.047
Test	86.727 %	1.053

Tabelle 8: Resultate in Out-Of-Sample Test (Retrained EfficientNet-B0 Model)

Das Resultat des Out-Of-Sample Tests, mit einer Accuracy von 86.7 Prozent, und einem Loss von 1.053 unterscheidet sich dabei kaum vom In-Sample Testresultat.

6.2.3.2 K-Fold Cross Validation

Auch in der K-Fold Cross Validation hat das Modell mit Underfitting zu kämpfen, wie der Trainingsverlauf, dargestellt in Abbildung 24 und Abbildung 25, aufzeigt. Während der Validation Loss in Abbildung 25 jeweils in allen Einzeltests nahe beieinander liegt, weist die Accuracy im Training, wie in Abbildung 24 dargestellt, eine minimale Abweichung auf.

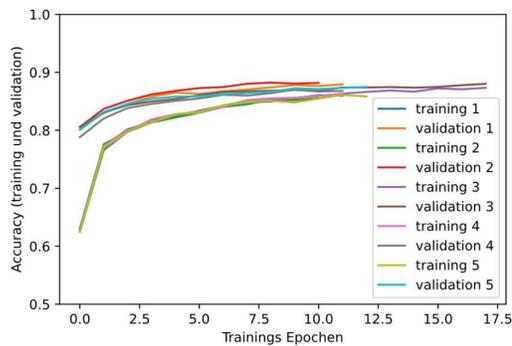


Abbildung 24: Accuracy in K-Fold Cross Validation mit dem Retrained EfficientNet-B0 Model

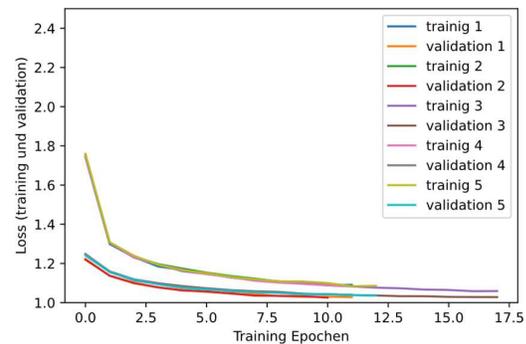


Abbildung 25: Loss in K-Fold Cross Validation mit dem Retrained EfficientNet-B0 Model

Mit einem Blick auf die Resultate des Tests in Tabelle 9, widerspiegeln sich diese oben erwähnten Abweichungen. Rund 1.2 Prozent Unterschied besteht zwischen dem besten Training Nr. 2 mit einer Genauigkeit von 88.2 Prozent und dem schlechtesten, Nr. 4 mit der Genauigkeit von 87.0 Prozent.

	Accuracy	Loss
1	87.548 %	1.032
2	88.186 %	1.037
3	87.800 %	1.029
4	86.970 %	1.045
5	87.391 %	1.037
Overall	87.579 %	1.036

Tabelle 9: Resultate der K-Fold Cross Validation mit dem Retrained EfficientNet-B0 Model

Mit einem Mittelwert von 87.58 Prozent, welcher das Gesamtergebnis der K-Fold Cross Validation darstellt, kann dieses Modell ohne Fine-Tuning und Augmentation bereits ähnliche Resultate erzielen wie das in Kapitel 6.2.2 vorgestellte Retrained Model mit MobileNetV3.

6.3 Mobile App

In der Mobile App namens «AI-Recycling», wurde das Retrained Model auf Basis von MobileNetV3 verwendet. Der fertige Prototyp in Form einer Android Mobile Applikation ist lauffähig auf einem Android Smartphone ab Android Version 5.0 Lollipop mit integrierter Kamera. Beim ersten Start der Applikation wird der User gebeten, der Applikation Zugriff auf die Kamera zu geben. Diese Rechte sind Voraussetzung für die weitere Nutzung der Applikation.

Der Prototyp besteht aus zwei Views. Auf dem Main-Screen kann der User wie in Abbildung 26 ersichtlich, ein Foto eines zu analysierenden Objektes aufnehmen. Dies geschieht über den einzigen Button, welcher unten mittig platziert wurde. Sobald der User diesen Button betätigt, wird ein Foto aufgenommen und im darauffolgenden Schritt automatisch die Vorhersage über die Recyclingkategorie ermittelt. Dabei wird die Recyclingkategorie mit der höchsten Wahrscheinlichkeit, zusammen mit dem Aufgenommenen Foto an die zweite View übergeben und wie in Abbildung 27 dargestellt, dem User präsentiert.



Abbildung 26: Prototyp Main-View



Abbildung 27: Prototyp Result-View

Möchte der Benutzer weitere Objekte analysieren, kann er über die Schaltfläche mit «OK» die Resultate schliessen und gelangt wieder zur Main-View. Mit diesem Prototyp können so beliebig viele Objekte nacheinander analysiert werden.

Wie der Prototyp im Recyclinghof Hagenholz im Praxistest abgeschnitten hat, wird im nachfolgenden Kapitel aufgezeigt.

6.4 Feldtest

Nach einem ersten internen Test der Mobile-App mit Objekten aus dem Haushalt, wie im Anhang D. beschrieben, wurde ein abschliessender Feldtest vorgenommen. Im Recyclinghof Hagenholz wurde der Prototyp dafür 40 Einzeltests unterzogen, welche in einem Testprotokoll (siehe Anhang E.) festgehalten wurden. Zudem wurden die jeweiligen Resultate per Screenshot festgehalten, wie in Anhang F. zu sehen ist.

In Anbetracht der Resultate des Modells in der Confusion Matrix in Kapitel 6.2.2.1, wurde ein spezielles Augenmerk auf die Kategorien «Elektro», «Metall» und «Buntmetall» aber auch auf «CD & DVD», sowie «Bücher» gelegt, da diese die Kategorien mit der schlechtesten Erkennungsrate sind. Auffällig war, dass die Tests mit Objekten der Kategorie «Elektro» im Feldtest trotzdem gut abgeschlossen haben. In Testfall 5, 6, 7, 8 und 38 konnten die Elektro-Objekte ohne weiteres erkannt werden. Ebenso konnten Objekte in den Tests 27, 28, 29, 30 und 31 der Kategorie «Metall» auf Anhieb erkannt werden. Der Test 26, ebenfalls der Kategorie «Metall» angehörig, konnte jedoch nicht erkannt werden. Es handelt sich dabei um einen länglichen Metallgegenstand, welcher ein wenig einem Kabel ähnelt. Das könnte allenfalls ein Grund sein, wieso dieser als «Elektro»-Objekt klassifiziert wurde. Generell wurden 32 von 40 Objekte korrekt klassifiziert und drei der acht falsch erkannten Objekte konnten in einem zweiten Versuch dennoch richtig erkannt werden. Im Test 10 wurde eine Art Vase, welche der Kategorie «Bauschutt» angehört, als «Buntmetall» erkannt. Gemäss Rücksprache mit Herrn Frei, Sachbearbeiter Recyclinghof Hagenholz und Herrn Di Fonzo, Gruppenleiter Recyclinghof Hagenholz (persönliche Kommunikation, 27.05.21) sieht das Objekt einem Messinggegenstand sehr ähnlich. Dies könnte ein möglicher Grund sein, wieso das Objekt als «Buntmetall» klassifiziert wurde.

In einem zweiten Versuch wurde das Objekt aus einem anderen Blickwinkel fotografiert und konnte in diesem Versuch (10.2) diesmal korrekt zugewiesen werden. Im Test 39 wurde eine PET-Flasche getestet. Anstatt «PET» wurde sie als «Alu-Dose» erkannt. Diese Fehlerkennung konnte nicht weiter begründet werden. In einem zweiten Versuch konnte dieselbe PET-Flasche jedoch korrekt erkannt werden. Dasselbe Phänomen trat auch im Test 3 auf. Dort wurde ein Buch getestet und anstatt der Kategorie «Bücher», wurde die Kategorie «CD&DVD» erkannt. Dabei handelt es sich aber gemäss Confusion Matrix um ein bekanntes Erkennungsproblem zwischen diesen beiden Kategorien. Auch hier konnte das Buch in einem zweiten Versuch (4) dann korrekt erkannt werden. Eine sehr schlechte Erkennungsrate wurde in der Sammelkategorie «Sperrgut» festgestellt. Dabei wurden die Tests 32, 34 und 37 falsch klassifiziert. Nur in Test 33, in welchem ein Stuhl getestet wurde, konnte das Objekt korrekt erkannt werden.

Es liegt nahe, dass das Modell zu wenig gut auf diese Klasse trainiert wurde. Denn gerade diese Sperrholzobjekte in den Tests 32, 34 und 37 wurden zu wenig im Dataset abgebildet. Der Grossteil der Bilderdaten der Kategorie «Sperrgut» besteht nämlich aus ganzen Möbeln und Matratzen und nicht aus einzelnen Holzresten. In den übrigen Tests anderer Kategorien wie «Plastikflaschen», «Batterien», «Buntmetall» oder «CD&DVD» konnte der Prototyp die Objekte erkennen. Auch die Objekte der Kategorie «Lithium-Akku» konnten die üblichen Objekte wie Smartphone, Tablets und Notebooks ohne weiteres erkannt werden. Eine Ausnahme bildet jedoch Test 22, in welchem ein Akku einer Bohrmaschine getestet wurde. Dieser wurde als «Elektro» klassifiziert. Diese Art Objekt wurde auch nicht im Dataset abgebildet und wurde dem Anschein nach dadurch auch nicht erkannt.

7 Diskussion

In der Confusion Matrix des Retrained Models auf Basis von MobileNetV3, konnten einige Schwachstellen des Modells aufgezeigt werden. Trotz der Begründbarkeit der beschriebenen Fehlerquellen, wie der Ähnlichkeit in Form und Farbe, oder dem Vorkommen von Materialien wie, Metall in verschiedenen Kategorien, ist die falsche Erkennung dieser Kategorien nicht vernachlässigbar. Denn gerade diese Kategorien Altmittel, Buntmetall, Mischmaterialien, Sperrgut, Elektro und Bauschutt sind auch für die Kundschaft des Recyclinghofs Hagenholz schwierig einzuordnen. Gerade für diese Kategorien könnten Kunden von einer Mobile App profitieren, welche diese Kategorien besser klassifiziert als der Kunde selbst.

Im Feldtest konnte der Prototyp mehrheitlich überzeugen. Überraschenderweise stimmen aber die Schwachstellen des Modells gemäss der Confusion Matrix nicht direkt mit denen des Feldtests überein. Wie bereits erwähnt, hat der Prototyp im Feldtest zum Beispiel Schwierigkeiten mit der Klasse Sperrgut, kommt aber ziemlich gut damit zurecht, Metallobjekte zu erkennen. Wichtig anzumerken ist jedoch, dass der Feldtest, bestehend aus 40 Einzeltest noch keine repräsentative Antwort über die Leistungsfähigkeit des Modells liefert. Dafür sind es einerseits zu wenige Tests und andererseits wird das Resultat zu einem gewissen Grad dadurch verfälscht, dass ich als Entwickler und als Tester der Applikation agiere. Andere Personen würden allenfalls andere Distanzen zwischen Smartphone und Objekt wählen oder die Objekte nicht genau mittig in die Kamera halten. Da es sich bei der Mobile-App aber um einen Prototyp handelt, steht in erster Linie die generelle Machbarkeit und die stichprobenhafte Überprüfung der Erkennung des Modells im Vordergrund.

Im Vergleich der entwickelten drei Modelle kann klar festgestellt werden, dass mit dem Retrainning eines vortrainierten Modells in der gegebenen Zeit, viel bessere Resultate erzielt werden konnten, als mit dem von Grund auf neu trainierten einfachen Modell. Mit einer Genauigkeit von 46.8 Prozent im Out-Of-Sample Test, beziehungsweise 46.2 Prozent in der K-Fold Cross

Validation, konnte das trainierte einfache Modell nur etwas mehr als halb so genaue Resultate liefern wie das Retrained Model mit MobileNetV3, mit einer Genauigkeit von 85.9 Prozent im Out-Of-Sample Test, beziehungsweise 88.34 Prozent in der K-Fold Cross Validation. Erstaunlich ist zudem, wie gut das auf EfficientNet-B0 aufbauende Modell, mit einer Out-Of-Sample Genauigkeit von 86.7 Prozent, beziehungsweise 87.6 Prozent in der K-Fold Cross Validation, ohne jegliches Fine-Tuning und Augmentation, bereits ähnliche Resultate erzielen kann, wie das optimierte Modell mit MobileNetV3.

Im Vergleich mit den Ergebnissen von Ruiz, Sánchez, Vélez und Raducanu (2019), mit einer durchschnittliche Genauigkeit von 88.66 Prozent mit einer ResNet-Architektur, oder mit den Ergebnissen von Olugboja und Zenghui (2019), mit einer Erkennungsrate von 87 Prozent mit ResNet-50 und SVM, schneiden die eigenen Modelle mit MobileNetV3 und EfficientNet-B0 ähnlich gut ab. Kennedy, Kennedy und Knowles (2018), haben den gleichen Ansatz gewählt, vortrainierte Modelle zu verwenden. Im Gegensatz zur vorliegenden Arbeit, haben sie jedoch VGG19 als Basis Modell gewählt und erreichen damit eine Erkennungsrate von 88.42 Prozent. Die Resultate von Yang und Thung (2016), den Erschaffern des TrashNet-Datasets sind jedoch mit ihrem Klassifikator mit rund 22 Prozent Erkennungsrate deutlich schlechter positioniert. Mit der Erkennungsrate von 97,63 Prozent mit SSD, beziehungsweise 95,76 Prozent Genauigkeit mit Faster R-CNN, welche Melinte, Travediu und Dumitriu (2020) erreichen, können die Modelle dieser Arbeit jedoch nicht mithalten.

In Anbetracht, dass in der vorliegenden Arbeit mit 24 Klassen gearbeitet wurde und Ruiz et al. (2019), Olugboja und Zenghui (2019), Kennedy et al. (2018), sowie Melinte et al. (2020) das TrashNet-Dataset verwenden und demnach mit nur 6 Klassen arbeiten, sind die Ergebnisse der eigenen Modelle mit MobileNetV3, beziehungsweise EfficientNet-B0 sehenswert.

8 Konklusion

Diese Arbeit zeigt, dass mittels vortrainierter Modelle, auf Basis eines eigenen Datasets ein Objektklassifikator entwickelt werden kann, um Objekte aus dem Recyclinghof Hagenholz zu klassifizieren. Zudem wird dargelegt wie mit dem AI-Recycling App, ein einfacher Prototyp geschaffen wurde, welche das Modell in der Praxis anwenden kann. Obwohl die Resultate der Modellvalidierung gewisse Schwachstellen aufweisen, ist die Erkennungsrate dennoch akzeptabel.

Das erarbeitete Dataset bestehend aus 24 Recyclingklassen mit jeweils 600 Bildern je Klasse, enthält nur ca. 4.5 Prozent Fotos vom Recyclinghof Hagenholz selbst. Durch die zeitliche Limitierung dieser Arbeit, wurde nur ein begrenzter Aufwand in die Sammlung von Bildern im Recyclinghof Hagenholz selbst gesteckt. Das Dataset bietet Erweiterungspotential in diesem Bereich, mit welchem die Genauigkeit des Modells für den Einsatz vor Ort sicherlich verbessert

würde. Ebenfalls im Training des Modells wurde durch die zeitliche Limitierung nicht alle Möglichkeiten ausgeschöpft. Das Problem des Overfittings wurde nicht vollständig gelöst. In einem weiteren Schritt wäre es möglich, anstatt das Fine-Tuning auf das gesamte Modell anzuwenden, dies nur auf einzelnen Layer des Basis Netzwerks anzuwenden. Ebenfalls wäre eine Möglichkeit mit L1-Regularisierung zu arbeiten.

Bezüglich der Mobile App wurde die Funktionalität stark auf das Minimum der Anforderungen reduziert, was für einen Prototyp völlig in Ordnung ist. Für einen nächsten Schritt wäre es interessant, den jeweilig erkannten Container dem Benutzer auf einer Karte zu präsentieren, damit er nicht nur weiss, zu welchem Container, sondern auch wohin im Recyclinghof er gehen muss, um sein Objekt abzugeben. Als weiteren Input nennt Herr Di Fonzo, Gruppenleiter Recyclinghof Hagenholz (persönliche Kommunikation, 27.05.21) die Erweiterung der Resultate des Prototyps, um die jeweiligen Piktogramme der Recyclingkategorien, welche vom Swiss Recycling (Swiss Recycling, o. J.) neu zu Verfügung gestellt werden. Damit könnte der Benutzer anhand dieser Piktogramme, den Container einfacher finden, welcher ebenfalls mit diesem Piktogramm ausgestattet sein wird. Der Recyclinghof Hagenholz befindet sich gemäss Herr Di Fonzo (persönliche Kommunikation, 27.05.21) in einer Umstrukturierung. Dabei werden einerseits die Container neu angeordnet und andererseits ist das Ziel, jeden Container mit dem jeweiligen Piktogramm der Sammelkategorie auszustatten.

Zudem könnte die Funktionalität der Mobile-App, insofern erweitert werden, dass bei Resultaten mit zu ungenauer Klassifizierung, eine Meldung an den Benutzer ausgegeben wird. Dabei könnte der Benutzer bei einem bestimmten Grenzwert darüber informiert werden, dass das Objekt nicht zugeordnet werden konnte und daher das Personal um Hilfe gebeten werden soll, oder allenfalls das Objekt als Abfall zu klassifizieren ist. Diese Arbeit hat sich bisher mit der Objektklassierung beschäftigt, als weiterer Schritt wäre auch denkbar, diese um die Funktionalität der Segmentierung und Objektlokalisierung, zu ergänzen.

Sowohl das erschaffene Dataset, wie auch die trainierten Modelle und der Prototyp, bieten zudem eine Grundlage für weitere Machine-Learning-Projekte im Themenbereich Recycling.

IV. Literaturverzeichnis

- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, S. 1–6. DOI.10.1109/ICEngTechnol.2017.8308186.
- Aloysius, N., & Geetha, M. (2017). A review on deep convolutional neural networks. *2017 International Conference on Communication and Signal Processing (IC-CSP)*, S. 588–592. DOI.10.1109/ICCSP.2017.8286426.
- Bundesamt für Umwelt (BAFU) (2020). *Emissionen von Treibhausgasen nach revidiertem CO₂-Gesetz und Kyoto-Protokoll, 2. Verpflichtungsperiode (2013–2020)*. Bern: Bundesamt für Umwelt. Abgerufen von https://www.bafu.admin.ch/dam/bafu/de/dokumente/klima/fachinfo-daten/CO2_Statistik.pdf.download.pdf/CO2_Publication_de_2020-07.pdf
- Bundesamt für Umwelt (BAFU) (2016). *Ent-Sorgen? Abfall in der Schweiz illustriert*. (Umwelt-Zustand Nr. 1615). Bern: Bundesamt für Umwelt. Abgerufen von https://www.bafu.admin.ch/dam/bafu/de/dokumente/abfall/uz-umwelt-zustand/ent-sorgen_-_abfallinderschweizillustriert.pdf.download.pdf/ent-sorgen_-_abfallinderschweizillustriert.pdf
- Bundesamt für Umwelt (BAFU) (2014). *Erhebung der Kehrichtzusammensetzung 2012*. Bern: Bundesamt für Umwelt. Abgerufen von: https://www.bafu.admin.ch/dam/bafu/de/dokumente/abfall/fachinfo-daten/erhebung_der_kehrrechtzusammensetzung2012.pdf.download.pdf/erhebung_der_kehrrechtzusammensetzung2012.pdf
- Confusion Matrix (2017). In C. Sammut & G. I. Webb (Hrsg.): *Encyclopedia of Machine Learning and Data Mining* (S. 260–260). Boston: Springer. DOI.10.1007/978-1-4899-7687-1_50.
- Cunningham, P., Cord, M., & Delany, S. (2008). Supervised Learning. In M. Cord & P. Cunningham P. (Hrsg.): *Machine Learning Techniques for Multimedia. Cognitive Technologies*. S. 21–49. Berlin, Heidelberg: Springer. DOI.10.1007/978-3-540-75171-7_2.
- Druzhkov, P. N., & Kustikova, V. D. (2016). A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognition and Image Analysis*, 26(1), S. 9–15. DOI.10.1134/S1054661816010065.
- Fujiyoshi, H., Hirakawa, T., & Yamashita, T. (2019). Deep learning-based image recognition for autonomous driving. *IATSS Research*, 43(4), S. 244–252. DOI.10.1016/j.iatssr.2019.11.008.

- Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. 2. Auflage. Kalifornien: O'Reilly.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. S. 770–778. DOI.10.1109/CVPR.2016.90.
- Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.-C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., & Le, Q. (2019). Searching for MobileNetV3. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, S. 1314–1324. DOI.10.1109/ICCV.2019.00140.
- Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*.
- Noman, N. (2020). A Shallow Introduction to Deep Neural Networks. In H. Iba & N. Noman (Hrsg.): *Deep Neural Evolution. Natural Computing Series*. S. 35-63. Singapur: Springer. DOI.10.1007/978-981-15-3685-4.
- Katariya, Y., & Lam, H. (o. J.). *Recognize Flowers with TensorFlow Lite on Android*. Abgerufen von <https://codelabs.developers.google.com/codelabs/recognize-flowers-with-tensorflow-on-android>
- Kennedy, T., Kennedy, S., & Knowles, J. (2018). *OscarNet: Using transfer learning to classify disposable waste*. CS230 Report: Deep Learning. Kalifornien: Stanford University, Department of Computer Science.
- Keras (o. J. -a). *Image data preprocessing*. Abgerufen von <https://keras.io/api/preprocessing/image/>
- Keras (o. J. -b). *EarlyStopping*. Abgerufen von https://keras.io/api/callbacks/early_stopping/
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, S. 1097–1105.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521 (7553), S. 436–444. DOI.10.1038/nature14539.

- LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1990). Handwritten Digit Recognition with a Back-Propagation Network. In D. Touretzky (Hrsg.): *Advances in Neural Information Processing Systems 2 (NIPS 1989)*, S. 396-404. Denver: Morgan Kaufmann.
- Lin, Y., Lv, F., Zhu, S., Yang, M., Cour, T., Yu, K., Cao, L., & Huang, T. (2011). Large-scale image classification: Fast feature extraction and SVM training. *Conference on Computer Vision and Pattern Recognition - CVPR 2011*. S. 1689–1696. Colorado Springs, CO: IEEE. DOI.10.1109/CVPR.2011.5995477.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Hrsg.): *Computer Vision – ECCV 2016*. S. 21–37. Cham: Springer. DOI.10.1007/978-3-319-46448-0_2.
- Melinte, D. O., Travediu, A.-M., & Dumitriu, D. N. (2020). Deep Convolutional Neural Networks Object Detector for Real-Time Waste Identification. *Applied Sciences*, 10(20), 7301. DOI.10.3390/app10207301.
- Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, 24(12), S. 1565–1567.
- Olugboja, A., & Zenghui, W. (2019). Intelligent Waste Classification System Using Deep Learning Convolutional Neural Network. *Procedia Manufacturing*, 35, S. 607–612. DOI.10.1016/j.promfg.2019.05.086
- Out-of-Sample Evaluation (2017). In C. Sammut & G. I. Webb (Hrsg.): *Encyclopedia of Machine Learning and Data Mining*. Boston: Springer. DOI.10.1007/978-1-4899-7687-1_621.
- Perez, L., & Wang, J. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv preprint arXiv:1712.04621*.
- Proença, P. F., & Simões, P. (2020). TACO: Trash Annotations in Context for Litter Detection. *arXiv preprint arXiv:2003.06975*.
- Raut, S. A., Raghuvanshi, M., Dharaskar, R., & Raut, A. (2009). Image Segmentation – A State-Of-Art Survey for Prediction. *2009 International Conference on Advanced Computer Control*. S. 420–424. DOI.10.1109/ICACC.2009.78.
- Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6), S. 1137-1149. DOI.10.1109/TPAMI.2016.2577031.

- Rezatofghi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). 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*. S. 658–666. Long Beach, CA: IEEE. DOI.10.1109/CVPR.2019.00075.
- Ridwang, Ilham, A. A., Nurtanio, I., & Syafaruddin. (2020). Image search optimization with web scraping, text processing and cosine similarity algorithms. *2020 IEEE International Conference on Communication, Networks and Satellite (Comnetsat)*. S. 346–350. DOI.10.1109/Comnetsat50391.2020.9328982.
- Ruiz, V., Sánchez, Á., Vélez, J. F., & Raducanu, B. (2019). Automatic Image-Based Waste Classification. In J. M. Ferrández Vicente, J. R. Álvarez-Sánchez, F. de la Paz López, J. Toledo Moreo, & H. Adeli (Hrsg.): *From Bioinspired Systems and Biomedical Applications to Machine Learning. IWINAC 2019. Lecture Notes in Computer Science, vol 11487*. S. 422–431. Cham: Springer. DOI.10.1007/978-3-030-19651-6_41
- Ruizendaal, R. (2017). *Deep Learning #3: More on CNNs & Handling Overfitting. Towards Data Science*. Abgerufen von <https://towardsdatascience.com/deep-learning-3-more-on-cnns-handling-overfitting-2bd5d99abe5d>.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3), S. 211–252. DOI.10.1007/s11263-015-0816-y.
- Salimi, I., Bayu Dewantara, B. S., & Wibowo, I. K. (2018). Visual-based trash detection and classification system for smart trash bin robot. *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*. S. 378–383. DOI.10.1109/KCIC.2018.8628499.
- Schweizerischer Bundesrat (2018). Umwelt Schweiz 2018. A Natürliche Ressourcen unter Druck. *Umwelt-Zustand*, UI-1831-D, S. 16-54. Bern: Bundesrat. Abgerufen von <https://www.bafu.admin.ch/bafu/de/home/zustand/publikationen-zum-umweltzustand/umwelt-schweiz-2018.html>
- Scikit Learn (o. J). 3.1. *Cross-validation: evaluating estimator performance*. Abgerufen von https://scikit-learn.org/stable/modules/cross_validation.html
- Shao, L., Zhu, F., & Li, X. (2015). Transfer Learning for Visual Categorization: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 26(5), S. 1019–1034. DOI.10.1109/TNNLS.2014.2330900.

- Shi, J., & Zhao, Y. (2020). Image classification optimization and results analysis based on VGG network. *2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*. S. 1090–1099. DOI.10.1109/ICIBA50161.2020.9277329.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Spanhol, F. A., Oliveira, L. S., Petitjean, C., & Heutte, L. (2016). Breast cancer histopathological image classification using Convolutional Neural Networks. *2016 International Joint Conference on Neural Networks (IJCNN)*. S. 2560–2567. DOI.10.1109/IJCNN.2016.7727519.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), S. 1929-1958.
- Stadt Zürich (o. J. -a). *App Sauberes Zürich*. Abgerufen von https://www.stadt-zuerich.ch/ted/de/index/entsorgung_recycling/sauberes_zuerich/app.html
- Stadt Zürich (o. J. -b). *Was entsorgen*. Abgerufen von https://www.stadt-zuerich.ch/ted/de/index/entsorgung_recycling/sauberes_zuerich/was_entsorgen.html
- Stadt Zürich (o. J. -c). *Wo + wann entsorgen*. Abgerufen von https://www.stadt-zuerich.ch/ted/de/index/entsorgung_recycling/sauberes_zuerich/wo_%2B_wann_entsorgen.html
- Su, B., Chen, H., Zhu, Y., Liu, W., & Liu, K. (2019). Classification of Manufacturing Defects in Multicrystalline Solar Cells With Novel Feature Descriptor. *IEEE Transactions on Instrumentation and Measurement*, 68(12), 4675–4688. DOI.10.1109/TIM.2019.2900961
- Swiss Recycling (o. J.). *Piktogramme*. Abgerufen von <http://www.swissrecycling.ch/piktogramme/>
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going Deeper With Convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, S. 1–9. DOI.10.1109/CVPR.2015.7298594.
- Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th International Conference on Machine Learning (PMLR)*. S. 6105–6114.

- TensorFlow (2021a). *Load images*. Abgerufen von https://www.tensorflow.org/tutorials/load_data/images
- TensorFlow (2021b). *Retraining an Image Classifier*. Abgerufen von https://www.tensorflow.org/hub/tutorials/tf2_image_retraining
- TensorFlow (2020). *Overfit and underfit*. Abgerufen von https://www.tensorflow.org/tutorials/keras/overfit_and_underfit#add_weight_regularization
- TensorFlow (o. J.). *GPU support*. Abgerufen von <https://www.tensorflow.org/install/gpu>
- Wang, T., Cai, Y., Liang, L., & Ye, D. (2020). A Multi-Level Approach to Waste Object Segmentation. *Sensors*, 20(14), 3816. DOI.10.3390/s20143816
- Wong, T.-T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, 48(9), S. 2839–2846. DOI.10.1016/j.pat-cog.2015.03.009
- Yang, M., & Thung, G. (2016). Classification of trash for recyclability status. *CS229 Project Report*, 2016. Kalifornien: Stanford University.
- Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. *arXiv preprint arXiv:1411.1792*.

V. Anhang

A. Container in Recyclinghof Hagenholz

#	Kategorie	Klasse in Dataset	Inhalt
1	Abfall	trash	<p>Die Kategorie Abfall beschreibt Haushaltsabfälle, meist in Form von Abfallsäcken. Ebenso nicht wiederverwendbarer Plastik gehört in diese Kategorie.</p> <ul style="list-style-type: none">▪ Glühbirne▪ Halogenlampe▪ Hygieneartikel▪ Kunststoffmaterial▪ Kunststoffverpackung▪ Papierschnipsel▪ Papiertragetasche▪ Staubsaugerbeutel▪ Tetrapak▪ Tiefkühlverpackung▪ Wischgut▪ Zigarettenstummel▪ Katzensand▪ Nylonstrumpf▪ Rasierklinge (gut verpackt)
2	Sperrgut	bulky_rubbish	<p>Sperrgut besteht mehrheitlich aus Holz und wie beispielsweise Möbel. Auch bezogene Sofas oder alte Matratzen gehören hier hin. Grössere Metalteile sollen hier nicht enthalten sein.</p> <ul style="list-style-type: none">▪ Koffer▪ Korbwaren▪ Matratze▪ Möbel▪ Plastik▪ Ski▪ Teppich
3	Karton	cardboard	<p>Zur Kategorie Karton gehören die haushaltsüblichen Kartonabfälle von Verpackungsmaterialien wie beispielsweise Kartonskisten.</p> <ul style="list-style-type: none">▪ Früchtekarton▪ Gemüsekarton▪ Papiertragetasche gefaltet▪ Postpaket▪ Schachtel▪ Verpackung▪ Waschmittelkarton (ohne Plastikgriff)
4	Papier	paper	<p>Zur Kategorie Papier gehören:</p> <ul style="list-style-type: none">▪ Kuvert▪ Prospekt (nicht beschichtet)▪ Zeitschrift▪ Zeitung

5	Styropor	styrofoam	Zur Kategorie Styropor gehören Verpackungsmaterial aus Styropor.
			
6	Alu-Dosen	alu_can	Die Kategorie Alu-Dosen beinhaltet handelsübliche Getränkedosen. Blechbüchsen gehören jedoch nicht in diese Kategorie, sondern zum Metall.
			
7	PET	pet	Zu PET gehören nur die PET-Getränkeflaschen. Andere Plastikflaschen oder Plastikmaterialien sind hier ausgeschlossen.
			
8	Textilien	textile	Zur Kategorie Textilien gehören wiederverwendbare Kleider und Textilmaterialien wie folgende:
<ul style="list-style-type: none"> ▪ Bettwaren ▪ Bettwäsche ▪ Gürtel ▪ Handtasche ▪ Kleider ▪ Lederbekleidung ▪ Pelz ▪ saubere Unterwäsche ▪ Schuhe (paarweise zusammengebunden) ▪ Tischwäsche 			
9	Metall	metal	Die Kategorie Metall beinhaltet allerlei Objekte aus Eisen und Blech ab. Folgende Beispiele sind zu nennen:
			
<ul style="list-style-type: none"> ▪ Draht ▪ kleine Pfanne ▪ Konservendose ▪ Kuchenblech ▪ Nagel ▪ Schraube ▪ Tube aus Aluminium ▪ Backform ▪ Felge ▪ Grill ▪ Metallgestell/-schrank ▪ Metallwerkzeug ohne Motor ▪ Pfanne ▪ Skistock ▪ Velo 			

10	Buntmetall	nonferrous_metal	Zur Kategorie Buntmetall gehören Objekte aus Kupfer, Zink, Nickel oder Aluminium, wie folgende Beispiele:		
			<ul style="list-style-type: none"> ▪ Skistöcke ▪ Golfschläger ▪ Fahrradteile ▪ Kupferrolle ▪ Ruder ▪ Buntmetallkleinteile 		
11	Elektro	electro	Die Kategorie Elektro beinhaltet allerlei Elektronische Geräte und Komponenten. Generell alle Objekte, welche ein Kabel integriert haben, gehören in diese Kategorie. Ausnahme bilden Fernseher und Geräte mit Lithium-Akkus. Folgende Beispiele für diese Kategorie:		
			<table border="0"> <tr> <td data-bbox="779 672 1055 1029"> <ul style="list-style-type: none"> ▪ Computer und Zubehör (Tastatur, Maus, Datenträger) ▪ Kabel ▪ Drucker, Fax- und Kopiergerät ▪ Modem & Netzwerkgerät ▪ Telefon ▪ Boiler & Klimagerät ▪ Fön ▪ Lampen ▪ Nähmaschine ▪ CD- & DVD-Gerät </td> <td data-bbox="1055 672 1334 1029"> <ul style="list-style-type: none"> ▪ Spiel- und Sportgerät ▪ Staubsauger ▪ Bügeleisen, Tumbler, Waschmaschine ▪ Projektor ▪ Radio ▪ Fernbedienung ▪ Küchengeräte (Mixer, Kaffeemaschine, Toaster, Kochherd, Geschirrspüler) ▪ Bohrmaschine ▪ Motorsäge ▪ Rasenmäher </td> </tr> </table>	<ul style="list-style-type: none"> ▪ Computer und Zubehör (Tastatur, Maus, Datenträger) ▪ Kabel ▪ Drucker, Fax- und Kopiergerät ▪ Modem & Netzwerkgerät ▪ Telefon ▪ Boiler & Klimagerät ▪ Fön ▪ Lampen ▪ Nähmaschine ▪ CD- & DVD-Gerät 	<ul style="list-style-type: none"> ▪ Spiel- und Sportgerät ▪ Staubsauger ▪ Bügeleisen, Tumbler, Waschmaschine ▪ Projektor ▪ Radio ▪ Fernbedienung ▪ Küchengeräte (Mixer, Kaffeemaschine, Toaster, Kochherd, Geschirrspüler) ▪ Bohrmaschine ▪ Motorsäge ▪ Rasenmäher
<ul style="list-style-type: none"> ▪ Computer und Zubehör (Tastatur, Maus, Datenträger) ▪ Kabel ▪ Drucker, Fax- und Kopiergerät ▪ Modem & Netzwerkgerät ▪ Telefon ▪ Boiler & Klimagerät ▪ Fön ▪ Lampen ▪ Nähmaschine ▪ CD- & DVD-Gerät 	<ul style="list-style-type: none"> ▪ Spiel- und Sportgerät ▪ Staubsauger ▪ Bügeleisen, Tumbler, Waschmaschine ▪ Projektor ▪ Radio ▪ Fernbedienung ▪ Küchengeräte (Mixer, Kaffeemaschine, Toaster, Kochherd, Geschirrspüler) ▪ Bohrmaschine ▪ Motorsäge ▪ Rasenmäher 				
12	Bauschutt/ Steingut	construction_waste	Zu dieser Kategorie gehören sowohl Bauschutt und Steinmaterialien wie auch Glas. Einige Beispiele sind:		
			<ul style="list-style-type: none"> ▪ Blumentopf ▪ Bauschutt ▪ Eternit ▪ Geschirr ▪ Hydrokugeln ▪ Keramik ▪ Tonwaren ▪ Trinkglas 		
			Glas soll in Zukunft separat gesammelt werden können.		
13	Batterien	battery	Zur Kategorie Batterien gehören handelsübliche Batterien, ohne Lithium-Akkus.		
					

14	Lithium-Akkus	lithium_battery	Diese Kategorie beinhaltet alle Geräte, welche einen Lithium-Akku verbaut haben. Dazu gehören beispielsweise:
			<ul style="list-style-type: none"> ▪ Akku ▪ Smartphone ▪ Tablet ▪ Elektrische Zahnbürste ▪ Notebook
15	Kaffeekapseln	coffee_capsule	In der Sammelkategorie Kaffeekapseln werden ausschließlich alte Kaffeekapseln gesammelt.
			
16	Bildschirme	screen	Zur Kategorie Bildschirme gehören alle folgende Objektarten:
			<ul style="list-style-type: none"> ▪ Bildschirm ▪ Monitore ▪ TV-Gerät <p>Von dieser Kategorie ausgeschlossen sind Röhrenbildschirme. Da diese giftige Gase enthalten, welche bei Zerschlagen, austreten könnten. Röhrenbildschirme gehören deshalb zum Sonderabfall.</p>
17	Bücher	book	Bücher bilden eine eigene Kategorie und gehören nicht zu der Kategorie Papier.
			
18	CD-DVD	cd_dvd	Zu dieser Kategorie gehören:
			<ul style="list-style-type: none"> ▪ CDs ▪ DVDs ▪ Videokassetten ▪ Schallplatten <p>Zudem gehören die jeweiligen Verpackungen, bzw. Hüllen der jeweiligen Medien dazu.</p>

19	Kork	cork	<p>Zu dieser Kategorie gehören Materialien aus Kork wie Beispielsweise:</p> <ul style="list-style-type: none"> • Weinkorken • Korkuntersätze
			
20	Pfannen	pan	<p>Ausschliesslich Pfannen werden in dieser Kategorie gesammelt.</p>
			
21	Mischmaterialien	mixed_material	<p>Objekte, aus mehreren Materialien inklusive Metalteilen gehören zu dieser Kategorie wie beispielsweise:</p> <ul style="list-style-type: none"> ▪ Bürostuhl ▪ Autositze ▪ Tische ▪ Kinderwagen
			
22	Sonderabfall	special	<p>Sonderabfall wird generell ausserhalb des Recyclinghofs, in einer Sonderabteilung entsorgt. Dennoch werden gewisse Materialien in kleinen Mengen im Recyclinghof entgegengenommen.</p>
			<p>Beispiele von im Recyclinghof entgegen-genommener Materialien:</p> <ul style="list-style-type: none"> ▪ Autobatterie ▪ Deosprühdose ▪ Spraydose ▪ Haarspraydose ▪ Feuerlöscher ▪ Röhrenfernseher ▪ Druckerpatrone ▪ Gasflasche ▪ Leuchtstoffröhre
			<p>Weitere Materialien welche direkt in der Sonderabfallstelle abgegeben werden:</p> <ul style="list-style-type: none"> ▪ Reinigungsmittel ▪ Röntgenbild ▪ Säure ▪ Asbesthaltiges Material ▪ Chemikalie ▪ Gift ▪ Holzschutzmittel ▪ Lösungsmittel ▪ Medikament ▪ Mineralöl ▪ Lauge ▪ Lack ▪ Farbe

23	Plastikflaschen (nicht PET)	plastic_bottle	Zu dieser Kategorie gehören Plastikflaschen, welche nicht zu PET-Getränkeflaschen gehören wie: <ul style="list-style-type: none"> ▪ Flaschen von Sonnencremes ▪ Shampoos ▪ Duschmitteln ▪ Wasch- und Reinigungsmitteln ▪ Pflanzendünger ▪ Milchflaschen ▪ Joghurt-Getränkeflaschen ▪ weitere Flaschen von Lebensmitteln
			
24	Pneu	tire	In den Container der Kategorie Pneu gehören beispielsweise: <ul style="list-style-type: none"> ▪ Autopneu ▪ Motorradpneu ▪ Velopneu <p>Zu beachten ist, dass für Pneus eine extra Gebühr fällig wird.</p>
			
25	Kompressoren	compressor	Diese Kategorie beinhaltet Geräte mit integriertem Kompressor dazu gehören beispielsweise: <ul style="list-style-type: none"> ▪ Tiefkühler ▪ Kühlschrank
			

B. Verwendete Suchbegriffe im Image-Scraping

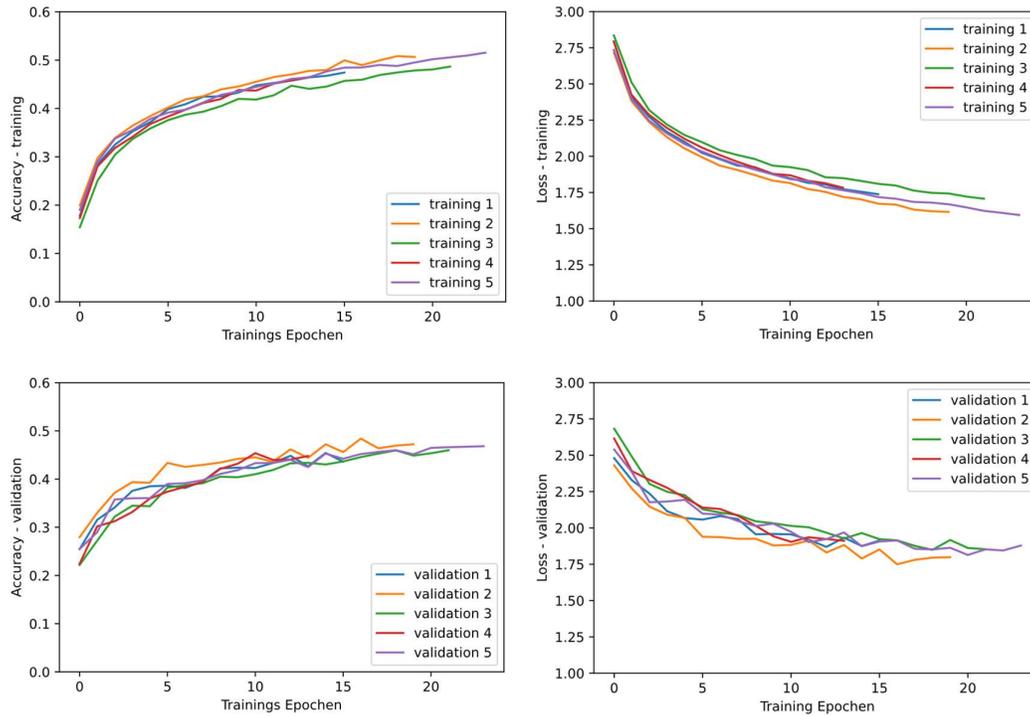
Label	Suchstring
metal	'metal%20chair' 'metal%20table' 'tin%20can' 'kleinanzeigen fahrrad' 'kleinanzeigen besteck' 'eisenstange' 'eisenteile' 'blechbüchse'
bulky_rubbish	'wood%20chair' 'mattress' 'wood%20table' 'metal%20table' 'wood%20cabinet' 'old%20mattress' 'alte%20couch' 'kleinanzeigen ski' 'kleinanzeigen snowboard'
alu_can	'aluminium%20can' 'getränkedose' 'bierdose' 'coladose'
electro	'cable' 'mixer' 'charger' 'kleinanzeigen telefon' 'kleinanzeigen kabel' 'kleinanzeigen lampe' 'kleinanzeigen stehlampe' 'defekte glühbirne' 'kleinanzeigen toaster' 'kleinanzeigen radio' 'kleinanzeigen mikrowelle' 'kleinanzeigen stereoanlage' 'kleinanzeigen fernbedienung' 'kleinanzeigen staubsauger' 'kleinanzeigen drucker'
cork	'cork' 'wine cork' 'weinkork'
cardboard	'cardboard' 'karton kiste'
textile	'kleinanzeigen pullover' 'kleinanzeigen shirt' 'kleinanzeigen hose' 'kleinanzeigen kleid'
book	'book'

	'kleinanzeigen buch'
cd_dvd	'kleinanzeigen cd' 'kleinanzeigen dvd' 'schallplatte'
battery	'battery' 'alkaline batterie' 'duracell batterie' 'alkaline battery leak'
mixed_material	'kinderwagen kleinanzeigen' 'office%20chair'
lithium_battery	'kleinanzeigen notebook' 'kleinanzeigen samsung' 'kleinanzeigen iphone' 'kleinanzeigen desktop' 'kleinanzeigen elektrische zahnbrüste' 'gebraucht+tablet+defekt' 'smartphone%20broken'
paper	'Papier' 'papiersammlung' 'zeitung'
trash	'Plastik' 'abfallsack%20schwarz%20voll' 'tetrapack%20leer'
construction_waste	'Glas' 'kleinanzeigen teller' 'kleinanzeigen tasse' 'kleinanzeigen keramik' 'scherben' 'glasscherben' 'kleinanzeigen blumenwase' 'ziegelsteine' 'zerbrochene ziegel'
styrofoam	'styropor' 'styrofoam packaging' 'white styrofoam packaging' 'styrofoam peanuts' 'styrofoam%20used'
special	'alte gassflasche' 'spraydose' 'gebrauchte shampoflasche' 'röhrenfernseher' 'röhrenfernseher kleinanzeige' 'feuerl%C3%B6scher%20gebraucht'
pan	'kleinanzeigen pfanne' 'alte%20teflonpfanne'
nonferrous_metal	'kleinanzeigen messing' 'messing teile' 'kleinanzeigen kupfer' 'kupfer teile'

	'kleinanzeigen kickboard'
	'kleinanzeigen golfset'
coffee_capsule	'gebrauchte nespresso kapsel'
	'nespresso kapsel'
	'nespresso%20capsule'
pet	'leere petflasche'
	'petflasche'
	'PET%20flasche'
screen	'kleinanzeigen fernseher'
	'kleinanzeigen samsung tv'
	'kleinanzeigen panaonic tv'
	'kleinanzeigen sony tv'
	'lcd fernseher kleinanzeige'
	'broken%20tv'
tire	'alte%20reifen'

C. Details K-Fold Cross Validation mit «Model from Scratch»

Nachfolgend sind die Verläufe der einzelnen Trainings ersichtlich. Die beiden linken Diagramme zeigen die Veränderung der Accuracy im Verlauf der Epochen und die beiden rechten Diagramme zeigen den Loss derselben Trainingsverlaufs. Oberhalb sind jeweils die Trainingswerte ersichtlich, während unten die jeweiligen Validierungsergebnisse abgebildet wurden.



D. Vortest

Erste Testresultate des Prototyps mit Haushaltsgegenständen.

#	Test-Case	Resultat	#	Test-Case	Resultat
1		Die erkannte Kategorie Elektroschrott ist korrekt!	4		Die erkannte Kategorie Bauschutt ist korrekt!
2		Die erkannte Kategorie Metall ist korrekt!	5		Die erkannte Kategorie Textilien ist falsch! Die korrekte Kategorie wäre Karton.
3		Die erkannte Kategorie Alu-Dose ist korrekt!	6		Ein zweiter Test mit demselben Objekt wurde vorgenommen. Die erkannte Kategorie Karton ist diesmal korrekt!

E. Feldtest: Testprotokoll

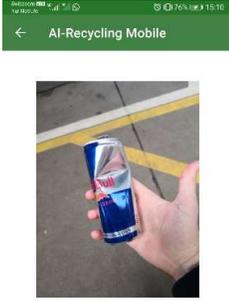
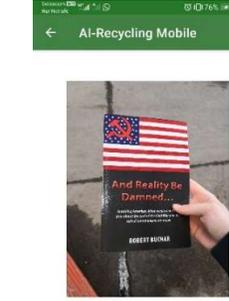
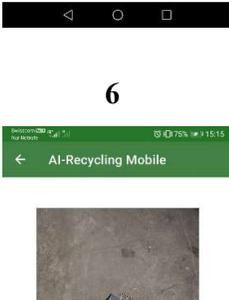
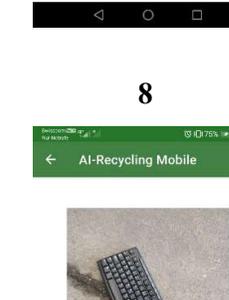
Testprotokoll-Prototyp

Testdatum: 27.05.21

#	SOLL-KATEGORIE	IST-KATEGORIE	KOMMENTAR	I.O.
1	Batterien	Batterien		✓
2	Alu-Dosen	Alu-Dosen		✓
3	Bücher	CD-DVD		X
4	Bücher	Bücher	Im 2ten Versuch OK.	✓
5	Elektro	Elektro		✓
6	Elektro	Elektro		✓
7	Elektro	Elektro		✓
8	Elektro	Elektro		✓
9	Bauschutt	Bauschutt		✓
10	Bauschutt	Buntmetall	Sieht aus wie Messing Im 2. Versuche OK.	X
11	TV-Bildschirm	TV-Bildschirm		✓
12	TV-Bildschirm	TV-Bildschirm		✓
13	Pfanne	Pfanne		✓
14	Buntmetall	Buntmetall		✓
15	Buntmetall	Buntmetall		✓
16	Plastikflasche	Plastikflasche		✓
17	Plastikflasche	Plastikflasche		✓
18	CD/DVD	CD/DVD		✓
19	CD/DVD	CD/DVD		✓

20	CD/DVD	CD/DVD		✓
21	Lithium-Akku	Lithium-Akku		✓
22	Lithium-Akku	Elektro		x
23	Lithium-Akku	Lithium-Akku		✓
24	Lithium-Akku	Lithium-Akku		✓
25	Reifen	Reifen		✓
26	Altmittel	Elektro	ähnlich wie ein Kabel	x
27	Altmittel	Altmittel		✓
28	Altmittel	Altmittel		✓
29	Altmittel	Altmittel		✓
30	Altmittel	Altmittel		✓
31	Altmittel	Altmittel		✓
32	Sperrgut	Karton		x
33	Sperrgut	Sperrgut		✓
34	Sperrgut	Styropor		x
35	Mischmaterialien	Mischmaterialien		✓
36	Mischmaterialien	Mischmaterialien		✓
37	Sperrgut	Styropor		x
38	Elektro	Elektro		✓
39	PET	Aku-Dose	Im 2ten Versuch ok.	x
40	Karton	Karton		✓

F. Feldtest: Resultate als Screenshots

1	2	3	4
 <p>Das Objekt gehört in den Container: [Batterien]</p> <p>OK</p>	 <p>Das Objekt gehört in den Container: [Alu-Dosen]</p> <p>OK</p>	 <p>Das Objekt gehört in den Container: [CD & DVD]</p> <p>OK</p>	 <p>Das Objekt gehört in den Container: [Bücher]</p> <p>OK</p>
5	6	7	8
 <p>Das Objekt gehört in den Container: [Elektroschrott]</p> <p>OK</p>	 <p>Das Objekt gehört in den Container: [Elektroschrott]</p> <p>OK</p>	 <p>Das Objekt gehört in den Container: [Elektroschrott]</p> <p>OK</p>	 <p>Das Objekt gehört in den Container: [Elektroschrott]</p> <p>OK</p>
9	10.1	10.2	11
 <p>Das Objekt gehört in den Container: [Bauschutt]</p> <p>OK</p>	 <p>Das Objekt gehört in den Container: [Buntmetall]</p> <p>OK</p>	 <p>Das Objekt gehört in den Container: [Bauschutt]</p> <p>OK</p>	 <p>Das Objekt gehört in den Container: [TV & Bildschirme]</p> <p>OK</p>

12

← AI-Recycling Mobile



Das Objekt gehört in den Container: [TV & Bildschirme]

OK



13

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Pfannen]

OK



14

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Buntmetall]

OK



15

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Buntmetall]

OK



16

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Plastikflaschen]

OK



17

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Plastikflaschen]

OK



18

← AI-Recycling Mobile



Das Objekt gehört in den Container: [CD & DVD]

OK



19

← AI-Recycling Mobile



Das Objekt gehört in den Container: [CD & DVD]

OK



20

← AI-Recycling Mobile



Das Objekt gehört in den Container: [CD & DVD]

OK



21

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Lithium-Akkus]

OK



22

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Elektroschrott]

OK



23

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Lithium-Akkus]

OK



24

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Lithium-Akkus]

OK

25

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Reifen]

OK

26

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Elektroschrott]

OK

27

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Altmittel]

OK

28

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Altmittel]

OK

29

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Altmittel]

OK

30

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Altmittel]

OK

31

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Altmittel]

OK

32

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Karton]

OK

33

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Sperrgut]

OK

34

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Styropor]

OK

35

← AI-Recycling Mobile



Das Objekt gehört in den Container: [Mischmaterialien]

OK

36



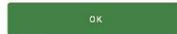
Das Objekt gehört in den Container: [Mischmaterialien]



37



Das Objekt gehört in den Container: [Styropor]



38



Das Objekt gehört in den Container: [Elektroschrott]



39.1



Das Objekt gehört in den Container: [Alu-Dosen]



39.2



Das Objekt gehört in den Container: [PET-Flaschen]



40



Das Objekt gehört in den Container: [Karton]

