



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Justus Börms

Entwicklung einer Methode zur Erweiterung des Hamburger 3D-Stadtmodells um Fensterflächen und Baujahre für detaillierte Wärmebedarfsprognosen

*Fakultät Technik und Informatik
Department Maschinenbau und Produktion*

*Faculty of Engineering and Computer Science
Department of Mechanical Engineering and
Production Management*

Justus Börms

**Entwicklung einer Methode zur
Erweiterung des Hamburger 3D-
Stadtmodells um Fensterflächen und
Baujahre für detaillierte
Wärmebedarfsprognosen**

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang Nachhaltige Energiesysteme im Maschinenbau
am Department Maschinenbau und Produktion
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

in Zusammenarbeit mit:
Competence Center für Erneuerbare Energien und Energieeffizienz (CC4E)
Am Schleusengraben 24
21029 Hamburg

Erstprüfer: Prof. Dr.-Ing. Hans Schäfers
Zweitprüfer: M. Sc. Jan Trosdorff

Abgabedatum: 25.05.2023

Zusammenfassung

Justus Börms

Thema der Masterarbeit

Entwicklung einer Methode zur Erweiterung des Hamburger 3D-Stadtmodells um Fensterflächen und Baujahre für detaillierte Wärmebedarfsprognosen

Stichworte

3D-Stadtmodell, LoD2, CityGML, Open Data, Wärmebedarfsprognosen, Machine Learning

Kurzzusammenfassung

In dieser Arbeit werden zwei Ansätze vorgestellt, mit denen zum einen das Verhältnis von Fenster- zu Wandflächen und zum anderen die Baualtersklassen von Hamburger Gebäuden bestimmt werden können. Hierzu werden das 3D-Stadtmodell von Hamburg, die ALKIS-Datenbank, digitale Schrägluftbilder und historische Karten von Hamburg verwendet. Die Bestimmung des Verhältnisses der Fenster- zur Wandfläche eines Gebäudes erfolgt mithilfe des Machine-Learning-Modells Faster R-CNN. Die Baualtersklasse wird über den Vergleich des Gebäudegrundrisses mit historischen Karten bestimmt.

Justus Börms

Title of the paper

Development of a method for extending the Hamburg 3D city model by window areas and years of construction for detailed heat demand predictions

Keywords

3D city model, LoD2, CityGML, open data, heat demand predictions, machine learning

Abstract

This work presents two approaches to determine the window-wall-ratios of buildings in the city of Hamburg on the one hand and their construction year categories on the other hand. The 3D city model of Hamburg, the ALKIS-database, digital oblique aerial images and historical maps are used. The window-wall-ratio is determined via the machine learning model Faster R-CNN. The construction year category is derived from a comparison of the building footprint with historical maps.

Inhaltsverzeichnis

Symbolverzeichnis.....	I
Abkürzungsverzeichnis.....	IV
Abbildungsverzeichnis	V
Tabellenverzeichnis.....	X
1 Einleitung.....	- 1 -
1.1 Überblick über die Arbeit.....	- 2 -
2 Theorie und Stand der Technik.....	- 3 -
2.1 Arbeiten zum Thema.....	- 3 -
2.1.1 Wärmebedarfsprognosen anhand von 3D-Stadtmodellen.....	- 3 -
2.1.2 Erkennung von Fenstern in Gebäudefassaden	- 4 -
2.1.3 Bestimmung des Baujahres	- 5 -
2.2 Datengrundlage in Hamburg.....	- 6 -
2.2.1 ALKIS.....	- 6 -
2.2.2 Das 3D-Stadtmodell von Hamburg, CityGML und CityJSON	- 7 -
2.2.3 Wohngebäudetypologie der IWU	- 9 -
2.2.4 Luftaufnahmen und Karten von Hamburg	- 10 -
2.3 Aufbereitung von Luftbildern.....	- 13 -
2.3.1 Georeferenzierung.....	- 13 -
2.3.2 Rektifikation.....	- 15 -
2.4 Machine Learning-Methoden zur Objekterkennung.....	- 16 -
2.4.1 Künstliche Intelligenz, Machine Learning und Deep Learning.....	- 16 -
2.4.2 Das Objekterkennungsmodell Faster R-CNN.....	- 19 -
2.4.3 Bewertungsgrößen für Klassifizierungsaufgaben.....	- 22 -
2.4.4 Clusteranalyse	- 25 -
2.4.5 Datensätze.....	- 27 -
3 Methodik.....	- 30 -
3.1 Bestimmung des Verhältnisses der Fenster- zur Wandfläche eines Gebäudes	- 30 -

3.1.1	Die Machine Learning Methode	- 31 -
3.1.2	Nachbearbeitung der Ergebnisse des ML-Modells.....	- 35 -
3.1.3	Der Trainingsdatensatz	- 39 -
3.1.4	Der Testdatensatz	- 41 -
3.1.5	Auswahl und Aufbereitung der Bilder.....	- 43 -
3.1.6	Berechnung des Verhältnisses der Fenster- zur Wandfläche.....	- 50 -
3.2	Bestimmung der Baualtersklasse eines Gebäudes	- 52 -
3.2.1	Auswahl und Aufbereitung der Karten	- 54 -
3.2.2	Bestimmung des Baujahres und der Baualtersklasse.....	- 56 -
3.2.3	Der Testdatensatz	- 59 -
4	Ergebnisse	- 64 -
4.1	Bestimmung des Verhältnisses der Fenster- zur Wandfläche eines Gebäudes	- 64 -
4.1.1	Ergebnisse des ML-Trainings.....	- 64 -
4.1.2	Validierung der Ergebnisse.....	- 71 -
4.1.3	Diskussion der Ergebnisse	- 76 -
4.2	Bestimmung der Baualtersklasse eines Gebäudes	- 81 -
4.2.1	Validierung der Ergebnisse.....	- 82 -
4.2.2	Diskussion der Ergebnisse	- 89 -
5	Fazit.....	- 94 -
6	Ausblick.....	- 96 -
7	Literaturverzeichnis.....	- 97 -
Anhang.....		A-1
Anhang A.....		A-1
Datengrundlage in den deutschen Bundesländern.....		A-1
Anhang B.....		B-1
Umformung der Kameragleichung.....		B-1
Anhang C.....		C-1
Fenster- und Wandflächen für die IWU-Wohngebäudetypen		C-1
Anhang D.....		D-1

Trainingsdatensätze für Fensterflächenerkennung.....	D-1
Anhang E	E-1
Beispielbilder der Gebäude des Hamburg Image Datasets	E-1
Plakette zu Zerstörung und Wiederaufbau.....	E-3
Sturmflutgebiet Hamburg 1962	E-4
Gebäude Mengestraße 6 und Hermann-Westphal-Straße 2a-c	E-5
Anhang F	F-1
Einfluss des Score-Grenzwertes auf mAP	F-1
Ergebnisse für $\Delta R_{\text{Gebäude, mean}}$	F-2

Symbolverzeichnis

Lateinische Symbole	Einheit	Bedeutung
A	[m ²]	Fläche
A_{Fenster}	[Pixel ²]	Fläche eines Fensters im rektifizierten Bild
AP	[-]	Durchschnittliche Präzision (Average Precision)
A_{Wand}	[Pixel ²]	Fläche einer Wand im rektifizierten Bild
a_p	[°]	Drehung der Kamera um die y-Achse (pitch)
a_r	[°]	Drehung der Kamera um die x-Achse (roll)
a_y	[°]	Drehung der Kamera um die z-Achse (yaw)
$B(C)$	[-]	Between-Cluster-Point-Scatter
BJ	[a]	Tatsächliches Baujahr eines Gebäudes
$BJ_{\text{geschätzt}}$	[a]	Geschätztes Baujahr eines Gebäudes
C	[-]	Encoder
c	[-]	Cosinus
c_{cam}	[mm]	Kamerakonstante
d	[-]	Euklidische Distanz
E	[m]	Ost-Koordinate im UTM-Koordinatensystem
$F1$	[-]	Harmonisches Mittel von Präzision und Recall
FP	[-]	False Positive (Falsch erkannte Objekte)
FN	[-]	False Negative (Nicht erkannte Objekte)
H	[-]	Projektive Transformationsmatrix
$h_1 - h_8$	[-]	Elemente der projektiven Transformationsmatrix
IoU	[-]	Intersection over Union
K	[-]	Kamera-Matrix
K_{Cluster}	[-]	Anzahl der Cluster für k-means-Clusteranalyse
L	[-]	Loss-Funktion
mAP	[-]	Mittlere durchschnittliche Präzision (Mean Average Precision)
N	[m]	Nord-Koordinaten im UTM-Koordinatensystem
N_{Adressen}	[-]	Anzahl der Adressen, die einem Gebäude zugeordnet sind
N_{Batch}	[-]	Anzahl der Batches pro Epoche
N_{Class}	[-]	Anzahl der Klassen in einem Datensatz

N_{Fenster}	[-]	Anzahl der Fenster in einer Wand
$N_{\text{Wände}}$	[-]	Anzahl der Wände eines Gebäudes
$N_{\text{Gebäude}}$	[-]	Anzahl der Gebäude im Hamburg Image Dataset
P	[-]	Präzision
P_i	[-]	Eckpunkt einer Wand
p	[-]	Punkt
R	[-]	Recall
$R_{\text{Gebäude}}$	[-]	Verhältnis der Fenster- zu Wandflächen für ein Gebäude auf Basis der ML-Ergebnisse
$R_{\text{Gebäude, HI}}$	[-]	Verhältnis der Fenster- zu Wandflächen für ein Gebäude im Hamburg Image Dataset
$R_{\text{Gebäude, IWU}}$	[-]	Verhältnis der Fenster- zu Wandflächen für ein Gebäude der IWU-Gebäudetypologie
R_{rot}	[-]	Rotationsmatrix
R_{Wand}	[Pixel ²]	Verhältnis der Fenster- zur Wandfläche im rektifizierten Bild
S	[-]	Seitenverhältnis einer Wand
s	[-]	Sinus
sign	[-]	Vorzeichen in der Gleichung zur Rektifikation
T	[-]	Transformationsmatrix
TP	[-]	True Positive (korrekt erkannte Objekte)
$T_1 - T_{12}$	[-]	Elemente der Transformationsmatrix
t	[-]	Translationsvektor
$W(C)$	[-]	Within-Cluster-Point-Scatter
X	[m]	Geografische Koordinate
X_c	[-]	Vektor der geografischen Koordinaten
X_t	[m]	Geografische Koordinate der Kamera
x	[Pixel]	Bildkoordinate
x_c	[-]	Vektor der Bildkoordinaten
x_i	[Pixel]	Koordinate eines Eckpunktes einer Wand
x_{rect}	[Pixel]	Koordinate des rektifizierten Bildes
x_K	[mm]	Koordinate der Hauptpunktlage des Bildes
Y	[m]	Geografische Koordinate
Y_t	[m]	Geografische Koordinate der Kamera

y	[Pixel]	Bildkoordinate
y_i	[Pixel]	Koordinate eines Eckpunktes einer Wand
y_{rect}	[Pixel]	Koordinate des rektifizierten Bildes
y_K	[mm]	Koordinate der Hauptpunktlage des Bildes
Z	[m]	Geografische Koordinate
Z_t	[m]	Geografische Koordinate der Kamera

Griechische Symbole	Einheit	Bedeutung
ΔB_J	[a]	Abweichung zwischen $B_{J_{geschätzt}}$ und B_J
ΔE	[m]	Geografischer Abstand zwischen zwei Punkten in Ost-Richtung
ΔN	[m]	Geografischer Abstand zwischen zwei Punkten in Nord-Richtung
$\Delta R_{Gebäude}$	[%]	Abweichung zwischen $R_{Gebäude}$ und $R_{Gebäude, HI}$
Δx	[Pixel]	Abstand zwischen zwei Bildpunkte in x-Richtung
Δy	[Pixel]	Abstand zwischen zwei Bildpunkte in y-Richtung
λ	[-]	Skalierungsfaktor
σ	[%]	empirische Standardabweichung von $\Delta R_{Gebäude}$

Indizes	Bedeutung
<i>bild</i>	In einem Bild
<i>Class</i>	Klassifizierung
<i>geo</i>	Geografisch
<i>HI</i>	Hamburg Image Dataset
<i>i</i>	Laufender Index
<i>max</i>	Maximalwert
<i>mean</i>	Mittelwert
<i>min</i>	Minimalwert
<i>ML</i>	Machine Learning
<i>rect</i>	Rektifiziert
<i>Reg</i>	Regression
<i>RPN</i>	Region Proposal Network

Abkürzungsverzeichnis

Abkürzung	Bedeutung
ALKIS	Amtliches Liegenschaftskatasterinformationssystem
API	Application Programming Interface
CNN	Convolutional Neural Network
EPSG	European Petroleum Survey Group Geodesy
FPN	Feature Pyramid Network
GML	Geography Markup Language
IWU	Institut für Wohnen und Umwelt
JSON	JavaScript Object Notation
KI	Künstliche Intelligenz
LGV	Landesbetrieb Geoinformation und Vermessung
LoD	Level of Detail
MetaVer	Metadatenverbund der Länder Brandenburg, Bremen, Hamburg, Hessen, Mecklenburg- Vorpommern, Saarland, Sachsen und Sachsen- Anhalt
ML	Machine Learning
NMS	Non-Maximum Suppression
R-CNN	Region-based Convolutional Network Method
RoI	Region of Interest
RPN	Region Proposal Network
SGD	Stochastisches Gradientenverfahren
SSWD	Street Scene Window Detection Dataset
WMS	Web Map Service
XML	Extensible Markup Language

Abbildungsverzeichnis

Abbildung 1: Beispielhafte Darstellung der 5 Detailgrade (LoDs) eines 3D-Gebäudemodells. Übernommen aus [7].	- 7 -
Abbildung 2: Baualters- und Größenklassen entsprechend der IWU-Wohngebäudetypologie inklusive Beispielgebäuden. Es werden folgende Größenklassen unterschieden: EFH = Einfamilienhaus, RH = Reihenhaushaus, MFH = Mehrfamilienhaus, GMH = Großes Mehrfamilienhaus, HH = Hochhaus. Übernommen aus [64].	- 10 -
Abbildung 3: Schematische Bodenabdeckung durch eine Malteserkreuz-Kamerakonfiguration. Übernommen aus [91].	- 11 -
Abbildung 4: Parameter der inneren und äußeren Orientierung eines Luftbildes	- 14 -
Abbildung 5: Darstellung eines Deep Learning Netzwerkes zur Objekterkennung in einem Bild mit sichtbarer Schicht (visible layer), versteckten Schichten (hidden layers) und dem Output. Übernommen aus [31].	- 18 -
Abbildung 6: Verlauf von Trainings- und Validierungs-Loss in Abhängigkeit der Epochen. Das linke Bild stellt einen optimalen Verlauf dar, bei dem die beiden Kurven konvergieren. Im mittleren Bild konnte das Modell am Trainings-Datensatz nicht ausreichend gut trainiert werden, es kommt zum Underfitting. Im rechten Bild wurde das Modell im Training zu sehr auf den Trainingsdatensatz spezialisiert, was zu Overfitting führt. Übernommen aus [12].	- 19 -
Abbildung 7: Schematische Ansicht des Faster R-CNN Algorithmus. Die schwarzen Balken stellen Schichten eines Neuronalen Netzes dar. In dem hier dargestellten Fall identifiziert Faster R-CNN im Eingangsbild zwei Objekte, deren Positionen durch die gestrichelten Rahmen im unteren rechten Bild markiert sind. Den Objekten wird mit einer Sicherheit (Score) von 0,94, beziehungsweise 0,96, die Klasse „Fenster“ zugeordnet.	- 20 -
Abbildung 8: Schematische Darstellung des Region Proposal Network. Schematisch nachgezeichnet aus [75].	- 21 -
Abbildung 9: Intersection over Union. Schematisch nachgezeichnet aus [92].	- 23 -
Abbildung 10: Beispielbild für die Bestimmung von Präzision und Recall (linke Seite) und schematische Darstellung einer PR-Kurve (rechte Seite). Die Fläche unter der Kurve entspricht der durchschnittlichen Präzision AP.	- 24 -
Abbildung 11: Beispiel für die Clusteranalyse eines Datensatzes mit $K_{\text{Cluster}} = 2$, $K_{\text{Cluster}} = 3$ und $K_{\text{Cluster}} = 4$. Die roten Kreuze markieren die Cluster-Mittelpunkte.	- 26 -
Abbildung 12: Beispiele für das Between-Cluster-Point-Scatter $B(C)$ (links) und das Within-Cluster-Point-Scatter $W(C)$ (rechts) in Abhängigkeit der Anzahl der Cluster K_{Cluster} .	- 27 -
Abbildung 13: Objektbezogene Bildsegmentierung im CMP-Datensatz [88] (links) und semantische Bildsegmentierung im eTRIMS-Datensatz [38] (rechts).	- 29 -

Abbildung 14: Darstellung der Methodik zur Bestimmung des Verhältnisses der Fenster- zur Wandfläche von Gebäuden als Flussdiagramm.....	31 -
Abbildung 15: Beispiel für die Ermittlung des optimalen K_{Cluster} für das Clustern der Fenster nach Breite und Höhe. Links oben ist die Fassade eines Gebäudes aus dem CMP-Datensatz [88] dargestellt, dessen Fenster geclustert wurden. Rechts oben ist die entsprechende $B(C)$ -Kurve dargestellt, rechts unten die $W(C)$ -Kurve. Links unten ist ein Streudiagramm der transformierten Fensterhöhe und -breite zu sehen. Die roten Kreuze markieren die Mittelpunkte der ermittelten Cluster.....	36 -
Abbildung 16: Beispiel für das Ergänzen von Fenstern, die vom ML-Modell nicht erkannt wurden. Die gepunkteten Fenster wurden vom Modell erkannt, das Fenster mit der Strichpunkt-Linie nicht. Die Kreuze markieren die möglichen Kombinationen aller einzigartigen x- und y-Werte.	38 -
Abbildung 17: Links: Darstellung einer Fassade aus dem CMP-Datensatz [88]. Mitte: Die Pixel des linken Bildes wurden entsprechend ihrer Klasse koloriert [88]. Rechts: Die Rahmen der Objekte, mit denen das linke Bild annotiert wurde.....	40 -
Abbildung 18: Beispiele für Augmentationen.....	41 -
Abbildung 19: Ein gelabeltes Bild aus dem Hamburg Image Dataset. Gelabelte Fenster sind rot umrandet, die gelabelte Fassade ist grün umrandet. Im oberen linken Teil des Bildes ist ein gerundetes Fenster zu erkennen, das rechteckig gelabelt wurde. Die Doppelfenster rechts im Bild wurden jeweils zu einem Label zusammengefasst. Das Schaufenster in der linken unteren Ecke wurde nicht als Fenster gelabelt, genauso wie die Dachfenster. Das Dach wurde nicht als Fassade gelabelt.....	43 -
Abbildung 20: Bestimmung der nach außen zeigenden Normalenvektoren eines Gebäudes. Normalenvektor 1 weist mit 0 eine gerade Anzahl an Schnittpunkten mit anderen Wänden desselben Gebäudes auf und zeigt dementsprechend nach außen. Normalenvektor2 weist mit 3 eine ungerade Anzahl an Schnittpunkten auf und zeigt nach innen.....	45 -
Abbildung 21: Sektoren (gestrichelte Linien) und Sichtachsen (durchgezogene Linien) zur Bestimmung sichtbarer Wände eines Gebäudes in einem Bild. Das Rechteck in der Mitte stellt ein Gebäude dar, die 4 Pfeile die Normalenvektoren der Wände.	46 -
Abbildung 22: Schematische Darstellung einer Wand in der Realität, in einem nicht rektifizierten Bild und in einem rektifizierten Bild.....	47 -
Abbildung 23: Alle Schritte der Aufbereitung des Schrägluftbildes eines Gebäudes.	50 -
Abbildung 24: Darstellung der Methodik zur Bestimmung der Baualtersklassen von Gebäuden als Flussdiagramm.....	53 -
Abbildung 25: Schematische Darstellung der Transformation von geografischen (UTM) Koordinaten einer Raster-Datei in Bild-Koordinaten.....	55 -

Abbildung 26: Ausschnitt einer historischen Karte. Grün umrandet sind die Konturen, die von OpenCV erkannt wurden.	- 56 -
Abbildung 27: Beispielhafte Darstellung eines Gebäudes, dem 4 Adressen zugewiesen sind. Die Adressen sind durch dünne schwarze Linien voneinander getrennt und nummeriert. Das Grundriss-Polygon des Gebäudes ist dick schwarz umrandet. Die erkannten Konturen sind schraffiert dargestellt. Die IoUs zwischen den erkannten Konturen und dem Grundriss-Polygon sind ganz rechts aufgelistet.	- 57 -
Abbildung 28: Ausschnitt der Karte „Jahrgang 1940-1950“ mit dem blau eingefärbten Grundriss des Gebäudes Weimarer Straße 43.	- 60 -
Abbildung 29: Verteilung der Gebäude des Testdatensatzes im Hamburger Stadtgebiet.	- 61 -
Abbildung 30: Verteilung der Baujahre der Gebäude im Testdatensatz bezogen auf die IWU-Baualtersklassen, die durch die Buchstaben über den Balken repräsentiert werden.	- 62 -
Abbildung 31: Abweichung (Delta) zwischen den Baujahren des Testdatensatzes und den Baujahren des ALKIS-Datensatzes. Ein negatives Delta bedeutet, dass das Gebäude im ALKIS-Datensatz älter ist als im Testdatensatz.	- 63 -
Abbildung 32: Verlauf von Trainings- und Validierungs-Loss für 12 Faster R-CNN-Modelle, die mit unterschiedlichen Hyperparametern trainiert wurden.	- 64 -
Abbildung 33: Verlauf von Trainings-Loss und Validierungs-Loss für das beste ML-Modell.	- 66 -
Abbildung 34: Vergleich der Ergebnisse des ML-Modells ohne Nachbearbeitung mit den nachbearbeiteten Ergebnissen für zwei unterschiedliche Score-Grenzwerte. Abgebildet ist eine Fassade des Gebäudes Veringstraße 149.	- 69 -
Abbildung 35: Fenster der Fassade aus Abbildung 34, wie sie im Hamburg Image Dataset gelabelt sind.	- 70 -
Abbildung 36: Abweichung $\Delta R_{\text{Gebäude, mean}}$ auf Basis der vom ML-Modell erkannten Fensterflächen mit und ohne Nachbearbeitung sowie auf Basis der IWU-Gebäudetypologie.	- 71 -
Abbildung 37: Verhältnis der Fenster- zur Wandfläche für die 6 Gebäude des Hamburg Image Datasets: Manuell ermittelt ($R_{\text{Gebäude, HI}}$, gestrichelt), auf Basis der nachbearbeiteten Ergebnisse des ML-Modells ($R_{\text{Gebäude, graustufen}}$) und auf Basis der IWU-Gebäudetypologie ($R_{\text{Gebäude, IWU}}$, schwarz). SG steht für den Score-Grenzwert, welcher für die Bestimmung der Fensterfläche variiert wurde.	- 73 -
Abbildung 38: Abweichung der auf Basis des 3D-Stadtmodells bestimmten gesamten Wandfläche A_{Wand} und der vom ML-Modell vorhergesagten gesamten Fensterfläche A_{Fenster} jedes Gebäudes zu den auf Basis des Hamburg Image Datasets manuell berechneten Werten. SG steht für den Score-Grenzwert, welcher für die Bestimmung der Fensterfläche variiert wurde.	- 74 -

Abbildung 39: Abweichung ΔR_{Wand} des vom ML-Modell vorhergesagten zum im Hamburg Image Dataset definierten Verhältnisses der Fenster- zur Wandfläche für alle Wände des Hamburg Image Datasets in Abhängigkeit der Ausrichtung der Wände. Der Score-Grenzwert beträgt 0,25.....	75 -
Abbildung 40: Fensterfläche der Gebäude des Hamburg Image Datasets, summiert entsprechend ihrer Ausrichtung. Es wird bei der Ausrichtung zwischen Nord, Ost, Süd und West unterschieden.	80 -
Abbildung 41: Durchschnittliche Abweichung ΔB_{Jmean} der geschätzten Baujahre von den tatsächlichen Baujahren der 247 Test-Gebäude, deren Standardabweichung σ sowie mAP bezüglich der Baualtersklassen in Abhängigkeit unterschiedlicher IoU-Grenzwerte IoU_{min}	82 -
Abbildung 42: Anzahl der Gebäude, die den IWU-Baualtersklassen auf Basis der tatsächlichen und geschätzten Baujahre zugewiesen wurden. Die Zahlen über den Balken stehen für das AP der jeweiligen Klasse.....	83 -
Abbildung 43: Maximal berechnet IoU-Werte für die geschätzten Baujahre des Testdatensatzes (Punkte). Die Kreuze sind die IoU-Mittelwerte der einzigartigen geschätzten Baujahre, über die eine schwarze Regressionsgerade gelegt wurde.	85 -
Abbildung 44: Grundriss-Polygon der Mengestraße 6 auf zwei Kartenausschnitten unterschiedlicher Jahrgänge.	86 -
Abbildung 45: Grundriss-Polygon der Weimarer Straße 43 auf drei Kartenausschnitten unterschiedlicher Jahrgänge.	87 -
Abbildung 46: Grundriss-Polygon (blau) der Weimarer Straße 49 auf drei Kartenausschnitten unterschiedlicher Jahrgänge. In Grün ist jeweils das Polygon der Kontur mit dem größten IoU dargestellt.	87 -
Abbildung 47: Grundriss-Polygon der Fährstraße 73 (blau) auf einem Ausschnitt der Karte „Jahrgang 1925-1930“. In Grün ist das erkannte Polygon mit dem größten IoU dargestellt.	88 -
Abbildung 48: Grundriss-Polygon der Veringstraße 149 auf zwei Kartenausschnitten unterschiedlicher Jahrgänge.	89 -
Abbildung 49: Grundriss-Polygon der Georg-Wilhelm-Straße 141 auf zwei Kartenausschnitten unterschiedlicher Jahrgänge.	89 -
Abbildung 50: Fassade der Mengestraße 6.....	E-1
Abbildung 51: Fassade der Weimarer Straße 43.....	E-1
Abbildung 52: Fassade der Weimarer Straße 49.....	E-1
Abbildung 53: Fassade der Fährstraße 73.....	E-2
Abbildung 54: Fassade der Veringstraße 149.....	E-2
Abbildung 55: Fassade der Georg-Wilhelm-Straße 141.....	E-3
Abbildung 56: Plakette mit Daten zur Zerstörung und dem Wiederaufbau des Gebäudes Weimarer Straße 43	E-3

Abbildung 57: Von der Sturmflut 1962 betroffenes Gebiet im Hamburger Stadtteil Wilhelmsburg. Das überflutete Gebiet ist blau hinterlegt. Die Positionen der 6 Gebäude des Hamburg Image Datasets sind durch rote Punkte markiert. Erstellt auf der Internetseite der Stadt Hamburg [32].	E-4
Abbildung 58: Luftbild der Gebäude Mengestraße 6 (links) und Hermann-Westphal-Straße 2a-c (rechts) [25].	E-5
Abbildung 59: PR-Kurve für unterschiedliche Score-Grenzwerte für das beste Modell aus Tabelle 6.	F-1

Tabellenverzeichnis

Tabelle 1: Auswahl von Metadaten des Hamburger Schrägluftbilddatensatzes	12 -
Tabelle 2: Hyperparameter des in dieser Arbeit verwendeten Faster R-CNN-Modells.	34 -
Tabelle 3: Einzigartige Koordinaten und Anzahl der dazugehörigen Fenster-Mittelpunkte für das Beispiel aus Abbildung 16.	38 -
Tabelle 4: Gebäude des Testdatensatzes (Hamburg Image Dataset)	42 -
Tabelle 5: Zulässige Sektoren für Normalenvektoren sichtbarer Wände in Abhängigkeit der Blickrichtung des Bildes.	46 -
Tabelle 6: Berechnete mAP für 12 Faster R-CNN-Modelle, die mit unterschiedlichen Hyperparametern trainiert wurden.	65 -
Tabelle 7: Unterschiedliche Nachbearbeitungen, die auf die Ergebnisse des ML-Modells angewendet wurden.	67 -
Tabelle 8: Berechnete mAP für unterschiedliche Score-Grenzwerte und Nachbearbeitungs-Varianten (siehe Tabelle 7).	67 -
Tabelle 9: Geschätzte und tatsächliche Baujahre und Baualtersklassen für die 6 Gebäude des Hamburg Image Datasets.	85 -
Tabelle 10: Vergleich der IWU-Baualtersklassen mit den geschätzten Baujahren, die den Gebäuden auf Basis der historischen Karten zugewiesen werden können.	90 -
Tabelle 11: Übersicht über die Verfügbarkeit von 3D-Gebäudemodellen, Schrägluftbildern und historischen Karten oder Luftbildern in Deutschland.	A-1
Tabelle 12: Fenster- und Wandflächen für die IWU-Wohngebäudetypen [64], erweitert um das Verhältnis der Fenster- zur Wandfläche.	C-1
Tabelle 13: Auswahl von Bild-Datensätzen, die zum Training von ML-Algorithmen zur Fensterflächenerkennung genutzt wurden.	D-1
Tabelle 14: Ergebnisse für $\Delta R_{\text{Gebäude, mean}}$ und σ auf Basis der vom ML-Modell erkannten Fenster mit und ohne Nachbearbeitung für unterschiedliche Score-Grenzwerte sowie auf Basis der IWU-Gebäudetypologie.	F-2

1 Einleitung

Der Klimawandel fordert einen Wandel in der Energieerzeugung und -nutzung. Deshalb hat die Bundesregierung die Klimaneutralität für Deutschland bis zum Jahr 2045 als Ziel ausgegeben [74]. Der Weg dorthin soll in Form der Energiewende beschritten werden, in deren Rahmen die deutsche Energieerzeugung auf erneuerbare Energien umgestellt wird, wo dies möglich ist. Dies betrifft die Sektoren Energiewirtschaft, Industrie, Verkehr, Gebäudebereich und Landwirtschaft.

Der Gebäudesektor war im Jahr 2021 mit 27,8 % ungefähr für ein Viertel des gesamten Endenergieverbrauches in Deutschland verantwortlich [90]. Über 90 % der Endenergie, die im Gebäudesektor verbraucht werden, wurden hierbei für Wärmeanwendungen verbraucht [89]. Dies macht deutlich, dass für die Energiewende der Gebäudesektor und die Erzeugung und der Verbrauch von Wärme einen entscheidenden Faktor darstellen. Die zum Erreichen der Klimaneutralität notwendige Transformation der Wärmeerzeugung, -verteilung und -abnahme wird als Wärmewende bezeichnet. Ziel der Wärmewende ist es, die Wärmeerzeugung klimaneutral zu gestalten und den Wärmebedarf so weit wie möglich zu senken. Dieses Ziel kann durch den Ausbau und die Modernisierung von Fernwärmenetzen erreicht werden. Fernwärme ermöglicht eine effizientere Wärmeerzeugung und einen einfacheren Umstieg auf neue Energieträger als die dezentrale Wärmeversorgung [71]. Um die Wärmenetze optimal betreiben zu können, sind Wärmebedarfsprognosen notwendig. In diesen Prognosen wird der Wärmebedarf der Abnehmer eines Fernwärmenetzes vorhergesagt. So können aktuelle und zukünftige Verbräuche und Lastspitzen abgeschätzt und für die Wärmebereitstellung im Fernwärmenetz berücksichtigt werden [37]. Hierbei gilt, dass eine hohe räumliche und zeitliche Auflösung der Wärmebedarfsprognosen eine effizientere Nutzung des Wärmenetzes ermöglicht [84]. Die Daten für den Wärmebedarf einzelner Gebäude sind jedoch selten öffentlich zugänglich [37]. Um trotzdem Wärmebedarfsprognosen durchführen zu können, können die Gebäude simuliert oder ihr Wärmebedarf über Machine Learning-Modelle angenähert werden. In beiden Fällen ist für eine genaue Wärmebedarfsprognose ein hoher Detailgrad der Eingangsgrößen notwendig. Hierbei sind die Fensterflächen der Gebäude von entscheidender Bedeutung, da sie eine Abschätzung der solaren Gewinne zulassen. Diese wiederum sind eine wichtige Grundlage für räumlich und zeitlich hoch aufgelöste Wärmebedarfsprognose, da sie ursächlich für kurzfristige Schwankungen im Wärmebedarf sein können [20]. Statt der Fensterfläche wird für Wärmebedarfsprognosen das Verhältnis der Fenster- zur Wandfläche verwendet, da es dimensionslos ist. Darüber hinaus sind auch die Baujahre wichtige Eingangsgrößen für Wärmebedarfsprognosen. Aus ihnen kann auf Baualtersklassen geschlossen werden, auf deren Basis die energetischen Eigenschaften eines Gebäudes bestimmt werden können [37].

1.1 Überblick über die Arbeit

Aus dem oben beschriebenen Problem der unzureichenden Datenlage für detaillierte Wärmebedarfsprognosen lassen sich für diese Masterarbeit zwei Forschungsfragen ableiten:

1. Wie kann das Verhältnis $R_{\text{Gebäude}}$ der Fenster- zur Wandfläche eines beliebigen Wohngebäudes in Hamburg auf Basis öffentlich zugänglicher Daten bestimmt werden?
2. Wie kann die Baualtersklasse eines beliebigen Wohngebäudes in Hamburg auf Basis öffentlich zugänglicher Daten bestimmt werden?

Die Beantwortung dieser Fragen soll als Basis für räumlich und zeitlich hochaufgelöste Wärmebedarfsprognosen dienen. Die Fragen beziehen sich auf Hamburger Wohngebäude, da die Ergebnisse dieser Arbeit im Rahmen des Projektes „Integrierte Wärmewende Wilhelmsburg“ weiterverwendet werden sollen.

Den Einstieg in diese Masterarbeit bildet die Vorstellung von Arbeiten in Kapitel 2.1, die sich mit Energiebedarfsprognosen, der Erkennung von Fenstern in Gebäudefassaden oder der Bestimmung des Baujahres eines Gebäudes befassen. In Anlehnung an diese Arbeiten wurden auch für diese Masterarbeit öffentlich zugängliche Daten genutzt, die in Kapitel 2.2 beschrieben werden. Es handelt sich hierbei um die ALKIS-Datenbank, das 3D-Stadtmodell von Hamburg, die Gebäudetypologie der IWU und Schrägluftbilder und historische Karten von Hamburg. Während die Karten zur Bestimmung der Baualtersklasse genutzt wurden, waren die Schrägluftbilder für die Abschätzung des Verhältnisses der Fenster- zur Wandfläche notwendig. Hierzu mussten sie jedoch aufbereitet werden, was in Kapitel 2.3 beschrieben ist. Zur Erkennung der Fensterflächen wurden die Schrägluftbilder in ein Machine Learning-Modell gegeben. Die Grundlagen zu Machine Learning-Methoden zur Objekterkennung sind in Kapitel 2.4 beschrieben. Das konkrete Vorgehen zur Bestimmung des Verhältnisses der Fenster- zur Wandfläche eines Gebäudes wird in Kapitel 3.1 vorgestellt. Im darauffolgenden Kapitel 4 wird die Methodik zur Bestimmung der Baualtersklasse eines Gebäudes beschrieben. Die jeweiligen Ergebnisse werden in Kapitel 5 vorgestellt und diskutiert. In Kapitel 6 werden schließlich die Erkenntnisse dieser Arbeit in einem Fazit zusammengefasst, bevor in Kapitel 7 ein Ausblick auf mögliche Anwendungsbereiche und offene Fragen gegeben wird.

2 Theorie und Stand der Technik

Der Einsatz von Gebäudemodellen zur Erstellung von Wärmebedarfsprognosen ist ein in der Literatur viel diskutiertes Thema. Eine besondere Rolle nimmt hierbei die Beschaffung wärmebedarfsrelevanter Informationen der Gebäude ein. Zu diesen Informationen zählen auch das Verhältnis von Fenster- zu Wandflächen und Baualtersklassen der Gebäude. Die Basis für die Informationsbeschaffung bilden 3D-Stadtmodelle, die neben geometrischen auch semantische Informationen zu den Gebäuden enthalten können. Informationen, die sich nicht aus den 3D-Stadtmodellen entnehmen lassen, werden in der Literatur häufig über öffentlich zugängliche Quellen, zum Beispiel in Form von digitalen Katastern oder Luftbildern, ermittelt. Die Auswertung dieser Quellen erfolgt in vielen Fällen über Machine Learning Algorithmen. In diesem Kapitel werden einige Ansätze aus der Literatur zu Wärmebedarfsprognosen und der Abschätzung von Fensterflächen und Baujahren von Gebäuden vorgestellt. Darüber hinaus enthält es eine Beschreibung der öffentlich zugänglichen Daten für Hamburg, die für diese Aufgaben in Frage kommen, sowie einen Überblick über geeignete Machine Learning-Methoden.

2.1 Arbeiten zum Thema

Die Grundlage für die in dieser Masterarbeit entwickelte Methodik bilden eine Reihe von Veröffentlichungen, die sich mit Wärmebedarfsprognosen anhand von 3D-Stadtmodellen, der Erkennung von Fenstern in Gebäudefassaden sowie der Bestimmung von Baujahren beschäftigen. Diese Arbeiten werden im Folgenden kurz vorgestellt.

2.1.1 Wärmebedarfsprognosen anhand von 3D-Stadtmodellen

3D-Stadtmodelle sind eine Möglichkeit, bei geringem personellem und zeitlichem Aufwand die Genauigkeit von Wärmebedarfsprognosen für Gebäude zu erhöhen. Vier Arbeiten, die dieses Thema behandeln, sollen hier vorgestellt werden. Kaden [37] beschäftigte sich mit der gebäudescharfen und großräumigen Berechnung von Energiebedarfen für Wärme, Strom und Warmwasser sowie der Ermittlung von Einsparpotentialen. Er verwendete hierfür frei zugängliche Geobasisdaten, 3D-Stadtmodelle im CityGML-Format, statistische Werte des Instituts für Wohnen und Umwelt (IWU), sowie den Kartendienst DualMaps. Die untersuchten Gebiete waren Berlin, London, Pullach und Trento. Die Gebäudeinformationen, die Kaden aus den zuvor genannten Quellen nicht beziehen konnte, ermittelte er durch Begehungen vor Ort. Kaden kam zu dem Ergebnis, dass sich 3D-Stadtmodelle im CityGML-Format für die Energiebedarfsabschätzung eignen. Nichtsdestotrotz traten ungenaue Prognosen in Folge von fehlerhaften Gebäudemodellen auf, bei denen z.B. Dachflächen oder Kellerwände als Außenwände definiert worden waren. Die Verwendung eines 3D-Stadtmodells für Wärmebedarfsprognosen ist auch Thema der Arbeit von

Nippgen [69]. Sie beschäftigte sich mit der Aufbereitung des Hamburger 3D-Stadtmodells für Wärmebedarfsprognosen in „SimStadt“, einem Programm für Energie-Analysen, sowie der Durchführung einer Wärmebedarfsprognose. Hierbei ergänzte sie nicht vorhandene Daten aus dem amtlichen Liegenschaftskatasterinformationssystem (ALKIS) mit Hilfe statistischer Werte des IWU oder durch manuelle Vorgabe von Werten. Eine Wärmebedarfsabschätzung ohne 3D-Stadtmodell, dafür aber mit dem digitalen Kataster ALKIS, wurde von Dochev et al. [21] durchgeführt. Hierzu verwendeten die Autor*innen Referenzgebäude, für die die energetischen Eigenschaften bekannt waren. Diese Gebäude wurden einer Gebäudeklasse zugeordnet und in die Hamburger ALKIS-Datenbank integriert. Die übrigen Gebäude wurden ebenfalls einer Gebäudeklasse und damit einem Referenzgebäude zugeordnet, sodass ihr Wärmebedarf bestimmt werden konnte. Dochev et al. kommen jedoch zu dem Schluss, dass eine Zuordnung von Referenzgebäuden in ein digitales Kataster schnell zu vielen Annahmen führen und komplex werden kann, was wiederum Einfluss auf die Qualität der Wärmebedarfsprognose hat. In einer weiteren Arbeit [20] verbanden Dochev et al. die Ansätze von Dochev et al. [21] und Nippgen [69], um die Wärmebedarfe von Gebäuden in Berlin zu bestimmen. Hierzu erstellten die Autor*innen ein 3D-Stadtmodell auf Basis von Schrägluftbildern, welches sie um Fensterflächen ergänzten. Diese wurden in den rektifizierten Schrägluftbildern anhand von Farbgradienten und geometrischen Eigenschaften automatisiert erkannt. Zum Vergleich wurden die Fensterflächen von 13 zufällig ausgewählten Gebäuden im untersuchten Gebiet manuell in Google Maps ausgemessen. Es zeigte sich, dass mit den IWU-Referenzgebäuden die Fensterflächen der Gebäude im Untersuchungsgebiet systematisch überschätzt wurden, während die automatische Fenstererkennung zu einer systematischen Unterschätzung führte.

In den vorgestellten Arbeiten wird deutlich, dass 3D-Stadtmodelle und digitale Kataster eine gute Basis für Wärmebedarfsprognosen bilden. In allen Arbeiten wurden jedoch mit unterschiedlichen Ansätzen zusätzliche Informationen zu den Gebäuden gesammelt, um die Genauigkeit der Prognosen zu erhöhen. Im Folgenden soll hiervon auf die Bestimmung der Fensterflächen und Baujahre von Gebäuden näher eingegangen werden.

2.1.2 Erkennung von Fenstern in Gebäudefassaden

Während Dochev et al. [20] zur Erkennung von Fenstern in Gebäudefassaden auf Farbgradienten und geometrische Eigenschaften setzten, wird für diese Aufgabe in der Literatur zunehmend auf Machine Learning Algorithmen zurückgegriffen. Dies zeigen die in diesem Kapitel vorgestellten Arbeiten, die sich ausschließlich dem Thema der Fenstererkennung widmen. So ließ Lippoldt Schrägluftbilder von Berlin von einem Mask R-CNN Algorithmus pixelweise (semantisch) segmentieren, um Fensterflächen zu erkennen [63]. Sie wählte diese Methode, da die Bilder perspektivisch verzerrt waren. Das Ergebnis des Algorithmus versuchte sie durch Clustern zu

verbessern, indem sie sich die festen Muster, in denen Fenster häufig angeordnet sind, zu Nutze machte. Diese Nachbearbeitung nahmen auch Ma und Ma in ihrer Arbeit [65] vor. Allerdings nutzten sie zur Erkennung der Fenster die objektbezogene Segmentierung mittels eines modifizierten Faster R-CNN Algorithmus, da sie, anders als Lippoldt, mit rektifizierten Bildern arbeiteten. Trotz zufriedenstellender Ergebnisse zeigten Ma und Ma die vielfältigen Probleme bei der Erkennung von Fenstern in Bildern auf, zum Beispiel in Form von Reflektionen oder Verschattungen. Nordmark und Ayenew verwendeten in ihrer Arbeit [70] den Mask R-CNN Algorithmus, mit dem sie dank *Transfer Learning* trotz eines kleinen Trainings-Datensatzes gute Ergebnisse erzielten. Hierbei beschreibt Transfer Learning das Verwenden eines bereits trainierten Algorithmus, den die Autoren durch Training an ihrem Datensatz spezialisierten. Eine Nachbearbeitung der Ergebnisse nahmen sie nicht vor. Die Autoren trainierten den Algorithmus nur für die Klasse „Fenster“, merkten jedoch an, dass sich die Ergebnisse womöglich verbessern ließen, wenn mehr Klassen verwendet würden. Dies ist auch das Fazit von Sun et al [83], die einen modifizierten Mask R-CNN Algorithmus verwendeten. Ihr Ansatz war es, nicht jedes Fenster für sich zu betrachten, sondern die Positionen und Verhältnisse der Fenster einer Fassade untereinander zu berücksichtigen, indem sie den Mask R-CNN Algorithmus um zwei Module ergänzten. Etwas einfacher ist der Ansatz von Wang [92], der den Faster R-CNN Algorithmus verwendete und die Ergebnisse nachträglich mittels Clustern und manueller Sortierung bezüglich Größe und Ausrichtung korrigierte. Wang kam zu dem Schluss, das Fenster besser erkannt werden, wenn die Fassade einfach strukturiert ist.

Mit den in diesem Kapitel beschriebenen Methoden ist es also möglich, die Fenster einer Gebäudefassade in Bildern zu erkennen und dementsprechend das Verhältnis der Fenster- zur Wandfläche des Gebäudes zu berechnen. Hierbei können unterschiedliche Algorithmen und Nachbearbeitungen angewendet werden. Auch für die Bestimmung der Baujahre von Gebäuden gibt es unterschiedliche Ansätze, von denen einige im Folgenden vorgestellt werden.

2.1.3 Bestimmung des Baujahres

Nicht immer sind die Baujahre von Gebäuden öffentlich zugänglich oder überhaupt bekannt. In diesem Fall gibt es mehrere Möglichkeiten, auf das Alter eines Gebäudes zu schließen. Eine dieser Möglichkeiten stellen Biljecki und Sindram vor [8]. In ihrer Arbeit ermittelten die Autoren das Alter eines Gebäudes auf Basis anderer, öffentlich zugänglicher Daten eines LoD1-Modells. Hierzu verwendeten sie eine Random-Forest Regression. Biljecki und Sindram war es so möglich, die Entstehungsperiode eines Gebäudes mit einer Genauigkeit von ca. 10 Jahren vorherzusagen. Wichtige Eingabeparameter hierbei waren die Deckenhöhe, die Nutzungsart des Gebäudes, Geometrische Eigenschaften des Gebäudes und das Alter der umstehenden Gebäude, sowie weitere Eigenschaften der Nachbarhäuser. Einen großen Einfluss bei der Vorhersage des

Baujahres hatten hierbei die umstehenden Gebäude. Wenn das Alter dieser Gebäude bekannt war, war die Vorhersage des Baujahres am genauesten. Das Alter der Nachbargebäude wurde unter anderem auch von Nippgen in [69] herangezogen, um mit Hilfe der Software FME (Feature Manipulation Engine) auf das Alter eines Gebäudes zu schließen. Sie bildete hierbei den Mittelwert der Baujahre aller Gebäude innerhalb eines festgelegten Radius um das Gebäude, dessen Baujahr gesucht wurde. Dieser Wert wurde letzterem Gebäude dann zugeschrieben. Eine alternative Methode ist der Ansatz von Kaden [37]: Dieser lud offizielle Karten des Senats Berlin aus dem Internet herunter, in denen händisch unterschiedliche Baualtersklassen in unterschiedlichen Farben markiert worden waren. Da diese Karten jedoch nicht überall mit dem 3D-Stadtmodell übereinstimmten, entwickelte Kaden einen Workflow, um die Gebäude des 3D-Stadtmodells auf den Karten lokalisieren zu können. Das Baujahr der Gebäude bestimmte er anschließend, indem er die Gebäude anhand des Farbcodes für die Baualtersklassen mit Hilfe eines Machine Learning Modells klassifizierte.

In den in diesem Kapitel vorgestellten Arbeiten wird deutlich, dass zur Bestimmung des Baujahres (und damit der Baualtersklasse) eines Gebäudes unterschiedliche Datenquellen herangezogen werden können, wobei die Kenntnis über Eigenschaften von Nachbargebäuden die Genauigkeit der Abschätzung verbessern kann.

2.2 Datengrundlage in Hamburg

Neben den unterschiedlichen Ansätzen, die in den vorgestellten Veröffentlichungen angewendet werden, kommt dort auch eine Vielzahl an unterschiedlichen Datenquellen zum Einsatz. Da die vorgestellten Arbeiten die Vorlage für die in dieser Arbeit entwickelte Methodik bilden, wurde die Verfügbarkeit öffentlich zugänglicher Daten für die Bestimmung des Verhältnisses von Fenster- zu Wandflächen und Baualtersklassen von Gebäuden für Hamburg überprüft. Die Ergebnisse werden in diesem Kapitel vorgestellt.

2.2.1 ALKIS

Das Amtliche Liegenschaftskatasterinformationssystem (ALKIS) enthält die Basisdaten des amtlichen Vermessungswesens und ist deutschlandweit standardisiert. Die ALKIS-Daten jedes Bundeslandes bestehen aus einem Grunddatenbestand sowie länderspezifischen zusätzlichen Daten. Sie sind online öffentlich zugänglich und können in Form von GML-Dateien unter anderem von der Urban Data Platform der Stadt Hamburg heruntergeladen werden. Die in dieser Arbeit verwendeten ALKIS-Datensätze enthalten die Adressen der Hamburger Gebäude sowie Informationen über Funktionen, Baujahre und Grundrisse. Hierbei muss angemerkt werden, dass die Baujahre nicht für alle Gebäude vorliegen. In den Hamburger ALKIS-Datensätzen sind die Baujahre für ca. 50 % der Gebäude aufgeführt, wobei fast alle Wohngebäude ein Baujahr

angegeben haben. Eine Verpflichtung zum Dokumentieren des Baujahres eines Gebäudes gibt es in Hamburg seit 1998. [21, 27]

Im Rahmen dieser Masterarbeit wurden Abweichungen bezüglich der Baujahre zwischen ALKIS und historischen Karten Hamburgs (siehe Kapitel 2.2.4), Plaketten an Gebäuden sowie Angaben des Denkmalschutzamtes entdeckt, die in Kapitel 3.2.3 ausführlicher beschrieben sind. Die im ALKIS angegebenen Baujahre wurden daher im Folgenden als nicht verlässlich bewertet.

2.2.2 Das 3D-Stadtmodell von Hamburg, CityGML und CityJSON

Im ALKIS enthalten ist auch das 3D-Stadtmodell Hamburgs. 3D-Stadtmodelle können bei der Erstellung einer Energiebedarfsprognose hilfreich sein (siehe Kapitel 2.1.1). Hierbei spielt es jedoch eine Rolle, wie detailliert die Gebäude in dem Modell abgebildet werden [66]. Deshalb werden bei der Darstellung von 3D-Stadtmodellen 5 Detailgrade unterschieden, die als „Level of Detail“ (LoD) bezeichnet werden [9, 69]. Ein LoD0-Modell eines Gebäudes besteht ausschließlich aus dem Grundriss. Mit steigender Ziffer nimmt auch der Detailgrad des Modells zu: Beim LoD1-Modell wird das Gebäude als Block dargestellt, das LoD2-Modell bildet zusätzlich das Dach ab, wobei komplexe Dachformen vereinfacht werden. Beim LoD3-Modell kommen Strukturen und Fenster hinzu. Das LoD4-Modell ist das detaillierteste und enthält neben der Außenhülle auch eine Darstellung der Innenräume des Gebäudes. Die 5 Detailgrade sind in Abbildung 1 beispielhaft für ein Wohnhaus abgebildet.



Abbildung 1: Beispielhafte Darstellung der 5 Detailgrade (LoDs) eines 3D-Gebäudemodells. Übernommen aus [7].

Ein gängiges Format für die Speicherung und Weitergabe von 3D-Stadtmodellen ist das CityGML-Format (City Geography Markup Language), das auf dem XML-Format (Extensible Markup Language) basiert [69]. CityGML enthält Informationen über die 3D-Geometrie, Topologie, Semantik und das Erscheinungsbild von städtischen Objekten, zu denen nicht nur Gebäude zählen, sondern auch Infrastruktur, Wasserflächen, Stadtmöbel und Vegetation [35]. CityGML wurde 2008 als Standard vom Open Geospatial Consortium anerkannt [10]. Für viele Städte und Gemeinden weltweit, darunter auch das gesamte Bundesgebiet, gibt es CityGML-Datenbanken.

Diese enthalten Daten in den Auflösungen LoD1 bis LoD3, wobei LoD1 und LoD2 am häufigsten vorliegen, so auch für Hamburg und die anderen deutschen Bundesländer [66]. Im Oist in Tabelle 11 eine Übersicht über die Verfügbarkeit der 3D-Gebäudemodelle in Form von CityGML-Dateien für alle Bundesländer aufgeführt, die Auskunft über die Übertragbarkeit der Ergebnisse dieser Masterarbeit auf andere Bundesländer gibt. Eine CityGML-Datei kann folgende Informationen über ein Gebäude enthalten [6, 37, 69]:

- Den Namen des Gebäudes. Mit diesem Namen kann die Verbindung zur ALKIS-Datenbank hergestellt werden [69].
- Die Gebädefunktion, die anhand eines Schlüssels beschrieben wird, der vom Institut für Wohnen und Umwelt (IWU) vorgegeben wird [64].
- Das Baujahr
- Die Bauweise. Auch hier werden Kürzel verwendet, die vom IWU vorgegeben werden.
- Bei $\text{LoD} \geq 2$: Die Grund-, Seiten- und Dachflächen (inkl. Dachform) eines Gebäudes, inkl. Öffnungen für Fenster und Türen
- Die Anzahl an Stockwerken über der Erdoberfläche
- Die Identifikationsnummer des Gebäudes.
- Das verwendete Koordinatenreferenzsystem

Die CityGML-Datei mit dem LoD2-Modell Hamburgs kann über die Internetseite des Metadatenverbundes der Länder Brandenburg, Bremen, Hamburg, Hessen, Mecklenburg-Vorpommern, Saarland, Sachsen und Sachsen-Anhalt (MetaVer) als Teil der ALKIS-Datenbank heruntergeladen werden [26].

Eine Alternative zum CityGML-Format bietet das CityJSON-Format. CityJSON ist eine auf JSON (JavaScript Object Notation) basierende Variante des CityGML-Formats. Sie ist ebenfalls vom Open Geospatial Consortium als Standard anerkannt. Im Vergleich zu CityGML soll CityJSON dieselben Informationen kompakter und übersichtlicher darstellen. So kann eine CityGML-Datei mit Hilfe von CityJSON um das 6-fache komprimiert werden, indem die Strukturen der CityGML abgeflacht werden [15]. Allerdings unterstützt CityJSON nicht alle Funktionen von CityGML [59]. So kann in CityJSON zum Beispiel kein LoD4-Modell erstellt werden. Die Konvertierung einer CityGML-Datei in eine CityJSON-Datei ist mit frei zugänglicher Software möglich und in der Arbeit von Ledoux et al. beschrieben [59]. Da die beiden Formate CityGML und CityJSON ohne Verlust von Information in beide Richtungen konvertiert werden können, ist es möglich, einige Aufgaben im GML-Format auszuführen und andere im JSON-Format.

2.2.3 Wohngebäudetypologie der IWU

Informationen, die zum Beispiel im 3D-Modell eines Gebäudes nicht enthalten sind, können mit Hilfe einer Gebäudetypologie abgeschätzt werden. Im Rahmen des EU-Projekts TABULA („Typology Approach for Building Stock Energy Assessment“) erstellte das Institut für Wohnen und Umwelt (IWU) eine Wohngebäudetypologie für Deutschland [64]. Diese Typologie aus dem Jahr 2014 unterteilt den deutschen Wohngebäudebestand in 5 Größen- und 12 Baualtersklassen. Hinzu kommen Sonderfälle für Fertighäuser und Wohngebäude, die in der DDR errichtet wurden. Für jede Kombination aus Größen- und Baualtersklasse wurde hierbei ein Beispielgebäude definiert, das exemplarisch für alle Gebäude der entsprechenden Klassen steht und als Grundlage für Energiebedarfsprognosen und andere Modellierungen dienen kann. Für diese Beispielgebäude sind Dach-, Wand-, Fenster-, Boden- und Türflächen sowie die dazugehörigen U-Werte angegeben. Eine Übersicht über die Größen- und Baualtersklassen sowie die dazugehörigen Beispielgebäude ist in Abbildung 2 dargestellt.

Baualtersklasse			EFH	RH	MFH	GMH	HH
			Basis-Typen				
A	... 1859		EFH A		MFH A		
B	1860 ... 1918		EFH B	RH B	MFH B	GMH B	
C	1919 ... 1948		EFH C	RH C	MFH C	GMH C	
D	1949 ... 1957		EFH D	RH D	MFH D	GMH D	
E	1958 ... 1968		EFH E	RH E	MFH E	GMH E	HH E
F	1969 ... 1978		EFH F	RH F	MFH F	GMH F	HH F
G	1979 ... 1983		EFH G	RH G	MFH G		
H	1984 ... 1994		EFH H	RH H	MFH H		
I	1995 ... 2001		EFH I	RH I	MFH I		
J	2002 ... 2009		EFH J	RH J	MFH J		
K	2010 ... 2015		EFH K	RH K	MFH K		
L	2016 ...		EFH L	RH L	MFH L		
Sonderfälle	F/F	1969 ... 1978 Fertig- haus	EFH F/F				
	NBL_D	1946 ... 1960			MFH D		
	NBL_E	1961 ... 1969			MFH E		
	NBL_F	1970 ... 1980				GMH F	HH F
	NBL_G	1981 ... 1985				GMH G	HH G
	NBL_H	1986 ... 1990				GMH H	HH H

Abbildung 2: Baualters- und Größenklassen entsprechend der IWU-Wohngebäudetypologie inklusive Beispielgebäuden. Es werden folgende Größenklassen unterschieden: EFH = Einfamilienhaus, RH = Reihenhause, MFH = Mehrfamilienhaus, GMH = Großes Mehrfamilienhaus, HH = Hochhaus. Übernommen aus [64].

Eine Auflistung der Fenster- und Wandflächen sowie des Verhältnisses von Fenster- zu Wandflächen für alle Gebäudetypen ist im Anhang C in Tabelle 12 angefügt.

2.2.4 Luftaufnahmen und Karten von Hamburg

Die in den Kapiteln 2.2.1 bis 2.2.3 vorgestellten Informationsquellen für wärmebedarfsbezogene Parameter können durch Luftbilder und Karten ergänzt werden. In dieser Masterarbeit werden

hierzu ein Schrägluftbilddatensatz sowie ein Datensatz mit historischen Karten verwendet. Beide Datensätze bilden das Hamburger Stadtgebiet ab und sind öffentlich zugänglich.

Das **digitale Schrägluftbild Hamburg** kann über das Geoportal der Hansestadt Hamburg aufgerufen werden. Es handelt sich hierbei um einen Datensatz mit über 300.000 JPEG-Bildern, die von einem Flugzeug aus aufgenommen wurden. Die Aufnahme erfolgte mit einer Malteserkreuz-Kamerakonfiguration. Bei dieser Art der Konfiguration werden 5 Kameras so am Flugzeug installiert, dass sie gleichzeitig Bilder zu allen 4 Seiten und senkrecht nach unten aufnehmen können [91]. Die Bodenfläche, die in dieser Konfiguration von den Kameras abgedeckt wird, ist schematisch in Abbildung 3 dargestellt. Der Datensatz besteht nur aus den Schrägluftbildern, die zu den 4 Seiten aufgenommen werden und in Abbildung 3 als „oblique“ gekennzeichnet sind. Die Bilder des Datensatzes decken das gesamte Hamburger Stadtgebiet ab. Der zum Zeitpunkt des Verfassens dieser Arbeit aktuelle Datensatz ist aus dem Jahr 2020. Die Bilder können in 5 verschiedenen Auflösungen abgerufen werden, wobei die höchste Auflösung 20 bis 40 $\frac{\text{cm}}{\text{Pixel}}$ entspricht. Die Bilder wurden zu einem Zeitpunkt aufgenommen, als die Bäume unbelaubt waren, was eine Untersuchung der Gebäudefassaden in den Bildern erleichtert.

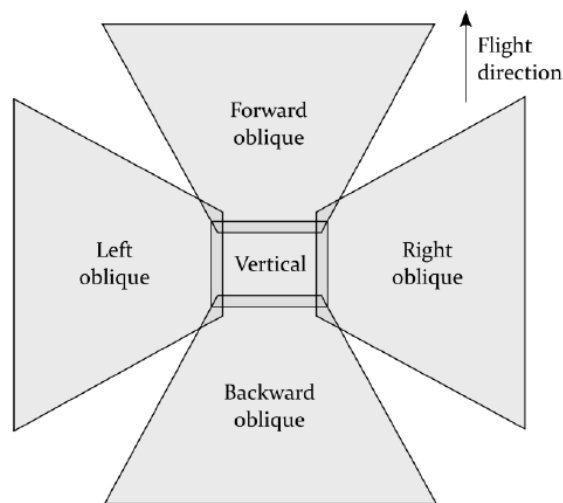


Abbildung 3: Schematische Bodenabdeckung durch eine Malteserkreuz-Kamerakonfiguration. Übernommen aus [91].

Neben den Bildern selbst enthält der Datensatz auch Metadaten zu jedem zusammengesetzten Bild. Eine Auswahl dieser Metadaten ist in Tabelle 1 aufgelistet. Die geografischen Koordinaten des Schrägluftbilddatensatzes beziehen sich auf das Koordinatenreferenzsystem „EPSG: 25832“. [24, 25]

Tabelle 1: Auswahl von Metadaten des Hamburger Schrägluftbilddatensatzes

Bezeichnung der Metadaten	Bedeutung	Einheit
name	Name des Bildes (z.B. „B204_04948“)	[-]
width, height	Breite und Höhe des Bildes	[Pixel]
view-direction	Blickrichtung der Kamera bei der Aufnahme des Bildes, die in Form eines Indexes zwischen 1 und 4 angegeben wird (1: Nord, 2: Ost, 3: Süd, 4: West)	[-]
groundCoordinates	Koordinaten der Eckpunkte der Bodenfläche, die auf dem Bild dargestellt wird.	[m]
p-to-image	Transformationsmatrix für die Transformation von geografischen Koordinaten in Bildkoordinaten	[-]

Der Datensatz „**Historische Karten 1:5000 – Auswahl**“ wird vom Landesbetrieb Geoinformation und Vermessung (LGV) Hamburg über einen Web Map Service (WMS) öffentlich zur Verfügung gestellt. Er beinhaltet historische Karten des Hamburger Stadtgebietes. Die älteste enthaltene Karte ist aus dem Jahr 1925, die neueste aus dem Jahr 2020. Für den Zeitraum dazwischen enthält der Datensatz eine Karte pro Jahrzehnt, wobei die Karten nach den Grenzen des jeweiligen Jahrzehnts benannt werden. Die älteste Karte des Datensatzes heißt demnach „Jahrgang 1925-1930“, die darauffolgende „Jahrgang 1930-1940“. Ab dem Jahr 2005 repräsentieren die Karten keinen Zeitraum mehr, sondern jeweils nur ein Jahr. Die Karte für 2005 heißt zum Beispiel „Jahrgang 2005“, die Karte für 2010 „Jahrgang 2010“. Ab der Karte „Jahrgang 1960-1970“ decken die Karten das gesamte Hamburger Stadtgebiet ab. Die Karten werden in Form von Raster-Dateien zugänglich gemacht, die Informationen über die geografischen Koordinaten des abgebildeten Stadtgebietes enthalten. Das Koordinatenreferenzsystem ist auch hier „EPSG: 25832“. [23, 55]

Um die Übertragbarkeit der Ergebnisse dieser Masterarbeit zu verdeutlichen ist, wie für die 3D-Stadtmodelle, in Tabelle 11 die Verfügbarkeit vergleichbarer Schrägluftbilder und historischer Karten oder Luftbilddaufnahmen in anderen Bundesländern aufgelistet. Neben den in

der Tabelle aufgeführten Quellen können Schrägluftbilder über den Kartendienst Bing Maps von Microsoft abgerufen werden. Dieser stellt unter der Funktion „Vogelperspektive“ Schrägluftbilder zur Verfügung, die entweder über die Webanwendung [68] angesehen oder über eine Programmierschnittstelle (API) [67] abgerufen werden können. Die Abdeckung Deutschlands durch die Vogelperspektive von Bing Maps konnte im Rahmen dieser Arbeit für die 16 deutschen Landeshauptstädte bestätigt werden.

Die öffentlich zugänglichen Daten für Hamburg bilden eine gute Basis, um das Verhältnis von Fenster- zu Wandflächen und die Baualtersklassen von Gebäuden bestimmen zu können. Die Informationen, die in den Luftaufnahmen enthalten sind, können diesen jedoch nicht direkt entnommen werden. Deshalb werden im Folgenden Methoden zur Aufbereitung und Auswertung der Aufnahmen vorgestellt.

2.3 Aufbereitung von Luftbildern

Um die in Luftbildern enthaltenen Informationen auswerten zu können, müssen die Bilder in ihren geografischen Kontext eingeordnet werden. Dies geschieht durch Georeferenzierung. Eine weitere Möglichkeit zur Aufbereitung von Luftbildern ist die Rektifikation, bei der die in den Bildern auftretenden Verzerrungen behoben werden. Beide Methoden werden im Folgenden vorgestellt.

2.3.1 Georeferenzierung

Bei der Georeferenzierung eines Luftbildes werden dem Bild räumliche Informationen zugewiesen, sodass die Position des Bildes in einem geografischen Koordinatensystem bestimmt werden kann. Bei diesen räumlichen Informationen handelt es sich häufig um die innere und äußere Orientierung des Bildes [91]. Die Innere Orientierung beschreibt hierbei die Orientierung der Bildfläche im Inneren der Kamera und wird durch die Matrix K beschrieben [39, 91]:

$$K = \begin{bmatrix} c_{cam} & 0 & x_K \\ 0 & c_{cam} & y_K \\ 0 & 0 & 1 \end{bmatrix} \quad [2.1]$$

Hierbei beschreibt c_{cam} die Kamerakonstante, die vereinfachend mit der Brennweite der Kamera gleichgesetzt werden kann. Die Hauptpunktlage des Bildes, die im Idealfall dem Bildmittelpunkt entspricht, wird durch x_K und y_K dargestellt. Die äußere Orientierung eines Luftbildes wird durch die Lage der Bildebene im entsprechenden geografischen Koordinatensystem definiert. Hierzu wird die Rotation und Translation der Kamera, mit der das Bild aufgenommen wurde, in Form der Rotationsmatrix R_{rot} und des Translationsvektors t beschrieben [58, 79]:

$$R_{rot} \quad [2.2]$$

$$= \begin{bmatrix} c a_y * c a_p & s a_y * c a_p & -s a_p \\ c a_y * s a_p * s a_r - c a_r * s a_y & s a_y * s a_p * s a_r + c a_r * c a_y & c a_p * s a_r \\ c a_r * s a_p * c a_y + s a_r * s a_y & s a_y * s a_p * c a_r - s a_r * c a_y & c a_p * c a_r \end{bmatrix}$$

$$t = \begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix} \quad [2.3]$$

Aufgrund des Umfangs der Matrix R_{rot} wurde in Gleichung [2.2] der Cosinus durch c und der Sinus durch s ersetzt. Die Variablen a_y , a_p , und a_r beschreiben die Drehung der Kamera in drei Richtungen: Die Drehung um die z-Achse wird Azimut genannt (englisch: yaw), die Drehung um die y-Achse ist die Steigung (englisch: pitch) und die Drehung um die x-Achse wird als Neigung bezeichnet (englisch: roll). Der Translationsvektor t der Kamera enthält die geografischen Koordinaten X_t , Y_t und Z_t der Kamera.

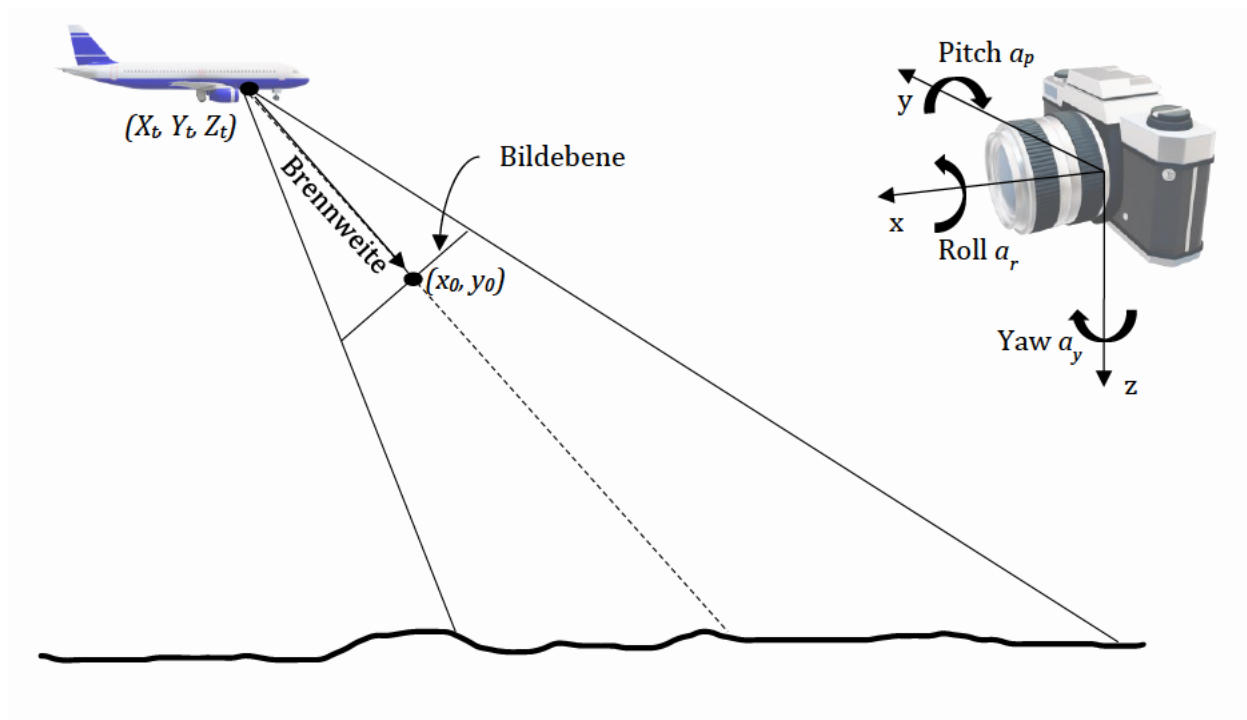


Abbildung 4: Parameter der inneren und äußeren Orientierung eines Luftbildes

Die Parameter der inneren und äußeren Orientierung sind in Abbildung 4 dargestellt. Sie können in der Kameragleichung zusammengefasst werden [22]:

$$\lambda x_c = K * R_{rot}^T * [I | -t] * X_c \quad [2.4]$$

Mit den Bildkoordinaten

$$x_c = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad [2.5]$$

Und den geografischen Koordinaten

$$X_c = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad [2.6]$$

Sowie dem Skalierungsfaktor λ . Mit Hilfe von Gleichung [2.4] kann ein dreidimensionales Objekt auf eine zweidimensionale Fläche projiziert werden. Wenn also alle Parameter der inneren und äußeren Orientierung für ein Luftbild bekannt sind, kann mit der Kameragleichung die Position eines Punktes mit den geografischen Koordinaten X , Y und Z auf dem Bild in Form der Bildkoordinaten x und y bestimmt werden. Hierzu kann Gleichung [2.4] vereinfacht werden zu:

$$\lambda x_c = T * X_c \quad [2.7]$$

Mit der Transformationsmatrix T

$$T = K * R_{rot}^T * [I | -t] = \begin{bmatrix} T_1 & T_2 & T_3 & T_4 \\ T_5 & T_6 & T_7 & T_8 \\ T_9 & T_{10} & T_{11} & T_{12} \end{bmatrix} \quad [2.8]$$

In der Transformationsmatrix sind alle Informationen über die Orientierung des Bildes zusammengefasst. Mit ihr können die Bildkoordinaten x und y folgendermaßen berechnet werden [16]:

$$x = \frac{T_1 X + T_2 Y + T_3 Z + T_4}{T_9 X + T_{10} Y + T_{11} Z + T_{12}} \quad [2.9]$$

$$y = \frac{T_5 X + T_6 Y + T_7 Z + T_8}{T_9 X + T_{10} Y + T_{11} Z + T_{12}} \quad [2.10]$$

Eine Herleitung der Gleichungen [2.9] und [2.10] ist im Anhang angefügt.

2.3.2 Rektifikation

Für manche Anwendungen ist es notwendig, die in Schrägluftbildern auftretenden Verzerrungen von Flächen durch Rektifikation zu entzerren. Eine Möglichkeit hierfür bietet die projektive Transformation. Mit ihr können Punkte zwischen zwei Darstellungen derselben Ebene in unterschiedlichen Perspektiven transformiert werden [39]. Die Transformation erfolgt mit Hilfe

der projektiven Transformationsmatrix H , welche eine Sonderform der Transformationsmatrix T aus Gleichung [2.8] für zweidimensionale Fälle ist [14, 39]:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \quad [2.11]$$

Die Bildkoordinaten des Ausgangsbildes x und y können mit H umgerechnet werden zu den Bildkoordinaten des rektifizierten Bildes x_{rect} und y_{rect} [91]:

$$x_{rect} = \frac{h_1x + h_2y + h_3}{h_7x + h_8y + 1} \quad [2.12]$$

$$y_{rect} = \frac{h_4x + h_5y + h_6}{h_7x + h_8y + 1} \quad [2.13]$$

Beispiele für ein rektifiziertes und ein nicht-rektifiziertes Bild sind in Abbildung 13 dargestellt. Die georeferenzierten und rektifizierten Luftbilder können genutzt werden, um mit den in Kapitel 2.1.2 beschriebenen Methoden die Fensterfläche der abgebildeten Gebäude zu bestimmen. Hierzu ist der Einsatz von Machine Learning-Algorithmen notwendig, auf die im Folgenden Kapitel näher eingegangen wird.

2.4 Machine Learning-Methoden zur Objekterkennung

In diesem Kapitel werden die Grundlagen des Machine Learning im Allgemeinen und anhand zweier Machine Learning-Methoden im Besonderen vorgestellt. Außerdem werden die Anforderungen an Datensätze, die zum Training von Machine Learning-Algorithmen verwendet werden, beschrieben.

2.4.1 Künstliche Intelligenz, Machine Learning und Deep Learning

Machine Learning (ML) ist ein Teilbereich der künstlichen Intelligenz (KI). Die ersten KI-Systeme konnten für Menschen anspruchsvolle Probleme lösen, solange sich diese mathematisch genau beschreiben ließen. Ein Beispiel hierfür sind Schach-Computer. Je mehr die Lösung eines Problems jedoch intuitives Handeln verlangte, desto aufwendiger musste das entsprechende KI-System gestaltet werden. Das Ergebnis dieser Bemühungen sind Machine Learning-Systeme. Es handelt sich hierbei um KI-Systeme, die nicht auf Wissen angewiesen sind, das ihnen durch den Menschen vorgegeben wird, sondern dieses Wissen aus Daten ableiten (oder „lernen“) können. Sie nutzen hierzu *Features*, die in den Daten enthalten sind. Klassische ML-Systeme werden zum Beispiel zum Filtern von Spam-E-Mails genutzt. Es gibt jedoch Problemstellungen, deren Komplexität sich nicht mit einfachen ML-Systemen beschreiben lässt. Beispiele hierfür sind das

Erkennen und Deuten von Sprache oder die Objekterkennung in einem Bild. In diesen Fällen kommt *Deep Learning* zum Einsatz, welches wiederum ein Teilbereich des Machine Learning ist. Mit Hilfe von Deep Learning löst der Computer ein komplexes Problem, indem er viele einfache Probleme hintereinander löst. Grafisch dargestellt bildet jedes dieser einfachen Probleme eine Schicht, welche wiederum zusammen mit allen anderen Schichten ein Netzwerk bildet, das in die Tiefe geht, weshalb das Konzept „tiefes Lernen“ (Deep Learning) genannt wird. Für die Objekterkennung in einem Bild bedeutet das, dass das Bild in jeder Schicht des Netzwerkes auf andere Merkmale untersucht wird, die für sich allein gesehen einfach zu ermitteln sind, zusammengenommen jedoch eine komplexe Beschreibung des Bildes ermöglichen. Die erste Schicht wird hierbei als *sichtbare Schicht (visible layer)* bezeichnet, weil sie die sichtbaren Informationen des Bildes in Form der Pixel enthält. Alle drauf folgenden Schichten werden *versteckte Schichten (hidden layers)* genannt, weil sie aus der jeweils vorherigen Schicht immer abstraktere Informationen ermitteln, die in den Eingangsdaten nicht enthalten waren. In der ersten versteckten Schicht können zum Beispiel mit Hilfe der Pixelinformationen Kanten im Bild ermittelt werden. Diese Kanten können in der zweiten versteckten Schicht zu Ecken und Umrissen zusammengeführt werden, welche wiederum in der dritten versteckten Schicht zu Objekten zugeordnet werden. Im Output werden schließlich die Informationen aus den versteckten Schichten zusammengefasst und gewichtet. Ein Deep Learning-Netzwerk, das auf diese Weise aufgebaut ist, wird *Neurales Netz (Neural Network)* genannt, da die Schichten in ihrer Funktionsweise an Neuronen im biologischen Gehirn angelehnt sind: In ein Neuron eingehende Informationen werden gewichtet, summiert und mittels einer Aktivierungsfunktion verstärkt. Der Aufbau eines Deep Learning Netzwerkes zur Objekterkennung ist in Abbildung 5 beispielhaft dargestellt, die Kreise repräsentieren hier die einzelnen Neuronen.

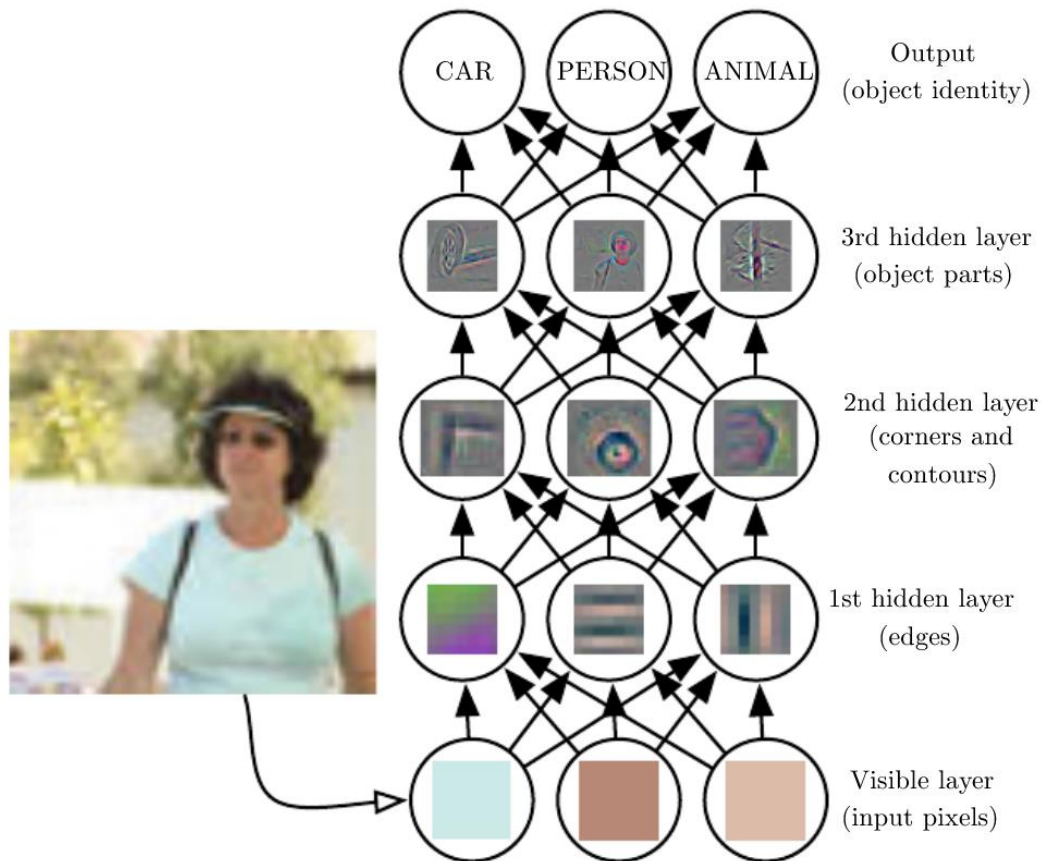


Abbildung 5: Darstellung eines Deep Learning Netzwerkes zur Objekterkennung in einem Bild mit sichtbarer Schicht (visible layer), versteckten Schichten (hidden layers) und dem Output. Übernommen aus [31].

Bei dem hier beschriebenen ML-Algorithmus handelt es sich um eine Klassifizierung, welche zum *überwachten Lernen (supervised Learning)* gezählt wird. Hierbei wird dem Algorithmus ein Datensatz zur Verfügung gestellt, der aus einzelnen Daten besteht, welche wiederum einzelne Features enthalten. Zu jedem dieser Features ist ein Label zugeordnet, welches im Fall der Klassifizierung eine Klasse (z.B. „CAR“, siehe Abbildung 5) enthält. Der Algorithmus wird darauf trainiert, möglichst korrekt jedem Feature ein Label zuzuordnen, indem er das Netzwerk mehrmals durchläuft und jedes Mal die Gewichtungen der Features im Netzwerk anpasst. Jede Iteration durch das neuronale Netzwerk wird dabei als *Epoche (epoch)* bezeichnet. Das Ziel ist es, eine Kostenfunktion zu minimieren, die als *Loss Function* bezeichnet wird und deren Ergebnis, der *Loss*, eine Messgröße für die Leistung des Netzwerkes ist. Der Loss wird sowohl während des Trainings des Modells als auch während der Validierung berechnet. Bei letzterer durchläuft ein Validierungsdatensatz das Modell, ohne dass die Gewichtungen angepasst werden. Der Vergleich von Trainings- und Validierungs-Loss kann Aufschluss darüber geben, wie robust das Modell trainiert wurde. Ein Beispiel für einen optimalen Verlauf von Trainings- und Validierungs-Loss, bei dem beide Kurven konvergieren, ist in Abbildung 6 dargestellt. Weicht eine der Kurven von dem in Abbildung 6 dargestellten Verlauf ab, kann dies je nach Verlauf darauf hindeuten, dass das

Modell entweder zu sehr auf den Trainingsdatensatz spezialisiert ist (*Overfitting*) oder zu sehr verallgemeinert (*Underfitting*). Für beide Fälle ist ebenfalls ein beispielhafter Verlauf in Abbildung 6 dargestellt. Ob das Training erfolgreich war, wird abschließend ermittelt, indem der trainierte Algorithmus einem Datensatz Label zuordnen muss, die ihm zuvor nicht bekannt waren. [31]

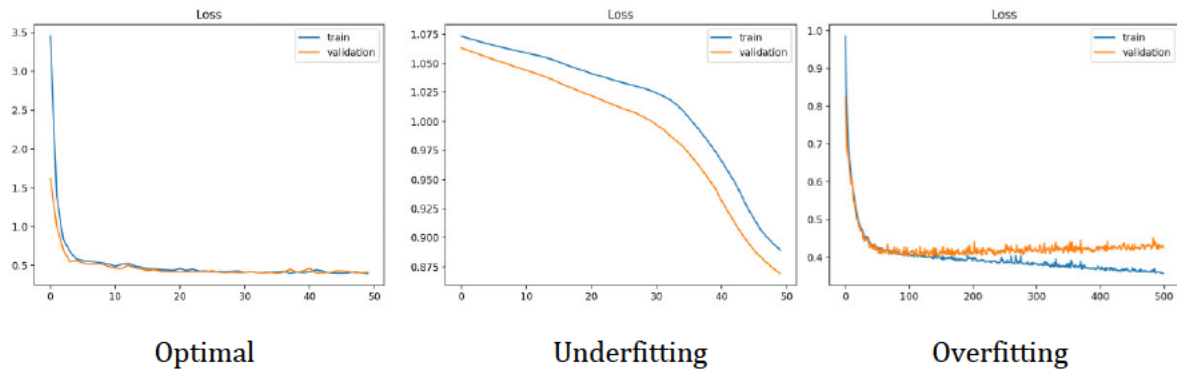


Abbildung 6: Verlauf von Trainings- und Validierungs-Loss in Abhängigkeit der Epochen. Das linke Bild stellt einen optimalen Verlauf dar, bei dem die beiden Kurven konvergieren. Im mittleren Bild konnte das Modell am Trainings-Datensatz nicht ausreichend gut trainiert werden, es kommt zum Underfitting. Im rechten Bild wurde das Modell im Training zu sehr auf den Trainingsdatensatz spezialisiert, was zu Overfitting führt. Übernommen aus [12].

2.4.2 Das Objekterkennungsmodell Faster R-CNN

Ein Anwendungsfeld der Objekterkennung mittels Deep Learning ist das Erkennen von Fassadenelementen, wie zum Beispiel Fenstern oder Türen. Zu den großen Herausforderungen hierbei zählen Verschattungen und Reflektionen, aber auch die große Anzahl an Architekturstilen, die ein erfolgreiches Training des ML-Algorithmus erschweren [92]. Um trotzdem eine möglichst genaue Bestimmung der Fassadenelemente mithilfe von Machine Learning zu erzielen, muss das bereits beschriebene neuronale Netzwerk um weitere Elemente ergänzt werden. Dies wurde in der Vergangenheit in Form unterschiedlicher Algorithmen umgesetzt, die alle auf die Objekterkennung spezialisiert sind [30, 63, 75]. Hierbei wird zwischen den schnelleren, einstufigen Ansätzen, wie YOLO, RetinaNet oder SSD, und langsameren, aber dafür präziseren zweistufigen Ansätzen wie R-CNN, Fast R-CNN, Faster R-CNN oder Mask R-CNN unterschieden [65]. Zur Erkennung von Fassadenelementen werden häufig die letzteren verwendet [63, 65, 70, 83, 92]. Während R-CNN, Fast R-CNN und Faster R-CNN Objekte innerhalb eines rechteckigen Rahmens identifizieren (objektbezogene Segmentierung), ist Mask R-CNN auch in der Lage, die Umrisse eines Objektes zu erkennen und wiederzugeben, indem es jedem Pixel eine Klasse (z.B. „Fenster“ oder „Tür“) zuordnet (semantische Segmentierung). Im Folgenden soll von den genannten Methoden jedoch nur *Faster R-CNN* vorgestellt werden, da diese Methode gut für die Erkennung von Fenstern in Fassaden geeignet ist [65, 92]. Faster R-CNN steht für „Faster Region-

based Convolutional Network Method“, also ein schnelleres, auf Regionen basierendes Convolutional Neural Network [75]. In einem *Convolutional Neural Network (CNN)* wird in mindestens einer Schicht die lineare mathematische Operation der Faltung (oder Konvolution) angewendet. Eine solche Schicht, im Folgenden *Convolutional Layer* genannt, filtert die Eingangsdaten und erzeugt so eine *Feature Map* [31]. Das CNN in Faster R-CNN wird mit dem „R“ um ein Netzwerk erweitert, das Regionen vorschlägt, in denen die gesuchten Objekte wahrscheinlich auftreten. Dieses Netzwerk wird *Region Proposal Network (RPN)* genannt. Ein Faster R-CNN ist ein zweistufiges Netzwerk, da es in einer ersten Stufe mit Hilfe des RPN Regionen vorschlägt, in denen in einer zweiten Stufe Objekte identifiziert werden [65]. Der Aufbau des Faster R-CNN ist schematisch in Abbildung 7 dargestellt und soll im Folgenden kurz beschrieben werden.

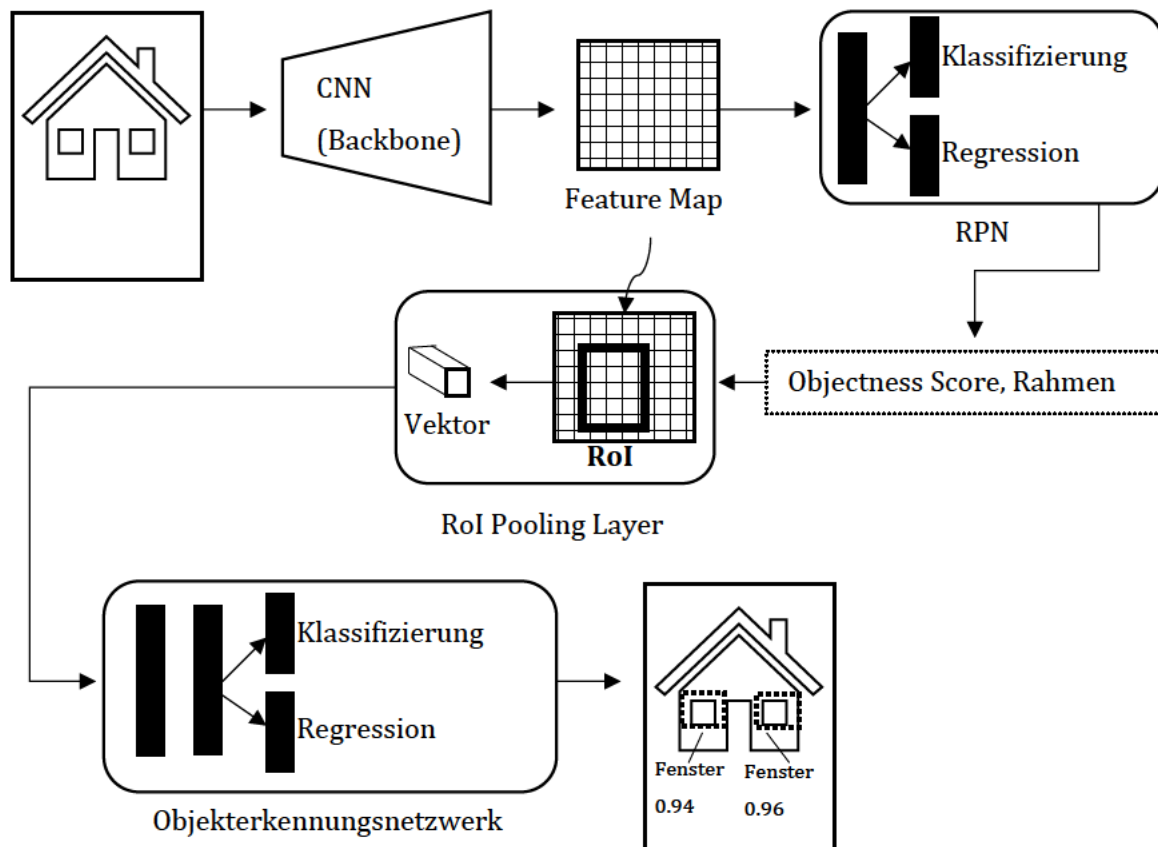


Abbildung 7: Schematische Ansicht des Faster R-CNN Algorithmus. Die schwarzen Balken stellen Schichten eines Neuronalen Netzes dar. In dem hier dargestellten Fall identifiziert Faster R-CNN im Eingangsbild zwei Objekte, deren Positionen durch die gestrichelten Rahmen im unteren rechten Bild markiert sind. Den Objekten wird mit einer Sicherheit (Score) von 0,94, beziehungsweise 0,96, die Klasse „Fenster“ zugeordnet.

Ein Bild, das in das Faster R-CNN eingespeist wird, passiert zunächst ein CNN. Hierbei handelt es sich um den *Backbone*, also das „Rückgrat“ des Faster R-CNN Algorithmus. Der Backbone kann bereits an einem Datensatz auf Objekterkennung (z.B. ImageNet [17] oder COCO [62]) trainiert

worden sein (Transfer Learning), ist jedoch nicht auf eine bestimmte Art von Objekten spezialisiert. Beispiele für Backbones sind Resnet101, Resnet50 oder VGGNet [63, 65, 92]. Die Leistung des Backbone kann gesteigert werden, indem ein *Feature Pyramid Network (FPN)* verwendet wird [70]. Der Backbone wandelt das Bild in eine Feature Map um, die eine geringere Höhe und Breite als das Eingangsbild aufweist, dafür aber eine größere Tiefe hat. Die Feature Map enthält die Features, mit deren Hilfe später die Objekte erkannt werden sollen. Im RPN wird die Feature Map von einem gleitenden Fenster abgetastet. Im Zentrum des Fensters sind die *Anker (Anchors)* fixiert. Hierbei handelt es sich um Rechtecke, deren Größe über Maßstab und Seitenverhältnis definiert ist. Die Anker bilden den Rahmen, in dem ein Objekt lokalisiert wird. Die Anzahl der Anker kann frei gewählt werden, indem die Maßstäbe und Seitenverhältnisse bestimmt werden. Ren et al., die die Methode Faster R-CNN in [75] vorgestellt haben, wählten jeweils 3 Maßstäbe und Seitenverhältnisse, sodass sie insgesamt 9 Anker erhielten. An jedem Punkt, den das gleitende Fenster abtastet, wird es zusammen mit den Ankern in einer Convolutional Layer zu einem Vektor transformiert, bevor es in zwei weitere Convolutional Layers übergeben wird. Die eine dieser Schichten führt eine Klassifizierung durch, an deren Ende jeder Anker entweder als „Vordergrund“ oder als „Hintergrund“ markiert wurde, je nachdem, ob im Anker ein Objekt vermutet wird. Die andere Schicht passt mit Hilfe einer Regression die Umrisse der Anker, die dem Vordergrund zugeordnet wurde, an. Hierbei bleibt die rechteckige Form der Anker erhalten. In Abbildung 8 ist ein RPN schematisch dargestellt. [75]

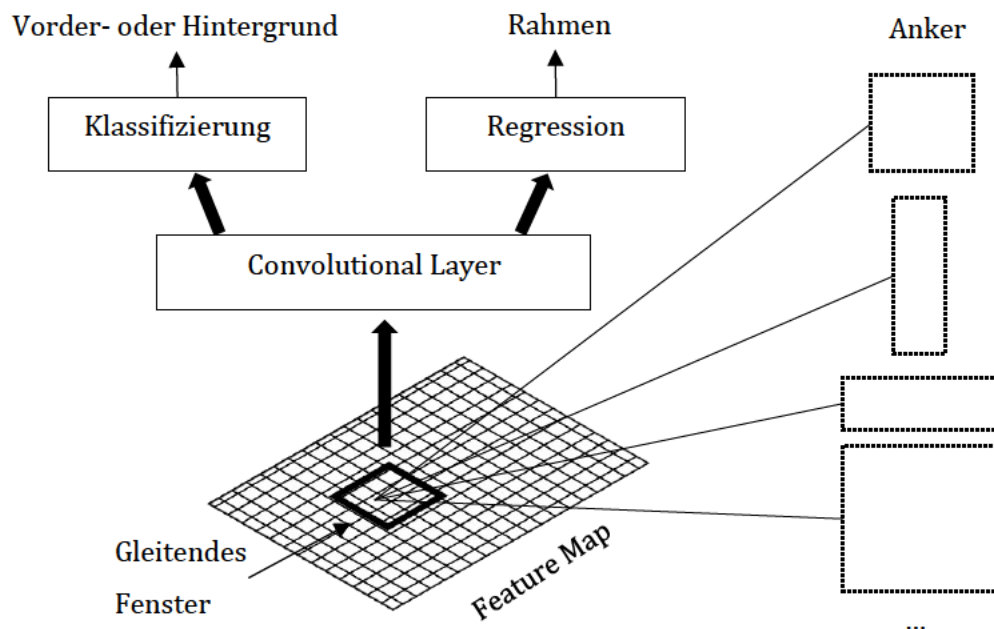


Abbildung 8: Schematische Darstellung des Region Proposal Network. Schematisch nachgezeichnet aus [75].

Mit Hilfe des RPN lassen sich demnach rechteckige Regionen bestimmen, in denen wahrscheinlich ein Objekt vorhanden ist, wobei die Wahrscheinlichkeit, mit der ein Objekt dort auftritt, als *Objectness Score* bezeichnet wird. Wenn sich mehrere dieser Regionen großflächig überschneiden, werden sie in der letzten Stufe des RPN mit einer *Non-Maximum Suppression (NMS)* gefiltert. Hierbei werden nur die Regionen mit dem höchsten Objectness Score erhalten, alle anderen werden verworfen [70]. Jede der verbleibenden Regionen wird *Region of Interest (RoI)* genannt. Da alle RoIs unterschiedliche Größen und Seitenverhältnisse aufweisen, können sie nicht direkt zur Identifikation von Objekten mit einem CNN genutzt werden. Um sie zu vereinheitlichen, werden sie deshalb in der *RoI Pooling Layer* bearbeitet. Dort werden ihre Informationen mit der Feature Map, die der Backbone ausgegeben hatte, vereint. Die Features der entsprechenden Regionen werden jeweils zu einem Vektor mit einer vordefinierten Größe zusammengefasst [30]. So wird aus einer Anzahl unterschiedlich großer Rechtecke in Form der RoIs dieselbe Anzahl gleich großer Vektoren. Diese Vektoren, die alle wesentlichen Features der jeweiligen RoI enthalten, können nun in das Objekterkennungsnetzwerk gegeben werden. Dort werden die Vektoren in zwei Schichten transformiert, bevor sie in zwei weitere Schichten übergeben werden, die erneut eine Klassifizierung und eine Regression durchführen. Bei der Klassifizierung wird die Klasse des Objektes bestimmt, wobei nun nicht mehr zwischen Vorder- und Hintergrund unterscheiden wird, sondern zwischen einer vordefinierten Auswahl an Klassen sowie dem Hintergrund. Die Regression passt den Rahmen der Objekte an, die nicht als Hintergrund identifiziert wurden. Das Ergebnis des Faster R-CNN Algorithmus sind die Positionen der Rahmen sowie die zugeordneten Klassen und die berechnete Sicherheit für alle Objekte, die im Eingangsbild erkannt werden konnten. Diese Sicherheit wird als *Score* bezeichnet und bezieht sich, im Gegensatz zum Objectness Score, nicht auf die Unterscheidung von Objekt und Hintergrund, sondern auf die Unterscheidung einzelner Klassen. [75]

Ein Vorteil des Faster R-CNN ist, dass sich das Objekterkennungsnetzwerk Convolutional Layers mit dem CNN teilen kann [75]. Dies ist vor allem während des Trainings des Modells von Nutzen, da die Trainingszeit so minimiert werden kann.

2.4.3 Bewertungsgrößen für Klassifizierungsaufgaben

Der Erfolg des Trainings eines ML-Modells zur Klassifizierung von Objekten lässt sich unter anderem über die *Intersection over Union (IoU)* bestimmen. Diese beschreibt die Übereinstimmung zwischen einem vom Modell erkannten Rahmen und dem vorgegebenen Rahmen (*Ground-Truth*) über das Verhältnis der Schnittmenge zur Vereinigungsmenge (siehe Abbildung 9) [92].

$$IoU = \frac{\text{Schnittmenge}}{\text{Vereinigungsmenge}} \quad [2.14]$$

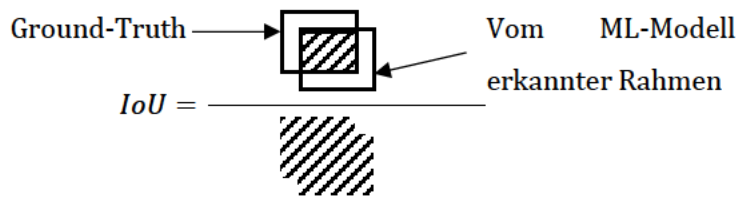


Abbildung 9: Intersection over Union. Schematisch nachgezeichnet aus [92].

Weitere Messwerte zur Beurteilung des Trainingserfolges sind der *Recall* R und die *Präzision* P [1]:

$$P = \frac{TP}{TP + FP} \quad [2.15]$$

$$R = \frac{TP}{TP + FN} \quad [2.16]$$

In den Gleichungen [2.15] und [2.16] steht TP für die korrekt einer Klasse zugeordneten Elemente (True Positive), FP für die fälschlicherweise einer Klasse zugeordneten Elemente (False Positive) und FN für die fälschlicherweise nicht der Klasse zugeordneten Elemente (False Negative). Die Präzision beschreibt also, wie viele der einer Klasse zugeordneten Elemente tatsächlich dieser Klasse angehören, der Recall hingegen, wie viele der insgesamt bekannten Elemente einer Klasse (zum Beispiel in einem Bild) dieser Klasse zugeordnet wurden. Dies wird im Beispiel in Abbildung 10 deutlich. Dort sind in einem Bild 10 Objekte vorhanden, die entweder der Klasse 1 oder der Klasse 2 zugeordnet werden können. Die gestrichelte Linie repräsentiert hierbei den Grenzwert, über den die Zuordnung der Klassen definiert ist. Dies kann zum Beispiel ein IoU -Grenzwert sein, der überschritten werden muss, damit ein Element der entsprechenden Klasse zugeordnet werden kann. Die Präzision und der Recall können für jede der Klassen bestimmt werden. Im Beispiel erfolgt die Bestimmung nur für Klasse 1.

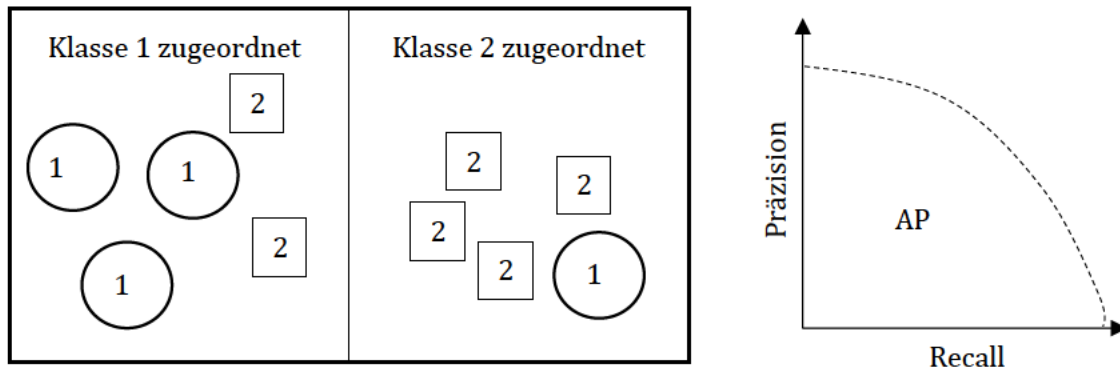


Abbildung 10: Beispielbild für die Bestimmung von Präzision und Recall (linke Seite) und schematische Darstellung einer PR-Kurve (rechte Seite). Die Fläche unter der Kurve entspricht der durchschnittlichen Präzision AP.

Der Klasse 1 wurden 5 Elemente zugeordnet, von denen 3 tatsächlich dieser Klasse angehören. 2 Elemente hingegen gehören eigentlich zur Klasse 2. Demnach ist in Gleichung [2.15] $TP = 3$, $FP = 2$ und somit die Präzision $P = \frac{3}{5}$. Damit kann die Aussage getroffen werden, dass $\frac{3}{5}$ der Elemente, die das Klassifizierungs-Modell der Klasse 1 zuordnet, korrekt zugeordnet wurden. Eine Beurteilung des Modells kann allein auf Basis der Präzision jedoch nicht erfolgen, da dieser Wert nicht berücksichtigt, wie viele Elemente der Klasse 1 anderen Klassen zugeordnet wurden. So wäre es zum Beispiel möglich, dass alle Elemente der Klasse 1 korrekt dieser Klasse zugeordnet wurden, allerdings auch alle Elemente der Klasse 2. Die Präzision wäre in diesem Fall genauso groß, wie wenn nur zwei Elemente der Klasse 1 und 3 Elemente der Klasse 2 zur Klasse 1 zugeordnet worden wären. Für die vollständige Bewertung des Modells ist daher zusätzlich der Recall notwendig. Im Beispiel in Abbildung 10 wurden 3 der 4 Elemente, die der Klasse 1 angehören, korrekt dieser Klasse zugeordnet. Damit gilt in Gleichung [2.16], dass $TP = 3$ und $FN = 1$ ist. Demnach ist der Recall $R = \frac{3}{4}$. Zur Bewertung des Modells wird die Präzision über dem Recall in der PR-Kurve aufgetragen (siehe Abbildung 10, rechte Seite). Hierzu werden die Elemente anhand ihrer Bewertungsgröße (z.B. Score oder IoU-Grenzwert) zunächst in absteigender Reihenfolge sortiert. Anschließend werden TP , FP und FN bestimmt, indem die sortierten Elemente nacheinander betrachtet werden. Für jedes Element werden P und R berechnet, welche dann in der PR-Kurve dargestellt werden können. Häufig führt hierbei eine hohe Präzision zu einem niedrigen Recall und umgekehrt [31], so auch in der schematischen Darstellung in Abbildung 10. Die Fläche unter der PR-Kurve entspricht der *durchschnittlichen Präzision (Average Precision) AP* [1]:

$$AP = \int_0^1 P(R) dR \quad [2.17]$$

Der Wert AP kann die Leistungsfähigkeit des Klassifizierungs-Modells insgesamt beschreiben, indem er P über R integriert und so den Mittelwert der Präzision für unterschiedliche Recalls bestimmt. Die AP kann in Abhängigkeit einer IoU angegeben werden [83]. Wenn der zulässige Grenzwert der IoU , bei der ein Element als korrekt erkannt gilt, zum Beispiel auf 0,5 festgelegt wird, wird die entsprechende AP als AP_{50} bezeichnet. In den Kriterien zur Auswertung des Microsoft Common Objects in Context Datensatzes (COCO) wird festgelegt, dass der IoU -Grenzwert in zehn Schritten von 0,5 auf 0,95 erhöht werden soll. Die AP bildet dann den Mittelwert aus AP_{50} bis AP_{95} [83]. So können Fehler vermieden werden, die durch Betrachtung einer AP bei nur einem IoU -Grenzwert auftreten könnten [93]. Um die durchschnittliche Präzision über alle Klassen zu beschreiben, wird die *mittlere durchschnittliche Präzision (mean Average Precision) mAP* verwendet, die als Standard für die Bewertung von Objekterkennungs-Aufgaben gilt [1, 75]:

$$mAP = \frac{1}{N_{class}} \sum_{i=1}^{N_{class}} AP_i \quad [2.18]$$

Mit der Anzahl der Klassen N_{class} . Eine weitere Bewertungsgröße für Objekterkennungs-Aufgaben ist $F1$, das das harmonische Mittel zwischen P und R beschreibt [31]:

$$F1 = \frac{2 * P * R}{P + R} \quad [2.19]$$

2.4.4 Clusteranalyse

Im Gegensatz zu den bisher beschriebenen Klassifizierungen handelt es sich bei der Clusteranalyse um eine *unüberwachte Machine Learning Methode (unsupervised Learning)*. Das bedeutet, dass das ML-Modell in einem Datensatz, für den keine Label existieren, nach unentdeckten Mustern sucht. Dies geschieht bei der Clusteranalyse, indem die Datenpunkte Clustern zugeordnet werden. Hierbei werden die Informationen verwendet, die dem Cluster-Modell zur Verfügung gestellt wurden. Ein typischer Algorithmus zur Clusteranalyse quantitativer Daten ist der *k-means-Algorithmus*. Dieser wählt eine Anzahl von $K_{cluster}$ zufälligen Datenpunkten eines Datensatzes aus, die er zunächst als Cluster-Mittelpunkte annimmt. Die Zuordnung der Datenpunkte zu einem Cluster erfolgt über die Berechnung der euklidischen Distanz d zwischen dem Datenpunkt und dem Cluster-Mittelpunkt im zweidimensionalen Raum mit

$$d(p_i, p_j) = \sqrt{\sum_{n=1}^2 (p_{i,n} - p_{j,n})^2} \quad [2.20]$$

Wobei p_i und p_j die beiden Punkte darstellen, zwischen denen d berechnet wird. Die Datenpunkte werden dem Cluster zugeordnet, zu dessen Mittelpunkt d am geringsten ist. Für die auf diese Weise gebildeten Cluster wird der Mittelwert aus allen dem Cluster zugehörigen Datenpunkten berechnet, welcher fortan als Mittelpunkt des Clusters gilt. Dann wird erneut d zwischen den Datenpunkten und den Cluster-Mittelpunkten berechnet und die Cluster angepasst. Diese Schritte werden so lange wiederholt, bis die Abweichung der Datenpunkte zum Mittelpunkt ihres Clusters ein Minimum erreicht. Ein Beispiel für die Ergebnisse einer Clusteranalyse mit variierendem $K_{Cluster}$ ist in Abbildung 11 gegeben.

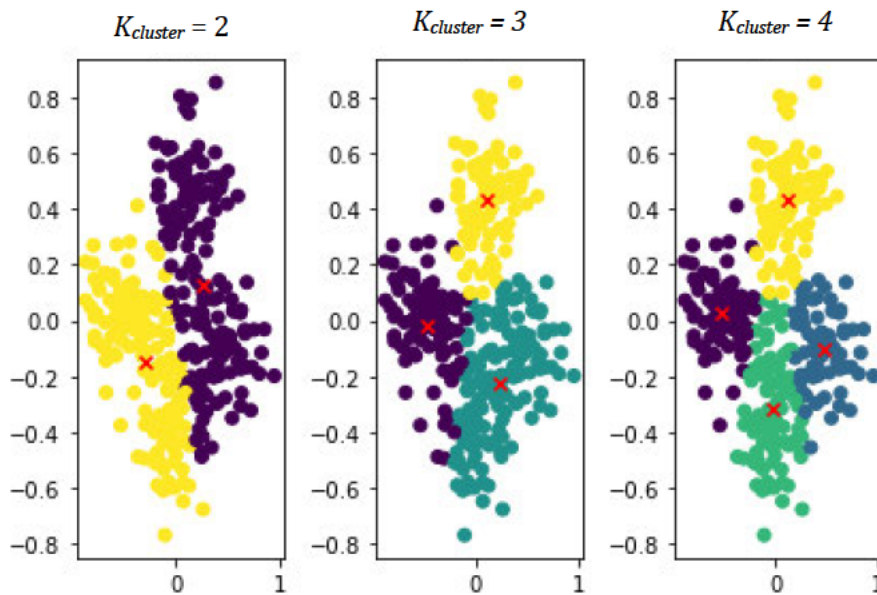


Abbildung 11: Beispiel für die Clusteranalyse eines Datensatzes mit $K_{Cluster} = 2$, $K_{Cluster} = 3$ und $K_{Cluster} = 4$. Die roten Kreuze markieren die Cluster-Mittelpunkte.

Der Wert $K_{Cluster}$, welcher die Anzahl der Cluster definiert, in die die Datenpunkte sortiert werden, muss dem Algorithmus vorgegeben werden. Hierzu können das *Within-Cluster-Point-Scatter* $W(C)$ und das *Between-Cluster-Point-Scatter* $B(C)$ verwendet werden, welche für

$$0 \leq K_{Cluster} \leq K_{Cluster,max}$$

berechnet werden. $W(C)$ beschreibt, wie eng die Datenpunkte eines Clusters beisammen liegen. Je kleiner $W(C)$ ist, desto dichter liegen die Punkte zusammen und desto dichter sind die Cluster gepackt. $W(C)$ wird berechnet mit

$$W(C) = \frac{1}{2} \sum_{K_{Cluster}=1}^{K_{Cluster,max}} \sum_{C(i)=K_{Cluster}} \sum_{C(j)=K_{Cluster}} d(p_i, p_j) \quad [2.21]$$

Wobei der Encoder $C(i)$ den i -ten Datenpunkt dem $K_{Cluster}$ -ten Cluster zuordnet. Im Gegensatz dazu beschreibt $B(C)$, wie weit die Datenpunkte unterschiedlicher Cluster auseinander liegen. Je größer

$B(C)$ ist, desto weiter liegen die Punkte auseinander und desto klarer sind die einzelnen Cluster voneinander getrennt. $B(C)$ wird folgendermaßen berechnet:

$$B(C) = \frac{1}{2} \sum_{K_{Cluster}=1}^{K_{Cluster,max}} \sum_{C(i)=K_{Cluster}} \sum_{C(j) \neq K_{Cluster}} d(p_i, p_j) \quad [2.22]$$

Beispiele für $W(C)$ und $B(C)$ sind in Abbildung 12 dargestellt.

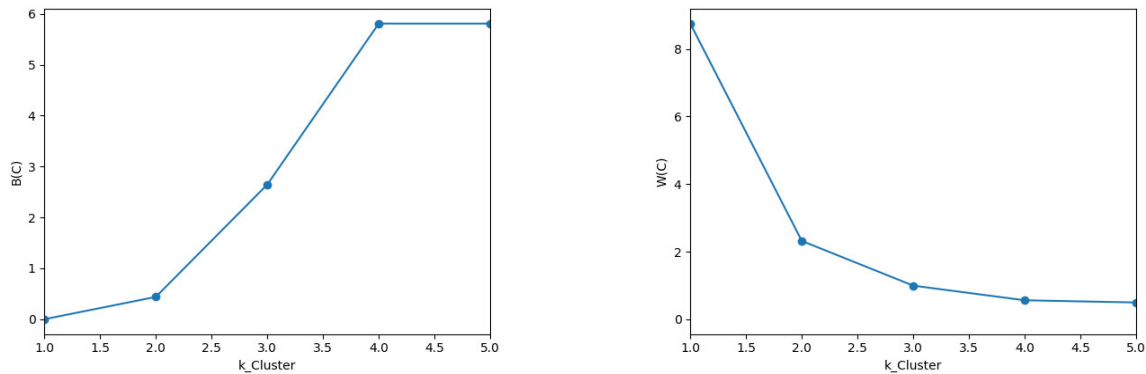


Abbildung 12: Beispiele für das Between-Cluster-Point-Scatter $B(C)$ (links) und das Within-Cluster-Point-Scatter $W(C)$ (rechts) in Abhängigkeit der Anzahl der Cluster $K_{Cluster}$.

Für das optimale $K_{Cluster}$ gilt, dass der Verlauf von $B(C)$ oder $W(C)$ für dieses $K_{Cluster}$ einen Knick aufweist. In Abbildung 12 ist so ein Knick im $B(C)$ -Diagramm bei $K_{Cluster} = 4$ zu sehen. Da auch das $W(C)$ -Diagramm ab $K_{Cluster} = 4$ keine großen Veränderungen mehr aufweist, kann davon ausgegangen werden, dass sich der entsprechende Datensatz am besten in 4 Cluster aufteilen lässt. [33]

2.4.5 Datensätze

Das Training und die Beurteilung eines ML-Modells zur Objekterkennung (wie z.B. Faster R-CNN) geschieht mit Hilfe jeweils eines Trainings-, Validierungs- und Testdatensatzes. Im Falle eines Objekterkennungs-Modells sind diese Datensätze Teil eines größeren Datensatzes von Bildern, die um Informationen erweitert sind, die für das Training des Modells relevant sind. Dieser Datensatz wird zunächst in einen Test- und einen Trainingsdatensatz aufgeteilt. Der Validierungsdatensatz ist wiederum ein Teil des Trainingsdatensatzes. Er macht üblicherweise 20 % des Trainingsdatensatzes aus [31]. Größe und Qualität des Datensatzes können den Erfolg des Trainings maßgeblich beeinflussen. So gaben Goodfellow et al. 2016 an, dass ungefähr 5000 gelabelte Daten benötigt werden, um einen ML-Algorithmus zu trainieren [31]. Mit einem zweistufigen Objekterkennungsalgorithmus wie Faster R-CNN oder Mask R-CNN ist dies auch mit einem deutlich kleineren Datensatz möglich. Dies wird bei Betrachtung einiger Bild-Datensätze im Anhang D in Tabelle 13 deutlich, die in den in Kapitel 2.1.2 vorgestellten Veröffentlichungen

verwendet wurden. Von den aufgelisteten Datensätzen weist nur der von Lippoldt [63] verwendete Datensatz CityGML Berlin eine Größe von mehr als 5000 Bildern auf. Alle anderen Datensätze liegen deutlich darunter. Sun et al. kombinierten die Datensätze 4 bis 9 mit eigenen Bildern, um ihren Mask R-CNN Algorithmus zu trainieren. Sie erhielten so einen Datensatz mit 12.000 Bildern. Ma und Ma [65] verwendeten für ihren Algorithmus, der auf Faster R-CNN beruht, das Street Scene Window Detection Dataset sowie den ECP-Datensatz, welche zusammen auf eine Anzahl von 1404 Bildern kommen. Wang [92] wendete das Faster R-CNN Netzwerk an und trainierte es mit dem 606 Bilder umfassenden CMP Datensatz. Nordmark und Ayenew [70] trainierten ihr Mask R-CNN Netzwerk mit dem gleichnamigen Datensatz (Nr. 3 in Tabelle 13), der aus 100 Bildern besteht.

Eine Vergrößerung des Datensatzes kann das Trainingsergebnis gegebenenfalls verbessern. Dies kann auch durch eine Verbesserung der Qualität der Informationen des Datensatzes geschehen. Die in einem Datensatz für die Objekterkennung an Fassaden enthaltenen Informationen können zum Beispiel die Umrisse und Bezeichnungen (Klassen) von Fassadenelementen wie Türen, Fenstern oder Balkonen sein. Nordmark und Ayenew [70] stellten bezüglich der Qualität solcher Informationen in Datensätzen zur Objekterkennung an Fassaden fest, dass viele der Datensätze ungenaue Definitionen enthalten. Dies zeigte sich zum Beispiel darin, dass Fenster nicht durch den Rahmen begrenzt werden, sondern durch Zierelemente, die den Rahmen umschließen (siehe Abbildung 13). Je nach Anwendungsfall kann dies das Trainingsergebnis verschlechtern. Um das Ergebnis des ML-Trainings zu verbessern, kann die Genauigkeit der Label angepasst oder der Datensatz vergrößert werden. Beides ist jedoch mit großem zeitlichen Aufwand verbunden, da die Erstellung oder Korrektur der Label für jedes Bild manuell erfolgen muss [65, 70]. Eine Alternative hierzu bietet die Augmentation eines Datensatzes. Hierbei werden aus den vorhandenen Daten neue, „künstliche“ Daten generiert, an denen der ML-Algorithmus trainiert wird [31]. Dies ist vor allem für Objekterkennung sehr effektiv, da die Eigenschaften von Bildern leicht simuliert werden können. Beispiele für Bild-Augmentationen sind Spiegeln, Drehen, Verzerren oder das Hinzufügen von Rauschen.

Bezüglich ihres Inhaltes können die Datensätze für die Objekterkennung in zwei Kategorien eingeteilt werden. Diese orientieren sich an der Unterteilung eines Bildes in Elemente (z.B. Fenster), welche als Bildsegmentierung bezeichnet wird. Es wird hierbei zwischen semantischer und objektbezogener Bildsegmentierung unterschieden (vergleiche Kapitel 2.4.2). Bei der semantischen Bildsegmentierung sind die Umrisse der Objekte in den Bildern des Datensatzes genau abgebildet. Jedem Pixel wird hierbei eine Klasse (z.B. „Fenster“) zugeordnet. Bei der objektbezogenen Bildsegmentierung hingegen werden die Objekte in den Bildern unabhängig von ihrer eigentlichen Form durch rechteckige Rahmen lokalisiert. Die Klasse des Objektes wird dabei

dem jeweiligen Rahmen zugeordnet. Beispiele für beide Arten von Bildsegmentierungen sind in Abbildung 13 dargestellt. Dort wird ebenfalls der Unterschied zwischen rektifizierten und nicht rektifizierten Bildern deutlich (siehe Kapitel 2.3.2). Das linke Bild stammt aus dem CMP-Datensatz [88] und wurde rektifiziert. Das rechte Bild hingegen stammt aus dem eTRIMS-Datensatz [38], dessen Bilder nicht rektifiziert wurden. Die Erkennung von rechteckigen Objekten (wie z.B. Fenstern) in Bildern mittels eines ML-Algorithmus ist für rektifizierte Bilder einfacher als für nicht rektifizierte [65].

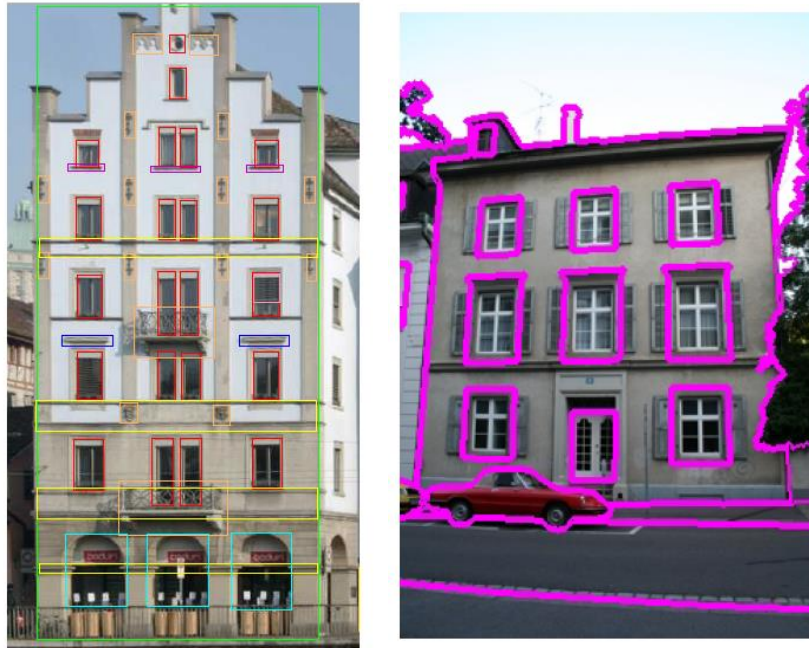


Abbildung 13: Objektbezogene Bildsegmentierung im CMP-Datensatz [88] (links) und semantische Bildsegmentierung im eTRIMS-Datensatz [38] (rechts).

3 Methodik

In Kombination mit den vorgestellten Methoden zur Aufbereitung von Luftbildern und Machine Learning-Methoden bilden die für Hamburg verfügbaren öffentlich zugänglichen Daten eine gute Basis, um das Verhältnis von Fenster- zu Wandflächen und Baualtersklassen von Gebäuden zu bestimmen. Die in diesem Kapitel beschriebene Methodik zur Bestimmung des Verhältnisses der Fenster- zur Wandfläche orientiert sich an den in Kapitel 2.1.2 vorgestellten Arbeiten, wobei eine Anpassung an die Hamburger Gebäude erfolgte. Die Methodik zur Bestimmung der Baualtersklasse erweitert die in Kapitel 2.1.3 beschriebenen Methoden um einen neuen Ansatz.

Die Anwendung der in diesem Kapitel beschriebenen Methoden erfolgte in Python. Die verwendeten Skripte sind in auf dem digitalen Datenträger dieser Arbeit beigefügt.

3.1 Bestimmung des Verhältnisses der Fenster- zur Wandfläche eines Gebäudes

Zur Bestimmung des Verhältnisses $R_{\text{Gebäude}}$ der Fenster- zur Wandfläche ist zunächst die Bestimmung der Fensterfläche notwendig. Die in Kapitel 2.1.2 vorgestellten Arbeiten zur Fenstererkennung in Gebäudefassaden weisen alle gute Ergebnisse auf. Nichtsdestotrotz ist keine der dort angewendeten Methoden auf Norddeutschland spezialisiert. Darüber hinaus wurden die Bilder, in denen die Gebäude dargestellt sind, in fast allen Arbeiten aus Straßenansichten generiert, die keine Auswertung der Gebäuderückseiten erlauben. Einzige Ausnahme bildet die Arbeit von Lippoldt [63]. Um eine möglichst genaue Bestimmung der Fensterflächen in Hamburg aus allen vier Himmelsrichtungen zu erzielen, wurde deshalb in dieser Arbeit keine Methodik aus Kapitel 2.1.2 in Gänze übernommen. Stattdessen wurde eine Methodik entwickelt, die sich an den vorgestellten Arbeiten orientiert und Elemente unterschiedlicher Ansätze vereint sowie neue hinzufügt. Die Methodik dieser Masterarbeit zur Ermittlung der Fensterfläche und des Verhältnisses der Fenster- zur Wandfläche ist in Abbildung 14 schematisch dargestellt.

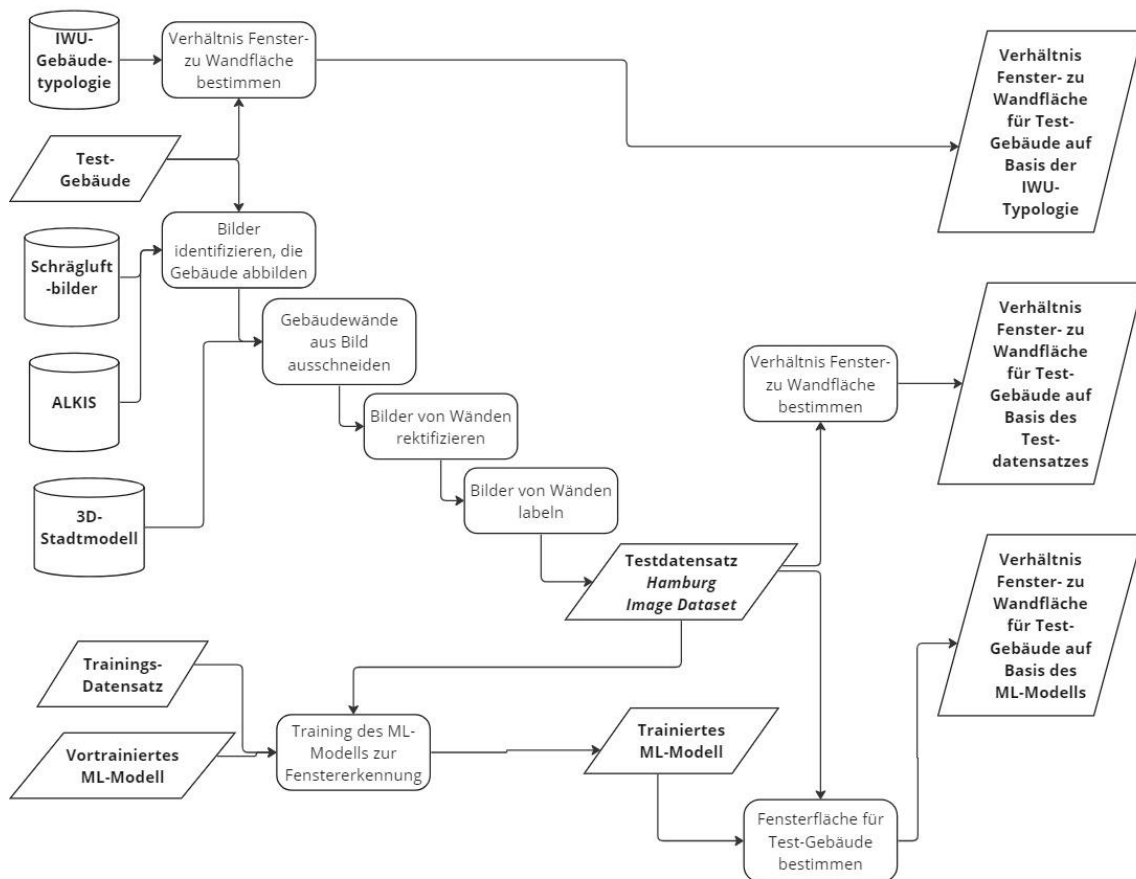


Abbildung 14: Darstellung der Methodik zur Bestimmung des Verhältnisses der Fenster- zur Wandfläche von Gebäuden als Flussdiagramm.

Die Basis bilden die in Kapitel 2.2 vorgestellten öffentlich zugänglichen Daten für Hamburg. Es wurden Test-Gebäude ausgewählt, für die $R_{\text{Gebäude}}$ über drei Wege berechnet und abschließend verglichen wurde. Die Berechnung erfolgte zum einen über die IWU-Gebäudetypologie. Darüber hinaus wurde ein Testdatensatz erstellt, der Bilder der Fassaden der Test-Gebäude enthielt. Das Verhältnis $R_{\text{Gebäude}}$ wurde zusätzlich aus den Labels dieses Datensatzes berechnet. Die Ergebnisse wurden als Grundwahrheit für den Vergleich verwendet. Als dritte Berechnungsgrundlage dienten die Vorhersagen eines ML-Modells, das für die Fenstererkennung trainiert und am Testdatensatz getestet wurde. Der Vergleich der drei Berechnungsmethoden soll Aufschluss darüber geben, wie gut die IWU-Gebäudetypologie und die Ergebnisse des ML-Modells die Grundwahrheit annähern können. Eine entscheidende Rolle spielt hierbei die Auswahl der Machine Learning-Methode.

3.1.1 Die Machine Learning Methode

Um eine Machine Learning Methode zu wählen, muss zunächst festgelegt werden, ob die Schrägluftbilder semantisch oder objektbezogen segmentiert werden sollen (siehe Kapitel 2.4.5). Für eine semantische Segmentierung spricht, dass Fensterflächen genauer abgebildet werden

können, da diese auf der Zuordnung jedes einzelnen Pixels zu einer Klasse basiert. Auch nicht-rechtwinklige Formen wie Dreiecke oder Bögen sowie verzerrte Fenster können mit dieser Methode genauer dargestellt werden. Von den in Kapitel 2.4.2 aufgelisteten zweistufigen Methoden ist nur der Mask R-CNN Algorithmus in der Lage, Bilder semantisch zu segmentieren. Die objektbezogene Segmentierung hingegen funktioniert nur für rektifizierte Bilder zufriedenstellend und kann mit R-CNN, Fast R-CNN oder Faster R-CNN durchgeführt werden. Das Ziel dieser Arbeit ist die Schaffung einer Grundlage für detaillierte Wärmebedarfsprognosen, mit deren Hilfe sich die Wärmebedarfe der einzelnen Gebäude im Hamburger Stadtgebiet, vor allem im Stadtteil Wilhelmsburg, vorhersagen lassen. Der Fokus liegt hierbei auf den Wohngebäuden, von denen im Stadtteil Wilhelmsburg im Jahr 2020 86 % Mehrfamilienhäuser waren [82]. Diese wiederum weisen erfahrungsgemäß häufig rechteckige Fenster ohne aufwendige Dekoration auf. Darüber hinaus muss ein vom Algorithmus erkanntes Fenster einer Wand zugeordnet werden können, damit es nicht doppelt gezählt wird, wenn es in zwei Ansichten des Gebäudes zu sehen ist. Das bedeutet, dass jede Fassade eines Gebäudes in den Schrägluftbildern identifiziert werden muss. Sobald dies geschehen ist, kann eine Rektifikation der Bilder ohne großen Mehraufwand erfolgen. Die Vorteile, die die semantische Segmentierung gegenüber der objektbezogenen Segmentierung bietet, sind für diese Arbeit dementsprechend nicht relevant, da sich die Fenster der Mehrfamilienhäuser durch Rechtecke darstellen lassen und die Bilder aufgrund der Rektifikation nicht verzerrt sind. Deshalb wird in dieser Arbeit die objektbezogene Segmentierung durchgeführt. Hierzu wird, in Anlehnung an die Arbeit von Wang [92], der in Kapitel 2.4.2 beschriebene Faster R-CNN Algorithmus verwendet, der im Vergleich zu Mask R-CNN etwas einfacher aufgebaut ist. Auf eine Modifikation des Algorithmus, wie sie von Ma und Ma [65] vorgenommen wurde, wurde in dieser Arbeit verzichtet, da diese laut den Autor*innen für rektifizierte Bilder ähnliche Ergebnisse hervorbrachte wie der einfache Faster R-CNN Algorithmus.

Die Implementierung von Faster R-CNN erfolgte in Python mit PyTorch [72]. Als Backbone wurde Resnet50 mit einem FPN gewählt, da dieser auch in den Arbeiten von Wang [92] und Sun et al. [83] gute Ergebnisse lieferte. In Anlehnung an Nordmark und Ayenew [70] wurde „Transfer Learning“ angewendet, um auch mit wenigen Trainingsdaten gute Ergebnisse zu erzielen und die Trainingszeit zu minimieren. Hierzu wurden Gewichtungen mittels einer in PyTorch enthaltenen Funktion in das Faster R-CNN Modell geladen, die zuvor am COCO-Datensatz [62] trainiert worden waren. Neben den am COCO-Datensatz trainierten Gewichtungen stellt PyTorch keine weiteren vortrainierten Gewichtungen zur Verfügung. Das Training des Faster R-CNN Modells von PyTorch erfolgte wie von Ren et al. [75] beschrieben. Es wurden alle Klassen des Trainingsdatensatzes für das Training verwendet, da sowohl Nordmark und Ayenew [70] als auch Sun et al. [83] in ihren

Arbeiten vermuteten, dass das Training des Modells an mehreren Klassen das Modell verbessern kann, auch wenn die Auswertung nur anhand einer Klasse erfolgt (siehe Kapitel 2.1.2). Die Loss-Funktion L des gesamten Netzwerkes wurde, wie von Ren et al. vorgegeben, in jeder Epoche mit dem *stochastischen Gradientenverfahren* (*SGD, Stochastic Gradient Descent*) optimiert, mit

$$L = \frac{\sum_{i=1}^{N_{Batch}} L_{RPN,Class,i} + L_{RPN,Reg,i} + L_{Class,i} + L_{Reg,i}}{N_{Batch}} \quad [3.1]$$

Das Faster R-CNN-Modell von PyTorch gibt den Loss jeweils für die Klassifizierung und Regression des RPN ($L_{RPN, Class}$ und $L_{RPN, Reg}$) und die Klassifizierung und Regression des Objekterkennungsnetzwerkes (L_{Class} und L_{Reg}) aus. Für die Optimierung des Modells wurden diese vier einzelnen Loss-Werte addiert und anschließend über die Anzahl aller Batches einer Epoche N_{Batch} gemittelt. Dies geschah sowohl während des Trainings als auch während der Validierung. Der Erfolg des Trainings wurde mit mAP gemessen, um einen Vergleich mit den ML-Modellen anderer Arbeiten (siehe Kapitel 2.1.2) zu ermöglichen. Das mAP wurde mit Gleichung [2.18] berechnet. Das Training des Modells erfolgte zwar mit allen Klassen des Trainingsdatensatzes, für die Auswertung wurde jedoch nur die Klasse „Fenster“ berücksichtigt, da für diese Arbeit nur die Leistung des ML-Modells bezüglich der Fenster relevant ist. Das bedeutet, dass N_{Class} in Gleichung [2.18] zu 1 wird. Damit ist $mAP = AP$. Die Berechnung von AP mit Gleichung [2.17] erfolgte nach den Kriterien zur Auswertung des COCO-Datensatzes, um eine Bewertung der Leistung des Modells über mehrere IoU -Grenzwerte hinweg zu erlauben (siehe Kapitel 2.4.2). Das Training des in dieser Arbeit verwendeten Faster R-CNN Modells kann die in Tabelle 2 aufgeführten Parameter beeinflusst werden, die zur Optimierung des Ergebnisses variiert werden können und als Hyperparameter bezeichnet werden.

Tabelle 2: Hyperparameter des in dieser Arbeit verwendeten Faster R-CNN-Modells.

Hyperparameter	Wert
Batch-Größe Training	1
Batch-Größe Validierung	1
Batch-Größe Test	2
Learning Rate	0,001
Momentum	0,9
Weight Decay	0,0005
Maximale Anzahl der Epochen	Variiert zwischen 20, 30, 40 und 50
Score-Grenzwert	0,05
<i>IoU</i> -Grenzwerte	Variiert zwischen 0,5 und 0,95 in 0,05-Schritten
Größe des Validierungsdatensatzes	Variiert zwischen 0,1, 0,2, 0,3
Wahrscheinlichkeit für Augmentationen	0,5

Die Batch-Größe beschreibt, wie viele Bilder auf einmal in das Modell geladen werden können. Sie wird durch die Leistung des verwendeten Rechners vorgegeben. In dieser Arbeit wurde für das Training ein Rechner mit einer leistungsstarken Grafikkarte (Nvidia Quadro RTX 4000) verwendet. Die Werte für Learning Rate, Momentum und Weight Decay beeinflussen die Optimierung der Loss-Funktion L . Sie wurden in Anlehnung an die Arbeiten von Ren et al. [75], Nordmark und Ayenew [70], Wang [92] und Ma und Ma [65] gewählt, da eine Untersuchung variabler Werte für diese Hyperparameter den Rahmen dieser Arbeit überstieg. Die maximale Anzahl der Epochen wurde zwischen 20, 30, 40 und 50 variiert, da diese Größe einen entscheidenden Einfluss auf den Erfolg des Trainings hat, der zu untersuchende Bereich jedoch klein ist. Der Score-Grenzwert, der die Sicherheit beschreibt, ab der ein Objekt als erkannt gilt wurde in Anlehnung an Ma und Ma auf 0,05 festgelegt. Ein niedriger Score-Grenzwert führt dazu, dass für die Berechnung von mAP ein größerer Bereich unter der PR-Kurve (siehe Abbildung 10) berücksichtigt wird und mAP dementsprechend größer wird. Die Leistung des ML-Modells wird mit einem niedrigeren Score-Wert also umfassender bewertet. Da die Berechnung von mAP nach den Kriterien zur Auswertung des COCO-Datensatzes erfolgte, wurden die *IoU*-Grenzwerte entsprechend gewählt. Die Bedeutung der Hyperparameter „Größe des Validierungsdatensatzes“ und „Wahrscheinlichkeit für Augmentationen“ wird in Kapitel 3.1.3 erläutert.

Das Modell wurde mit der Python-Bibliothek Ray Tune [61] trainiert. Da einige Hyperparameter mehr als einen Wert enthalten, wurde für jede mögliche Kombination von Hyperparametern ein

Modell trainiert. Während des Trainings bewertete Ray Tune die Modelle anhand des Validierungs-Losses. Dabei bestand die Möglichkeit, dass das Training bereits vor Erreichen der maximalen Anzahl an Epochen abgebrochen wird, wenn der Verlauf des Validierungs-Losses keine Verbesserung mehr erkennen ließ. Für jedes der Modelle wurde mAP berechnet und so das am besten geeignete Modell bestimmt.

3.1.2 Nachbearbeitung der Ergebnisse des ML-Modells

Um die Ergebnisse des ML-Modells zu optimieren, wurden diese nachbearbeitet. Hierbei wurden Bedingungen, die erfahrungsgemäß für die meisten Fenster in Gebäudefassaden gültig sind, auf die Ergebnisse angewendet. In einem ersten Schritt wurden Überlappungen zwischen vom ML-Modell erkannten Fenstern mithilfe einer NMS (Non-Maximum Suppression) beseitigt. Hierzu wurde ein maximal zulässiger IoU -Grenzwert definiert, bis zu dem keine NMS durchgeführt wird. Um das Entfernen von korrekt erkannten Fenstern zu vermeiden, die minimale Überschneidungen mit anderen, sehr dicht daneben liegenden Fenstern aufweisen, wurde der IoU -Grenzwert auf 0,05 festgelegt.

Anschließend wurden die verbleibenden Fenster entsprechend ihrer Breite und Höhe geclustert. Hierzu wurde der k-means-Algorithmus verwendet (siehe Kapitel 2.4.4). Da die Anzahl unterschiedlicher Fensterformen je nach Bild variiert, wurde eine Methode zur Bestimmung des optimalen $K_{Cluster}$ implementiert. Für jedes Bild wurden die erkannten Fenster geclustert, wobei $K_{Cluster}$ zwischen 1 und 10 variiert wurde, da angenommen wurde, dass eine Fassade nicht mehr als 10 unterschiedliche Fenstergrößen aufweist. Für jedes $K_{Cluster}$ wurden $W(C)$ und $B(C)$ entsprechend den Gleichungen [2.21] und [2.22] berechnet. Für den Fall, dass bei einem $K_{Cluster}$ von 1 $W(C)$ kleiner als 1 ist, wurde davon ausgegangen, dass in dem entsprechenden Bild nur eine Fensterform existiert und das optimale $K_{Cluster}$ entsprechend 1 ist. Für $K_{Cluster} = 2$ wurden $W(C)$ und $B(C)$ für $K_{Cluster} = 1$ und $K_{Cluster} = 2$ verglichen. Wenn die Veränderung für mindestens einen der beiden Werte weniger als 5 % betrug, wurde davon ausgegangen, dass in der entsprechenden Kurve ein Knick auftritt und das optimale $K_{Cluster}$ 2 ist (vergleiche hierzu Kapitel 2.4.4 und Abbildung 12). Für alle $K_{Cluster}$, die größer als 2 sind, wurde das $\Delta W(C)$ und $\Delta B(C)$ jeweils zwischen $K_{Cluster}$ und $K_{Cluster} - 1$ sowie zwischen $K_{Cluster} - 1$ und $K_{Cluster} - 2$ berechnet. So wurde die Steigung der $W(C)$ - und $B(C)$ -Kurve zwischen $K_{Cluster}$ und $K_{Cluster} - 2$ bestimmt. Wenn die Steigung der $W(C)$ -Kurve um weniger als 1 % abgenommen hatte, wurde das optimale $K_{Cluster}$ auf das aktuelle $K_{Cluster}$ festgelegt. Gleiches gilt für den Fall, dass die Steigung der $B(C)$ -Kurve um weniger als 1 % zugenommen hatte. Ein Beispiel für die Ermittlung des optimalen $K_{Cluster}$ ist in Abbildung 15 dargestellt. Auf der linken Seite ist eine Fassade aus dem CMP-Datensatz zu sehen, deren Fenster geclustert wurden. Auf Basis der in diesem Kapitel beschriebenen Methode wurde hierzu ein optimales $K_{Cluster}$ von 2 ermittelt. Unter Vernachlässigung der Rundbögen der Fenster des obersten

Stockwerkes, die von den Labels des Datensatzes nicht wiedergegeben werden können, entspricht dies dem optischen Eindruck, dass in der Fassade zwei Fenstertypen auftreten. Die $W(C)$ - und $B(C)$ -Kurve sowie ein Streudiagramm der transformierten Fensterbreite und -höhe sind ebenfalls in Abbildung 15 abgebildet. In der $B(C)$ -Kurve wird hier der Knick bei $K_{Cluster} = 2$ deutlich.

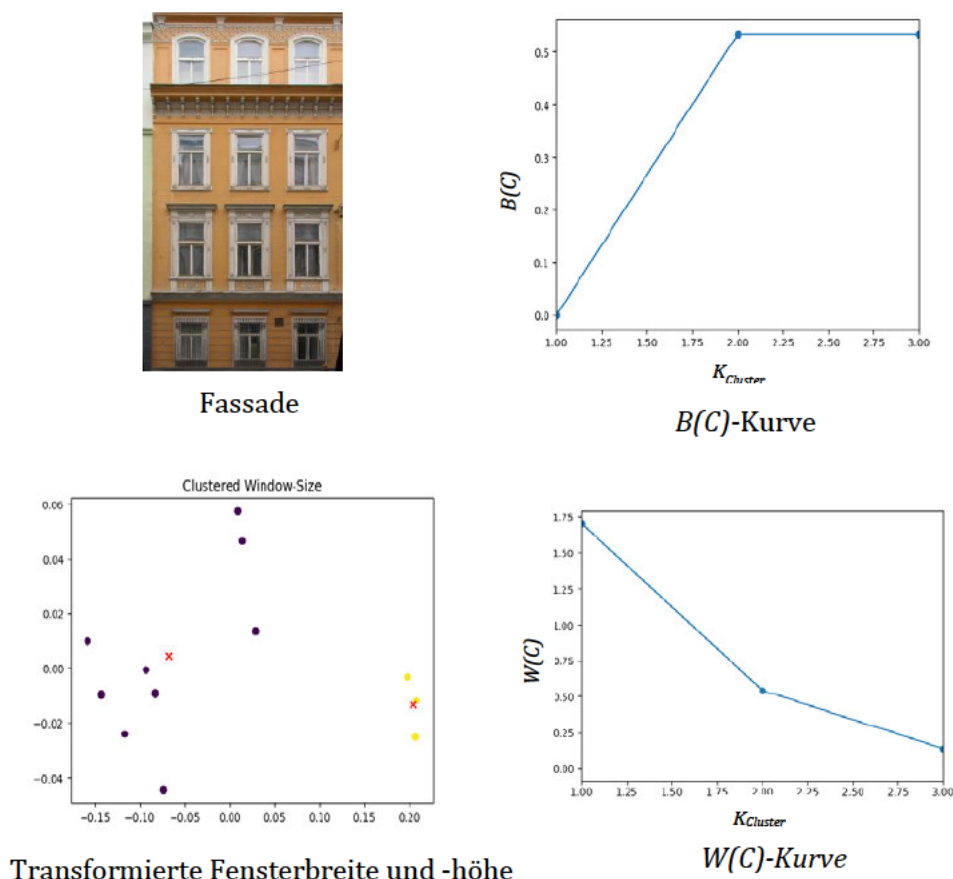


Abbildung 15: Beispiel für die Ermittlung des optimalen $K_{Cluster}$ für das Clustern der Fenster nach Breite und Höhe. Links oben ist die Fassade eines Gebäudes aus dem CMP-Datensatz [88] dargestellt, dessen Fenster geclustert wurden. Rechts oben ist die entsprechende $B(C)$ -Kurve dargestellt, rechts unten die $W(C)$ -Kurve. Links unten ist ein Streudiagramm der transformierten Fensterhöhe und -breite zu sehen. Die roten Kreuze markieren die Mittelpunkte der ermittelten Cluster.

Für jedes der mit dem optimalen $K_{Cluster}$ ermittelten Cluster wurde eine mittlere Höhe und eine mittlere Breite aller dem Cluster zugehörigen Fenster berechnet. Da das ML-Modell Fenster des gleichen Typs nicht zwangsläufig auch als gleich groß erkennt, wurde anschließend die Höhe und Breite aller Fenster eines Clusters auf diese Mittelwerte angepasst, wobei ein Überlappen der angepassten Fenster vermieden wurde. Beruhend auf der Annahme, dass in Fassaden derselbe Fenstertyp häufiger auftritt, kann das Ergebnis des ML-Modells so näher an die Realität herangeführt werden.

Gleiches gilt für die Ausrichtung der Fenster. Es kann davon ausgegangen werden, dass in einer Fassade Fenster gleichen Typs, die nebeneinander liegen, in horizontaler Richtung auf derselben Linie ausgerichtet sind, sowie dass alle Fenster, die übereinander liegen, in vertikaler Richtung auf derselben Linie ausgerichtet sind. Diese Regelmäßigkeit wird vom ML-Modell nicht zwingend erkannt. Deshalb wurden die erkannten Fenster, nachdem sie in der Größe angepasst worden waren, entsprechend ausgerichtet. Hierzu wurde für jedes Fenster der Mittelpunkt berechnet und dieser dann mit den Mittelpunkten aller anderen Fenster verglichen. Wenn die Mittelpunkte in vertikaler oder horizontaler Richtung eine Abweichung von weniger als 15 % aufwiesen, wurden sie in eine Liste mit Kandidaten für eine Ausrichtung in der jeweiligen Richtung aufgenommen. Anschließend wurde der Mittelwert der Mittelpunkte aller Kandidaten berechnet und alle Kandidaten an diesem Mittelpunkt neu ausgerichtet. Jedes Fenster wurde nur einmal neu ausgerichtet und danach für weitere Neuausrichtungen nicht weiter berücksichtigt.

Der letzte Schritt der Nachbearbeitung der Ergebnisse des ML-Modells ist das Ergänzen von Fenstern, die vom ML-Modell nicht erkannt wurden. Die Basis für diese Nachbearbeitung bildet erneut die Annahme, dass die Anordnung von Fenstern eine Regelmäßigkeit aufweist. Um vom ML-Modell übersehene Fenster zu finden, wurden die Mittelpunkte aller Fenster eines Clusters miteinander verglichen. Aus den Koordinaten der Mittelpunkte wurden alle einzigartigen x- und y-Werte ermittelt. Für jeden einzigartigen x- und y-Wert wurde die Anzahl an Fenster-Mittelpunkten gezählt, die den entsprechenden Wert in ihren Koordinaten enthalten. Anschließend wurden aus allen möglichen Kombinationen der einzigartigen x- und y-Werte Punkte gebildet. Es wurde geprüft, ob an jedem dieser Punkte ein Fenster-Mittelpunkt liegt. Die Punkte, bei denen dies nicht der Fall ist, wurden als Kandidaten definiert. Anschließend wurde für jeden Kandidaten überprüft, ob jeweils mindestens 50 % der anderen einzigartigen x- und y-Werte mehr Mittelpunkte zugeordnet haben als die x- und y-Werte des Kandidaten. Wenn dies der Fall war, wurde geprüft, ob ein Fenster, dessen Mittelpunkt der Kandidat ist und das die Höhe und Breite des entsprechenden Clusters hat, keine Überschneidungen mit angrenzenden Fenstern hat, die ein *IoU* von mehr als 0,05 aufweisen. Wenn der *IoU*-Grenzwert eingehalten wurde, wurde an der Stelle des Kandidaten ein neues Fenster mit der Größe des entsprechenden Clusters eingesetzt. Da mehrere Bedingungen zum Hinzufügen eines neuen Fensters erfüllt werden müssen, wurde davon ausgegangen, dass das neue Fenster mit hoher Wahrscheinlichkeit tatsächlich an dieser Stelle vorhanden ist. Der Score des neuen Fensters wurde deshalb auf 1 gesetzt. Der Prozess des Ergänzens von Fenstern soll an einem Beispiel in Abbildung 16 verdeutlicht werden.

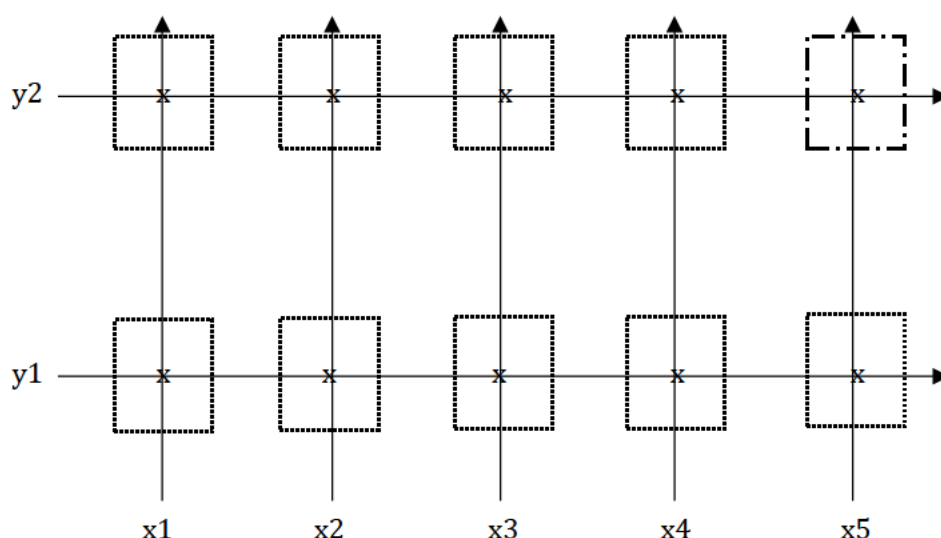


Abbildung 16: Beispiel für das Ergänzen von Fenstern, die vom ML-Modell nicht erkannt wurden. Die gepunkteten Fenster wurden vom Modell erkannt, das Fenster mit der Strichpunkt-Linie nicht. Die Kreuze markieren die möglichen Kombinationen aller einzigartigen x- und y-Werte.

Die Koordinaten der Mittelpunkte der in Abbildung 16 dargestellten Fenster haben die einzigartigen x-Werte x1, x2, x3, x4 und x5 und die einzigartigen y-Werte y1 und y2. Die Anzahl an Fenster-Mittelpunkten, die den einzigartigen x- und y-Werten zugeordnet werden können, sind in Tabelle 3 aufgelistet.

Tabelle 3: Einzigartige Koordinaten und Anzahl der dazugehörigen Fenster-Mittelpunkte für das Beispiel aus Abbildung 16.

Einzigartige Koordinate	Anzahl an Fenster-Mittelpunkten mit der Koordinate
x1	2
x2	2
x3	2
x4	2
x5	1
y1	5
y2	4

Aus allen einzigartigen x- und y-Werten werden Punkte gebildet. Beim Vergleich mit den Mittelpunkten der Fenster fällt auf, dass am Punkt $\begin{pmatrix} x5 \\ y2 \end{pmatrix}$ kein Fenster-Mittelpunkt liegt. Dieser Punkt wird nun als Kandidat gehandelt. Der x-Koordinate des Kandidaten, x5, wird in Tabelle 3 nur ein Mittelpunkt zugeordnet, bei allen anderen x-Koordinaten sind es 2. Das heißt, dass mehr

als 50 % der anderen x-Koordinaten mehr Mittelpunkte aufweisen als x5. Die y-Koordinate des Kandidaten, y2, hat 4 Mittelpunkte zugeordnet, die andere y-Koordinate y1 hat 5 Mittelpunkte zugeordnet. Dementsprechend hat y1 mehr zugeordnete Mittelpunkte als y2 und an der Position des Kandidaten wird ein Fenster eingesetzt. In Abbildung 16 ist dieses Fenster durch die Strichpunkt-Linie dargestellt. Da es keine Überschneidungen mit anderen Fenstern aufweist wird es in die Reihe der vom ML-Modell erkannten Fenster mit aufgenommen.

Die Grenzwerte, die in der Nachbearbeitung zur Entscheidungsfindung genutzt werden, wurden willkürlich gewählt, an Beispielen getestet und auf die in diesem Kapitel aufgeführten Werte angepasst. Eine systematische Optimierung der Werte konnte aus Zeitgründen nicht erfolgen.

3.1.3 Der Trainingsdatensatz

Die Nachbearbeitung der Ergebnisse des ML-Modells kann diese positiv beeinflussen. Einen deutlich größeren Einfluss hat jedoch der Trainingsdatensatz, da über ihn definiert wird, auf welche Aufgabe das ML-Modell spezialisiert werden soll. Für das Training des in dieser Arbeit verwendeten Faster R-CNN Algorithmus kommen von den in Tabelle 13 aufgelisteten Datensätzen nur diejenigen in Frage, die eine objektbezogene Segmentierung erlauben. Dies ist für die Datensätze „Street Scene Window Detection Dataset (SSWD)“ und „CMP“ der Fall. Der SSWD-Datensatz besteht aus einer großen Anzahl an Bildern, die nicht rektifizierte Straßenansichten von Paris enthalten. Die einzige Klasse des Datensatzes sind Fenster. Der CMP-Datensatz besteht im Vergleich dazu aus deutlich weniger Bildern. Allerdings beinhaltet er Bilder aus vielen verschiedenen Städten aus unterschiedlichen Ländern und ist nicht nur auf eine Stadt spezialisiert. Außerdem listet er neben der Klasse „Fenster“ 11 weitere Klassen auf, was eine erfolgreiche Fenstererkennung wahrscheinlicher macht (siehe Kapitel 2.1.2). Daher wurde für das Training des Modells in dieser Arbeit der CMP-Datensatz verwendet. Es muss hierzu jedoch angemerkt werden, dass der Datensatz keine Aufnahmen aus Hamburg enthält.

Der CMP-Datensatz ist in zwei Datensätze aufgeteilt. Der Basis-Datensatz besteht aus 378 rektifizierten Bildern, der erweiterte Datensatz aus weiteren 228 rektifizierten Bildern. In dieser Arbeit wurden alle 606 Bilder des kombinierten Basis- und Erweiterungs-Datensatzes verwendet. Die 12 Klassen des Datensatzes sind: Hintergrund, Fassade, Fenster, Tür, Sims, Fensterbank, Balkon, Jalousie, Deko, Fries, Säule, Geschäft. Der Datensatz beinhaltet die Bilder der Fassaden im JPEG-Format, die Objekt-Annotationen im XML-Format und PNG-Bilder, auf denen alle Pixel des dazugehörigen JPEG-Bildes entsprechend ihrer Klasse koloriert wurden. In der XML-Datei sind die Rahmen der Objekte im Bild, die jeweils durch den minimalen und maximalen x- und y-Wert definiert sind, sowie die dazugehörigen Klassen-Label enthalten. In Abbildung 17 sind beispielhaft ein Bild einer Fassade sowie die entsprechend den Klassen pixelweise kolorierte Fassade

dargestellt. Außerdem sind in einem dritten Bild die Rahmen der Objekte aus der XML-Datei abgebildet.

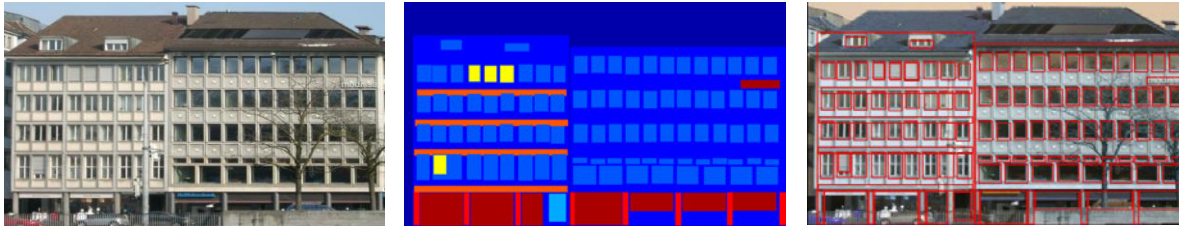


Abbildung 17: Links: Darstellung einer Fassade aus dem CMP-Datensatz [88]. Mitte: Die Pixel des linken Bildes wurden entsprechend ihrer Klasse koloriert [88]. Rechts: Die Rahmen der Objekte, mit denen das linke Bild annotiert wurde.

Der CMP-Datensatz wurde ausschließlich für Training und Validierung des ML-Modells verwendet, da zum Testen des Modells ein Hamburg-spezifischer Datensatz erstellt wurde (siehe Kapitel 3.1.4). Der CMP-Datensatz wurde in einen Trainings- und einen Validierungsdatsatz aufgeteilt. Die Zuordnung erfolgte zufällig mithilfe der Python-Bibliothek Scikit-learn [73], die eine Reproduzierbarkeit dieser Zuordnung ermöglicht. So konnte in allen Sitzungen dieselbe Zuordnung der Bilder in Trainings- und Validierungsdatsatz erfolgen. Die Größe des Validierungsdatsatzes, die als Hyperparameter in Tabelle 2 aufgeführt ist, wurde zwischen 0,1, 0,2 und 0,3 variiert, was bedeutet, dass entweder 10 %, 20 % oder 30 % des CMP-Datensatzes für die Bildung des Validierungsdatsatzes verwendet wurden. Wie bei der Maximalen Anzahl an Epochen handelt es sich auch bei diesem Hyperparameter um eine Größe, die durch kleine Anpassungen einen großen Einfluss auf das Trainingsergebnis haben kann.

Um trotz der überschaubaren Größe des Datensatzes das Faster R-CNN Modell an einer möglichst großen Anzahl unterschiedlicher Bilder zu trainieren, wurden die Bilder des CMP-Datensatzes augmentiert. Ob ein Bild augmentiert wurde, wurde beim Laden eines jeden Batches anhand einer festgelegten Wahrscheinlichkeit ermittelt. Das bedeutet, dass ein Bild, das in einer Epoche unverändert in das Modell gegeben wird, in der nächsten Epoche augmentiert werden kann. Die Anzahl unterschiedlicher Bilder, an denen das Modell trainiert wird, wurde so erhöht. Folgende Augmentationen kamen zum Einsatz:

- Horizontale Drehung
- Vertikale Drehung
- Zuschneiden
- Verwischen
- Ändern von Helligkeit und Kontrast
- Einfügen künstlicher Schatten

Die Augmentationen wurden mit Hilfe der Python-Bibliothek Albumentations [13] implementiert. Die Wahrscheinlichkeiten, mit der die Augmentationen zur Anwendung kamen, wurden für alle oben genannten Augmentationen auf 0,5 festgelegt, da dies der für fast alle Augmentationen vorgegebene Standardwert ist. Dieser Wert ist ebenfalls als Hyperparameter in Tabelle 2 aufgelistet. Beispiele für die Augmentationen sind in Abbildung 18 dargestellt.

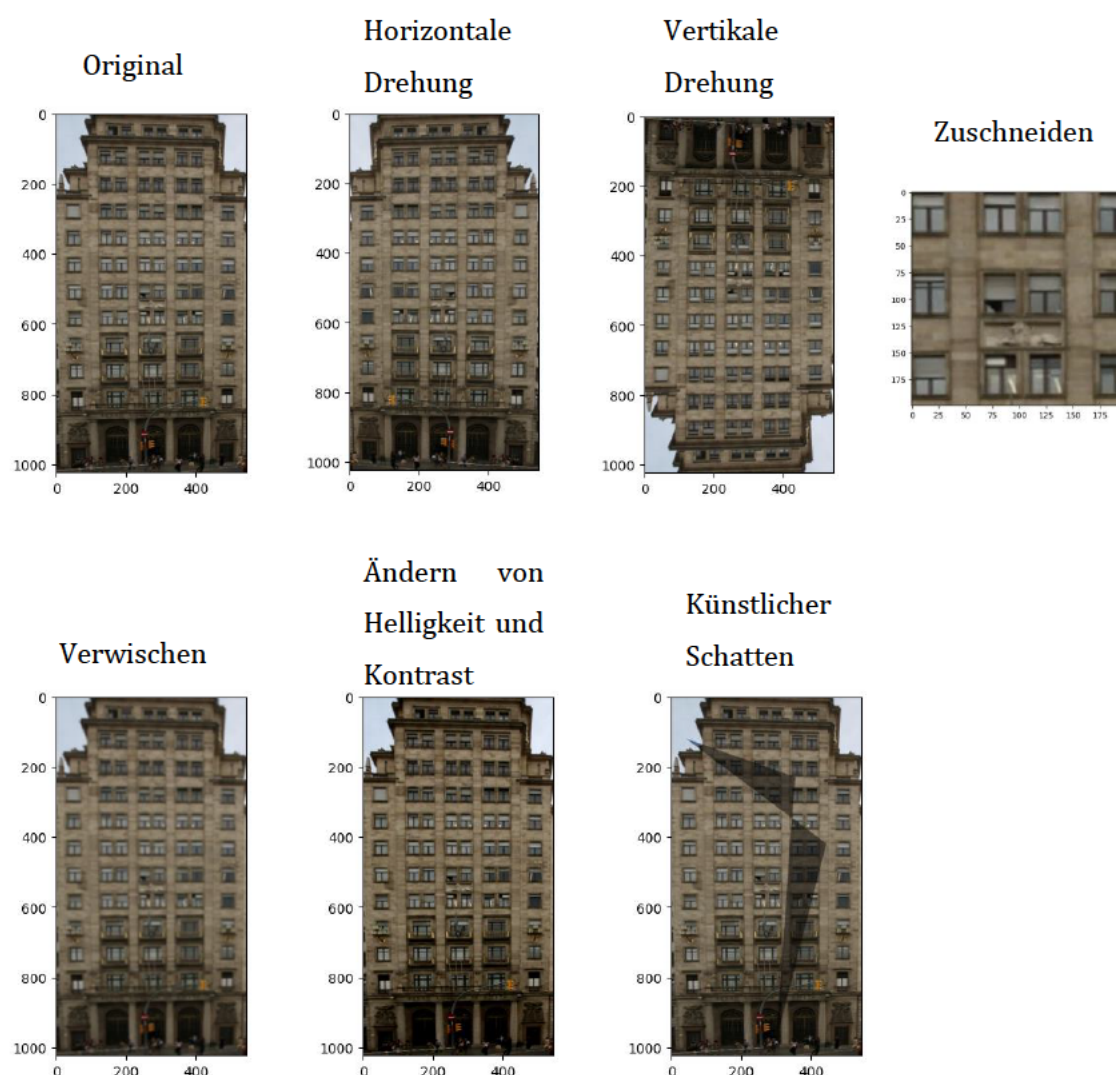


Abbildung 18: Beispiele für Augmentationen

3.1.4 Der Testdatensatz

Im Gegensatz zu dem in Kapitel 2.4.5 beschriebenen Vorgehen zum Aufteilen eines Datensatzes in Trainings-, Validierungs- und Testdatensatz wurde der oben beschriebene CMP-Datensatz nur für Training und Validierung genutzt. Zum Testen des Modells wurde ein Datensatz mit Bildern von Hamburger Gebäudefassaden erstellt, der im Folgenden als *Hamburg Image Dataset* bezeichnet wird. So konnte das ML-Modell mit dem größtmöglichen Datensatz trainieren, was bessere Ergebnisse versprach (siehe Kapitel 2.4.5). Gleichzeitig erlaubte das Testen der mit

unterschiedlichen Hyperparametern trainierten Modelle an einem Hamburg-spezifischen Datensatz die Auswahl des Modells, das am besten für Hamburger Gebäude geeignet ist. Zur Erstellung des Testdatensatzes wurden willkürlich 6 Test-Gebäude ausgewählt, die im Hamburger Stadtteil Wilhelmsburg liegen. Die Adressen der Gebäude inklusive Baujahr und der IWU-Größen- und Baualtersklasse sind in Tabelle 4 aufgelistet.

Tabelle 4: Gebäude des Testdatensatzes (Hamburg Image Dataset)

Adressen der Gebäude des Hamburg Image Datasets	Baujahr	IWU-Größenklasse	IWU-Baualtersklasse
Mengestraße 6	1966	GMH	E
Weimarer Straße 43	1949	MFH	D
Weimarer Straße 49	2014	MFH	K
Fährstraße 73	1904	MFH	B
Veringstraße 149	1952	MFH	D
Georg-Wilhelm-Straße 141	1956	MFH	D

Die Schrägluftbilder, auf denen diese Gebäude zu sehen sind, wurden so zugeschnitten, dass auf einem Bild immer nur ein Fassadenelement eines der Gebäude zu sehen ist. Hierzu wurde die in Kapitel 3.1.5 beschriebene Methodik angewendet. So entstand ein Datensatz von 44 Bildern einzelner Fassadenelemente, welche im Anschluss mit dem Web-basierten ImgLab-Tool [36] gelabelt wurden. Hierbei wurden aus zeitlichen Gründen nur die Klassen „Fenster“ und „Fassaden“ unterschieden. Als Außengrenzen eines Fensters wurde entsprechend der DIN EN ISO 10077-1 [19] der äußere Fensterrahmen angenommen. Dementsprechend wurden Doppel- oder Dreifachfenster zu einem „Fenster“-Label zusammengefasst. Da die Bilder anhand des 3D-Stadtmodells zugeschnitten wurden, welches zum Teil Dachflächen mit zu Fassaden zählt, sind auf einigen der Bilder Dachfenster zu sehen. Diese wurden nicht als „Fenster“ gelabelt und die sichtbaren Dachflächen nicht als „Fassade“. Auch Schaufenster und Fenster in Türen wurden nicht als „Fenster“ gelabelt, da davon ausgegangen werden kann, dass diese nicht an beheizte Wohnräume angrenzen und demnach für eine Wärmebedarfsprognose weniger relevant sind. Darüber hinaus werden diese Arten von Fenstern auch im CMP-Datensatz nicht zur Klasse „Fenster“ gezählt (siehe Kapitel 3.1.3). Verschattete Fenster, die auf den Bildern nicht sichtbar waren, wurden entsprechend der regelmäßigen Anordnung der sichtbaren Fenster extrapoliert. Gerundete Fenster wurden rechteckig gelabelt, gerundete Fassaden hingegen wurden durch Polygone angenähert. Die Begründung hierfür ist, dass die „Fenster“-Label zur Auswertung des ML-Modells genutzt wurden und daher mit den rechteckigen Vorhersagen des Faster R-CNN-

Modells vergleichbar sein müssen. Die „Fassade“-Label werden im Gegensatz dazu nur benötigt, um die Fläche der Fassade im Bild bestimmen und mit der Fensterfläche ins Verhältnis setzen zu können. Die Label eines jeden Bildes wurden zusammen mit der Adresse des dazugehörigen Gebäudes in einer XML-Datei im Pascal-VOC-Format mit demselben Namen wie das entsprechende Bild gespeichert. Da dieses Format nur rechteckige Label zulässt, wurden die „Fassaden“-Label gesondert in einer JSON-Datei im COCO-Format gespeichert. Von den 44 Bildern des Hamburg Image Datasets sind auf 30 Bildern Fenster abgebildet. Ein Bild aus dem Datensatz ist inklusive Labeln in Abbildung 19 dargestellt. Beispielbilder aller Gebäude aus Tabelle 4 sind im Anhang E angefügt.

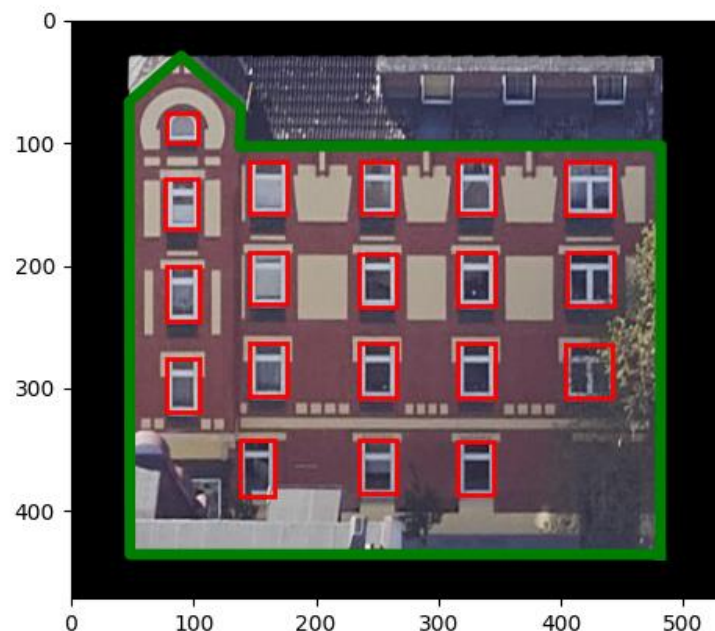


Abbildung 19: Ein gelabeltes Bild aus dem Hamburg Image Dataset. Gelabelte Fenster sind rot umrandet, die gelabelte Fassade ist grün umrandet. Im oberen linken Teil des Bildes ist ein gerundetes Fenster zu erkennen, das rechteckig gelabelt wurde. Die Doppelfenster rechts im Bild wurden jeweils zu einem Label zusammengefasst. Das Schaufenster in der linken unteren Ecke wurde nicht als Fenster gelabelt, genauso wie die Dachfenster. Das Dach wurde nicht als Fassade gelabelt.

3.1.5 Auswahl und Aufbereitung der Bilder

Um die Bilder der Gebäudefassaden zu erhalten, die die Grundlage für das Hamburg Image Dataset bilden, wurden das digitale Schrägluftbild der Stadt Hamburg aus dem Jahr 2020 sowie das Hamburger 3D-Stadtmodell aus dem Jahr 2022 im CityJSON-Format verwendet. Im Gegensatz zu den meisten Arbeiten aus Kapitel 2.1.2 wurden keine Straßenansichten verwendet, da die

gesamte Wandfläche der Gebäude aus allen 4 Himmelsrichtungen auf Fenster untersucht werden sollte.

Aus dem 3D-Stadtmodell wurde die Grundfläche jedes Gebäudes ermittelt und der Mittelpunkt dieser Fläche berechnet. Dieser Punkt wurde mit den Koordinaten verglichen, die die Fläche eines jeden Schrägluftbildes auf der Erdoberfläche beschreiben und in den Metadaten der Schrägluftbilder als „groundCoordinates“ aufgeführt sind (siehe Tabelle 1). So konnte ermittelt werden, auf welchen Bildern das zu einem Punkt gehörige Gebäude sichtbar ist. Es wurde für jede der vier Blickrichtungen Nord, Ost, Süd und West das Bild ausgewählt, auf dem das Gebäude am dichtesten am Bildmittelpunkt dargestellt ist. Die vier ausgewählten Schrägluftbilder wurden in der höchsten Auflösung heruntergeladen.

Um die Wandfläche eines Gebäudes ausschneiden zu können, mussten die Umrisse des Gebäudes im Bild identifiziert werden. Hierzu wurden die Eckpunkte des Gebäudes mit Hilfe der Gebäude-Identifikationsnummer (siehe Kapitel 2.2.2) aus der CityJSON-Datei des LoD2-Modells ermittelt. Die Eckpunkte sind in der CityJSON-Datei entweder Grund-, Wand- oder Dachflächen zugeordnet. Da eine Betrachtung sowohl der Wand- als auch der Dachflächen im Rahmen dieser Arbeit nicht möglich war, wurden nur die den Wandflächen zugeordneten Eckpunkte verwendet. Alle Wandflächen eines Gebäudes werden im LoD2-Modell durch Polygone dargestellt, die durch mindestens 3 Eckpunkte beschrieben werden. Da die Reihenfolge, in der diese Eckpunkte angegeben sind, jedoch nicht einheitlich ist, kann aus ihnen nicht direkt die Ausrichtung der Wand abgeleitet werden. Diese muss allerdings bekannt sein, um die in einem Bild sichtbaren Wandflächen identifizieren zu können. Um dieses Problem zu lösen, wurden die nach außen zeigenden Normalenvektoren aller Wandflächen bestimmt. Hierzu wurden für jede Wandfläche zwei Normalenvektoren bestimmt, die in entgegengesetzte Richtungen zeigen. Für beide Vektoren wurde überprüft, wie häufig sie andere Wandflächen desselben Gebäudes schneiden. Bei einer geraden Anzahl an Schnittpunkten wurde der Normalenvektor als nach außen zeigend definiert. In Abbildung 20 ist dieses Verfahren schematisch dargestellt.

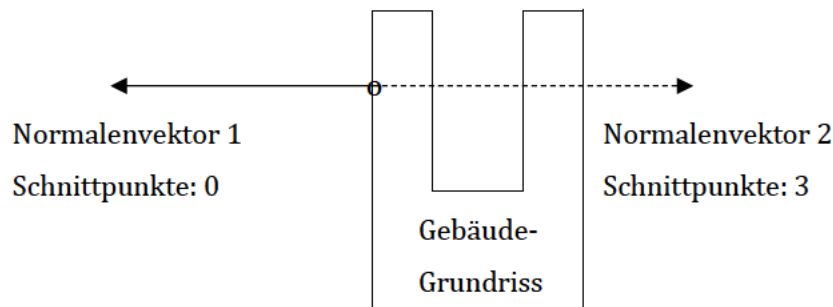


Abbildung 20: Bestimmung der nach außen zeigenden Normalenvektoren eines Gebäudes. Normalenvektor 1 weist mit 0 eine gerade Anzahl an Schnittpunkten mit anderen Wänden desselben Gebäudes auf und zeigt dementsprechend nach außen. Normalenvektor 2 weist mit 3 eine ungerade Anzahl an Schnittpunkten auf und zeigt nach innen.

Die Richtung der nach außen zeigenden Normalenvektoren wurde anschließend mit der Blickrichtung der 4 ausgewählten Schrägluftbilder verglichen. Es wurde angenommen, dass eine Wand dann als im Bild sichtbar gilt, wenn ihre Ausrichtung einen Winkel von maximal 45° zur Sichtachse aufweist. Daraus folgt, dass für ein Bild mit Blickrichtung nach Norden eine Wand sichtbar ist, wenn die Ausrichtung ihres Normalenvektors in einem Sektor zwischen 136° und 225° liegt. Das untere Ende des Sektors weist nur eine Abweichung zur Sichtachse von 44° auf, um eine Überschneidung der Sektoren und damit eine doppelte Zuordnung derselben Wand zu vermeiden. Die Sektoren und Sichtachsen sind in Abbildung 21 dargestellt und in Tabelle 5 aufgelistet.

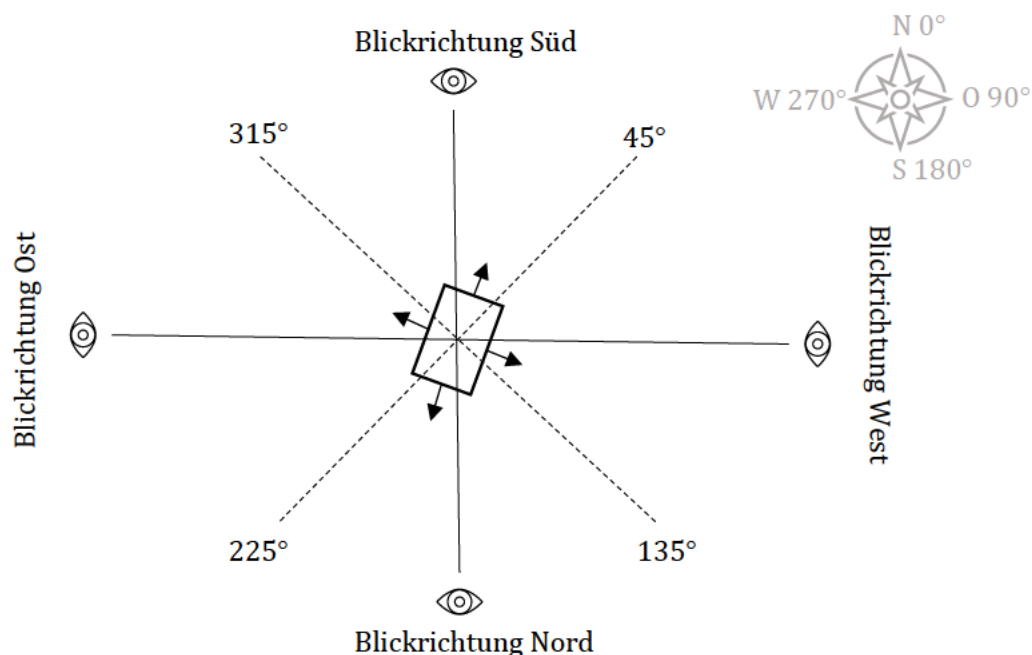


Abbildung 21: Sektoren (gestrichelte Linien) und Sichtachsen (durchgezogene Linien) zur Bestimmung sichtbarer Wände eines Gebäudes in einem Bild. Das Rechteck in der Mitte stellt ein Gebäude dar, die 4 Pfeile die Normalenvektoren der Wände.

Tabelle 5: Zulässige Sektoren für Normalenvektoren sichtbarer Wände in Abhängigkeit der Blickrichtung des Bildes.

Blickrichtung des Bildes	Ausrichtung der Normalenvektoren sichtbarer Wände
Nord	136° - 225°
Ost	226° - 315°
Süd	316° - 45°
West	46° - 135°

Mithilfe der Sektoren und der Ausrichtung der Normalenvektoren wurden für jedes der 4 Schrägluftbilder die sichtbaren Wände eines Gebäudes bestimmt. Die entsprechenden Eckpunkte der Wände wurden mit den Gleichungen [2.9] und [2.10] von geografischen Koordinaten in Bildkoordinaten umgerechnet. Als Transformationsmatrix T wurde hierbei die in Tabelle 1 aufgeführte p-to-image-Transformationsmatrix verwendet. Um die Erkennung der Fenster in den Wandflächen zu erleichtern, wurde im darauffolgenden Schritt die Umgebung der Wände im Bild schwarz gefärbt, sodass nur die Wände selbst sichtbar blieben. Hierzu wurden die transformierten Eckpunkte der Wände zu Linien verbunden, welche in einer Maske weiß koloriert wurden. Die Maske wurde weiß ausgefüllt, um als Vorlage zum Ausschneiden der Wände aus dem Bild zu dienen. Abschließend wurde die Maske invertiert und über das Bild gelegt, sodass nur die Wände sichtbar blieben. In Abbildung 23 sind diese Schritte dargestellt.

Da die Erkennung der Fenster mit dem Faster R-CNN Algorithmus erfolgen soll, wurden die ausgeschnittenen Wände anschließend rektifiziert. Hierzu wurden die ersten drei Eckpunkte jeder Wand betrachtet, von denen sowohl die geografischen Koordinaten ($P_{0,geo}$, $P_{1,geo}$, $P_{2,geo}$) als auch die Koordinaten im nicht rektifizierten Bild ($P_{0,bild}$, $P_{1,bild}$, $P_{2,bild}$) bekannt sind. Die Entfernung von $P_{0,bild}$ zu $P_{1,bild}$ und von $P_{1,bild}$ zu $P_{2,bild}$ wurde in x-Richtung gemessen, um zu ermitteln, ob es sich um eine horizontale oder eine vertikale Seite des Bildes handelt, da die Reihenfolge, in der die Koordinaten angegeben werden, nicht bekannt ist. Anschließend wurde das Seitenverhältnis S der Wand aus den geografischen Koordinaten berechnet, um damit die Eckpunkte der Wand in der rektifizierten Ebene ($P_{0,rect}$, $P_{1,rect}$, $P_{2,rect}$, $P_{3,rect}$, $P_{4,rect}$) zu bestimmen. Dieser Prozess soll an einem Beispiel erläutert werden. Hierzu ist in Abbildung 22 eine Wand schematisch in der Realität, in einem nicht rektifizierten Bild und in einem rektifizierten Bild dargestellt.

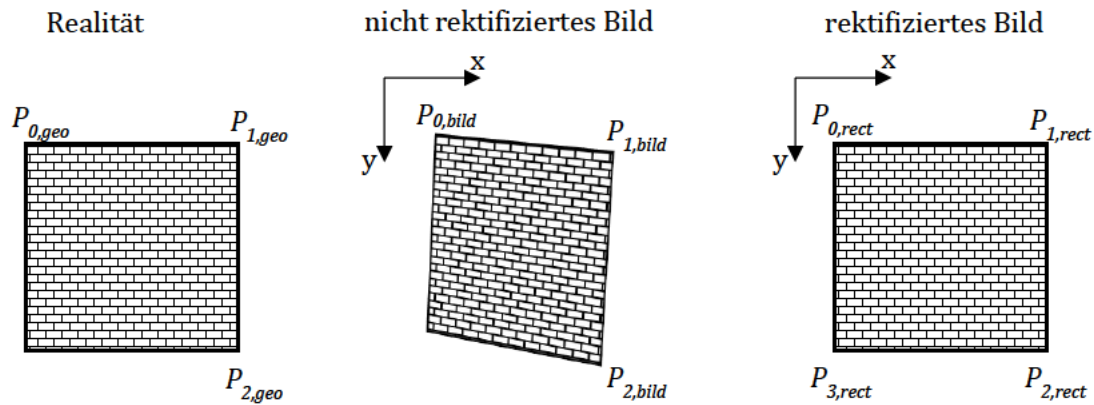


Abbildung 22: Schematische Darstellung einer Wand in der Realität, in einem nicht rektifizierten Bild und in einem rektifizierten Bild.

In x-Richtung betrachtet ist der Abstand Δx zwischen $P_{0,bild}$ und $P_{1,bild}$ in Abbildung 22 größer als der Abstand zwischen $P_{1,bild}$ und $P_{2,bild}$. Daraus folgt, dass die Linie zwischen $P_{0,bild}$ und $P_{1,bild}$ eine horizontale Seite der Wand darstellt. Entsprechend wird das Seitenverhältnis S in der Realität berechnet mit:

$$S = \frac{\text{horizontale Seite}}{\text{vertikale Seite}} = \frac{\overline{P_{0,geo}P_{1,geo}}}{\overline{P_{1,geo}P_{2,geo}}} \quad [3.2]$$

Hierbei stellt $\overline{P_{0,geo}P_{1,geo}}$ die Entfernung zwischen den Punkten $P_{0,geo}$ und $P_{1,geo}$ und $\overline{P_{1,geo}P_{2,geo}}$ die Entfernung zwischen den Punkten $P_{1,geo}$ und $P_{2,geo}$ dar. Um die Richtung zu bestimmen, in der die Eckpunkte in der Realität und im nicht rektifizierten Bild angegeben werden, wird der Abstand

Δy zwischen $P_{1,bild}$ und $P_{2,bild}$ berechnet. Die Eckpunkte im rektifizierten Bild werden anschließend folgendermaßen bestimmt:

$$P_{0,rect} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = P_{0,bild} = P_{4,rect} \quad [3.3]$$

$$P_{1,rect} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 - \Delta x \\ y_0 \end{pmatrix} \quad [3.4]$$

$$P_{2,rect} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 - sign * \frac{|\Delta x|}{S} \end{pmatrix} \quad [3.5]$$

$$P_{3,rect} = \begin{pmatrix} x_3 \\ y_3 \end{pmatrix} = \begin{pmatrix} x_2 + \Delta x \\ y_2 \end{pmatrix} \quad [3.6]$$

Mit

$$sign = \begin{cases} -1, & \Delta y < 0 \\ 1, & \Delta y > 0 \end{cases} \quad [3.7]$$

Und

$$\Delta x = \overline{P_{0,bild}P_{1,bild}} \quad [3.8]$$

$$\Delta y = \overline{P_{1,bild}P_{2,bild}} \quad [3.9]$$

Einen Sonderfall bilden sehr schmale, schräg orientierte Wände. Diese treten vor allem dort auf, wo Rundungen von Häusern im 3D-Stadtmodell durch viele schmale, rechteckige Polygone angenähert werden. Da in diesen Fällen Δx zwischen den vertikal orientierten Punkten größer sein kann als zwischen den horizontal orientierten, muss hier das Seitenverhältnis S für die Bestimmung der horizontalen Linie herangezogen werden. Hierzu wurde S abweichend zu [3.1] bestimmt mit:

$$S = \frac{\max(\overline{P_{0,geo}P_{1,geo}}, \overline{P_{1,geo}P_{2,geo}})}{\min(\overline{P_{0,geo}P_{1,geo}}, \overline{P_{1,geo}P_{2,geo}})} \quad [3.10]$$

Ein Wandsegment wurde als sehr schmal definiert, sobald S größer als 10 wird. Da derart schmale Wandsegmente aus Erfahrung nur dort im 3D-Stadtmodell auftreten, wo Rundungen angenähert

werden sollen, wurde davon ausgegangen, dass diese Segmente immer vertikal ausgerichtet sind. Daher wurde für den Fall, dass S größer als 10 ist, die Seite als horizontal definiert, die ein kleineres Δy aufweist. Anschließend konnten die Eckpunkte im rektifizierten Bild entsprechend den Gleichungen [3.3] bis [3.9] bestimmt werden. Ein weiterer Sonderfall sind dreieckige Wandflächen, zum Beispiel in Form von Dachgiebeln. Da für die hier beschriebene Methode zur Rektifikation 4 Eckpunkte notwendig sind, wurde aus den minimalen x- und y-Koordinaten der Eckpunkte des Dreiecks ein Rechteck aufgespannt, mit welchem die Rektifikation erfolgen konnte.

Nach der Bestimmung der Eckpunkte der Wände im rektifizierten Bild wurde die projektive Transformationsmatrix H aus den Koordinaten dieser Punkte sowie den Koordinaten der Eckpunkte im nicht rektifizierten Bild berechnet. Hierzu wurde die Bibliothek OpenCV [11] für Python verwendet, welche die in Kapitel 2.3.2 beschriebenen Grundlagen anwendet. Mit Hilfe der Matrix H wurde anschließend das nicht rektifizierte Bild auf eine Ebene projiziert, in der die Wand nicht verzerrt ist, und so rektifiziert. Die Projektion wurde mit der Funktion `cv2.warpPerspective` durchgeführt. Das rektifizierte Bild bildet den Abschluss der Bildaufbereitung und ist die Basis für die Fensterflächenerkennung.

Alle Schritte der in diesem Kapitel beschriebenen Aufbereitung des Schrägluftbildes eines Gebäudes sind in Abbildung 23 anhand der resultierenden Bilder dargestellt.

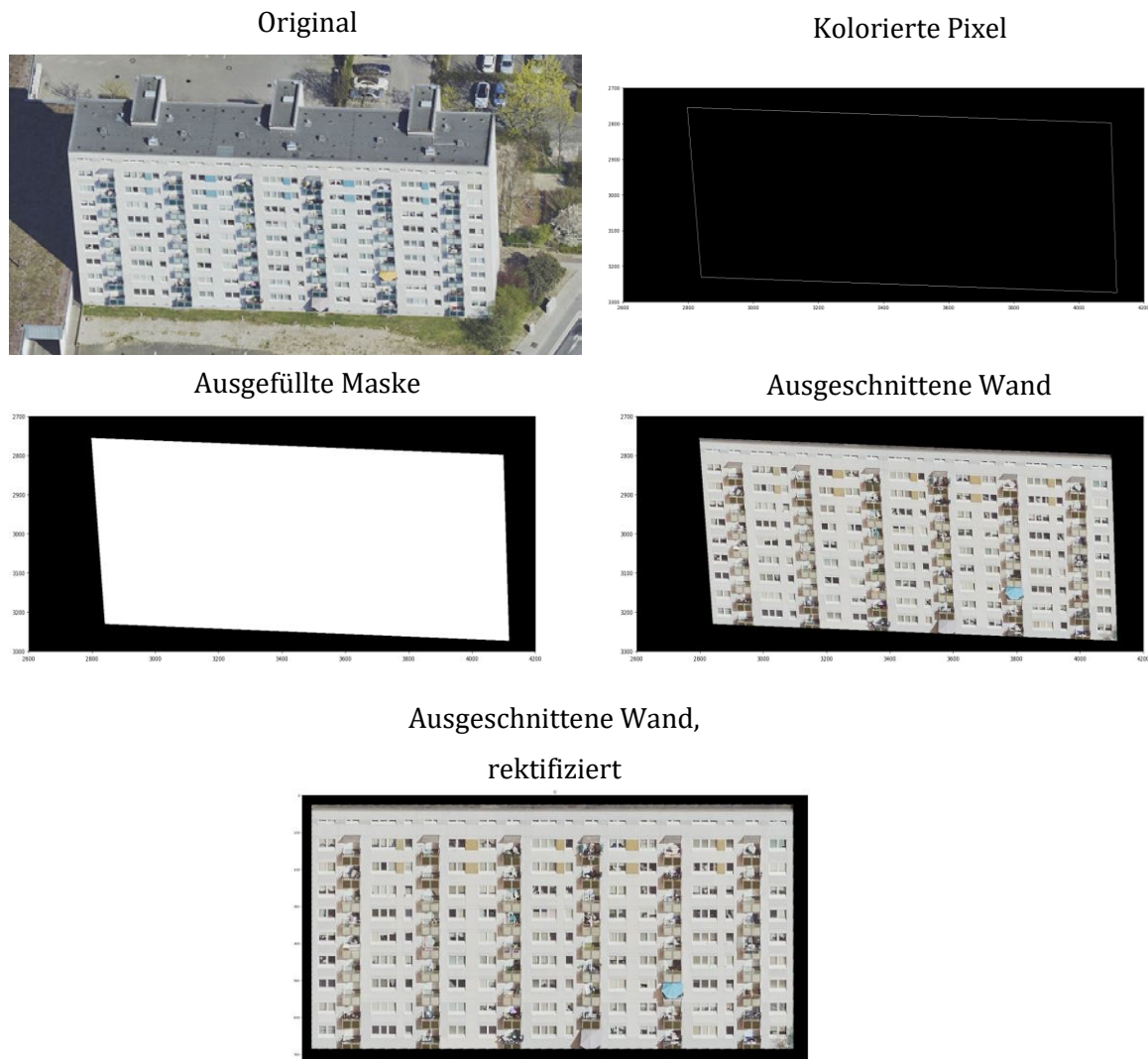


Abbildung 23: Alle Schritte der Aufbereitung des Schrägluftbildes eines Gebäudes.

3.1.6 Berechnung des Verhältnisses der Fenster- zur Wandfläche

Um die Fensterfläche einer Wand zu bestimmen, aus der $R_{\text{Gebäude}}$ berechnet werden kann, wurde das rektifizierte Bild in das trainierte Faster R-CNN-Modell gegeben. Der Score-Grenzwert, der für das Training des ML-Modells auf 0,05 festgelegt worden war (siehe Kapitel 3.1.1), wurde hierbei zwischen 0,05, 0,25, 0,5 und 0,8 variiert. Dies hat den Hintergrund, dass ein niedriger Score-Grenzwert zwar zu einem höheren mAP führt, dies jedoch nicht automatisch eine höhere Übereinstimmung der vorhergesagten mit der tatsächlichen Fensterfläche zur Folge hat. Ursächlich hierfür ist die Bestimmung von mAP über Präzision und Recall, die jeweils nur die Anzahl an falsch oder richtig erkannten Objekten berücksichtigen (siehe Gleichungen [2.15] und [2.16]) und nicht die Größe der erkannten Fenster. Die Grenzen des untersuchten Bereichs für den Score-Grenzwert orientieren sich an den Arbeiten von Ma und Ma [65] und Wang [92]. Das Modell berechnete Vorhersagen für alle 12 Klassen, auf die es trainiert worden war. Diese

Vorhersagen bestehen aus den Rahmen der erkannten Objekte in Form der minimalen und maximalen x- und y-Werte, den zugeordneten Labeln sowie der Sicherheit, mit der das Objekt dem entsprechenden Label zugeordnet wurde (Score, siehe Kapitel 2.4.2). Es wurden im Folgenden nur die Vorhersagen für die Klasse Fenster weiterverwendet. Aus den Koordinaten der Rahmen wurde die Fläche des entsprechenden Fensters $A_{Fenster}$ in Pixel² berechnet:

$$A_{Fenster} = (x_{max} - x_{min}) * (y_{max} - y_{min}) \quad [3.11]$$

Mit den minimalen und maximalen x- und y-Koordinaten x_{min} , y_{min} , x_{max} und y_{max} des vom Faster R-CNN vorhergesagten Rahmens, der das Fenster repräsentiert. Die Fensterflächen aller vorhergesagten Fenster wurden für eine Wand addiert. Das Verhältnis R_{Wand} der gesamten Fensterfläche einer Wand zur Fläche der rektifizierten Wand A_{Wand} wurde berechnet mit:

$$R_{Wand} = \frac{\sum_{i=1}^{N_{Fenster}} A_{Fenster,i}}{A_{Wand}} \quad [3.12]$$

Mit der Anzahl an Fenstern in der Wand $N_{Fenster}$. Die Wandfläche A_{Wand} wurde mit den in Kapitel 3.1.5 beschriebenen Eckpunkten der rektifizierten Wand bestimmt, indem aus den Eckpunkten mithilfe der Python-Bibliothek Shapely [29] ein Polygon erstellt und dessen Fläche berechnet wurde. Für dreieckige Wände, die für die Rektifikation der Bilder in Rechtecke umgewandelt worden waren (siehe Kapitel 3.1.5), wurde A_{Wand} mit der Hälfte der mit Shapely berechneten Fläche des Polygons gleichgesetzt. Für das gesamte Gebäude wurden alle Fenster- und Wandflächen des Gebäudes addiert und das Verhältnis $R_{Gebäude}$ berechnet mit:

$$R_{Gebäude} = \frac{\sum_{j=1}^{N_{Wände}} \sum_{i=1}^{N_{Fenster}} A_{Fenster,i,j}}{\sum_{j=1}^{N_{Wände}} A_{Wand,j}} \quad [3.13]$$

Mit der Anzahl an Wandflächen des Gebäudes $N_{Wände}$. Es wird davon ausgegangen, dass R_{Wand} und $R_{Gebäude}$ das tatsächliche Verhältnis der Fenster- zur Wandfläche repräsentieren, auch wenn die Berechnung im rektifizierten Bild erfolgte, da die Wand im rektifizierten Bild dasselbe Seitenverhältnis S aufweist wie die reale Wand (siehe Gleichung [3.2]). Um einordnen zu können, wie gut R_{Wand} und $R_{Gebäude}$ die Realität abbilden, wurden aus den Labeln des Hamburg Image Datasets die Verhältnisse $R_{Wand, HI}$ und $R_{Gebäude, HI}$ mit den Gleichungen [3.12] und [3.13] als Grundwahrheit berechnet. Hierbei wurden die x- und y-Koordinaten für $A_{Fenster}$ aus den XML-Dateien der zugehörigen Bilder entnommen. Außerdem wurde die Fläche A_{Wand} von der Software ImageLab bereits während des Labelns der Fassade berechnet und konnte dementsprechend aus

der JSON-Datei des Hamburg Image Datasets übernommen werden. Die Abweichung ΔR_{Wand} von R_{Wand} zu $R_{Wand, HI}$ wurde berechnet mit:

$$\Delta R_{Wand} = \left(1 - \frac{R_{Wand}}{R_{Wand, HI}}\right) * 100\% \quad [3.14]$$

Die Abweichung $\Delta R_{Gebäude}$ von $R_{Gebäude}$ zu $R_{Gebäude, HI}$ wurde folgendermaßen ermittelt:

$$\Delta R_{Gebäude} = \left(1 - \frac{R_{Gebäude}}{R_{Gebäude, HI}}\right) * 100\% \quad [3.15]$$

Anschließend wurde der Mittelwert $\Delta R_{Gebäude, mean}$ von $\Delta R_{Gebäude}$ über alle Gebäude des Hamburg Image Datasets ermittelt mit:

$$\Delta R_{Gebäude, mean} = \frac{\sum_{i=1}^{N_{Gebäude}} \Delta R_{Gebäude, i}}{N_{Gebäude}} \quad [3.16]$$

Mit der Anzahl der Gebäude $N_{Gebäude}$. Da die in diesem Kapitel beschriebenen Berechnungen nur für die Gebäude des Hamburg Image Datasets durchgeführt wurden, die als Stichprobe des gesamten Hamburger Gebäudebestandes gesehen werden können, wurde die empirische Standardabweichung σ für $\Delta R_{Gebäude, mean}$ folgendermaßen berechnet [78]:

$$\sigma = \sqrt{\frac{1}{N_{Gebäude} - 1} \sum_{i=1}^{N_{Gebäude}} (\Delta R_{Gebäude, i} - \Delta R_{Gebäude, mean})^2} \quad [3.17]$$

Neben dem Vergleich von $R_{Gebäude, HI}$ mit $R_{Gebäude}$ erfolgte auch ein Vergleich mit den Werten für $R_{Gebäude, IWU}$, die in der IWU-Typologie in Tabelle 12 im Anhang C aufgeführt werden. Die Gebäudedaten der Gebäude des Hamburg Image Datasets wurden dabei aus Tabelle 4 übernommen. Die Werte $\Delta R_{Gebäude}$, $\Delta R_{Gebäude, mean}$ und σ wurden auch für die Abweichung der Werte der IWU-Typologie zu den Labeln des Hamburg Image Datasets berechnet, wobei in die Gleichungen [3.15] bis [3.16] $R_{Gebäude, IWU}$ an Stelle von $R_{Gebäude}$ eingesetzt wurde.

3.2 Bestimmung der Baualtersklasse eines Gebäudes

Während sich die in dieser Arbeit entwickelte Methodik zur Erkennung von Fenstern in Gebäudefassaden an der vorgestellten Literatur orientiert, wurde zur Bestimmung des Baujahres, aus dem die Baualtersklasse abgeleitet werden kann, ein Ansatz gewählt, der sich von den in Kapitel 2.1.3 beschriebenen Ansätzen deutlicher unterscheidet. Dies hat den Hintergrund, dass für die Ansätze aus Kapitel 2.1.3 zum Teil umfassende Kenntnisse über Eigenschaften des

betrachteten Gebäudes und seiner Nachbargebäude notwendig sind. Die Verlässlichkeit dieser Informationen ist hierbei entscheidend. Für Hamburg zeigte sich jedoch, dass der ALKIS-Datensatz fehlerhafte Baujahre enthielt. Dies reduziert nicht nur die Anzahl an Gebäuden, für die eine detaillierte Wärmebedarfsprognose basierend auf dem Baujahr erstellt werden kann, sondern beeinträchtigt auch die Anwendbarkeit der Methoden aus der Literatur. Daher wurde in dieser Masterarbeit eine Bestimmung der Baualtersklasse auf Basis historischer Karten durchgeführt, die neben den Karten nur die Grundrisskoordinaten der Gebäude aus ALKIS benötigt. In Abbildung 24 ist die Vorgehensweise schematisch dargestellt.

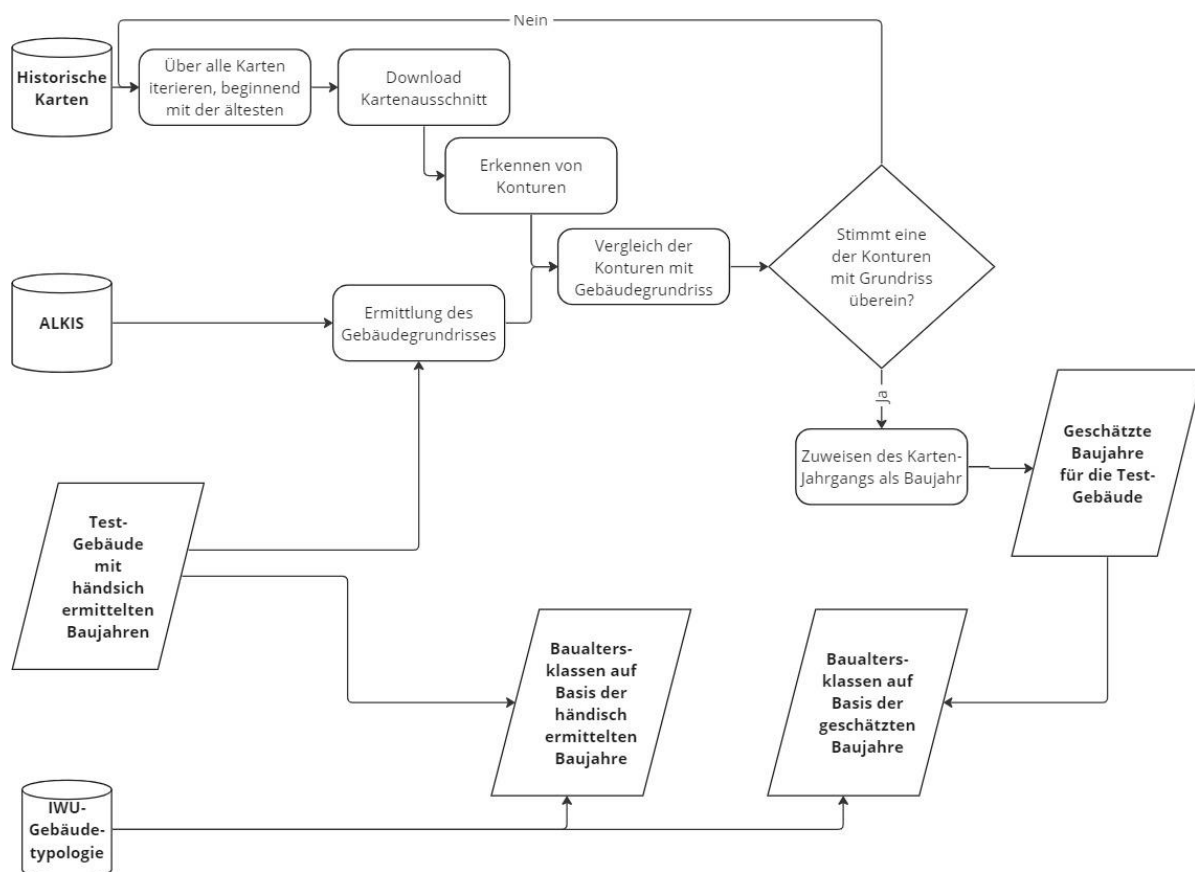


Abbildung 24: Darstellung der Methodik zur Bestimmung der Baualtersklassen von Gebäuden als Flussdiagramm.

Um die Methodik bewerten zu können wurde ein Datensatz mit Test-Gebäuden erstellt, deren Baujahre händisch zusammengetragen wurden. Im Folgenden werden die mit der vorgestellten Methodik bestimmten Baujahre als „geschätzte Baujahre“ bezeichnet, die Baujahre aus dem Testdatensatz als „tatsächliche Baujahre“. Auf Basis der tatsächlichen und geschätzten Baujahre wurden die Baualtersklassen der Gebäude bestimmt und verglichen. Die Unterteilung der Baualtersklassen erfolgte entsprechend der IWU-Wohngebäudetypologie. Neben den Baualtersklassen wurden auch die tatsächlichen und geschätzten Baujahre selbst miteinander verglichen.

3.2.1 Auswahl und Aufbereitung der Karten

Um das Baujahr eines Gebäude schätzen zu können, wurde eine Verbindung zwischen dem ALKIS-Datensatz „GebäudeBauwerk“ [27] und den historischen Karten von Hamburg hergestellt. Beide Datenquellen sind in Kapitel 2.2 beschrieben. Zunächst wurde hierzu das Grundriss-Polygon des Gebäudes anhand der Gebäudeidentifikationsnummer aus dem ALKIS-Datensatz ermittelt. Anschließend wurde der Mittelpunkt des Grundriss-Polygons berechnet. Um zu ermitteln, ob das Gebäude zu einem definierten Zeitpunkt bereits existierte, wurden daraufhin Ausschnitte der historischen Karten in Form quadratischer Kacheln über den WMS heruntergeladen. Hierbei bildete der zuvor berechnete Mittelpunkt des Polygons auch den Mittelpunkt der Kacheln. Breite und Höhe des in den Kacheln dargestellten Gebiets wurden auf 500 Meter festgelegt, um nicht nur das Gebäude selbst, sondern auch dessen Umgebung abzubilden. Die Karten wurden als Raster-Dateien heruntergeladen. Um eine Bearbeitung der Karten zu ermöglichen, wurden diese anschließend in Bild-Dateien umgewandelt. Da die geografischen Abmessungen der Karten-Kacheln bekannt sind, wurden anschließend auch die Koordinaten des Grundflächen-Polygons des Gebäudes in Bild-Koordinaten umgerechnet. Bei dieser Transformation von geografischen in Bild-Koordinaten liegt keine Rotation oder Translation vor, weshalb nicht auf die in Kapitel 2.3 beschriebenen Methoden zurückgegriffen wurde. Stattdessen erfolgte die Berechnung der Bild-Koordinaten mit Hilfe der minimalen und maximalen Nord-Koordinate (N_{min} und N_{max}) sowie der minimalen und maximalen Ost-Koordinate (E_{min} und E_{max}) der Karten-Kachel. Diese geografischen Koordinaten wurden mit den Bild-Koordinaten x_{min} , y_{min} , x_{max} und y_{max} der Kachel verglichen. Beliebige Koordinaten x und y im Bild der Karte, die den geografischen Koordinaten N und E entsprechen, lassen sich so berechnen mit:

$$x = x_{min} + \Delta x * \frac{E - E_{min}}{E_{max} - E_{min}} \quad [3.18]$$

$$y = y_{max} + \Delta y * \frac{N - N_{min}}{N_{max} - N_{min}} \quad [3.19]$$

mit

$$\Delta x = x_{max} - x_{min} \quad [3.20]$$

$$\Delta y = y_{max} - y_{min} \quad [3.21]$$

Zu Gleichung [3.19] muss angemerkt werden, dass hier der maximale y -Wert verwendet wird, da die Darstellung von Bildern mit einer invertierten y -Achse erfolgt. Zur Verdeutlichung ist in Abbildung 25 ein Beispiel dargestellt. Dort wird der Punkt $\begin{pmatrix} N \\ E \end{pmatrix}$ der Raster-Datei in den Punkt $\begin{pmatrix} x \\ y \end{pmatrix}$ in der Bild-Datei umgewandelt. Der untere linke Punkt der Raster-Datei, $\begin{pmatrix} N_{min} \\ E_{min} \end{pmatrix}$, wird dabei zum unteren linken Punkt der Bild-Datei, $\begin{pmatrix} x_{min} \\ y_{max} \end{pmatrix}$, transformiert. Der obere rechte Punkt $\begin{pmatrix} N_{max} \\ E_{max} \end{pmatrix}$ wird zum Punkt $\begin{pmatrix} x_{max} \\ y_{min} \end{pmatrix}$ in der Bild-Datei.

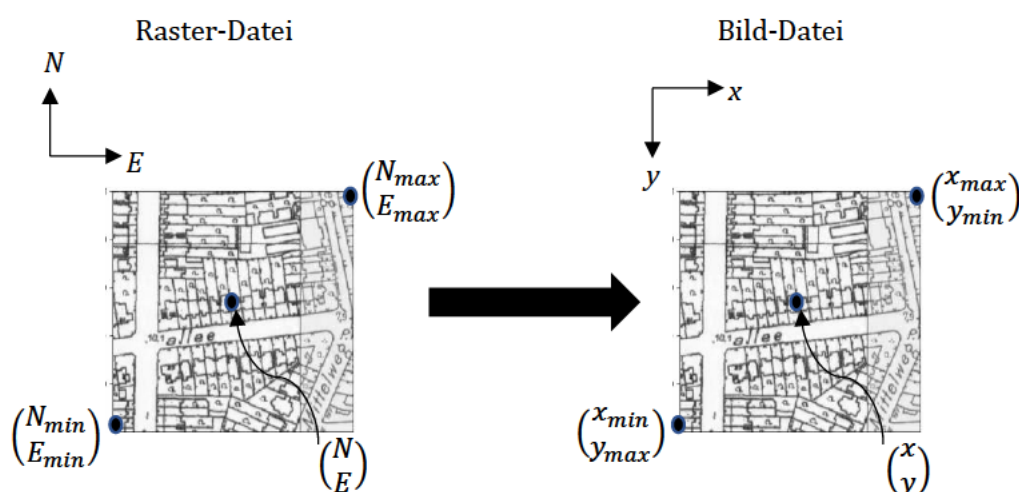


Abbildung 25: Schematische Darstellung der Transformation von geografischen (UTM) Koordinaten einer Raster-Datei in Bild-Koordinaten.

Zur Bestimmung des Baujahres des Gebäudes wurde dessen Grundriss-Polygon mit den auf der Karte aufgezeichneten Grundrissen verglichen. Die Grundrisse der Gebäude in den Karten wurden mit Hilfe der Python-Bibliothek OpenCV [11] ermittelt. Hierzu wurden das Bild einer Karte zunächst in Graustufen und anschließend in ein binäres schwarz-weißes Bild umgewandelt, welches zusätzlich invertiert wurden. Anschließend konnten innerhalb des Bildes mittels der OpenCV-Funktion „findContours“ Konturen ermittelt werden. Die Konturen wurden in Polygone umgewandelt, welche mit dem Grundriss-Polygon des Gebäudes verglichen werden konnten. In Abbildung 26 ist ein Kartenausschnitt zu sehen, auf dem alle erkannten Konturen grün umrandet sind.

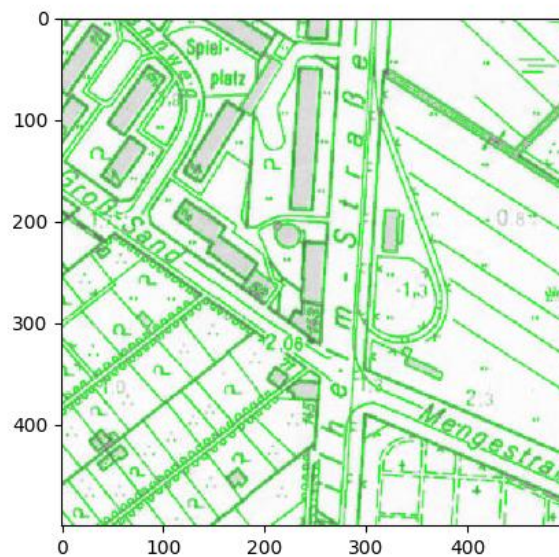


Abbildung 26: Ausschnitt einer historischen Karte. Grün umrandet sind die Konturen, die von OpenCV erkannt wurden.

3.2.2 Bestimmung des Baujahres und der Baualtersklasse

Das Baujahr eines Gebäudes wurde über den Jahrgang der Karte definiert, in der das Grundriss-Polygon mit einem von OpenCV erkannten Polygon übereinstimmt. Hierzu wurde die in Kapitel 2.4.2 beschriebene Messgröße der *Intersection over Union (IoU)* angewendet. Entsprechend Gleichung [2.14] wurden sowohl die Überschneidung als auch die vereinigte Fläche des Grundriss-Polygons mit allen im Bild gefundenen Polygonen berechnet und ins Verhältnis gesetzt. Es wurde ein Grenzwert IoU_{min} definiert, der überschritten werden muss, damit ein Grundriss-Polygon als übereinstimmend mit einem gefundenen Polygon gilt. Dieser Grenzwert wurde zwischen 0,9 und 0,1 mit einer Schrittweite von 0,05 variiert. Es wurde immer das maximal in einem Bild ermittelte IoU eines Gebäudes mit dem IoU_{min} verglichen. Eine Besonderheit stellen hierbei Grundriss-Polygone dar, die im ALKIS-Datensatz mehr als einer Adresse zugeordnet sind. In einigen Karten sind diese Grundrisse zum Teil entsprechend den Adressen getrennt aufgezeichnet. Die Erkennung der Konturen führt in diesem Fall dazu, dass der Grundriss des gesamten, zusammenhängenden Gebäudes nicht erkannt wird, sondern nur die einzelnen Gebäudeelemente. Zur Veranschaulichung ist hierzu in Abbildung 27 auf der linken Seite ein Gebäude dargestellt, das in dieser Form auf einer Karte abgebildet sein könnte und 4 Adressen beinhaltet. Das Grundriss-Polygon des Gebäudes ist dick schwarz umrandet und an der zweiten Position von links separat dargestellt, die Adressen sind durch dünne schwarze Linien voneinander getrennt. In dem dargestellten Fall ist ein Überschreiten von IoU_{min} unwahrscheinlich, auch wenn das Gebäude in der Karte vorhanden ist, da die erkannten Konturen nur die Gebäudeelemente entsprechend den Adressen abbilden. In Abbildung 27 sind die

erkannten Konturen durch schraffierte Rechtecke gekennzeichnet. Die $IoUs$, die aus den 4 erkannten Konturen und dem Grundriss-Polygon berechnet werden können, sind in der Abbildung ganz rechts aufgelistet. Wenn IoU_{min} nun zum Beispiel auf 0,5 festgelegt wurde, wird das Gebäude in der Karte nicht erkannt, da keins der aufgelisteten IoU den Grenzwert von 0,5 übersteigt. Um eine Erkennung derartiger Gebäude trotzdem zu ermöglichen, wurde für Grundriss-Polygone, die mehr als einer Adresse zugeordnet sind, überprüft, ob eine definierte Anzahl $N_{Adressen}$ an ermittelten IoU den Wert $\frac{IoU_{min}}{N_{Adressen}}$ überschreitet. $N_{Adressen}$ entspricht hierbei der Anzahl an Adressen, die dem Grundriss-Polygon zugeordnet sind. Die $IoUs$ wurden nach Größe sortiert, sodass immer die $N_{Adressen}$ besten $IoUs$ mit $\frac{IoU_{min}}{N_{Adressen}}$ verglichen wurden. Dem Beispielgebäude in Abbildung 27 können 4 Adressen zugeordnet werden, demnach ist $N_{Adressen} = 4$. Daraus folgt, dass die besten 4 IoU , die zwischen dem Grundriss-Polygon und den erkannten Konturen ermittelt wurden, jeweils den Wert von $\frac{IoU_{min}}{N_{Adressen}} = \frac{0,5}{4} = 0,125$ überschreiten müssen. Da im Beispiel nur 4 IoU vorliegen und alle größer sind als 0,125, gilt das Gebäude somit als erkannt.

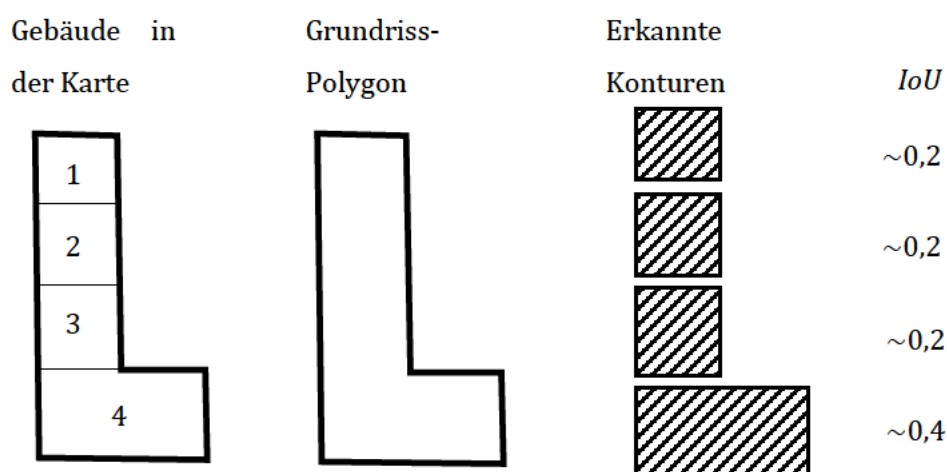


Abbildung 27: Beispielhafte Darstellung eines Gebäudes, dem 4 Adressen zugewiesen sind. Die Adressen sind durch dünne schwarze Linien voneinander getrennt und nummeriert. Das Grundriss-Polygon des Gebäudes ist dick schwarz umrandet. Die erkannten Konturen sind schraffiert dargestellt. Die $IoUs$ zwischen den erkannten Konturen und dem Grundriss-Polygon sind ganz rechts aufgelistet.

Das Grundriss-Polygon wurde nacheinander in den Jahrgängen der Karten gesucht. Hierbei wurde mit der ältesten Karte mit dem Jahrgang 1925-1930 begonnen. Sobald IoU_{min} für ein gefundenes Polygon überschritten wurde, wurde das Baujahr des Gebäudes entsprechend des Jahrganges der Karte festgelegt. Hierbei wurde für Karten, die einen Zeitraum repräsentieren, die untere Grenze dieses Zeitraumes als Baujahr angenommen. Diese Methodik wurde gewählt, da sich gezeigt hatte,

dass sie im Vergleich zum Zuweisen der oberen Grenze oder der Mitte eines Zeitraumes als Baujahr die besten Ergebnisse erzielte. Für Karten, die nur für ein Jahr gelten, wurde dieses Jahr als Baujahr definiert. Wenn IoU_{min} zum Beispiel in der Karte „Jahrgang 1970-1980“ überschritten wird, ist das geschätzte Baujahr 1970. Ist dies hingegen für die Karte „Jahrgang 2005“ der Fall, wird das Baujahr auf 2005 geschätzt. Falls für ein Grundriss-Polygon in keiner Karte IoU_{min} überschritten wurde, wurde das Baujahr auf 0 gesetzt, um es später leichter identifizieren zu können. Für den Fall, dass das Gebiet, auf dem ein Gebäude steht, auf einer Karte nicht dargestellt ist, das Gebäude auf der darauffolgenden Karte aber sichtbar ist, wurde das Baujahr auf die untere Grenze dieser Karte festgelegt. Demnach wurde das Baujahr eines Gebäudes, das auf der Karte des Jahrgangs 1930-1940 nicht sichtbar ist, auf der Karte des Jahrgangs 1950-1960 aber schon, auf 1950 festgelegt, wenn die Karte „Jahrgang 1940-1950“ das entsprechende Gebiet nicht abbildet. Wenn davon ausgegangen wird, dass das Gebäude nicht vor 1935 erbaut wurde, wenn es in der entsprechenden Karte nicht auftaucht, beträgt der maximale Fehler bei der Baujahresabschätzung 15 Jahre, was im Hinblick auf die zeitliche Auflösung der Karten als hinnehmbar klein angesehen wird.

Auf Basis des geschätzten Baujahres wurde den Gebäuden eine IWU-Baualtersklasse zugewiesen, die im Folgenden als „geschätzte Baualtersklasse“ bezeichnet wird. Außerdem erfolgte eine Einteilung anhand der tatsächlichen Baujahre, die entsprechende Baualtersklasse wird im Folgenden als „tatsächliche Baualtersklasse“ bezeichnet. Zur Einteilung der Gebäude in die Baualtersklassen wurden die in Abbildung 2 definierten Grenzen der IWU-Baualtersklassen verwendet.

Die in diesem Kapitel beschriebene Methodik wurde an einem Testdatensatz, der im folgenden Kapitel näher beschrieben wird, getestet. Zur Bewertung der Methodik wurde zum einen die Größe mAP verwendet. Mit ihr wurde die Genauigkeit bezüglich der Einteilung der Gebäude in die IWU-Baualtersklassen beurteilt, da sie zur Bewertung von Klassifizierungsaufgaben geeignet ist. (siehe Kapitel 2.4.3). Über den Vergleich der geschätzten und tatsächlichen Baualtersklasse jedes Test-Gebäudes wurden Präzision und Recall mit Hilfe der Gleichungen [2.15] und [2.16] berechnet. Zur Bewertung der Genauigkeit wurden daraus AP und mAP entsprechend den Gleichungen [2.17] und [2.18] berechnet. Die Anzahl der Klassen N_{class} entsprach hierbei der Anzahl der IWU-Baualtersklassen.

Die zweite Bewertungsgröße ist die Abweichung ΔB_j der geschätzten von den tatsächlichen Baujahren. Sie wurde berechnet mit:

$$\Delta BJ = BJ - BJ_{\text{geschätzt}} \quad [3.22]$$

Mit dem tatsächlichen Baujahr BJ und dem geschätzten Baujahr $BJ_{\text{geschätzt}}$. Die Abweichungen wurden über alle Gebäude des Testdatensatzes gemittelt und so die mittlere Abweichung ΔBJ_{mean} bestimmt. Hierzu wurde Gleichung [3.16] verwendet, wobei $\Delta R_{\text{Gebäude}}$ durch ΔBJ ersetzt wurde und $N_{\text{Gebäude}}$ der Anzahl der Gebäude des Testdatensatzes entsprach. Die Standardabweichung σ wurde entsprechend Gleichung [3.17] berechnet. Wie bei Gleichung [3.16] wurde auch hier $\Delta R_{\text{Gebäude}}$ durch ΔBJ ersetzt und $N_{\text{Gebäude}}$ entsprach der Anzahl der Gebäude des Testdatensatzes. Zusätzlich wurde $\Delta R_{\text{Gebäude, mean}}$ durch ΔBJ_{mean} ersetzt. Mithilfe von ΔBJ_{mean} und σ kann beurteilt werden, wie genau die Baujahre der Gebäude geschätzt werden. Neben dieser quantitativen Bewertung der Methodik wurde ebenfalls eine qualitative Bewertung vorgenommen, indem das Baujahr und die Baualtersklasse der 6 Gebäude des Hamburg Image Datasets aus Tabelle 4 bestimmt und mit den tatsächlichen Werten verglichen wurden.

Um den Einfluss der Reihenfolge der Karten auf die Bestimmung der Baualtersklasse zu überprüfen, wurde diese in einem Test geändert, sodass nicht mit der ältesten, sondern mit der jüngsten Karte begonnen wurde. In diesem Fall musste IoU_{min} nicht über-, sondern unterschritten werden, damit einem Gebäude ein geschätztes Baujahr zugewiesen werden kann. Es zeigte sich, dass die geschätzten Baualtersklassen für die Reihenfolge von der neuesten zur ältesten Karte ein geringeres mAP aufwiesen. Daher wurde für die weitere Untersuchung die Reihenfolge der Karten wie oben beschrieben von alt nach neu festgelegt.

3.2.3 Der Testdatensatz

Bei den tatsächlichen Baujahren handelt es sich in dieser Masterarbeit nicht um die im ALKIS angegebenen Werte, da diese, wie in Kapitel 2.2.1 bereits erwähnt, Abweichungen zu anderen Quellen aufweisen. Diese wurden beim stichprobenhaften Vergleich der im ALKIS angegebenen Baujahre mit den historischen Karten Hamburgs deutlich. So ist beispielsweise der blau markierte Grundriss des Gebäudes Weimarer Straße 43 aus Tabelle 4 in der Karte „Jahrgang 1940-1950“ in Abbildung 28 eindeutig zu erkennen, in der ALKIS-Datenbank ist das Baujahr des Gebäudes allerdings mit 1974 angegeben.

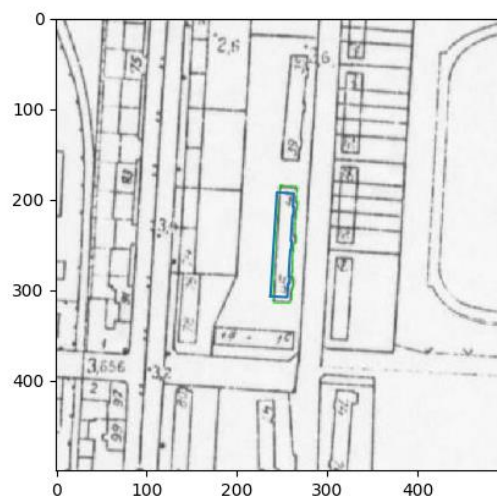


Abbildung 28: Ausschnitt der Karte „Jahrgang 1940-1950“ mit dem blau eingefärbten Grundriss des Gebäudes Weimarer Straße 43.

Bei einer Ortsbegehung wurde an dem Gebäude eine Plakette entdeckt, die Auskunft darüber gibt, dass das Gebäude 1945 zerstört und 1949 wieder aufgebaut wurde (siehe Abbildung 56 im Anhang E). Das Baujahr des Gebäudes liegt laut dem Denkmalschutzamt in den 1930er-Jahren [18]. Die Ursache für die Abweichung des im ALKIS aufgeführten Baujahres zu dem auf der Plakette vermerkten Jahr des Wiederaufbaus könnte die Zerstörung des Gebäudes während der Sturmflut 1962 sein. Eine Karte, die das Gebiet der Sturmflut sowie die Positionen der 6 Gebäude des Hamburg Image Datasets abbildet, ist im Anhang E in Abbildung 57 angefügt. Alle 6 Gebäude befinden sich auf dem Gebiet, das 1962 überschwemmt wurde. Nach Einschätzung des Denkmalschutzamtes Hamburg kann eine Zerstörung der Gebäude durch die Sturmflut, die einen Wiederaufbau in den 1970ern hätte zur Folge haben können, jedoch ausgeschlossen werden [18]. Daher bleibt nur die Vermutung, dass es sich bei dem abweichenden Baujahr um einen Fehler im ALKIS handelt. Um eine verlässlichere Validierungsgrundlage zu erhalten, wurde deshalb im Rahmen dieser Arbeit ein Datensatz erstellt, der die Baujahre von 247 Gebäuden enthält. Hierbei wurde auf Vorarbeit aus dem Institut, an dem diese Masterarbeit verfasst wurde, zurückgegriffen. Als Quelle für die ermittelten Baujahre dienten zum einen an den Gebäuden angebrachte Plaketten, die, wie im Fall der Weimarer Straße 43, Auskunft über Zerstörung und Wiederaufbau oder über das Baujahr eines Gebäudes geben. Zum anderen wurde auf die Internetseiten der Baugenossenschaften Bauverein Reiherstieg [5], Baugenossenschaft der Buchdrucker [2], Baugenossenschaft dhu [3] und Baugenossenschaft Süderelbe [4] zurückgegriffen, auf denen die Baujahre der Gebäude der jeweiligen Genossenschaft angegeben sind. Die Verteilung der Gebäude des Datensatzes im Hamburger Stadtgebiet ist in Abbildung 29 dargestellt. Die Gebäude sind

durch blaue Punkte markiert und im gesamten Stadtgebiet verteilt. Eine dichte Konzentration an Gebäuden ist in den Stadtteilen Sankt Pauli, Wilhelmsburg und Hamm zu erkennen. Die Auswahl der Gebäude, die in den Datensatz aufgenommen wurden, erfolgte willkürlich. 5 der 6 Gebäude des Hamburg Image Datasets aus Tabelle 4 sind im Datensatz enthalten. Das sechste Gebäude mit der Adresse Mengestraße 6, für das keine Informationen über Plaketten oder Baugenossenschaften eingeholt werden konnten, wurde als baugleich mit dem Nachbargebäude in der Hermann-Westphal-Straße 2a-c angenommen, welches im Datensatz enthalten ist. Ein Luftbild beider Gebäude ist im Anhang E in Abbildung 58 abgebildet.

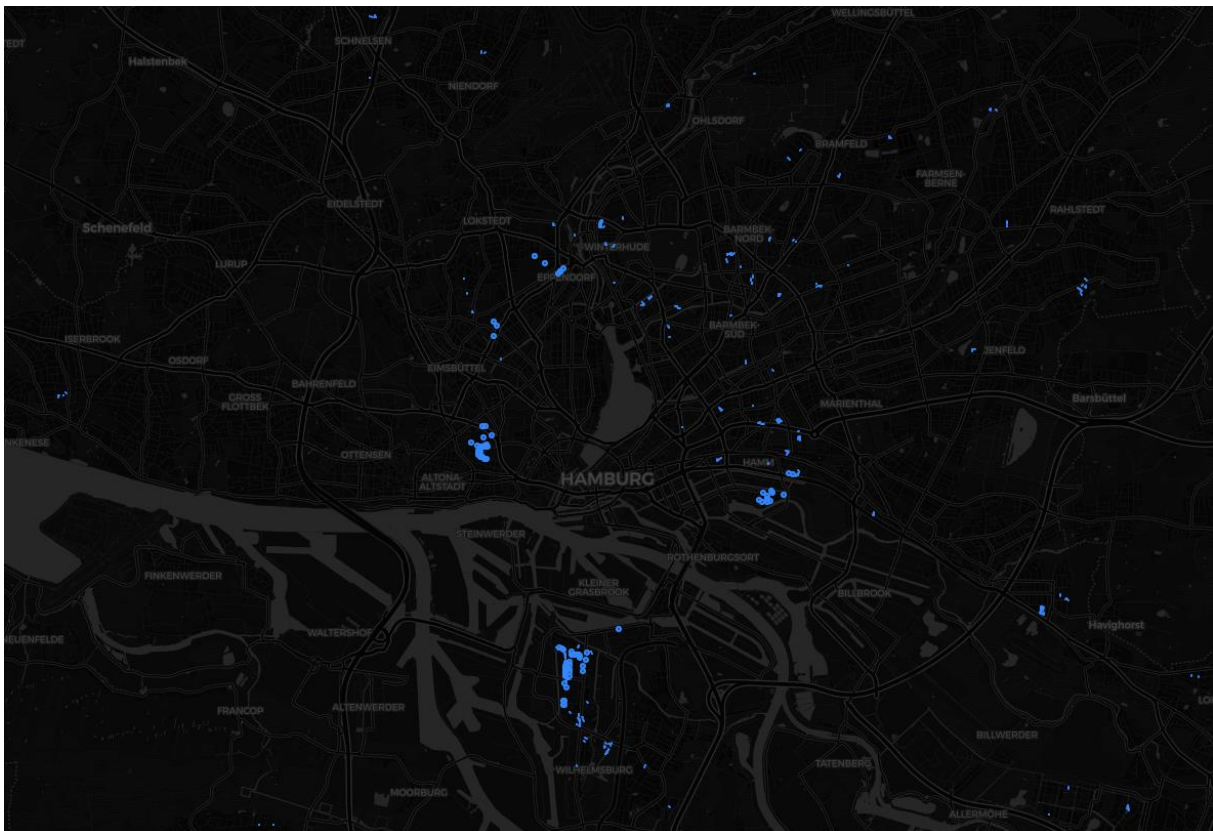


Abbildung 29: Verteilung der Gebäude des Testdatensatzes im Hamburger Stadtgebiet.

In Abbildung 30 ist die Verteilung der Baujahre im Testdatensatz dargestellt. Die Baujahre wurden hierzu in die IWU-Baualtersklassen (siehe Abbildung 2) eingeordnet. Im Datensatz ist kein Gebäude älter als 1860, demnach werden der Klasse A keine Gebäude zugeordnet. Allen anderen Klassen können Gebäude zugeordnet werden, wobei die Klasse D (Baujahr zwischen 1949 und 1957) mit Abstand die meisten Gebäude aufweist. Dies liegt daran, dass das Baujahr für 89 der 247 Gebäude anhand der Plaketten an den Gebäuden bestimmt wurde. Die meisten dieser Plaketten wurden an Gebäuden angebracht, die im zweiten Weltkrieg zerstört und dann im Zeitraum der Baualtersklasse D wieder aufgebaut wurden.

Verteilung der Baujahre im Testdatensatz bezogen auf die IWU-Baualtersklassen

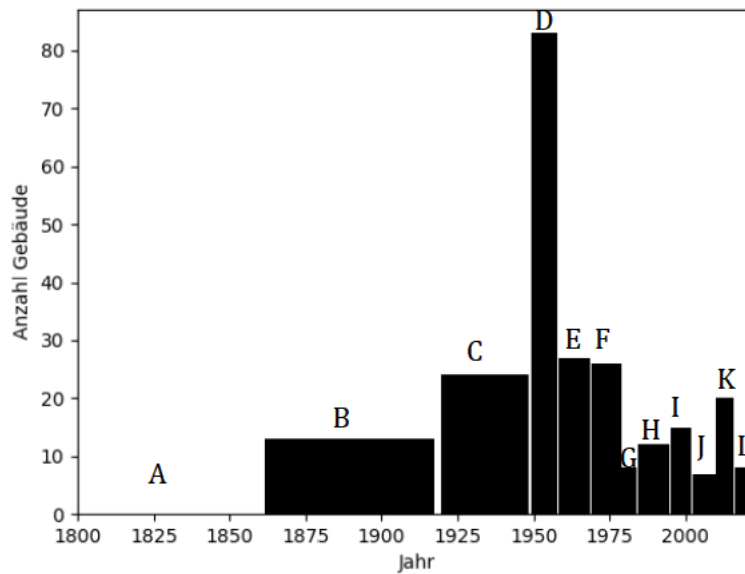


Abbildung 30: Verteilung der Baujahre der Gebäude im Testdatensatz bezogen auf die IWU-Baualtersklassen, die durch die Buchstaben über den Balken repräsentiert werden.

Die Diskrepanz zwischen den Baujahren des ALKIS-Datensatzes und den händisch ermittelten Baujahren des beschriebenen Testdatensatzes ist in Abbildung 31 sichtbar. Eine negative Abweichung bedeutet, dass das Gebäude im ALKIS-Datensatz älter ist als im Testdatensatz. Es wird deutlich, dass eine große Anzahl an Gebäuden eine Abweichung von wenigen oder gar keinen Jahren zwischen ALKIS und dem Testdatensatz aufweist. Nichtsdestotrotz werden Abweichungen von mehr als 25 Jahren vor allem im positiven Bereich sichtbar. Die größte positive Abweichung beträgt 88 Jahre. Im Durchschnitt sind Gebäude im Testdatensatz 9,5 Jahre älter als im ALKIS-Datensatz.

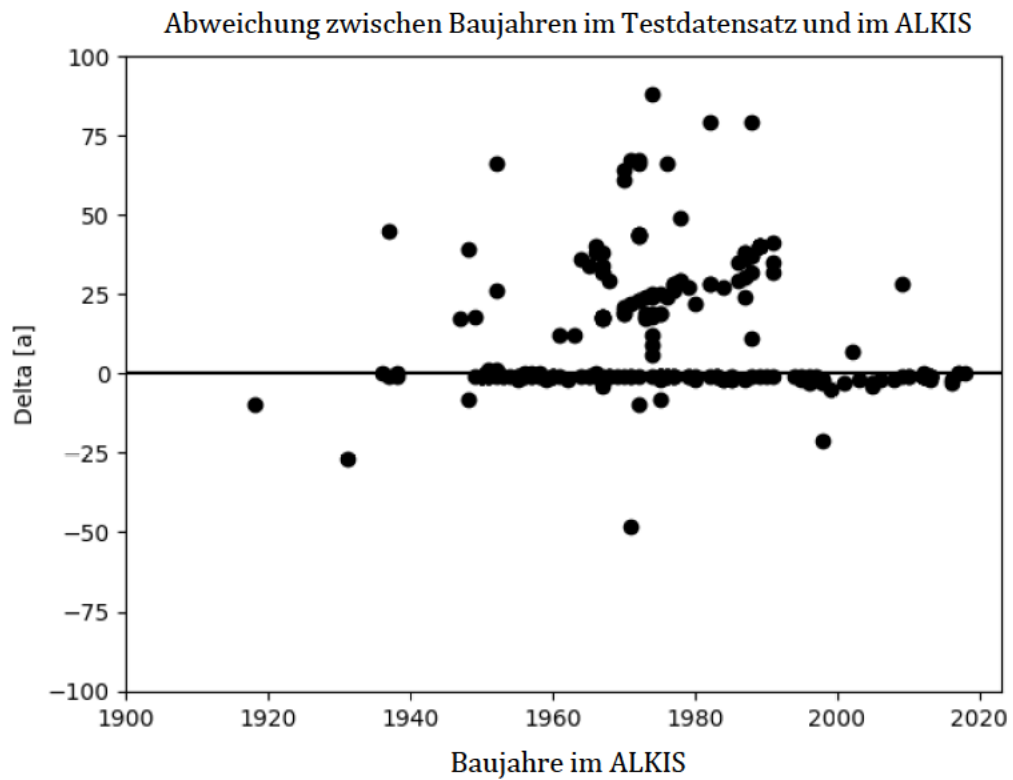


Abbildung 31: Abweichung (Delta) zwischen den Baujahren des Testdatensatzes und den Baujahren des ALKIS-Datensatzes.
Ein negatives Delta bedeutet, dass das Gebäude im ALKIS-Datensatz älter ist als im Testdatensatz.

4 Ergebnisse

Anhand der Testdatensätze wurde die Methodik, sowohl zur Bestimmung von $R_{\text{Gebäude}}$ als auch zur Bestimmung der Baualtersklasse eines Gebäudes, bewertet. Hierbei wurden auch die ermittelten Fensterflächen und Baujahre berücksichtigt, da diese zur Bestimmung von $R_{\text{Gebäude}}$ und Baualtersklasse notwendig sind.

4.1 Bestimmung des Verhältnisses der Fenster- zur Wandfläche eines Gebäudes

Die Bestimmung des Verhältnisses $R_{\text{Gebäude}}$ der Fenster- zur Wandfläche eines Gebäudes erfolgte auf Basis der mithilfe des Faster R-CNN-Modells erkannten Fenster. Daher werden im Folgenden zunächst die Ergebnisse des Trainings des ML-Modells beschrieben, bevor auf die Ergebnisse bezüglich $R_{\text{Gebäude}}$ eingegangen wird.

4.1.1 Ergebnisse des ML-Trainings

Das Faster R-CNN-Modell wurde mit den in Tabelle 2 spezifizierten Hyperparametern trainiert. Hierbei wurden die Parameter „Maximale Anzahl der Epochen“ und „Größe des Validierungsdatensatzes“ variiert. Insgesamt wurden so 12 Faster R-CNN-Modelle mit unterschiedlichen Hyperparametern trainiert. Der Verlauf von Trainings- und Validierungs-Loss für alle trainierten Modelle ist in Abbildung 32 dargestellt. Hierzu ist anzumerken, dass auf der x-Achse nicht die Epoche, sondern die $\frac{\text{Epoche}}{2}$ aufgeführt ist. Der Wert 25 auf der x-Achse steht dementsprechend für die 50. Epoche.

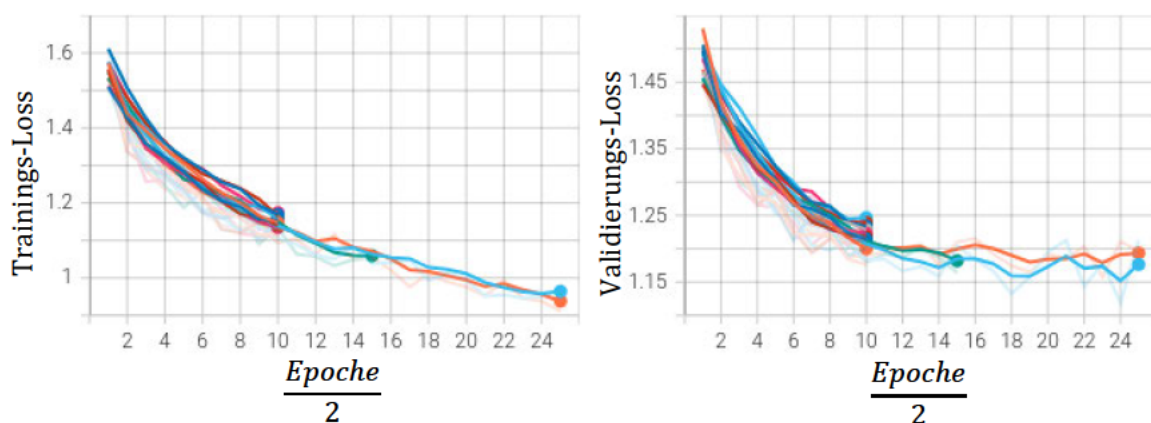


Abbildung 32: Verlauf von Trainings- und Validierungs-Loss für 12 Faster R-CNN-Modelle, die mit unterschiedlichen Hyperparametern trainiert wurden

In Abbildung 32 wird deutlich, dass Trainings- und Validierungs-Loss bis Epoche 20 exponentiell abfallen. Während der Trainings-Loss ab Epoche 20 fast kontinuierlich weiter sinkt, bleibt der

Validierungs-Loss in diesem Bereich ungefähr konstant. Von den 12 trainierten Modellen wurden nur 3 länger als 20 Epochen trainiert. Dies spricht dafür, dass eine höhere Anzahl an Epochen nicht zu einem besseren Modell, sondern zu Overfitting führt (vergleiche hierzu Abbildung 6). Um das für Hamburger Gebäude am besten geeignete Modell zu ermitteln, wurde für alle 12 trainierten Modelle *mAP* berechnet. In Tabelle 6 sind die Ergebnisse aufgeführt.

Tabelle 6: Berechnete *mAP* für 12 Faster R-CNN-Modelle, die mit unterschiedlichen Hyperparametern trainiert wurden.

Maximale Anzahl der Epochen	Größe des Validierungsdatensatzes	<i>mAP</i>
20	0,1	0,360
30	0,1	0,381
40	0,1	0,373
50	0,1	0,371
20	0,2	0,386
30	0,2	0,366
40	0,2	0,319
50	0,2	0,323
20	0,3	0,319
30	0,3	0,343
40	0,3	0,275
50	0,3	0,310

In Tabelle 6 wird deutlich, dass *mAP* tendenziell kleiner wird, je größer der Validierungsdatensatz ist. Dies kann daran liegen, dass ein kleiner Validierungsdatensatz mehr Daten für das Training des Modells übriglässt. Bezüglich der Anzahl der Epochen wird keine Tendenz deutlich, die sich über alle Größen des Validierungsdatensatzes abzeichnet. Das beste Modell wurde in Tabelle 6 fett hervorgehoben. Es hat eine maximale Anzahl an Epochen von 20 und der Validierungsdatensatz bestand aus 20 % des CMP-Datensatzes, was mit der in Kapitel 2.4.5 beschriebenen üblichen Größe des Validierungsdatensatzes übereinstimmt. Mit diesen Hyperparametern wurde ein *mAP* von 0,386 erreicht. Zusätzlich zu *mAP* wurde der Wert *F1* für dieses Modell für einen Score-Wert von 0,8 berechnet, um einen Vergleich mit der Arbeit von Wang [92] zu ermöglichen. Er beträgt 0,493 für ein *IoU* von 0,75. Da dieses Faster R-CNN-Modell das beste *mAP* aufweist, wurde es für alle im Folgenden beschriebenen Schritte verwendet.

Um zu prüfen, ob das Potential des ausgewählten Modells ausgeschöpft wurde, wurde der Verlauf des Trainings- und des Validierungs-Losses gegeneinander aufgetragen. In Abbildung 33 sind die beiden Verläufe dargestellt. Wie auch in Abbildung 32 ist die Einheit der x-Achse $\frac{\text{Epoche}}{2}$.

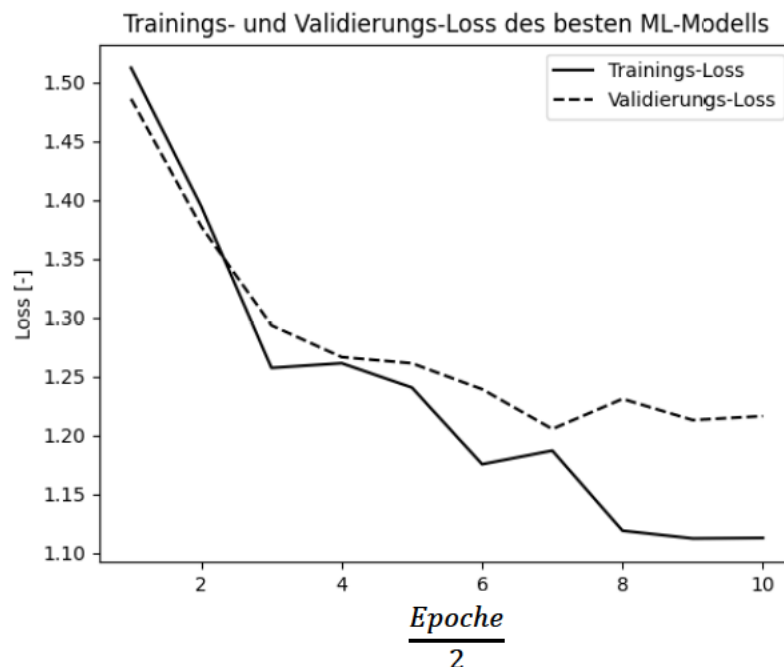


Abbildung 33: Verlauf von Trainings-Loss und Validierungs-Loss für das beste ML-Modell.

Der Trainings-Loss verläuft exponentiell und nähert sich ab Epoche 16 einem Wert von ca. 1,1 an. Der Validierungs-Loss hat einen ähnlichen Verlauf, allerdings verläuft er ungefähr ab Epoche 4 oberhalb des Trainings-Losses und endet bei ca. 1,2. Dies deutet darauf hin, dass das Modell zu Overfitting neigt. Trotz der Abweichung zwischen Trainings- und Validierungs-Loss in der 20. Epoche weisen beide Kurven zu diesem Zeitpunkt keine großen Veränderungen mehr auf, weshalb davon ausgegangen werden kann, dass ein Training des Modells über die 20. Epoche hinaus keinen Effekt auf die Genauigkeit mehr gehabt hätte.

Die Ergebnisse des trainierten ML-Modells wurden entsprechend den in Kapitel 3.1.2 beschriebenen Methoden nachbearbeitet. Um die Auswirkungen der Nachbearbeitung auf die Genauigkeit der Ergebnisse des ML-Modells bewerten zu können, wurde verglichen, wie sich *mAP* ändert, je nachdem welche der beschriebenen Methoden zur Nachbearbeitung angewendet wurden. Hierbei wurden 5 Fälle unterschieden, die in Tabelle 7 aufgelistet sind. Es muss hierzu angemerkt werden, dass für das Neuausrichten der Fenster eine vorherige Anpassung der Größe notwendig ist und das Hinzufügen von Fenstern wiederum nicht ohne vorherige Neuausrichtung erfolgen kann.

Tabelle 7: Unterschiedliche Nachbearbeitungen, die auf die Ergebnisse des ML-Modells angewendet wurden.

	NMS	Ändern von Breite und Höhe Fenster	Neuausrichten der Fenster	Hinzufügen von Fenstern
Ohne Nachbearbeitung	Nein	Nein	Nein	Nein
Nachbearbeitung 1	Ja	Ja	Ja	Ja
Nachbearbeitung 2	Ja	Ja	Ja	Nein
Nachbearbeitung 3	Ja	Ja	Nein	Nein
Nachbearbeitung 4	Ja	Nein	Nein	Nein

Der Einfluss der Nachbearbeitung wurde nicht nur für einen Score-Grenzwert von 0,05 untersucht, auf dessen Basis das ML-Modell ausgewählt wurde, sondern auch für die in Kapitel 3.1.6 beschriebenen Grenzwerte 0,25, 0,5 und 0,8, da davon ausgegangen werden kann, dass sich die Nachbearbeitung unterschiedlich stark auf das Ergebnis auswirkt, je nachdem, wie hoch der Score-Grenzwert gewählt wurde. Die *mAP*-Werte, die für die Kombination aus den Score-Grenzwerten und Nachbearbeitungs-Varianten berechnet wurden, sind in Tabelle 8 aufgelistet.

Tabelle 8: Berechnete *mAP* für unterschiedliche Score-Grenzwerte und Nachbearbeitungs-Varianten (siehe Tabelle 7).

Score-Grenzwert	Ohne Nachbearbeitung	Nachbearbeitung 1	Nachbearbeitung 2	Nachbearbeitung 3	Nachbearbeitung 4
0,05	0,386	0,264	0,267	0,263	0,381
0,25	0,370	0,277	0,282	0,277	0,365
0,5	0,341	0,312	0,311	0,306	0,337
0,8	0,268	0,247	0,247	0,245	0,266

Hier zeigt sich, dass eine Nachbearbeitung der Ergebnisse des ML-Modells *mAP* in keinem Fall verbessert. Einzig mit der Nachbearbeitungs-Variante 5, bei der nur NMS angewendet wird, wird für alle Score-Grenzwerte ein ähnliches *mAP* erzielt wie komplett ohne Nachbearbeitung. Ein zusätzliches Anwenden mindestens einer der drei anderen Methoden führt zu einer Abnahme des

mAP. Hierbei fällt die Abnahme je nach Größe des Score-Grenzwertes unterschiedlich groß aus. Für den niedrigsten Score-Grenzwert fällt *mAP* um ca. 0,12, für den größten nur noch um ca. 0,02. Neben den Auswirkungen der Nachbearbeitung auf das *mAP* wird in Tabelle 8 auch der in Kapitel 3.1.1 beschriebene Einfluss des Score-Grenzwertes auf das *mAP* deutlich. Mit dem niedrigen Score-Grenzwert von 0,05 beträgt *mAP* 0,386, mit zunehmendem Score-Grenzwert wird *mAP* jedoch immer kleiner, bis es bei einem Score-Grenzwert von 0,8 nur noch 0,268 beträgt.

Beispiele für die Ergebnisse der Nachbearbeitung sind in Abbildung 34 für zwei unterschiedliche Score-Grenzwerte (0,05 und 0,5) dargestellt.

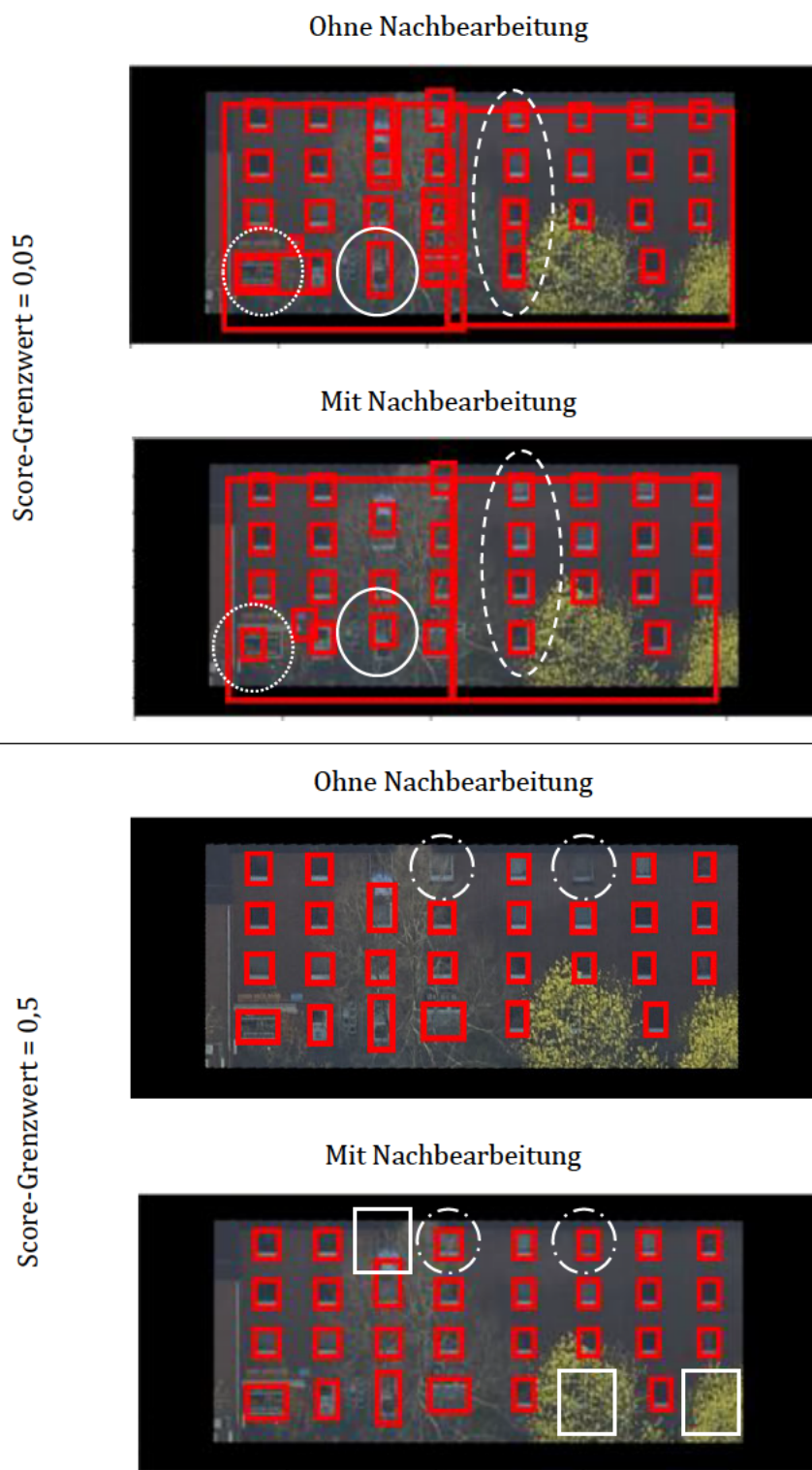


Abbildung 34: Vergleich der Ergebnisse des ML-Modells ohne Nachbearbeitung mit den nachbearbeiteten Ergebnissen für zwei unterschiedliche Score-Grenzwerte. Abgebildet ist eine Fassade des Gebäudes Veringstraße 149.

Im oberen Teil der Abbildung ist die Fassade des Gebäudes vor und nach der Nachbearbeitung für einen Score-Grenzwert von 0,05 zu sehen. Viele sich überschneidende Fenster, die das ML-Modell erkannt hatte, wurden durch die Nachbearbeitung entfernt. Trotzdem bleiben Überschneidungen

auch nach der Nachbearbeitung bestehen, da die beiden großen Rahmen, die die Fassade des Gebäudes fälschlicherweise als Fenster einordnen, nicht entfernt wurden. Die Größe der Fenster wurde zum Teil angepasst, zum Beispiel für das Fenster im weißen Kreis. Durch die Nachbearbeitung wurde es auf die korrekte Größe reduziert. Im gepunkteten weißen Kreis fällt hingegen ein Fenster auf, dass vom ML-Modell in der richtigen Größe erkannt, durch die Nachbearbeitung jedoch auf die falsche Größe getrimmt wurde. Eine Neuausrichtung der Fenster erfolgte ebenfalls, wie im gestrichelten weißen Kreis zu erkennen ist. Während das ML-Modell für den niedrigeren Score-Grenzwert von 0,05 viele überschneidende Fenster vorhersagte, ist dies für den Score-Grenzwert von 0,5 nicht der Fall, wie im unteren Teil der Abbildung 34 zu sehen ist. Eine Anpassung der Größe oder Neuausrichtung der Fenster ist ebenfalls in der Abbildung nicht zu erkennen. Die weißen Kreise mit Strichpunkt-Linie heben jedoch zwei Fenster hervor, die vom ML-Modell nicht erkannt, durch die Nachbearbeitung jedoch hinzugefügt wurden. In den weißen Rechtecken befinden sich hingegen Fenster, die auch durch die Nachbearbeitung nicht ergänzt wurden. Dies liegt zum einen daran, dass Türen und Schaufenster im Erdgeschoss als Fenster erkannt wurden. Die Anzahl unterschiedlicher Fenstergrößen in der Fassade ($K_{Cluster}$) wurde dadurch auf 4 festgelegt, obwohl die optische Auswertung der Fassade nur zwei unterschiedliche Fenstergrößen nahelegt. Dies wird in Abbildung 34 deutlich, die die Fenster der Fassade darstellt, wie sie im Hamburg Image Dataset gelabelt sind. Da $K_{Cluster}$ einen entscheidenden Einfluss auf die Größenanpassung, Neuausrichtung und Ergänzung von Fenstern hat (siehe Kapitel 3.1.2), wird durch das Überschätzen dieses Wertes die Wahrscheinlichkeit der Verbesserung des Ergebnisses durch Nachbearbeitung geringer. Auch die Fenster, die von den Bäumen verdeckt sind, werden in Abbildung 34 nicht erkannt. Ursache hierfür ist, dass die Fenster der entsprechenden Reihe eine andere Größe und Anordnung aufweisen als die darüberliegenden, weshalb sie durch die Nachbearbeitung nicht ergänzt werden können.

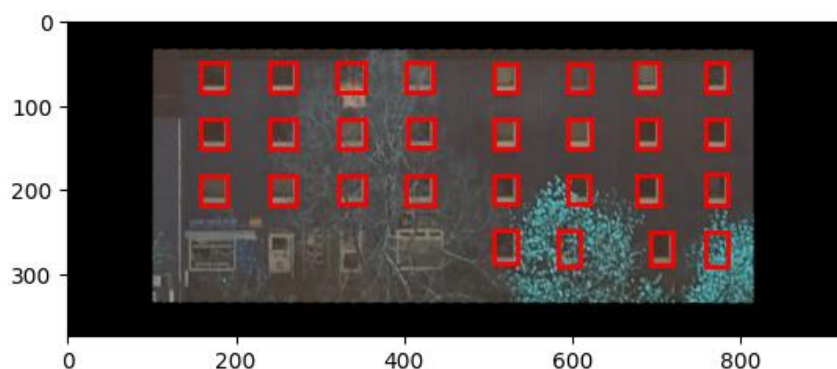


Abbildung 35: Fenster der Fassade aus Abbildung 34, wie sie im Hamburg Image Dataset gelabelt sind.

4.1.2 Validierung der Ergebnisse

Aus den Ergebnissen des ML-Modells sowie dem 3D-Stadtmodell wurde die Fenster- und Wandfläche sowie deren Verhältnis $R_{\text{Gebäude}}$ für alle Gebäude des Hamburg Image Datasets entsprechend den Gleichungen [3.11] bis [3.13] mit variablem Score-Grenzwert berechnet. Das Verhältnis der Fenster- zur Wandfläche $R_{\text{Gebäude}}$ wurde mit den Werten für $R_{\text{Gebäude, IWU}}$ verglichen. Als Referenzwerte für den Vergleich diente $R_{\text{Gebäude, HI}}$. Es wurde die über alle Gebäude des Testdatensatzes gemittelte Abweichung $\Delta R_{\text{Gebäude, mean}}$ von $R_{\text{Gebäude, HI}}$ zu $R_{\text{Gebäude}}$ und $R_{\text{Gebäude, IWU}}$ berechnet. Auch wenn die Ergebnisse aus Tabelle 8 dafürsprechen, dass eine Nachbearbeitung der Ergebnisse des ML-Modells keine Verbesserung bewirkt, wurde $R_{\text{Gebäude}}$ mit und ohne Nachbearbeitung berechnet und die Ergebnisse verglichen. Die Nachbearbeitung erfolgte mit der Variante 1 aus Tabelle 7, sodass der Einfluss aller Nachbearbeitungsmethoden untersucht werden konnte. In Abbildung 36 sind die berechneten Werte für $\Delta R_{\text{Gebäude, mean}}$ für unterschiedliche Score-Grenzwerte und mit und ohne Nachbearbeitung dargestellt. Die dazugehörigen Werte inklusive der mit Gleichung [3.17] berechneten Standardabweichung σ sind in Tabelle 14 im Anhang F aufgelistet.

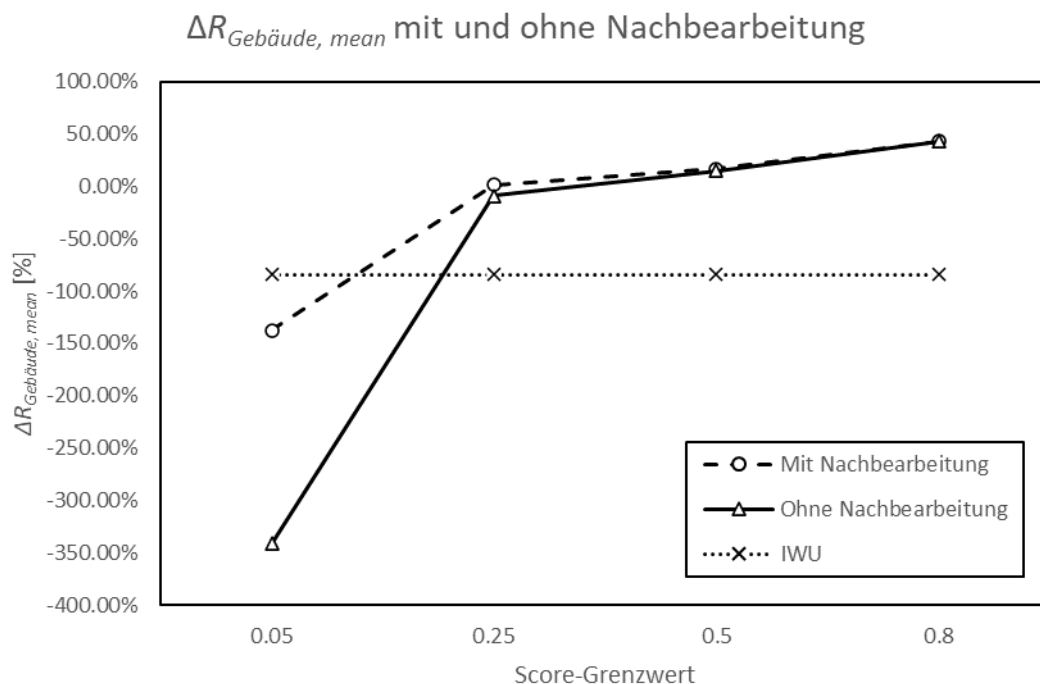


Abbildung 36: Abweichung $\Delta R_{\text{Gebäude, mean}}$ auf Basis der vom ML-Modell erkannten Fensterflächen mit und ohne Nachbearbeitung sowie auf Basis der IWU-Gebäudetypologie.

Das $\Delta R_{\text{Gebäude, mean}}$ auf Basis der IWU-Typologie ist für alle Score-Grenzwerte konstant bei -84,04 %, da es nicht auf den Ergebnissen des ML-Modells beruht und damit nicht vom Score-Grenzwert abhängig ist. Demnach wird das Verhältnis der Fenster- zur Wandfläche mit den Werten der IWU-

Gebäudetypologie im Durchschnitt um 84,04 % größer eingeschätzt als das manuell gemessene $R_{\text{Gebäude, HI}}$. Die Standardabweichung ist hier mit 71,07 % hoch. Für einen Score-Grenzwert von 0,05 weisen auch die mit den Ergebnissen des ML-Modells berechneten $\Delta R_{\text{Gebäude, mean}}$ negative Werte auf, die mit -137,8 % für die nachbearbeiteten und -341,24 % für die nicht nachbearbeiteten Ergebnisse noch größer sind als das $\Delta R_{\text{Gebäude, mean}}$ für die IWU-Werte und entsprechend größere Standardabweichungen aufweisen. Für einen Score-Grenzwert von 0,25 nähern sich die beiden Kurven für $\Delta R_{\text{Gebäude, mean}}$ in Abbildung 36 an. Das $\Delta R_{\text{Gebäude, mean}}$ der nachbearbeiteten Ergebnisse beträgt hier 1,21 % bei einem σ von 20,24 %. Für die nicht nachbearbeiteten Ergebnisse fällt die Abweichung mit -8,83 % und einem σ von 21,04 % etwas stärker aus. Für die Score-Grenzwerte 0,5 und 0,8 nimmt $\Delta R_{\text{Gebäude, mean}}$ für die nachbearbeiteten Ergebnisse weiter zu auf 16,89 %, beziehungsweise 43,18 %. Die Standardabweichungen betragen 21,27 % und 16,07 %. Ähnlich verhält es sich für die nicht nachbearbeiteten Ergebnisse, für die $\Delta R_{\text{Gebäude, mean}}$ auf 14,64 % und 43,13 % steigt, während σ auf 21,08 %, beziehungsweise 15,37 % sinkt. Im Gegensatz zu den Ergebnissen aus Tabelle 8 zeigt sich in Abbildung 36, dass die Nachbearbeitung der Ergebnisse des ML-Modells diese durchaus verbessern kann, wenn die Bewertungsgröße nicht mAP , sondern $\Delta R_{\text{Gebäude, mean}}$ ist. Die größten Verbesserungen treten hierbei bei geringen Score-Grenzwerten auf. Für einen Score-Grenzwert von 0,8 hingegen verändert die Nachbearbeitung das Ergebnis so gut wie nicht mehr. Die geringste mittlere Abweichung $\Delta R_{\text{Gebäude, mean}}$ beträgt 1,21 % für nachbearbeitete Ergebnisse des ML-Modells und einen Score-Grenzwert von 0,25. Das Verhältnis der Fenster- zur Wandfläche für alle Gebäude des Hamburg Image Datasets ist deshalb in Abbildung 37 nur für nachbearbeitete Ergebnisse dargestellt. Neben $R_{\text{Gebäude}}$ für unterschiedliche Score-Grenzwerte sind in der Abbildung $R_{\text{Gebäude, HI}}$ sowie $R_{\text{Gebäude, IWU}}$ dargestellt.

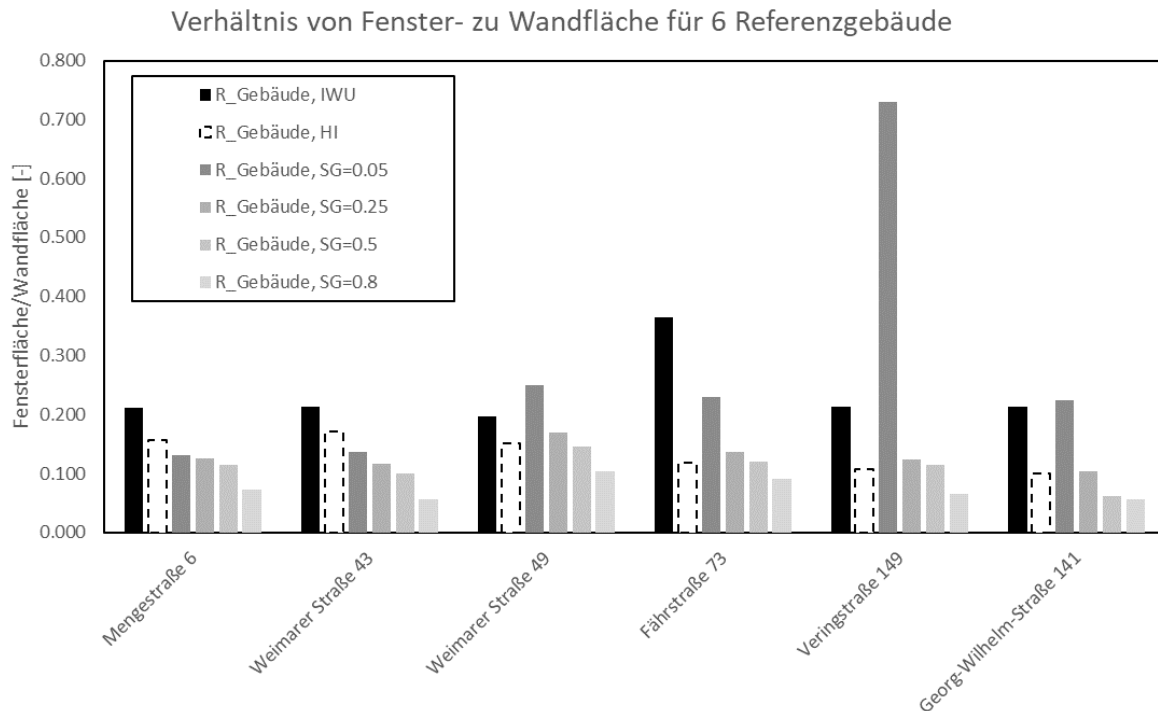


Abbildung 37: Verhältnis der Fenster- zur Wandfläche für die 6 Gebäude des Hamburg Image Datasets: Manuell ermittelt ($R_{\text{Gebäude, HI}}$, gestrichelt), auf Basis der nachbearbeiteten Ergebnisse des ML-Modells ($R_{\text{Gebäude}}$, graustufen) und auf Basis der IWU-Gebäudetypologie ($R_{\text{Gebäude, IWU}}$, schwarz). SG steht für den Score-Grenzwert, welcher für die Bestimmung der Fensterfläche variiert wurde.

Bezüglich der Ergebnisse auf Basis der IWU-Typologie zeigt sich in Abbildung 37, dass diese $R_{\text{Gebäude, HI}}$ für alle Gebäude überschätzen. Die Werte für $R_{\text{Gebäude}}$ nehmen mit zunehmendem Score-Grenzwert ab. Für einen Score-Grenzwert von 0,05 liegt $R_{\text{Gebäude}}$ für die linken beiden Gebäude in Abbildung 37 im Bereich von $R_{\text{Gebäude, HI}}$, für alle anderen Gebäude wird $R_{\text{Gebäude, HI}}$ überschätzt. Es fällt hier vor allem das Ergebnis für $R_{\text{Gebäude}}$ für die Veringstraße 149 auf, das deutlich über $R_{\text{Gebäude, HI}}$ und dem IWU-Wert liegt. Für einen Score-Grenzwert von 0,8 hingegen unterschätzt $R_{\text{Gebäude}}$ $R_{\text{Gebäude, HI}}$ für alle Gebäude.

Um die Ergebnisse aus Abbildung 37 besser nachvollziehen zu können, wurden neben $R_{\text{Gebäude}}$ auch die gesamte Wandfläche $\sum A_{\text{Wand}}$ und die gesamte Fensterfläche $\sum \sum A_{\text{Fenster}}$ jedes Gebäudes verglichen (siehe Kapitel 3.1.6). Für diesen Vergleich, der in Abbildung 38 aufgeführt ist, wurden nur die manuell ermittelten Werte aus dem Hamburg Image Dataset und die Ergebnisse des ML-Modells herangezogen. Die Abweichungen zwischen diesen Werten wurden mit Gleichung [3.15] berechnet, wobei für $R_{\text{Gebäude}}$ die vom ML-Modell erkannte Wand- oder Fensterfläche eingesetzt wurde und für $R_{\text{Gebäude, HI}}$ die entsprechende Fläche des Hamburg Image Datasets. So kann zum einen ermittelt werden, inwiefern sich die Wandflächen des 3D-Stadtmodells, die für die Berechnung von $R_{\text{Gebäude}}$ verwendet wurden, von den manuell definierten Wandflächen des

Hamburg Image Datasets unterscheiden. Zum anderen gibt der Vergleich Aufschluss über die Genauigkeit des ML-Modells hinsichtlich der gesamten Fensterfläche eines Gebäudes.

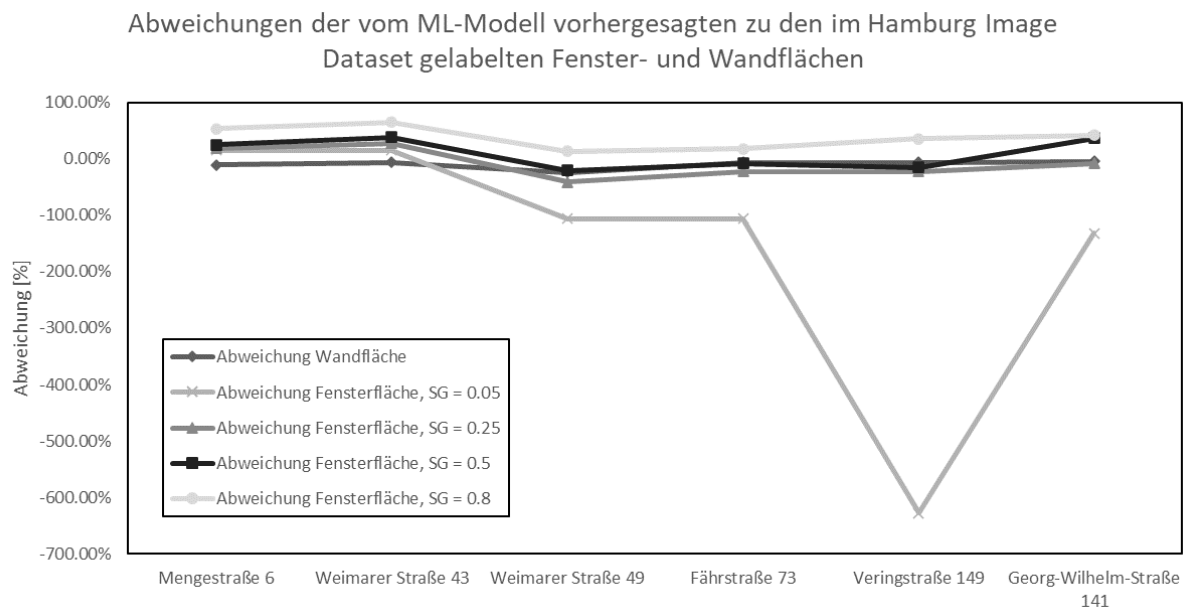


Abbildung 38: Abweichung der auf Basis des 3D-Stadtmodells bestimmten gesamten Wandfläche $\sum A_{Wand}$ und der vom ML-Modell vorhergesagten gesamten Fensterfläche $\sum \sum A_{Fenster}$ jedes Gebäudes zu den auf Basis des Hamburg Image Datasets manuell berechneten Werten. SG steht für den Score-Grenzwert, welcher für die Bestimmung der Fensterfläche variiert wurde.

Bezüglich der Wandflächen wird in Abbildung 38 nur eine geringe Abweichung zu den mit dem Hamburg Image Dataset berechneten Werten deutlich. Die Abweichung ist für alle Gebäude negativ, was bedeutet, dass die Wandflächen im 3D-Stadtmodell größer sind als die entsprechenden Label im Hamburg Image Dataset. Die durchschnittliche Abweichung über alle Gebäude beträgt hier -7,25 %. Für die Abweichung der vom ML-Modell vorhergesagten Fensterflächen zu den Fensterflächen der Gebäude im Hamburg Image Dataset lässt sich beobachten, dass diese für die niedrigeren Score-Grenzwerte 0,05 und 0,25 eher negativ sind, für die größeren Score-Grenzwerte 0,5 und 0,8 eher positiv. Dies spiegelt sich auch in der durchschnittlichen Abweichung über alle Gebäude, die für den Score-Grenzwert 0,05 -154,91 %, für den Score-Grenzwert 0,25 -6,32 %, für den Score-Grenzwert 0,5 10,46 % und für den Score-Grenzwert 0,8 38,74 % beträgt. Hier fällt vor allem der große negative Wert für den Score-Grenzwert von 0,05 auf, der auch in Abbildung 38 durch einen großen negativen Ausschlag der Abweichung für das Gebäude Veringstraße 149 erahnt werden kann. Insgesamt zeigt sich, dass das ML-Modell die Fensterfläche der Gebäude für die Score-Grenzwerte 0,05 und 0,25 über- und für die Score-Grenzwerte 0,5 und 0,8 unterschätzt. Die geringste durchschnittliche Abweichung über alle Gebäude weist hierbei die Vorhersage der Fenster mit einem Score-Grenzwert von 0,25 auf, für den auch $\Delta R_{Gebäude, mean}$ am niedrigsten ist.

Abschließend wurde die Abweichung des vom ML-Modell vorhergesagten zum im Hamburg Image Dataset definierten Verhältnisses der Fenster- zur Wandfläche für jede einzelne Wandfläche in Form von ΔR_{Wand} entsprechend Gleichung [3.14] berechnet, da die solaren Gewinne einer Gebäudefassade stark von der Ausrichtung der Wände abhängen. Auf diese Weise konnte ermittelt werden, ob die Ausrichtung der Wände einen Einfluss darauf hat, wie gut das ML-Modell die Fenster erkennt. Diese Untersuchung wurde nur für nachbearbeitete Ergebnisse mit einem Score-Grenzwert von 0,25 durchgeführt. In Abbildung 39 ist die Abweichung ΔR_{Wand} in Abhängigkeit der Ausrichtung der Wand dargestellt. Die Ausrichtung der Wand wurde entsprechend der in Abbildung 20 vorgestellten Methodik bestimmt.

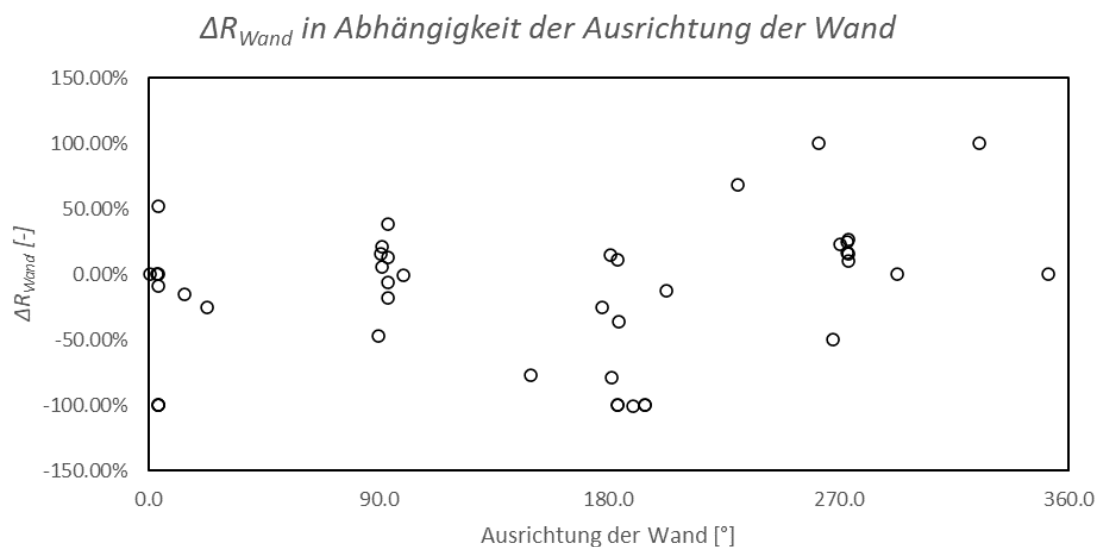


Abbildung 39: Abweichung ΔR_{Wand} des vom ML-Modell vorhergesagten zum im Hamburg Image Dataset definierten Verhältnisses der Fenster- zur Wandfläche für alle Wände des Hamburg Image Datasets in Abhängigkeit der Ausrichtung der Wände. Der Score-Grenzwert beträgt 0,25.

In Abbildung 39 wird deutlich, dass sich die Ausrichtung der Wände an den 4 Himmelsrichtungen orientiert. Für die nach Norden ausgerichteten Wände, die in der Abbildung durch Markierungen im Bereich von 0°, beziehungsweise 360° repräsentiert werden, zeigt sich, dass mehrere Wände Abweichungen um 0 % aufweisen. Gleichzeitig streuen die Werte aber auch stark im positiven und negativen Bereich von ΔR_{Wand} . Anders verhält es sich für die nach Osten ausgerichteten Wände, die sich in der Abbildung um den Wert von 90° konzentrieren. Bezüglich ΔR_{Wand} liegen diese Werte deutlich enger beisammen und weisen eine Tendenz ins Positive auf. Das ML-Modell schätzt demnach das Verhältnis der Fenster- zur Wandfläche für diese Wände als zu klein ein. Die nach Süden ausgerichteten Wände, die in der Abbildung durch eine Ausrichtung im Bereich von 180° dargestellt werden, sind bezüglich ihrer Ausrichtung und ΔR_{Wand} weit gestreut, wobei hier für ΔR_{Wand} eine Tendenz ins Negative erkennbar ist. Dies bedeutet, dass das ML-Modell für diese

Wände ΔR_{Wand} als zu groß einschätzt. Die nach Westen (270°) ausgerichteten Wände verhalten sich ähnlich wie die nach Norden ausgerichteten, allerdings liegen die Werte für ΔR_{Wand} eher im positiven Bereich.

4.1.3 Diskussion der Ergebnisse

Insgesamt kann aus den oben beschriebenen Ergebnissen geschlussfolgert werden, dass trotz eines geringen mAP -Wertes mit einer mittleren Abweichung von 1,21 % eine gute Genauigkeit bezüglich der Bestimmung des Verhältnisses der Fenster- zur Wandfläche erreicht werden kann, wenn der Score-Grenzwert auf 0,25 festgelegt wird. Dieses Ergebnis wird als für die Erstellung von Wärmebedarfsprognosen ausreichend angesehen, da es trotz der hohen Standardabweichung von 20,24 % für die betrachteten Gebäude eine höhere Genauigkeit aufweist, als auf Basis der Werte der IWU-Gebäudetypologie erreicht werden kann. Die Bewertungsgröße mAP wurde nichtsdestotrotz zur Bewertung der Leistung des Faster R-CNN-Modells genutzt, da sie Aufschluss darüber gibt, wie gut einzelne Fenster vom Modell erkannt werden und Vergleiche mit Werten aus der Literatur zulässt (siehe Kapitel 2.1.2). So konnte abgeschätzt werden, inwiefern das Potential des Modells ausgeschöpft wurde. Die 1. Forschungsfrage aus Kapitel 1.1 kann also folgendermaßen beantwortet werden: Das Verhältnis der Fenster- zur Wandfläche eines beliebigen Hamburger Wohngebäudes kann mit Hilfe des Objekterkennungsmodells Faster R-CNN auf Basis des 3D-Statmodells, ALKIS sowie des digitalen Schrägluftbildes von Hamburg mit einer Genauigkeit von $1,21 \% \pm 20,24 \%$ bestimmt werden. Im Folgenden werden die vorgestellten Ergebnisse diskutiert, wobei zunächst erörtert wird, inwiefern das Potential des ML-Modells ausgeschöpft wurde.

In Tabelle 6 weist das am besten trainierte Faster R-CNN-Modell ein mAP von 0,386 auf. Dieser Wert liegt deutlich unter den Werten, die in der Literatur erreicht werden. Ma und Ma [65] erreichten mit einem Faster R-CNN-Modell, das am ECP-Datensatz trainiert worden war, ein mAP von 0,912. Ihr Score-Grenzwert lag bei 0,05, einen IoU -Grenzwert gaben die Autor*innen nicht an. Wang erzielte mit einem am CMP-Datensatz trainierten Faster R-CNN-Modell ein $F1$ von 0,819, sein Score-Grenzwert lag bei 0,8, der IoU -Grenzwert bei 0,75. Der $F1$ -Wert des besten Modells dieser Masterarbeit betrug für denselben IoU - und Score-Grenzwert 0,493. Ein mAP wurde von Wang nicht berechnet. Es muss für beide Literaturwerte jedoch angemerkt werden, dass die jeweiligen Modelle an einem Teil des Datensatzes getestet wurden, aus dem auch der Trainingsdatensatz entnommen worden war. Die in Tabelle 6 aufgelisteten Werte wurden für das Hamburg Image Dataset berechnet, das unabhängig vom Trainingsdatensatz erstellt worden war. Da die Score- und IoU -Grenzwerte der genannten Autor*innen mit den in dieser Arbeit verwendeten Werten übereinstimmen, kann jedoch geschlussfolgert werden, dass selbst das beste Modell aus Tabelle 6 noch Potential für bessere Ergebnisse in der Fenstererkennung hat.

Dies wird durch den Verlauf des Trainings- und Validierungs-Losses in Abbildung 33 noch weiter bestärkt, der eine Tendenz zu Overfitting erkennen lässt und keine Konvergenz zwischen den beiden Losses aufweist (vergleiche hierzu Abbildung 6). Ein Grund hierfür könnte sein, dass die Gebäude des CMP-Datensatzes, an dem das Modell trainiert wurde, eine zu geringe Ähnlichkeit mit den ausgewählten Hamburger Gebäuden des Testdatensatzes haben. Ein Training des Faster R-CNN-Modells mit einem noch zu erstellenden Trainingsdatensatz mit Hamburger Bildern könnte das Ergebnis in diesem Fall verbessern. Eine andere mögliche Erklärung für die geringen *mAP*-Werte sind schlecht gewählte Hyperparameter. Es wurden im Training des Faster R-CNN-Modells in dieser Arbeit ausschließlich die maximale Anzahl der Epochen und die Größe des Validierungsdatensatzes variiert. Alle anderen Werte, die nicht durch die zur Verfügung stehende Rechenleistung vorgegeben waren, wurden aus der Literatur übernommen. Eine feinmaschigere Untersuchung aller Hyperparameter könnte unter Umständen das Ergebnis ebenfalls verbessern. Gleiches gilt für die in Kapitel 3.1.3 beschriebenen Augmentationen. Der Einfluss einzelner Augmentationen auf das *mAP* wurde nicht geprüft.

Die Ergebnisse der Variation eines zusätzlichen Hyperparameters, nämlich des Score-Grenzwertes, sind in Tabelle 8 aufgelistet. Eine Erklärung für die Abhängigkeit des *mAP* vom Score-Grenzwert ist im Anhang F aufgeführt: Durch Absenken des Score-Grenzwertes nimmt die Anzahl an Ergebnissen, die für *mAP* berücksichtigt werden, zu. Neben der Abhängigkeit des *mAP* vom Score-Grenzwert wird in Tabelle 8 auch der Einfluss der Nachbearbeitung auf die Bewertungsgröße deutlich. Keine der Nachbearbeitungsmethoden konnte das *mAP* verbessern. Dies spricht dafür, dass die Methoden zur Nachbearbeitung verbesserungswürdig sind. Bei Betrachten der Ergebnisse der Nachbearbeitung in Abbildung 34 fällt jedoch auf, dass die nachbearbeiteten Ergebnisse die Fenster der Fassade besser abbilden als die nicht nachbearbeiteten. Dies gilt gerade für geringe Score-Grenzwerte. Trotzdem wurden nur bei wenigen Bildern des Testdatensatzes neue Fenster hinzugefügt, was mit großer Wahrscheinlichkeit zu einer Verbesserung des *mAP* geführt hätte. Auch wurden für niedrige Score-Grenzwerte nicht alle Überschneidungen zwischen erkannten Fenstern behoben, wenn der Größenunterschied der Fenster zu groß war (wie in Abbildung 34). Ursächlich hierfür könnte sein, dass die in Kapitel 3.1.2 beschriebenen Regeln zur Ergänzung neuer Fenster nicht systematisch für das Hamburg Image Dataset optimiert wurden, genauso wie die Regeln zur Ermittlung von $K_{Cluster}$, der *IoU*-Grenzwert zur Anwendung der NMS und die maximale Abweichung, die Fenster-Mittelpunkte haben dürfen, damit sie neu ausgerichtet werden. Eine Untersuchung des Einflusses dieser Größen könnte gegebenenfalls das *mAP* verbessern. Darüber hinaus zeigt sich in Abbildung 34, dass das Einteilen der Fenstergrößen durch Clustern im Rahmen der Nachbearbeitung nicht fehlerfrei funktioniert. Die Auswahl von $K_{Cluster}$ (siehe Kapitel 3.1.2) weist akzeptable Ergebnisse

auf, dennoch treten in den Bildern Abweichungen zur tatsächlichen Anzahl unterschiedlicher Fenstergrößen auf, wie auch in Abbildung 34, in der $K_{Cluster}$ als 4 definiert wurde, während tatsächlich nur 2 unterschiedliche Fenstergrößen vorliegen. Wie in Kapitel 4.1.1 erwähnt, kann aufgrund dieser fehlerhaften Clusterung keine Ergänzung neuer Fenster erfolgen. All dies könnten Gründe sein, weshalb das mAP für alle Methoden der Nachbearbeitung sinkt, auch wenn die optischen Ergebnisse diesen Schluss nicht zwangsläufig zulassen. Eine weitere Ursache könnte sein, dass die Scores der Ergebnisse des ML-Modells in der Nachbearbeitung nur beim Hinzufügen neuer Fenster angepasst wurden. Auch wenn die nachbearbeiteten Fenster die gelabelten Fenster genauer abbilden, ist der entsprechende Score in allen anderen Fällen noch der, der vom ML-Modell berechnet wurde. Die Sortierung der Ergebnisse entsprechend des Scores zur Berechnung von mAP (siehe Kapitel 2.4.3) wird durch die Nachbearbeitung also nicht geändert, was wiederum das mAP negativ beeinflussen könnte.

Im Gegensatz zu mAP wird die Berechnung des Verhältnisses der Fenster- zur Wandfläche $R_{Gebäude}$ nicht durch den Score der vom ML-Modell vorhergesagten Fenster beeinflusst. Dementsprechend verringert sich die Abweichung von $R_{Gebäude}$ zu $R_{Gebäude, HI}$ auch durch Nachbearbeitung. Hinzu kommt, dass für $R_{Gebäude}$ das gesamte Gebäude betrachtet wird. Die Positionen einzelner Fenster, die das mAP maßgeblich beeinflussen, spielen hierbei keine Rolle. Fenster, die an einer Stelle nicht erkannt wurden, können so durch Fenster ausgeglichen werden, die an einer anderen Stelle zu viel erkannt wurden. Auch Dachfenster, Schaufenster oder Türen werden vom ML-Modell als Fenster identifiziert, sind im Hamburg Image Dataset jedoch nicht als solche gelabelt (siehe Abbildung 19 und Abbildung 34). Während dies zu einer weiteren Verschlechterung des mAP führt, können bezüglich $R_{Gebäude}$ auf diese Weise nicht erkannte Fenster ausgeglichen werden.

Dies zeigt sich auch in Abbildung 37, in der $R_{Gebäude}$, $R_{Gebäude, IWU}$ und $R_{Gebäude, HI}$ für alle 6 Gebäude des Hamburg Image Datasets gegenübergestellt sind. Hierbei wird $R_{Gebäude, HI}$ auf Basis der IWU-Gebäudetypologie für alle Gebäude überschätzt. Dies entspricht den Erkenntnissen von Dochev et al. [20] (siehe Kapitel 2.1.1). Im Gegensatz zur Arbeit von Dochev et al. wird in Abbildung 37 jedoch deutlich, dass $R_{Gebäude}$ auf Basis der Ergebnisse des ML-Modells die tatsächlichen Werte entweder über- oder unterschätzt, je nachdem, welches Gebäude betrachtet wird und wie hoch der Score-Grenzwert ist. Für die Gebäude Mengestraße 6 und Weimarer Straße 43 wird $R_{Gebäude, HI}$ für jeden Score-Grenzwert unterschätzt. Die erkannte Fensterfläche ist immer kleiner als die im Hamburg Image Dataset markierte. Für beide Gebäude gibt es im Vergleich zu den anderen Gebäuden weniger Bilder, die wenige oder gar keine Fenster enthalten, was die niedrigen $R_{Gebäude}$ -Werte erklären könnte. Für die übrigen 4 Gebäude gilt, dass die Fensterfläche für hohe Score-Grenzwerte eher unter-, für niedrige Score-Grenzwerte eher überschätzt wird. Insgesamt zeigt sich, dass zwar der Score einzelner erkannter Fenster keinen direkten Einfluss auf die

Abweichung von $R_{\text{Gebäude}}$ zu $R_{\text{Gebäude, HI}}$ hat, eine Zunahme des Score-Grenzwertes hingegen zu einer Abnahme von $R_{\text{Gebäude}}$ führt. Dies liegt daran, dass bei einem höheren Score-Grenzwert weniger erkannte Fenster mit in die Auswertung einbezogen werden (siehe Abbildung 59 im Anhang F). Die Größe der gesamten Fensterfläche einer Wand sinkt also, während die Wandfläche konstant bleibt, da sie nicht vom Score-Grenzwert abhängig ist. Für den Score-Grenzwert 0,05 fällt in Abbildung 37 der hohe Wert für $R_{\text{Gebäude}}$ für das Gebäude Veringstraße 149 auf. In Abbildung 34 ist zu erkennen, dass das ML-Modell die gesamte Fassadenfläche des Gebäudes als 2 große Fenster erkennt. Auch auf einer zweiten Fassade des Gebäudes wird eine Fläche als Fenster erkannt, die fast die gesamte Fassade einnimmt. Ursächlich hierfür könnte die Ziegelfassade des Gebäudes sein, die im CMP-Datensatz kaum repräsentiert ist. Dies führt dazu, dass die erkannte Fensterfläche viel größer ist als die tatsächliche, wodurch sich sowohl der hohe Wert für $R_{\text{Gebäude}}$ in Abbildung 37 als auch der Ausschlag der Abweichung der erkannten zur tatsächlichen Fensterfläche ins Negative in Abbildung 38 erklären lassen. Die oben beschriebene Einbindung von Bildern von Hamburger Gebäuden in das Training des ML-Modells könnte dem entgegenwirken.

Die Genauigkeit der Ergebnisse für das Verhältnis der Fenster- zur Wandfläche variiert nicht nur von Gebäude zu Gebäude, sondern auch innerhalb der Wände einzelner Gebäude. Für den Score-Grenzwert von 0,25, mit dem das beste Ergebnis bezüglich $R_{\text{Gebäude}}$ erzielt werden konnte, zeigt sich in Abbildung 39, dass die Ausrichtung der Gebäude einen Einfluss auf die Genauigkeit der Ergebnisse des ML-Modells hat. So ist die Abweichung ΔR_{Wand} für nach Süden ausgerichtete Wände weiter gestreut als für die anderen Himmelsrichtungen. Die negativen Werte für ΔR_{Wand} lassen darauf schließen, dass das ML-Modell für diese Wände Fenster erkennt, wo im Hamburg Image Dataset keine sind. Im Gegensatz dazu scheint das ML-Modell ΔR_{Wand} für Wände, die nach Osten oder Westen ausgerichtet sind, eher zu unterschätzen, was bedeutet, dass vorhandene Fenster vom Modell nicht erkannt werden. Dies lässt sich mit der Ausrichtung der 6 Gebäude des Hamburg Image Datasets erklären. In Abbildung 40 ist die im Hamburg Image Dataset definierte Fensterfläche der einzelnen Wände der Gebäude in Abhängigkeit der Himmelsrichtung, zu der die Wände ausgerichtet sind, dargestellt. Die Einteilung der Ausrichtungen der Wände in die 4 Himmelsrichtungen erfolgte dabei anhand der in Abbildung 21 dargestellten Sektoren.

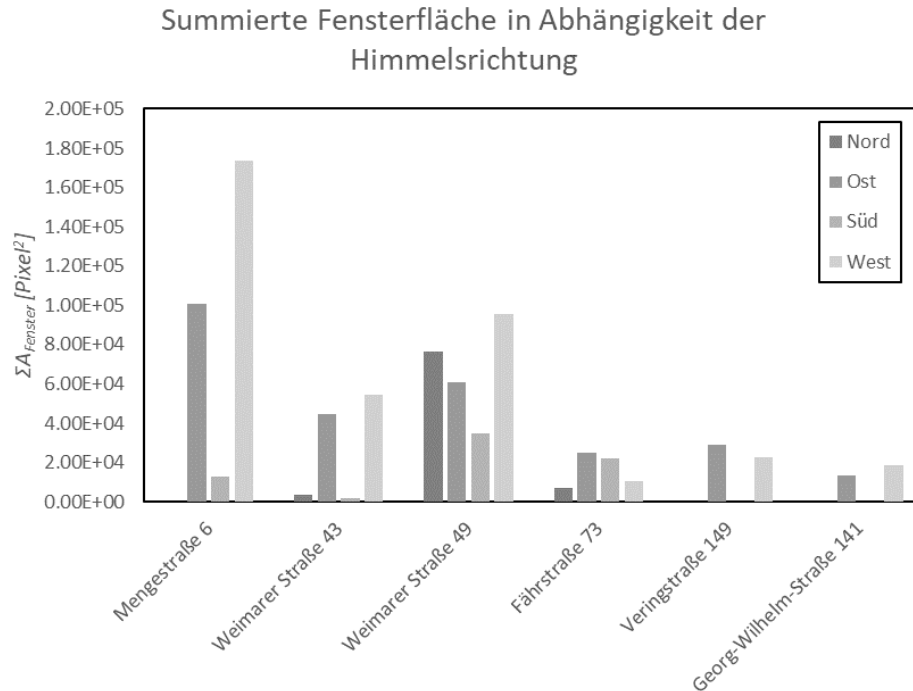


Abbildung 40: Fensterfläche der Gebäude des Hamburg Image Datasets, summiert entsprechend ihrer Ausrichtung. Es wird bei der Ausrichtung zwischen Nord, Ost, Süd und West unterschieden.

Hier wird deutlich, dass 4 der 6 Gebäude in Ost-West-Richtung eine größere Fensterfläche aufweisen als in Nord-Süd-Richtung. Die Gebäude Weimarer Straße 49 und Fährstraße 49 lassen diese Verteilung nicht erkennen, da es sich bei diesen Gebäuden um Eckgebäude handelt, die an Kreuzungen gelegen sind. Alle anderen Gebäude weisen eine eindeutige Vorder- und Rückseite auf, die größere Fensterflächen haben als die Seitenflächen, welche zum Teil gar keine Fenster aufweisen. Das ML-Modell erkennt an den meisten der Seitenflächen dennoch Fenster, wodurch es zu den negativen Werten für ΔR_{Wand} in Nord- und Südrichtung in Abbildung 39 kommt. Dass die Werte in Nordrichtung weniger stark gestreut sind als in Südrichtung könnte damit zusammenhängen, dass die Fassade der Weimarer Straße 49 nach Norden zeigt und viele Fenster aufweist. Für die Vorder- und Rückseiten der Gebäude, die eine größere Fensterfläche aufweisen als die Seitenflächen, kann das ML-Modell im Gegensatz dazu nicht alle Fenster erkennen, weil diese zum Beispiel verschattet sind (siehe Abbildung 34). Dies erklärt die positiven Werte für ΔR_{Wand} in Ost- und Westrichtung in Abbildung 39. Das ML-Modell neigt demnach dazu, die Fensterfläche der Seitenflächen von Gebäuden zu überschätzen und die Fensterflächen der Vorder- und Rückseite zu unterschätzen.

Hierbei ist jedoch zu beachten, dass das 3D-Modell größere Wandflächen aufweist als das Hamburg Image Dataset, was in Abbildung 38 deutlich wird. Dies scheint plausibel, da das 3D-Modell zum Teil Dachflächen mit zu den Wandflächen zählt, welche beim Labeln der Bilder des

Hamburg Image Datasets jedoch nicht berücksichtigt wurden (siehe Abbildung 19). Eine hohe Übereinstimmung zwischen $R_{\text{Gebäude}}$ und $R_{\text{Gebäude, HI}}$ wird jedoch demnach nur dann erzielt, wenn die erkannte Fensterfläche eines Gebäudes größer ist als die tatsächliche. Eine höhere Übereinstimmung bezüglich $R_{\text{Gebäude}}$ und ein höheres mAP könnten unter Umständen erzielt werden, wenn neben den Wandflächen auch die Dachflächen mit ins Hamburg Image Dataset einbezogen werden würden. Auf diese Weise würden die Dachfenster als korrekt erkannte Fenster zählen und die Fläche, auf die sich $R_{\text{Gebäude}}$ bezieht würde durch die Dachfläche vergrößert.

Bei der Auswertung aller beschriebenen Ergebnisse muss berücksichtigt werden, dass die Label des Hamburg Image Datasets, die als Referenzgröße sowohl für mAP als auch für $\Delta R_{\text{Gebäude, mean}}$ dienen, Ungenauigkeiten aufweisen. Diese sind zum einen über Ungenauigkeiten des 3D-Stadtmodells zu erklären, die bereits in der Arbeit von Kaden [37] beschrieben wurden (siehe Kapitel 2.1.1) und auch in Abbildung 19 deutlich werden. Sie führen dazu, dass Fassaden, die vom 3D-Modell als zu klein dargestellt werden, auch als zu klein gelabelt werden, da die Label auf dem sichtbaren Bereich des Gebäudes auf dem Bild basieren, welcher wiederum das 3D-Modell zur Grundlage hat. Zum anderen unterliegt die Genauigkeit der Label des Hamburg Image Datasets in Position und Größe Schwankungen, die der manuellen Annotation, der unterschiedlichen grafischen Qualität der Bilder und Verschattungen geschuldet sind. Demnach bildet das Hamburg Image Dataset die Realität nicht fehlerfrei ab. Eine ideale Referenzgröße für die Beurteilung der Methodik dieser Arbeit wären Baupläne von Gebäuden, die jedoch nicht beschafft werden konnten. Ein händisches Ausmessen der Fassaden vor Ort konnte aufgrund des hohen technischen und zeitlichen Aufwandes nicht realisiert werden.

Die beschriebene Methodik zur Bestimmung von $R_{\text{Gebäude}}$ kann auf Städte in allen anderen deutschen Bundesländern übertragen werden, da 3D-Stadtmodelle für alle Bundesländer vorhanden sind (siehe Tabelle 11 im O). Schrägluftbilder werden außer in Hamburg zwar nur in Bremen von Seite des Bundeslandes zur Verfügung gestellt (siehe Tabelle 11), doch mit der in Kapitel 2.2.4 beschriebenen Funktion „Vogelperspektive“ von Bing Maps können mindestens für alle deutschen Landeshauptstädte Schrägluftbilder aus vier Blickrichtungen heruntergeladen werden. Für die Bundesländer Baden-Württemberg, Rheinland-Pfalz und Saarland ist die Beschaffung des 3D-Stadtmodells mit Kosten verbunden, für alle anderen Bundesländer sind die 3D-Modelle frei zugänglich.

4.2 Bestimmung der Baualtersklasse eines Gebäudes

Während die Bestimmung von $R_{\text{Gebäude}}$ für die 6 Gebäude des Hamburg Image Datasets erfolgte, wurde die Baualtersklasse für die 247 Gebäude des in Kapitel 3.2.3 vorgestellten Testdatensatzes bestimmt. Dieser Prozess wird im Folgenden beschrieben. Hierbei wird neben der Bestimmung

der Baualtersklassen auch auf die Bestimmung der Baujahre eingegangen, da diese die Grundlage für die Einordnung der Gebäude in Baualtersklassen bilden.

4.2.1 Validierung der Ergebnisse

Für jeden IoU -Grenzwert IoU_{min} zwischen 0,9 und 0,1 wurden die durchschnittliche Abweichung ΔBJ_{mean} der geschätzten Baujahre von den tatsächlichen Baujahren aller 247 Test-Gebäude, deren Standardabweichung σ sowie mAP bezüglich der Baualtersklassen bestimmt. Zur Veranschaulichung sind die Ergebnisse in Abbildung 41 grafisch dargestellt. Auf der linken y-Achse ist hierbei die mittlere Abweichung der geschätzten von den tatsächlichen Baujahren aufgetragen, auf der rechten y-Achse das mAP .

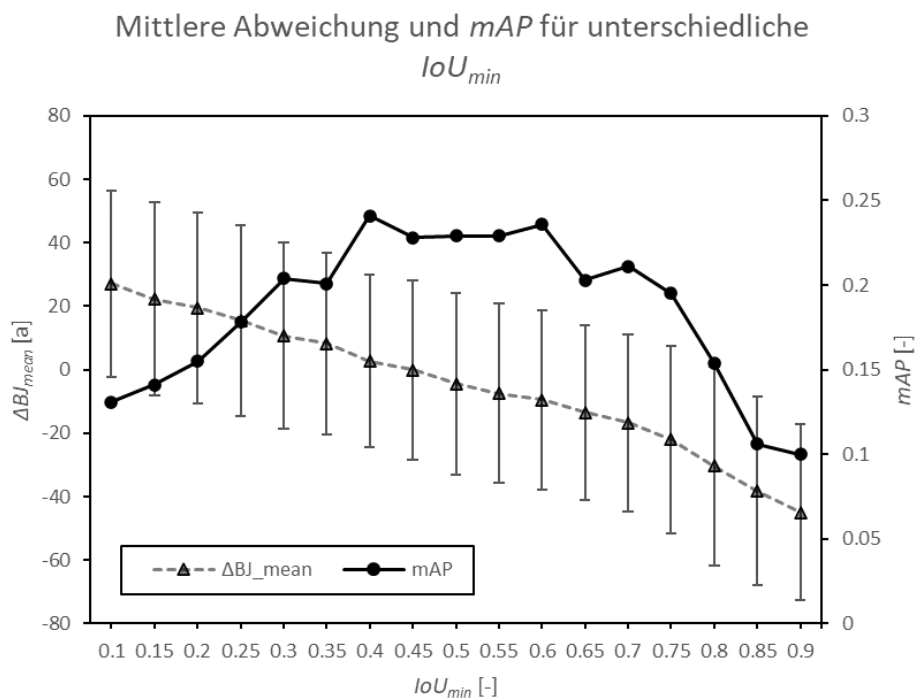


Abbildung 41: Durchschnittliche Abweichung ΔBJ_{mean} der geschätzten Baujahre von den tatsächlichen Baujahren der 247 Test-Gebäude, deren Standardabweichung σ sowie mAP bezüglich der Baualtersklassen in Abhängigkeit unterschiedlicher IoU -Grenzwerte IoU_{min} .

Es zeigt sich in Abbildung 41, dass die durchschnittliche Abweichung beinahe konstant von 27,1 Jahren für ein IoU_{min} von 0,1 auf -44,9 Jahre für ein IoU_{min} von 0,9 abfällt. Für ein IoU_{min} von 0,45 beträgt die mittlere Abweichung 0 Jahre. Die Standardabweichung σ ist über alle IoU_{min} -Werte beinahe konstant, sie schwankt zwischen 30,3 Jahren für $IoU_{min} = 0,15$ und 27,1 Jahren für $IoU_{min} = 0,4$. Diese Werte zeigen, dass die Abschätzung eines Baujahres mit der in dieser Arbeit beschriebenen Methodik große Unsicherheiten aufweist. Bezüglich der Einordnung der Gebäude des Testdatensatzes in die IWU-Baualtersklassen kann die Genauigkeit mithilfe des Wertes mAP

überprüft werden. Der Verlauf von mAP in Abbildung 41 kann grob vereinfacht als nach unten geöffnete Parabel beschrieben werden. Zwischen $IoU_{min} = 0,1$ und $IoU_{min} = 0,4$ steigt mAP annähernd quadratisch an, zwischen $IoU_{min} = 0,6$ und $IoU_{min} = 0,9$ fällt es ebenso ab. Dazwischen bildet mAP ein Plateau, wobei der maximale Wert von $mAP = 0,241$ bei einem IoU_{min} von 0,4 erreicht wird. Das mAP von 0,241 bedeutet, dass die Test-Gebäude mit einer Sicherheit von 24,1 % in die IWU-Baualtersklassen eingeordnet werden. Zwar ist ΔBJ_{mean} für $IoU_{min} = 0,45$ mit 0 Jahren am geringsten, da für $IoU_{min} = 0,4$ jedoch sowohl mAP als auch σ die geringsten Werte aufweisen und das entsprechende ΔBJ_{mean} mit 2,7 Jahren der zweitniedrigste Wert und nur minimal größer als das ΔBJ_{mean} für $IoU_{min} = 0,45$ ist, wird im Folgenden die Baujahresbestimmung mit einem IoU_{min} von 0,4 näher beschrieben. Es konnte mit dieser Methode jedem der 247 Gebäude des Testdatensatzes ein Baujahr zugewiesen werden.

Wie in Kapitel 2.4.3 beschrieben, wird mAP aus den APs der einzelnen Klassen berechnet. Diese APs sind in Abbildung 42 für jede Klasse aufgeführt. Darüber hinaus gibt die Abbildung Aufschluss über die Anzahl an Gebäuden, die den Klassen auf Basis der geschätzten Baujahre (weiße Balken) oder tatsächlichen Baujahre (gestreifte Balken) zugeordnet wurden.

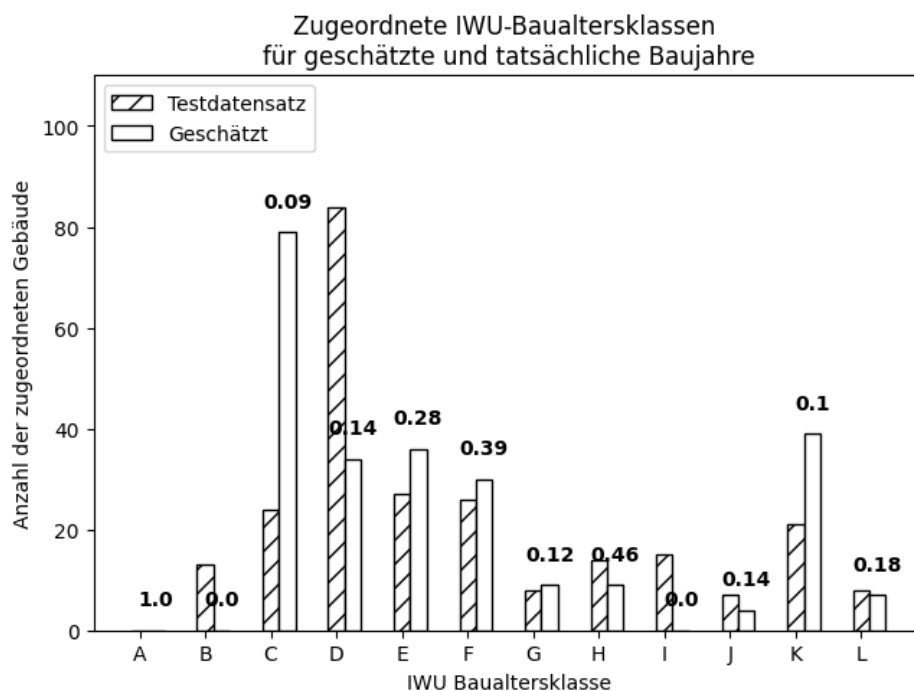


Abbildung 42: Anzahl der Gebäude, die den IWU-Baualtersklassen auf Basis der tatsächlichen und geschätzten Baujahre zugewiesen wurden. Die Zahlen über den Balken stehen für das AP der jeweiligen Klasse.

Hier zeigt sich, dass AP für die IWU-Baualtersklassen zwischen 0 und 1 schwankt. Für die Klasse A beträgt AP 1, da dieser Klasse weder auf Basis der geschätzten noch der tatsächlichen Baujahre Gebäude zugeordnet werden. Für die Klasse B hingegen ist AP 0, da mehrere Gebäude des Test-

Datensatzes dieser Klasse angehören, jedoch kein geschätztes Baujahr im Bereich dieser Klasse liegt. Der Klasse C wurden viele Gebäude zugewiesen. Da im Testdatensatz jedoch deutlich weniger Gebäude dieser Klasse angehören, beträgt *AP* hier nur 0,09. Im Gegensatz dazu weist die Klasse D einen deutlichen Überschuss an Gebäuden des Testdatensatzes gegenüber den auf Basis der geschätzten Baujahre zugeordneten Gebäude auf. *AP* ist mit 0,14 nur geringfügig höher als für Klasse C. Die Klassen E und F weisen mit *AP*s von 0,28, beziehungsweise 0,39 vergleichsweise hohe Genauigkeiten auf und auch die Anzahl der zugeordneten Gebäude ist für geschätzte Baujahre nur geringfügig höher als für tatsächliche. Ursache für die Abweichungen könnten hier Gebäude sein, die auf älteren Karten vorhanden waren, jedoch aufgrund schlechter grafischer Qualität nicht erkannt wurden. Für die Klasse G sind in Abbildung 42 die beiden Balken fast gleich groß, das *AP* mit 0,12 jedoch gering. Dies könnte an der geringen Anzahl an Gebäuden liegen, die der Klasse im Testdatensatz zugewiesen werden können. Gleiches gilt für das geringe *AP* von 0,14 für Klasse J. Im Gegensatz dazu weist Klasse H, bei ebenfalls geringer Anzahl an Gebäuden im Testdatensatz, mit 0,46 das höchste *AP* auf. Der Klasse K werden auf Basis der geschätzten Baujahre deutlich mehr Gebäude zugewiesen als auf Basis der tatsächlichen Baujahre. Das *AP* dieser Klasse ist mit 0,1 gering. Der Klasse L werden auf Basis der geschätzten Baujahre fast so viele Gebäude zugewiesen wie auf Basis der tatsächlichen Baujahre. Das *AP* ist auch hier mit 0,18 niedrig, was vor allem unter Berücksichtigung der guten grafischen Qualität der Karte und der zunehmenden Übereinstimmung der erkannten Konturen mit den tatsächlichen Grundrissen bemerkenswert ist. Diese zunehmende Übereinstimmung der Grundrisse kann aus Abbildung 43 abgeleitet werden. Dort sind die maximal ermittelten *IoUs* zwischen erkannten Konturen und den Grundriss-Polygonen in Abhängigkeit der geschätzten Baujahre abgebildet. In der Abbildung sind eindeutig die geschätzten Baujahre zu erkennen, die sich an den Grenzen der Jahrgänge der Karten orientieren. Für jedes der geschätzten Baujahre wurde der Mittelwert aller maximalen *IoU*-Werte gebildet. Die Mittelwerte sind in Abbildung 43 als Kreuze markiert. Die schwarze Linie stellt die Regressionsgerade durch die Mittelwerte der maximalen *IoU*-Werte dar. Der Verlauf dieser Linie lässt darauf schließen, dass die Übereinstimmung zwischen den Grundriss-Polygonen der ALKIS-Datenbank und den in der Karte gefundenen Polygonen zunimmt, je neuer die Karte ist. Dies scheint plausibel, da sowohl die Karten als auch die ALKIS-Datenbank vom Landesbetrieb für Geoinformation und Vermessung verwaltet werden und dementsprechend wahrscheinlich ist, dass die neueren Karten und die ALKIS-Datenbank dieselbe Grundlage haben.

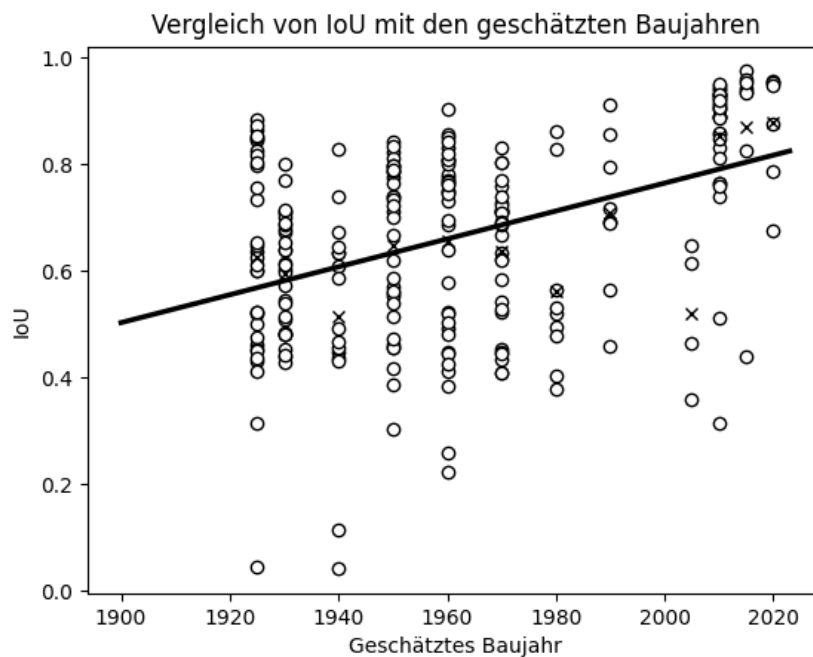


Abbildung 43: Maximal berechnet IoU-Werte für die geschätzten Baujahre des Testdatensatzes (Punkte). Die Kreuze sind die IoU-Mittelwerte der einzigartigen geschätzten Baujahre, über die eine schwarze Regressionsgerade gelegt wurde.

Weitere Erkenntnisse gibt die qualitative Analyse der Baualtersklassenbestimmung an den 6 Gebäuden des Hamburg Image Datasets aus Tabelle 4. Die geschätzten Baujahre und Baualtersklassen sind in Tabelle 9 den tatsächlichen Baujahren und Baualtersklassen gegenübergestellt.

Tabelle 9: Geschätzte und tatsächliche Baujahre und Baualtersklassen für die 6 Gebäude des Hamburg Image Datasets

Adressen der Gebäude des Hamburg Image Datasets	Tatsächliches Baujahr	Tatsächliche Baualtersklasse	Geschätztes Baujahr	Geschätzte Baualtersklasse
Mengestraße 6	1966	E	1970	F
Weimarer Straße 43	1949	D	1940	C
Weimarer Straße 49	2014	K	1980	G
Fährstraße 73	1904	B	1925	C
Veringstraße 149	1952	D	1950	D
Georg-Wilhelm-Straße 141	1956	D	1950	D

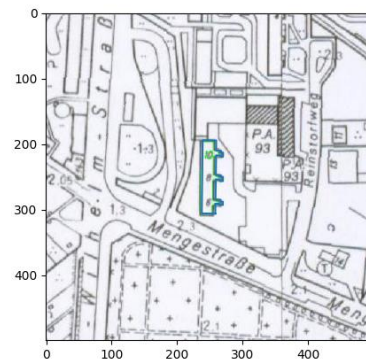
Die durchschnittliche Abweichung der geschätzten von den tatsächlichen Baujahren beträgt für die 6 Gebäude 4,3 Jahre und weicht damit nur geringfügig von der durchschnittlichen Abweichung von 2,7 Jahren aus Abbildung 41 ab. Darüber hinaus zeigt sich, dass auf Basis der geschätzten

Baujahre nur zwei der sechs Gebäude der korrekten Baualtersklasse zugeordnet werden konnten. Im Folgenden soll die Bestimmung der Baujahre und Baualtersklassen für die 6 Gebäude nachvollzogen werden.

Die Karten „Jahrgang 1960-1970“ und „Jahrgang 1970-1980“ sind zusammen mit dem Grundriss-Polygon des Gebäudes Mengestraße 6 ausschnittsweise in Abbildung 44 dargestellt. Auf dem linken Kartenausschnitt ist zu erkennen, dass an der Stelle des Grundriss-Polygons keine Gebäude aufgezeichnet sind. Auf der rechten Seite hingegen ist das Gebäude in der heutigen Form, wie sie im ALKIS-Datensatz hinterlegt ist, vorhanden. Die Entscheidung, dem Gebäude das Baujahr 1970 als unteren Grenzwert des Jahrgangs 1970-1980 zuzuweisen ist dementsprechend nachvollziehbar. Allerdings würde das Gebäude auf Basis des tatsächlichen Baujahres der Baualtersklasse E zugeordnet werden, auf Basis des geschätzten Baujahres der Klasse F.



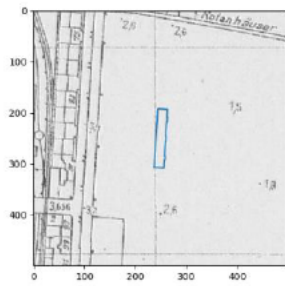
Jahrgang 1960-1970



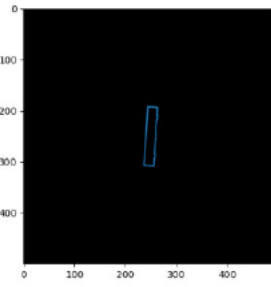
Jahrgang 1970-1980

Abbildung 44: Grundriss-Polygon der Mengestraße 6 auf zwei Kartenausschnitten unterschiedlicher Jahrgänge.

Das Grundriss-Polygon des Gebäudes Weimarer Straße 43 ist auf drei Kartenausschnitten in Abbildung 45 dargestellt. Im Testdatensatz wird das Baujahr mit 1949 angegeben, mit der in dieser Arbeit beschriebenen Methodik wird es auf 1940 geschätzt. Bei Betrachtung der Kartenausschnitte in Abbildung 45 wird deutlich, dass das Gebäude in der Karte „Jahrgang 1925-1930“ nicht vermerkt ist. In Kapitel 3.2.3 wurde angegeben, dass das Gebäude eigentlich in den 1930ern erbaut, im zweiten Weltkrieg zerstört und 1949 wiederaufgebaut wurde. Da die Karte „Jahrgang 1930-1940“ das Gebiet nicht abbildet, das Gebäude in der Karte „Jahrgang 1940-1950“ aber sichtbar ist, wurde das Baujahr auf 1940 geschätzt. Bezüglich der Baualtersklasse wird das tatsächliche Baujahr in Klasse D eingeordnet, das geschätzte Baujahr jedoch in Klasse C.



Jahrgang 1925-1930



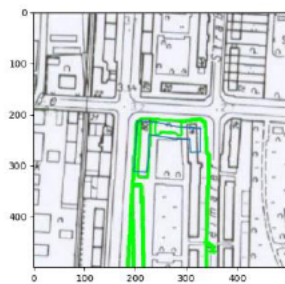
Jahrgang 1930-1940



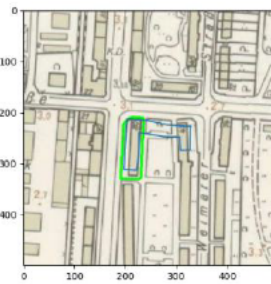
Jahrgang 1940-1950

Abbildung 45: Grundriss-Polygon der Weimarer Straße 43 auf drei Kartenausschnitten unterschiedlicher Jahrgänge.

Das Gebäude mit der Adresse Weimarer Straße 49 weist in Tabelle 9 die größte Abweichung zwischen tatsächlichem und geschätztem Baujahr auf. In Abbildung 46 ist das Grundriss-Polygon des Gebäudes in blau sowie das Polygon der Kontur mit dem größten IoU in grün für die Karten der Jahrgänge 1970-1980, 1980-1990 und 2015 dargestellt. Weder für die Karte „Jahrgang 1970-1980“ noch „Jahrgang 1980-1990“ wird das IoU_{min} von 0,4 überschritten. Da dem Gebäude allerdings mehrere Adressen zugeordnet werden können, wird das Gebäude bereits in der Karte „Jahrgang 1980-1990“ als erkannt definiert, obwohl es tatsächlich erst in der Karte „Jahrgang 2015“ zum ersten Mal auftaucht. Die geschätzte Baualtersklasse G weicht somit deutlich von der tatsächlichen Baualtersklasse K ab.



Jahrgang 1970-1980



Jahrgang 1980-1990



Jahrgang 2015

Abbildung 46: Grundriss-Polygon (blau) der Weimarer Straße 49 auf drei Kartenausschnitten unterschiedlicher Jahrgänge. In Grün ist jeweils das Polygon der Kontur mit dem größten IoU dargestellt.

Auch für das Gebäude Fährstraße 72 stimmen geschätzte und tatsächliche Baualtersklasse nicht überein. Das Baujahr des Gebäudes wird im Testdatensatz mit 1904 angegeben, was der Klasse B entspricht. Das geschätzte Baujahr hingegen ist mit 1925 der Klasse C zuzuordnen. In Abbildung 47 ist das Grundriss-Polygon des Gebäudes in der Karte „Jahrgang 1925-1930“ abgebildet. Die

Schätzung des Baujahres auf 1925 ist hier nachvollziehbar, da es sich bei der Karte um die älteste des Kartensatzes handelt. In der Abbildung wird deutlich, dass die grafische Qualität über die Karte variiert und im Bereich des Gebäudes schlecht ist. Darüber hinaus wird die Erkennung von Konturen durch in den Gebäudegrundrissen notierten Hausnummern erschwert.

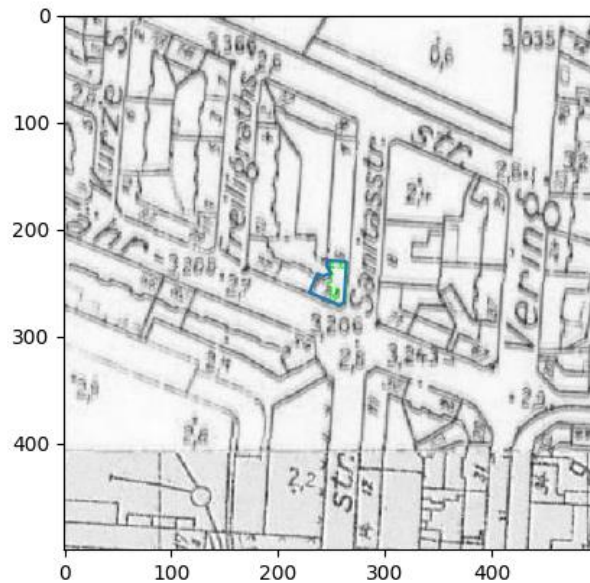


Abbildung 47: Grundriss-Polygon der Fährstraße 73 (blau) auf einem Ausschnitt der Karte „Jahrgang 1925-1930“. In Grün ist das erkannte Polygon mit dem größten IoU dargestellt.

Als einzige Gebäude aus Tabelle 9 weisen die Veringstraße 149 und Georg-Wilhelm-Straße 141 identische Baualtersklassen für geschätzte und tatsächliche Baujahre auf. Für die Veringstraße 149 sind Grundrisspolygon und Kartenausschnitte in Abbildung 48 abgebildet. Das Baujahr des Gebäudes wird vom Denkmalschutzamt in die 1930er datiert [18]. Das Gebäude taucht jedoch zum ersten Mal auf der Karte „Jahrgang 1940-1950“ auf, da die Karte „Jahrgang 1930-1940“ das Gebiet nicht abbildet. Aufgrund schlechter Bildqualität kann der Grundriss des Gebäudes jedoch erst auf der Karte „Jahrgang 1950-1960“ erkannt werden. Das Baujahr wird damit auf 1950 geschätzt, die Baualtersklasse ist D.



Jahrgang 1940-1950



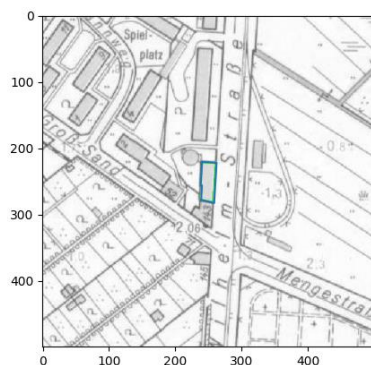
Jahrgang 1950-1960

Abbildung 48: Grundriss-Polygon der Veringstraße 149 auf zwei Kartenausschnitten unterschiedlicher Jahrgänge.

In Abbildung 49 ist der Grundriss des Gebäudes Georg-Wilhelm-Straße 141 auf zwei Karten abgebildet. Es ist zu erkennen, dass der Grundriss des Gebäudes zum ersten Mal in der Karte „Jahrgang 1950-1960“ auftaucht. Das geschätzte Baujahr ist demnach ebenfalls 1950, die Baualtersklasse ist D.



Jahrgang 1940-1950



Jahrgang 1950-1960

Abbildung 49: Grundriss-Polygon der Georg-Wilhelm-Straße 141 auf zwei Kartenausschnitten unterschiedlicher Jahrgänge.

4.2.2 Diskussion der Ergebnisse

Die größte Genauigkeit bezüglich der Abschätzung eines Baujahres als auch der Einordnung in die IWU-Baualtersklassen wird für ein IoU_{min} von 0,45 erzielt. Die hohe Standardabweichung von 27,1 Jahren für ein ΔB_{mean} von 2,7 Jahren sowie das geringe mAP von 0,241 zeigen jedoch, dass die in dieser Arbeit verwendete Methodik selbst im besten Fall noch zu großen Ungenauigkeiten führt. Die zweite Forschungsfrage aus Kapitel 1.1 kann demnach folgendermaßen beantwortet

werden: Die IWU-Baualtersklasse eines beliebigen Wohngebäudes in Hamburg kann über den Vergleich des Grundriss-Polygons aus ALKIS mit dem Kartensatz „Historische Karten 1:5000 – Auswahl“ mit einer Sicherheit von 24,1 % bestimmt werden. Mögliche Ursachen für die geringe Genauigkeit werden in diesem Kapitel erläutert.

Die Verteilung der Gebäude in Abbildung 42 bezüglich der IWU-Baualtersklassen zeigt zum Teil starke Abweichungen zwischen den geschätzten und den tatsächlichen Baualtersklassen. Eine Ursache hierfür kann die Schätzung der Baujahre auf Basis der Kartenjahrgänge sein. In Tabelle 10 sind die IWU-Baualtersklassen sowie die dazugehörigen geschätzten Baujahre auf Basis der historischen Karten aufgelistet.

Tabelle 10: Vergleich der IWU-Baualtersklassen mit den geschätzten Baujahren, die den Gebäuden auf Basis der historischen Karten zugewiesen werden können.

IWU-Baualtersklasse	Zeitraum	Geschätztes Baujahr
A	bis 1859	
B	1860 – 1918	
C	1919 – 1948	1925, 1930, 1940
D	1949 – 1957	1950
E	1958 – 1968	1960
F	1969 – 1978	1970
G	1979 – 1983	1980
H	1984 – 1994	1990
I	1995 – 2001	
J	2002 – 2009	2005
K	2010 - 2015	2010, 2015
L	seit 2016	2020

Den Klassen A, B und I können auf Basis der historischen Karten keine geschätzten Baujahre zugeordnet werden. So ist das AP von 0 für die Klassen B und I in Abbildung 42 zu erklären. Im Gegensatz dazu können der Klasse C auf Basis der historischen Karten drei geschätzte Baujahre zugewiesen werden, da die Klasse mit Ausnahme der Klassen A und B den größten Zeitraum abdeckt. Darüber hinaus werden alle Gebäude, die vor 1925 gebaut wurden und damit zum Teil der Klasse B angehören, ebenfalls der Klasse C zugeordnet. Dies sind zwei mögliche Erklärungen für die große Anzahl an Gebäuden, die der Klasse auf Basis der geschätzten Baujahre zugewiesen werden. Neben der Klasse C weist auch die Klasse K für die geschätzten Baujahre eine deutlich größere Anzahl an Gebäuden auf als für die tatsächlichen Baujahre. Dies könnte daran liegen, dass

der Klasse in Tabelle 10 zwei geschätzte Baujahre zugewiesen werden. Die zunehmende Qualität der entsprechenden Karten könnte darüber hinaus dazu führen, dass der Klasse ältere Gebäude zugeordnet werden, die vorher nicht erkannt wurden.

Die Auswertung der Baualtersklassenbestimmung der Gebäude Weimarer Straße 49 und Veringstraße 149 zeigt, dass die in Kapitel 3.2.2 vorgestellte Vorgehensweise zur Schätzung von Baujahren für Gebäude mit mehreren Adressen anfällig für Fehler ist. Während das Gebäude in der Weimarer Straße 43 bereits in einer Karte erkannt wird, in der es in seiner heutigen Form nicht vorkommt, wird das Gebäude Veringstraße 149 in der Karte, in der es zum ersten Mal zu sehen ist, nicht erkannt. Während letzterer Fehler auf die schlechte grafische Qualität der Karte zurückgeführt werden kann, auf die später noch eingegangen wird, ist die Ursache für das zu frühe Erkennen des Gebäudes Weimarer Straße 49 in der Methodik zu suchen. Die Schwelle des Grenzwertes IoU_{min} ist offensichtlich zu niedrig, um das Baujahr des Gebäudes korrekt bestimmen zu können. Es ist davon auszugehen, dass dies auch für andere Gebäude des Testdatensatzes gilt, die irgendwann an der Stelle abgerissener Gebäude neu errichtet wurden. Das Alter dieser Gebäude würde demnach systematisch zu hoch eingeschätzt werden. Entsprechend wäre es eine Erklärung für die *mAP*-Werte der Klassen J, K und L in Abbildung 42, die aufgrund der hohen grafischen Qualität der entsprechenden Karten und hohen Übereinstimmungen der Grundrisse in den Karten mit den im ALKIS aufgeführten Grundrissen deutlich größer sein müssten. Eine Anpassung der Methodik könnte demnach das *mAP* verbessern.

Auch eine höhere grafische Auflösung der älteren Karten bis einschließlich Jahrgang 1930-1940 könnte das *mAP* positiv beeinflussen. Dies zeigt die Baualtersklassenbestimmung für das Gebäude Veringstraße 149 (siehe Abbildung 48). Die Auflösung der Karte, in der das Gebäude zum ersten Mal auftaucht, ist so schlecht, dass es nicht erkannt wird. Das Gebäude Fährstraße 73 hingegen wurde zwar korrekt erkannt, die entsprechende Karte wies trotzdem ebenfalls eine schlechte Qualität auf und Hausnummern in den Gebäudegrundrissen erschwerten eine Erkennung zusätzlich (siehe Abbildung 47). Eine Aufbereitung der Bilder der Kartenausschnitte mit der Funktion „*dnn_superres*“ der Python-Bibliothek OpenCV [11], die die Auflösung eines Bildes erhöht, verbesserte die grafische Qualität der Karten nicht. Gleiches gilt für eine Bearbeitung der Bilder mit Kantenerkennung, Unschärfe, morphologischen Operationen und Linien-Segmentierung. Der Fehler, der durch die schlechte grafische Qualität der älteren Karten entsteht, fällt in Abbildung 42 nicht auf, weil der Klasse C nicht nur die Gebäude mit einem geschätzten Baujahr zwischen 1919 und 1948 zugeordnet werden, sondern alle Gebäude, die älter geschätzt werden als 1940. Das geringe *AP* der Klasse C in Abbildung 42 spricht jedoch dafür, dass eine bessere Qualität der Karten, die diesen Zeitraum abbilden, auch zu einer genaueren Baualtersklassenbestimmung führen könnte.

Hierbei muss jedoch berücksichtigt werden, dass die Baujahre von Gebäuden, die in den Karten „Jahrgang 1925-1930“ oder „Jahrgang 1930-1940“ zum ersten Mal auftauchen, nicht zwingend in diesem Zeitraum liegen. In Hamburg wurden im zweiten Weltkrieg Gebäude durch Bombenangriffe zerstört und in den späten 40ern und frühen 50ern wiederaufgebaut. Da bei den Angriffen zum Teil ganze Stadtteile zerstört wurden, kann davon ausgegangen werden, dass dies auf viele Gebäude in Hamburg zutrifft [60]. Auch die Gebäude Weimarer Straße 43 und Veringstraße 149 wurden nach dem Krieg wiederaufgebaut, was durch Plaketten an den Gebäuden belegt ist. Beide Gebäude wurden nach Informationen des Denkmalschutzamtes in den 1930er-Jahren erbaut [18]. Dass die Baujahre der Gebäude trotzdem auf 1940 und 1950 geschätzt werden, liegt daran, dass die Karte „Jahrgang 1930-1940“ den Stadtteil Wilhelmsburg nicht abbildet und die Gebäude demnach zum ersten Mal in der Karte „Jahrgang 1940-1950“ auftauchen (siehe Abbildung 45 und Abbildung 48). Die bereits erwähnte schlechte Qualität der Karten führt im Fall der Veringstraße 149 sogar dazu, dass das Baujahr auf 1950 geschätzt wird und so der Baualtersklasse D zugeordnet werden kann, welcher auch das Wiederaufbaudatum angehört. Anders ist es bei der Weimarer Straße 43, wo das geschätzte Baujahr 1940 der Klasse C angehört, während das Wiederaufbaudatum auch hier in die Klasse D eingeordnet wird. Würde die Zuweisung anhand der Karten ideal funktionieren, würden beide Gebäude ungeachtet der Zerstörung im zweiten Weltkrieg der Klasse C zugeordnet werden. Auf Basis der beiden Gebäude lässt sich zwar keine allgemeingültige Gesetzmäßigkeit ableiten, aber sie bestärken die Vermutung, dass weitere Gebäude, deren Baujahr im Testdatensatz auf Basis einer Plakette in die 50er-Jahre datiert wurde und damit im Bereich der Baualtersklasse D liegt, bereits vor dem zweiten Weltkrieg, im Bereich der Baualtersklasse C, errichtet wurden. Für die Verteilung der Gebäude in Abbildung 42 bedeutet das, dass der Klasse C auf Basis der historischen Karten Gebäude zugewiesen werden, die im Testdatensatz der Klasse D angehören. Die Auswertung der geschätzten Baujahre für die Gebäude Weimarer Straße 43 und Veringstraße 149 lässt jedoch vermuten, dass die schlechte grafische Qualität der Karten bis 1950 diesen Effekt abschwächt.

Demnach ist davon auszugehen, dass bei einer vollständigen Abdeckung des Hamburger Stadtgebietes durch historische Karten auch vor 1960 und einer höheren Auflösung der älteren Karten mehr Gebäude der Klasse C zugewiesen werden würden. Der Überhang an geschätzten Gebäuden in der Klasse C gegenüber den tatsächlichen in Abbildung 42 würde noch größer werden. Die Anzahl der falsch C zugeordneten Gebäude bliebe wahrscheinlich gleich, während die Anzahl der korrekt C zugeordneten Gebäude anstiege. Dies könnte zu einer Steigerung von mAP führen, da in den Gleichungen [2.15] und [2.16] TP steigen, FP konstant bleiben und FN sinken würde. Präzision und Recall stiegen also an und damit auch AP und mAP . Allerdings muss hierbei berücksichtigt werden, dass das Baujahr vieler Gebäude, die zum ersten Mal in den Karten vor

1950 auftauchen, durch das Wiederaufbaudatum nach dem zweiten Weltkrieg ersetzt worden sein könnte, was einer Steigerung des *mAP* entspricht. Eine Definition von Gebieten, in denen eine Zerstörung der Gebäude während des zweiten Weltkrieges wahrscheinlich ist, sowie deren Einbindung in die Baualtersklassenabschätzung könnte hier Abhilfe schaffen.

Die in dieser Arbeit beschriebene Methodik zur Bestimmung der Baualtersklasse von Gebäuden ist nur bedingt auf andere Bundesländer übertragbar, da nur die Hälfte von ihnen historische Karten oder Luftbilder zur Verfügung stellt (siehe Tabelle 11 im 0). Darüber hinaus decken nicht alle in Tabelle 11 aufgelisteten Karten und Luftbilder ein Bundesland vollständig ab oder liegen über einen ähnlich großen Zeitraum wie in Hamburg vor.

5 Fazit

In dieser Masterarbeit wurden zwei Methoden vorgestellt, mit denen sich zum einen das Verhältnis der Fenster- zur Wandfläche $R_{\text{Gebäude}}$ und zum anderen die Baualtersklasse von Hamburger Gebäuden auf Basis öffentlich zugänglicher Daten bestimmen lassen. Die in Kapitel 1.1 gestellten Forschungsfragen konnten im Verlauf der Arbeit beantwortet werden.

Zur Bestimmung des Verhältnisses der Fenster- zur Wandfläche eines Gebäudes wurde dessen Fensterfläche mit dem auf Objekterkennung spezialisierten Machine Learning-Modell Faster R-CNN bestimmt. Hierzu wurden das 3D-Stadtmodell von Hamburg, ALKIS und das digitale Schrägluftbild Hamburgs verwendet. Das ML-Modell wurde am CMP-Datensatz auf die Fenstererkennung trainiert. Die Bewertung des Trainings erfolgte mit der *Mean Average Precision (mAP)*, um eine gute Vergleichbarkeit mit der Literatur zu ermöglichen. Das Modell wurde an einem in dieser Arbeit erstellten Testdatensatz, dem Hamburg Image Dataset, getestet, der aus 44 gelabelten Bildern von Hamburger Gebäudefassaden besteht. Die Auswertung des Trainings zeigte, dass das Modell deutlich schlechtere Ergebnisse liefert als vergleichbare Modelle aus der Literatur und zu Overfitting neigt. Das Potential des Modells wurde demnach nicht vollständig ausgeschöpft, was entweder daran liegen könnte, dass der CMP-Datensatz nicht für das Untersuchungsgebiet in Hamburg geeignet ist oder dass die Hyperparameter ungünstig gewählt wurden. Auch eine Nachbearbeitung konnte das *mAP* nicht verbessern. Die für Wärmebedarfsprognosen relevante Größe ist jedoch nicht *mAP*, sondern die Abweichung des auf Basis des ML-Modells ermittelten $R_{\text{Gebäude}}$ von $R_{\text{Gebäude, HI}}$, welches auf den Labels des Hamburg Image Datasets beruht. Diese konnte durch Nachbearbeitung verringert werden. Sie beträgt $1,21 \% \pm 20,24 \%$ für die in dieser Arbeit vorgestellte Methodik und ist damit geringer als die Abweichung, die die Werte der IWU-Gebäudetypologie zu $R_{\text{Gebäude, HI}}$ aufweisen und die $-84,04 \% \pm 71,07 \%$ beträgt. Trotz schlechtem *mAP* ist die Methodik dieser Masterarbeit also für die Bestimmung von $R_{\text{Gebäude}}$ zur Vorbereitung einer Wärmebedarfsprognose geeignet. Da für die Ermittlung der solaren Gewinne, die für eine zeitlich hochaufgelöste Wärmebedarfsprognose notwendig sind, die Ausrichtung der Fenster entscheidend ist, muss jedoch berücksichtigt werden, dass die Fensterfläche in Seitenflächen eines Gebäudes tendenziell als zu groß eingeschätzt wird, die Fensterfläche in Vorder- und Rückseiten hingegen als zu klein. Darüber hinaus muss beachtet werden, dass die Ergebnisse der Bestimmung von $R_{\text{Gebäude}}$ durch Ungenauigkeiten des Hamburg Image Datasets beeinflusst wurden, die auf ein ungenaues 3D-Stadtmodell, eine schwankende grafische Qualität der Bilder der Gebäudefassaden und das händische Labeln zurückzuführen sind.

Die Bestimmung der Baualtersklassen von Gebäuden erfolgte über den Vergleich historischer Karten Hamburgs mit den im ALKIS vermerkten Gebäudegrundrissen. Hierbei wurden zunächst die Baujahre bestimmt, aus denen anschließend die Baualtersklassen entsprechend der IWU-Wohngebäudetypologie abgeleitet werden konnten. Die Bewertung der Methode erfolgte anhand eines in dieser Arbeit erstellten Testdatensatzes, der die händisch ermittelten Baujahre von 247 Hamburger Gebäuden umfasst. Bezüglich der Baujahresbestimmung konnte eine Genauigkeit von $2,7 \pm 27,1$ Jahren erreicht werden. Die Genauigkeit bezüglich der Einordnung der Gebäude in die Baualtersklassen wurde mit *mAP* bewertet und betrug 24,1 %. Als Ursachen für die großen Ungenauigkeiten bei der Baujahres- und Baualtersklassenbestimmung werden die schlechte grafische Qualität von Karten vor 1950 und eine unvollständige Abdeckung des Hamburger Stadtgebietes durch die Karten vor 1960 vermutet. Darüber hinaus stößt die Methodik bei Gebäuden an ihre Grenze, die im zweiten Weltkrieg zerstört und in den späten 1940er- und 1950er-Jahren wiederaufgebaut wurden. Auch abgerissene Gebäude stellen für die Methodik eine Herausforderung dar. Insgesamt zeigen die großen Unsicherheiten, sowohl bezüglich der Baujahres- als auch der Baualtersklassenabschätzung, dass die in dieser Arbeit verwendete Methodik unzureichende Ergebnisse für eine detaillierte Wärmebedarfsprognose liefert. In Kombination mit anderen Methoden aus der Literatur (siehe Kapitel 2.1.3) könnte sie jedoch wichtige Impulse geben, um einem Gebäude eine Baualtersklasse zuordnen können.

Die in dieser Masterarbeit vorgestellte Methode zur Bestimmung der Fensterfläche eines Gebäude lässt sich auf alle anderen deutschen Bundesländer übertragen, wobei die 3D-Stadtmodelle nicht immer kostenfrei zugänglich sind und Schrägluftbilder in allen Ländern außer Hamburg und Bremen über die Funktion „Vogelperspektive“ von Bing Maps bezogen werden müssen. Die Methode zur Bestimmung der Baualtersklasse eines Gebäudes lässt sich im Gegensatz dazu nur bedingt auf andere Bundesländer übertragen, da historische Karten oder Luftbildaufnahmen nur für 8 der 16 Bundesländer öffentlich zugänglich sind und nicht immer zeitlich oder räumlich lückenlos vorliegen.

6 Ausblick

Die in dieser Arbeit vorgestellten Ansätze zur Bestimmung des Verhältnisses der Fenster- zur Wandfläche und Baualtersklasse eines Gebäudes bieten viele Möglichkeiten zur Weiterentwicklung.

So kann das ML-Modell, das die Basis für die Bestimmung von $R_{\text{Gebäude}}$ bildet, durch weiteres Training noch besser an die Aufgabe angepasst werden. Hierzu können die Hyperparameter Learning Rate, Momentum und Weight Decay systematisch optimiert werden. Gleiches gilt für die Regeln und variablen Größen bei der Nachbearbeitung der Ergebnisse des ML-Modells. Es wird außerdem davon ausgegangen, dass sich die Ergebnisse des ML-Modells verbessern ließen, wenn neben den Wandflächen eines Gebäudes auch die Dachflächen mit einbezogen würden. Auf diese Weise könnte die Anzahl falsch erkannter Fenster reduziert und die Genauigkeit der gesamten Fensterfläche eines Gebäudes erhöht werden. Darüber hinaus könnte das Hamburg Image Dataset erweitert und in das Training des ML-Modells integriert werden. Auf diese Weise könnte eine Spezialisierung des Faster R-CNN-Modells auf Hamburger Gebäude erfolgen. Hierzu wäre es jedoch wichtig, auch die Genauigkeit der Label des Hamburg Image Datasets zu erhöhen. Dies könnte durch das Auswerten von Bauplänen oder das Ausmessen von Gebäuden vor Ort erfolgen.

Bezüglich der Bestimmung der Baualtersklasse eines Gebäudes könnte die Genauigkeit der Ergebnisse unter Umständen erhöht werden, indem alte historische Karten mit schlechter grafischer Qualität händisch aufbereitet oder in höherer Auflösung neu eingescannt würden. Auch die Richtung, in der die Karten auf der Suche nach einer Übereinstimmung zwischen Grundriss-Polygon und erkannten Konturen durchgegangen werden, könnte geändert werden. Auf diese Weise würden unter Umständen abgerissene Gebäude nicht fälschlicherweise erkannt werden. Zu diesem Zweck könnte auch zur Beurteilung der Übereinstimmung zwischen Grundriss-Polygon und Konturen statt dem absoluten IoU_{min} die Veränderung von IoU_{min} zwischen zwei Kartenjahrgängen herangezogen werden. Eine Einbeziehung der Nachbargebäude ist ebenfalls vielversprechend und wird in der Literatur häufig verwendet. Darüber hinaus könnte eine Definition von Gebieten in Hamburg, die im zweiten Weltkrieg Ziel von Bombenangriffen waren, sowie deren Einbeziehung in die Bestimmung der Baualtersklasse eine Möglichkeit sein, die Genauigkeit der Methodik zu erhöhen.

Unabhängig von den vorgestellten Methoden können das Hamburg Image Dataset und der Testdatensatz zur Baualtersklassenbestimmung als Basis für weitere Untersuchungen im norddeutschen Raum genutzt werden.

7 Literaturverzeichnis

- [1] Ayenew, M. *Towards Large Scale Façade Parsing: A Deep Learning Pipeline Using Mask R-CNN*.
- [2] Baugenossenschaft der Buchdrucker eG. 2023. *Übersicht - Baugenossenschaft der Buchdrucker eG. Günstige Wohnungen in Hamburg mieten!* <https://www.buchdrucker.de/de/gesamtbestand>. Accessed 18 April 2023.
- [3] Baugenossenschaft dhu eG. 2021. *Nord - Baugenossenschaft dhu eG*. <https://dhu.hamburg/wohnen/wohnungsbestand/nord/>. Accessed 18 April 2023.
- [4] Baugenossenschaft Süderelbe. 2018. *Wohnungsbestand - Baugenossenschaft Süderelbe*. <https://www.baugen-suederelbe.de/wohnungsbestand/>. Accessed 18 April 2023.
- [5] Bauverein Reiherstieg eG. 2023. *Objektbestand / Bauverein Reiherstieg eG*. <https://www.reiherstieg.de/objektbestand/>. Accessed 18 April 2023.
- [6] Biljecki, F. and Dehbi, Y. 2019. RAISE THE ROOF: TOWARDS GENERATING LOD2 MODELS WITHOUT AERIAL SURVEYS USING MACHINE LEARNING. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* IV-4/W8, 27–34.
- [7] Biljecki, F., Ledoux, H., and Stoter, J. 2016. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems* 59, 25–37.
- [8] Biljecki, F. and Sindram, M. 2017. ESTIMATING BUILDING AGE WITH 3D GIS. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* IV-4/W5, 17–24.
- [9] Boeters, R. 2013. *Automatic enhancement of CityGML LoD2 models with interiors and its usability for net internal area determination*, TU Delft.
- [10] Borrmann, A., König, M., Koch, C., and Beetz, J. 2018. *Building Information Modeling*. Springer International Publishing, Cham.
- [11] Bradski, G. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- [12] Brownlee, J. 2019. How to use Learning Curves to Diagnose Machine Learning Model Performance. *Machine Learning Mastery* (Feb. 2019).
- [13] Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. 2020. Albumentations: Fast and Flexible Image Augmentations. *Information* 11, 2, 125.
- [14] Carrillo-Perez, B., Barnes, S., and Stephan, M. 2022. Ship Segmentation and Georeferencing from Static Oblique View Images. *Sensors (Basel, Switzerland)* 22, 7.
- [15] CityJSON. 2022. *What is CityJSON?* <https://www.cityjson.org/about/>. Accessed 20 July 2022.
- [16] Cléri, I., Pierrot-Deseilligny, M., and Vallet, B. 2014. Automatic Georeferencing of a Heritage of old analog aerial Photographs. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* II-3, 33–40.

- [17] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009 ; 20 - 25 June 2009, Miami [Beach], FL, USA*. IEEE, Piscataway, NJ, 248–255. DOI=10.1109/CVPR.2009.5206848.
- [18] *Denkmalkartei des Referats Inventarisierung im Denkmalschutzamt Hamburg. Adressen: Weimarer Straße 43 u. Veringstraße 149 (Auskunft auf Anfrage im April 2023).*
- [19] 2017. *DIN EN ISO 10077-1:2020-10, Wärmetechnisches Verhalten von Fenstern, Türen und Abschlüssen - Berechnung des Wärmedurchgangskoeffizienten. Teil 1: Allgemeines, (ISO_10077-1:2017, korrigierte Fassung 2020-02); Deutsche Fassung EN_ISO_10077-1:2017*. Beuth Verlag GmbH, Berlin.
- [20] Dochev, I., Gorzalka, P., Weiler, V., Estevam Schmiedt, J., Linkiewicz, M., Eicker, U., Hoffschmidt, B., Peters, I., and Schröter, B. 2020. Calculating urban heat demands: An analysis of two modelling approaches and remote sensing for input data and validation. *Energy and Buildings* 226, 110378.
- [21] Dochev, I., Seller, H., and Peters, I. 2019. *Assigning energetic archetypes to a digital cadastre and estimating building heat demand. An example from Hamburg, Germany*. Unpublished.
- [22] 2005. *Emerging topics in computer vision*. Prentice Hall, Upper Saddle River, NJ.
- [23] Freie und Hansestadt Hamburg, Landesbetrieb Geoinformation und Vermessung. 1925 - 2016. *Historische Karten 1: 5000 - Auswahl*. <https://metaver.de/trefferanzeige?docuuiid=7A32A730-8B7A-4BB1-9D7F-D0BB37805696>. Accessed 08.11.2022.
- [24] Freie und Hansestadt Hamburg, Landesbetrieb Geoinformation und Vermessung. 2022. *Geo-Online*. <https://geoportal-hamburg.de/geo-online/>. Accessed 6 September 2022.
- [25] Freie und Hansestadt Hamburg, Landesbetrieb Geoinformation und Vermessung. 2022. *Digitales Schrägluftbild Hamburg*. <https://metaver.de/trefferanzeige?docuuiid=87D131E6-1D27-4C38-828E-B3DBEBFF8CE6>. Accessed 4 November 2022.
- [26] Freie und Hansestadt Hamburg, Landesbetrieb Geoinformation und Vermessung. 2022. *3D-Stadtmodell LoD2-DE Hamburg*. <https://metaver.de/trefferanzeige?docuuiid=2C1F2EEC-CF9F-4D8B-ACAC-79D8C1334D5E>. Accessed 21 November 2022.
- [27] Freie und Hansestadt Hamburg, Landesbetrieb Geoinformation und Vermessung. 2022. *Collections - ALKIS - Liegenschaften und Verwaltungseinheiten*. <https://qs-api.hamburg.de/datasets/v1/alkis vereinfacht/collections>. Accessed 21 November 2022.
- [28] Gadde, R., Marlet, R., and Paragios, N. 2016. Learning Grammars for Architecture-Specific Facade Parsing. *Int J Comput Vis* 117, 3, 290–316.
- [29] Gillies, S. 2023. *Shapely Python package for computational geometry*.

- [30] Girshick, R. 2015. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE. DOI=10.1109/iccv.2015.169.
- [31] Goodfellow, I., Courville, A., and Bengio, Y. 2016. *Deep learning*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts.
- [32] hamburg.de. 2023. *Hamburger Sturmflut 1962*. <https://www.hamburg.de/sturmflut-1962/>. Accessed 31 March 2023.
- [33] Hastie, T., Friedman, J., and Tibshirani, R. 2001. *The Elements of Statistical Learning*. Springer New York, New York, NY.
- [34] Hessische Verwaltung für Bodenmanagement und Geoinformation. 2023. *Geodaten online - Downloadcenter*. https://gds.hessen.de/INTERSHOP/web/WFS/HLBG-Geodaten-Site/de_DE/-/EUR/ViewDownloadcenter-Start. Accessed 9 May 2023.
- [35] Hong, T., Chen, Y., Luo, X., Luo, N., and Lee, S. H. 2020. Ten questions on urban building energy modeling. *Building and Environment* 168, 106508.
- [36] 2022. *ImgLab - Image Annotation tool*. <https://imglab.in/>. Accessed 15 March 2023.
- [37] Kaden, R. *Berechnung der Energiebedarfe von Wohngebäuden und Modellierung energiebezogener Kennwerte auf der Basis semantischer 3D-Stadtmodelle*, Technische Universität München.
- [38] Korc, F. and Förstner, W. 2009. *eTRIMS Image Database for interpreting images of man-made scenes*. University of Bonn, Bonn.
- [39] Kraus, K. 2007. *Photogrammetry. Geometry from images and laser scans*. De Gruyter Textbook. de Gruyter, Berlin.
- [40] Landesamt für Digitalisierung, Breitband und Vermessung Bayern. 2023. *OpenData*. <https://geodaten.bayern.de/opengeodata/OpenDataDetail.html?pn=lod2>. Accessed 9 May 2023.
- [41] Landesamt für Geobasisinformation Sachsen. 2021. *Downloadbereich Digitale 3D-Stadtmodelle - Offene Geodaten - sachsen.de*. <https://www.geodaten.sachsen.de/downloadbereich-digitale-3d-stadtmodelle-4875.html>. Accessed 9 May 2023.
- [42] Landesamt für Geoinformation und Landentwicklung Baden-Württemberg. 2023. *3D-Gebäudemodell LoD2*. <https://www.lgl-bw.de/Produkte/3D-Produkte/3D-Gebaeudemodelle/LoD2/>. Accessed 9 May 2023.
- [43] Landesamt für Geoinformation und Landentwicklung Baden-Württemberg. 2023. *Startseite - Geoportal BW*. <https://www.geoportal-bw.de/>. Accessed 9 May 2023.
- [44] Landesamt für Geoinformation und Landesvermessung Niedersachsen. 2023. *OpenGeoData.NI*. <https://opengeodata.lgln.niedersachsen.de/>. Accessed 9 May 2023.

- [45] Landesamt für innere Verwaltung Mecklenburg-Vorpommern. 2023. *Gebäudemodelle*. <https://www.laiv-mv.de/Geoinformation/Geobasisdaten/gebaeude%E2%80%93modelle/>. Accessed 9 May 2023.
- [46] Landesamt für innere Verwaltung Mecklenburg-Vorpommern. 2023. *Themenkarte - Details - GeoPortal Mecklenburg-Vorpommern*. <https://www.geoportal-mv.de/portal/Suche/Themenkarten/Details?id=11>. Accessed 9 May 2023.
- [47] Landesamt für Vermessung und Geobasisinformation Rheinland-Pfalz. 2022. *GeoBasisViewer RLP*. <https://maps.rlp.de/?layerIDs=172&visibility=true&transparency=0¢er=400215.02547731064,5579808.374800296&zoomlevel=9>. Accessed 9 May 2023.
- [48] Landesamt für Vermessung und Geobasisinformation Rheinland-Pfalz. 2023. *3D-Gebäudemodell*. <https://lvermgeo.rlp.de/de/produkte/geotopografie/3d-geodaten/3d-gebaeudemodell/>. Accessed 9 May 2023.
- [49] Landesamt für Vermessung und Geoinformation Sachsen-Anhalt. 2023. *Kostenfreie Geobasisdaten*. https://www.lvermgeo.sachsen-anhalt.de/de/kostenfreie_geobasisdaten_lvermgeo.html. Accessed 9 May 2023.
- [50] Landesamt für Vermessung und Geoinformation Sachsen-Anhalt. 2023. *Kostenfreies 3D-Gebäudemodell - LoD2*. https://www.lvermgeo.sachsen-anhalt.de/de/download_lod2.html. Accessed 9 May 2023.
- [51] Landesamt für Vermessung und Geoinformation Schleswig-Holstein. 2022. *Schleswig-Holstein Downloadportal*. https://geodaten.schleswig-holstein.de/gaialight-sh/_apps/dldownload/dl-lod2.html. Accessed 9 May 2023.
- [52] Landesamt für Vermessung, Geoinformation und Landentwicklung Saarland. 2023. *Digitale Geländemodelle*. https://www.shop.lvgl.saarland.de/index.php?option=com_virtuemart&view=category&virtuemart_category_id=1098&Itemid=156. Accessed 9 May 2023.
- [53] Landesamt Geoinformation Bremen. 2023. *3D-Gebäudemodelle*. <https://www.geo.bremen.de/produkte/3d-produkte/3d-gebaeudemodelle-11892>. Accessed 9 May 2023.
- [54] Landesamt Geoinformation Bremen. 2023. *Luftbildprodukte*. <https://www.geo.bremen.de/produkte/luftbildprodukte-11703>. Accessed 9 May 2023.
- [55] Landesbetrieb Geoinformation und Vermessung (LGV) Hamburg. 1925 - 2016. *WMS Historische Karte 1: 5000. Datenlizenz Deutschland – Namensnennung – Version 2.0*. <https://metaver.de/trefferanzeige?docuuid=7CF3384C-6709-4169-A048-2ADA6112A804>. Accessed 19 September 2022.

- [56] Landesbetrieb Information und Technik Nordrhein-Westfalen. 2023. *3D-Gebäudemodell LoD2 - Paketierung: Einzelkacheln*. https://www.opengeodata.nrw.de/produkte/geobasis/3dg/lod2_gml/lod2_gml/. Accessed 9 May 2023.
- [57] Landesvermessung und Geobasisinformation Brandenburg. 2023. *3D-Gebäude - Gebietsauswahl | Geobroker - Der Internetshop der LGB*. <https://geobroker.geobasis-bb.de/basiskarte.php?mode=startup&aProductId=0414a37a-a749-4ee6-9f59-a41226919c58>. Accessed 9 May 2023.
- [58] LaValle, S. M. 2009. *Planning Algorithms*. Cambridge University Press, Cambridge.
- [59] Ledoux, H., Arroyo Ohori, K., Kumar, K., Dukai, B., Labetski, A., and Vitalis, S. 2019. CityJSON: a compact and easy-to-use encoding of the CityGML data model. *Open geospatial data, softw. stand.* 4, 1, 1–12.
- [60] Lenner, B. and Luerweg, T. 2021. Juli 1943: Feuersturm vernichtet Hamburg. *NDR* (Feb. 2021).
- [61] Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., and Stoica, I. 2018. *Tune: A Research Platform for Distributed Model Selection and Training*.
- [62] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. 2014. *Microsoft COCO: Common Objects in Context*.
- [63] Lippoldt, F. 2019. Window Detection in Facades for Aerial Texture Files of 3D CityGML Models.
- [64] Loga, T., Stein, B., Diefenbach, N., and Born, R. 2015. *Deutsche Wohngebäudetypologie. Beispielhafte Maßnahmen zur Verbesserung der Energieeffizienz von typischen Wohngebäuden ; erarbeitet im Rahmen der EU-Projekte TABULA - "Typology approach for building stock energy assessment", EPISCOPE - "Energy performance indicator tracking schemes for the continuous optimisation of refurbishment processes in European housing stocks"*. IWU, Darmstadt.
- [65] Ma, W. and Ma, W. 2020. Deep Window Detection in Street Scenes. *KSII TIS* 14, 2.
- [66] Malhotra, A., Raming, S., Frisch, J., and van Treeck, C. 2021. Open-Source Tool for Transforming CityGML Levels of Detail. *Energies* 14, 24, 8250.
- [67] Microsoft. 2023. *Get a Static Map - Bing Maps*. <https://learn.microsoft.com/en-us/bingmaps/rest-services/imagery/get-a-static-map>. Accessed 9 May 2023.
- [68] Microsoft Bing. 2023. *Bing Karten - Wegbeschreibungen, Reiseplanung, Verkehrskameras und mehr*. <https://www.bing.com/maps?FORM=LGCYVD&cp=53.558512%7E10.010648&lvl=19.0&style=g>. Accessed 9 May 2023.

- [69] Nippgen, P. *Nutzung von 3D-Stadtmodellen für die Wärmebedarfsermittlung am Beispiel Hamburgs. Nutzung von 3D-Stadtmodellen für die Wärmebedarfsermittlung am Beispiel Hamburgs*, HafenCity Universität Hamburg.
- [70] Nordmark, N. and Ayenew, M. 2021. *Window Detection In Facade Imagery: A Deep Learning Approach Using Mask R-CNN*. arXiv.
- [71] Nussbaumer, T., Thalmann, S., Jenni, A., and Ködel, J. 2018. *Planungshandbuch Fernwärme*. EnergieSchweiz Bundesamt für Energie BFE, Ittigen.
- [72] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.
- [73] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, 85, 2825–2830.
- [74] Presse- und Informationsamt der Bundesregierung. 2023. *Klimaschutzgesetz: Klimaneutralität bis 2045 | Bundesregierung*. <https://www.bundesregierung.de/breg-de/themen/klimaschutz/klimaschutzgesetz-2021-1913672>. Accessed 4 May 2023.
- [75] Ren, S., He, K., Girshick, R., and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 6, 1137–1149.
- [76] Riemenschneider, H., Bódis-Szomorú, A., Weissenberg, J., and van Gool, L. Learning Where to Classify in Multi-view Semantic Segmentation. In *ECCV 2014, Part V*, 516–532. DOI=10.1007/978-3-319-10602-1_34.
- [77] Riemenschneider, H., Krispel, U., Thaller, W., Donoser, M., Havemann, S., Fellner, D., and Bischof, H. 2012. Irregular lattices for complex shape grammar facade parsing. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1640–1647. DOI=10.1109/cvpr.2012.6247857.
- [78] Ryan, T. P. 2007. *Modern engineering statistics*. Wiley-Interscience, Hoboken, N.J.
- [79] Schaub, H. and Junkins, J. L. 2018. *Analytical mechanics of space systems*. AIAA education series. American Institute of Aeronautics and Astronautics Inc(AIAA), Reston, VA.
- [80] Senatsverwaltung für Inneres, Digitalisierung und Sport Berlin. 2023. *LoD2 Gebäudedaten Berlin | Offene Daten Berlin*. <https://daten.berlin.de/datensaetze/lod2-geb%C3%A4udedaten-berlin>. Accessed 9 May 2023.

- [81] Senatsverwaltung für Stadtentwicklung, Bauen und Wohnen Berlin. *FIS Broker*. https://fbinter.stadt-berlin.de/fb/?loginkey=showMap&mapId=k_luftbild1994_rgb@senstadt. Accessed 9 May 2023.
- [82] 2020. *Statistisches Jahrbuch Hamburg, Berichtsjahr 2020. Kapitel 5: Bauen und Wohnen*.
- [83] Sun, Y., Malihi, S., Li, H., and Maboudi, M. 2022. DeepWindows: Windows Instance Segmentation through an Improved Mask R-CNN Using Spatial Attention and Relation Modules. *IJGI* 11, 3, 162.
- [84] Swan, L. G. and Ugursal, V. I. 2009. Modeling of end-use energy consumption in the residential sector: A review of modeling techniques. *Renewable and Sustainable Energy Reviews* 13, 8, 1819–1835.
- [85] Teboul, O., Simon, L., Koutsourakis, P., and Paragios, N. 2010. *Segmentation of Building Facades Using Procedural Shape Priors*. San Francisco, California, USA, 13 - 18 June 2010. IEEE, Piscataway, NJ.
- [86] Thüringer Landesamt für Bodenmanagement und Geoinformation. 2023. *Download 3D Gebäude*. <https://www.geoportal-th.de/de-de/Downloadbereiche/Download-Offene-Geodaten-Th%C3%BCrigen/Download-3D-Geb%C3%A4ude>. Accessed 9 May 2023.
- [87] Thüringer Landesamt für Bodenmanagement und Geoinformation. 2023. *Download Luftbilder und Orthophotos*. <https://www.geoportal-th.de/de-de/Downloadbereiche/Download-Offene-Geodaten-Th%C3%BCrigen/Download-Luftbilder-und-Orthophotos>. Accessed 9 May 2023.
- [88] Tyleček, R. 2013. *CMP Facade Database*. Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University.
- [89] Umweltbundesamt. 2023. *Energieverbrauch für fossile und erneuerbare Wärme*. <https://www.umweltbundesamt.de/daten/energie/energieverbrauch-fuer-fossile-erneuerbare-waerme>. Accessed 4 May 2023.
- [90] Umweltbundesamt. 2023. *Energieverbrauch nach Energieträgern und Sektoren*. <https://www.umweltbundesamt.de/daten/energie/energieverbrauch-nach-energietraegern-sektoren>. Accessed 4 May 2023.
- [91] Verykokou, S. A. 2020. *Georeferencing Procedures for Oblique Aerial Images*. National Technological University of Athens.
- [92] Wang, L. 2022. *Detailed Facade Reconstruction for Mahattan-world Buildings*.
- [93] Zhuo, X., Tian, J., and Fraundorfer, F. 2023. Cross Field-Based Segmentation and Learning-Based Vectorization for Rectangular Windows. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing* 16, 431–448.

Anhang

Anhang A

Datengrundlage in den deutschen Bundesländern

Tabelle 11: Übersicht über die Verfügbarkeit von 3D-Gebäudemodellen, Schrägluftbildern und historischen Karten oder Luftbildern in Deutschland.

Bundesland	3D-Gebäudemodelle				Schrägluftbilder	Historische Karten oder Luftbildaufnahmen
	LoD	Zugänglich via Download	Zugänglich via Bestellung	Kostenpflichtig		
Baden-Württemberg [42, 43]	2		x	x		x (Luftbilder)
Bayern [40]	2	x				
Berlin [80, 81]	2	x				x (Karten und Luftbilder)
Brandenburg [57]	2	x				x (Karten und Luftbilder)
Bremen [53, 54]	2		x		x	x (Luftbilder)
Hamburg [23, 25, 26]	2	x			x	x (Karten)
Hessen [34]	2	x				
Mecklenburg-Vorpommern [45, 46]	2		x			x (Karten)
Niedersachsen [44]	2	x				
Nordrhein-Westfalen [56]	2	x				

Rheinland-Pfalz [47, 48]	2		x	x		x (Karten und Luftbilder)
Saarland [52]	2		x	x		
Sachsen-Anhalt [49, 50]	2	x				x (Luftbilder)
Sachsen [41]	2	x				
Schleswig-Holstein [51]	2	x				
Thüringen [86, 87]	2	x				x (Luftbilder)

Anhang B

Umformung der Kameragleichung

Die Kameragleichung (Gleichung [2.4])

$$\lambda x_c = K * R_{rot}^T * [I | -t] * X_c$$

kann umgeformt werden zu:

$$\lambda \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} = \begin{bmatrix} c_{cam} & 0 & x_K \\ 0 & c_{cam} & y_K \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^T * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} -X_t \\ -Y_t \\ -Z_t \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Dies führt zu:

$$\begin{aligned} \lambda \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} &= \begin{bmatrix} ac_{cam} + cx_K & dc_{cam} + fx_K & gc_{cam} + ix_K & X_t(ac_{cam} + cx_K) + Y_t(dc_{cam} + fx_K) + Z_t(gc_{cam} + ix_K) \\ bc_{cam} + cy_K & ec_{cam} + fy_K & hc_{cam} + iy_K & X_t(bc_{cam} + cy_K) + Y_t(ec_{cam} + fy_K) + Z_t(hc_{cam} + iy_K) \\ c & f & i & X_t c + Y_t f + Z_t i \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\ &= \begin{bmatrix} T_1 & T_2 & T_3 & T_4 \\ T_5 & T_6 & T_7 & T_8 \\ T_9 & T_{10} & T_{11} & T_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \end{aligned}$$

Durch Ausmultiplizieren beider Seiten folgt:

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} T_1 X + T_2 Y + T_3 Z + T_4 \\ T_5 X + T_6 Y + T_7 Z + T_8 \\ T_9 X + T_{10} Y + T_{11} Z + T_{12} \end{bmatrix}$$

Die Gleichungen [2.9] und [2.10] können hieraus folgendermaßen gebildet werden:

$$\frac{\lambda x}{\lambda} = \frac{T_1 X + T_2 Y + T_3 Z + T_4}{T_9 X + T_{10} Y + T_{11} Z + T_{12}} = x$$

$$\frac{\lambda y}{\lambda} = \frac{T_5 X + T_6 Y + T_7 Z + T_8}{T_9 X + T_{10} Y + T_{11} Z + T_{12}} = y$$

Anhang C

Fenster- und Wandflächen für die IWU-Wohngebäudetypen

Die Flächen $A_{Fenster\ 1}$, $A_{Fenster\ 2}$, $A_{Wand\ 1}$, $A_{Wand\ 2}$ und $A_{Tür}$ sind in der IWU-Wohngebäudetypologie angegeben [64]. $A_{Fenster\ gesamt}$ und $A_{Wand\ gesamt}$ wurden berechnet mit

$$A_{Fenster\ gesamt} = A_{Fenster\ 1} + A_{Fenster\ 2}$$

$$A_{Wand\ gesamt} = A_{Wand\ 1} + A_{Wand\ 2} + A_{Tür}$$

Aus diesen Werten wurde wiederum das Verhältnis der Fenster- zur Wandfläche $R_{Gebäude, IWU}$ berechnet mit

$$R_{Gebäude, IWU} = \frac{A_{Fenster\ gesamt}}{A_{Wand\ gesamt}}$$

Tabelle 12: Fenster- und Wandflächen für die IWU-Wohngebäudetypen [64], erweitert um das Verhältnis der Fenster- zur Wandfläche.

Größen- klasse	Baualters- klasse	$A_{Fenster\ 1}$ [m²]	$A_{Fenster\ 2}$ [m²]	$A_{Wand\ 1}$ [m²]	$A_{Wand\ 2}$ [m²]	$A_{Tür}$ [m²]	$A_{Fenster\ gesamt}$ [m²]	$A_{Wand\ gesamt}$ [m²]	$R_{Gebäude, IWU}$ [m²/m²]
EFH_A	... 1859	28.78	0	169.78	0	2	28.78	171.78	0.168
EFH_B	1860 ... 1918	22.3	0	194.04	0	2	22.3	196.04	0.114
EFH_C	1919 ... 1948	52.35	0	235.3	0	2	52.35	237.3	0.221
EFH_D	1949 ... 1957	18.4	0	117.8	0	2	18.4	119.8	0.154
EFH_E	1958 ... 1968	27.1	0	141.2	8.7	2.1	27.1	152	0.178
EFH_F	1969 ... 1978	34.21	0	177.55	0	2	34.21	179.55	0.191
EFH_G	1979 ... 1983	27	0	159.4	0	2	27	161.4	0.167

EFH_H	1984 ... 1994	29.67	0	211.3	0	2	29.67	213.3	0.139
EFH_I	1995 ... 2001	32.46	0	126.6	0	2	32.46	128.6	0.252
EFH_J	2002 ... 2009	28.3	0	188.86	0	2	28.3	190.86	0.148
EFH_K	2010 ... 2015	42	0	227.6	0	2.6	42	230.2	0.182
EFH_L	2016 ...	42	0	227.6	0	2.6	42	230.2	0.182
RH_B	1860 ... 1918	18.091	0	74.47	0	2	18.091	76.47	0.237
RH_C	1919 ... 1948	21.48	0	64.14	0	2	21.48	66.14	0.325
RH_D	1949 ... 1957	46.67	0	134.66	0	2	46.67	136.66	0.342
RH_E	1958 ... 1968	13.52	0	40.42	0	2	13.52	42.42	0.319
RH_F	1969 ... 1978	23.36	0	53.72	0	2	23.36	55.72	0.419
RH_G	1979 ... 1983	20.25	0	54.1	0	2	20.25	56.1	0.361
RH_H	1984 ... 1994	18.75	0	50.9	0	2	18.75	52.9	0.354
RH_I	1995 ... 2001	22.4	0	45.2	13.9	2.5	22.4	61.6	0.364
RH_J	2002 ... 2009	36.3	0	140.7	0	2	36.3	142.7	0.254
RH_K	2010 ... 2015	25.5	2.6	137.8	69.2	2.7	28.1	209.7	0.134
RH_L	2016 ...	25.5	2.6	137.8	69.2	2.7	28.1	209.7	0.134
MFH_A	... 1859	106.97	0	749.31	0	2	106.97	751.31	0.142
MFH_B	1860 ... 1918	54.08	0	146	0	2	54.08	148	0.365
MFH_C	1919 ... 1948	71.16	0	323.54	0	2	71.16	325.54	0.219
MFH_D	1949 ... 1957	98.72	0	462	0	2	98.72	464	0.213
MFH_E	1958 ... 1968	507.48	0	2039	0	2	507.48	2041	0.249
MFH_F	1969 ... 1978	81.3	0	336	0	2	81.3	338	0.241

MFH_G	1979 ... 1983	99.37	0	447.13	0	2	99.37	449.13	0.221
MFH_H	1984 ... 1994	161	0	774.8	0	2	161	776.8	0.207
MFH_I	1995 ... 2001	162.8	0	695.8	0	2	162.8	697.8	0.233
MFH_J	2002 ... 2009	308.7	0	1698	0	2	308.7	1700	0.182
MFH_K	2010 ... 2015	243.6	0	1193.16	0	47.85	243.6	1241.01	0.196
MFH_L	2016 ...	243.6	0	1193.16	0	47.85	243.6	1241.01	0.196
GMH_B	1860 ... 1918	136.2	0	305.4	0	2	136.2	307.4	0.443
GMH_C	1919 ... 1948	278.52	0	1244	0	2	278.52	1246	0.224
GMH_D	1949 ... 1957	294.9	0	1376	0	2	294.9	1378	0.214
GMH_E	1958 ... 1968	687.03	0	3247.79	0	2	687.03	3249.79	0.211
GMH_F	1969 ... 1978	545	0	2130	0	2	545	2132	0.256
HH_E	1958 ... 1968	1947.19	0	6949.14	0	2	1947.19	6951.14	0.280
HH_F	1969 ... 1978	2580.52	0	14623.4	0	2	2580.52	14625.4	0.176
EFH_F/F	1969 ... 1978	32	0	117	0	2.2	32	119.2	0.268
NBL_MFH_D	1949 ... 1957	319.9	0	1158.16	0	2	319.9	1160.16	0.276
NBL_MFH_E	1958 ... 1968	547	0	1480.48	0	2	547	1482.48	0.369
NBL_GMH_F	1969 ... 1978	461	0	1599.73	0	2	461	1601.73	0.288
NBL_GMH_G	1979 ... 1983	387	0	1673.73	0	2	387	1675.73	0.231
NBL_GMH_H	1984 ... 1994	387	0	1673.73	0	2	387	1675.73	0.231
NBL_HH_F	1969 ... 1978	756	0	2992.09	0	2	756	2994.09	0.252
NBL_HH_G	1979 ... 1983	1471.06	0	4221.74	0	2	1471.06	4223.74	0.348

Anhang D

Trainingsdatensätze für Fensterflächenerkennung

Tabelle 13: Auswahl von Bild-Datensätzen, die zum Training von ML-Algorithmen zur Fensterflächenerkennung genutzt wurden.

Nr.	Name	Umfang in Bildern	Stadt / Land	Jahr	Art der Bilder	Bildsegmentierung
1	CityGML Berlin [63]	7000	Berlin	2006	Schrägluftbilder	Semantisch
2	Street Scene Window Detection Dataset [65]	1300	Paris	2012	Straßenansichten	Objektbezogen
3	Dataset Nordmark + Ayenew [70]	100	Göteborg	2020	Straßenansichten	Semantisch
4	CMP [88]	606	Prag, Bratislava, Buenos Aires, Frankfurt, Graz, London, Ostrava, Rom, Znojmo, Zürich, Barcelona, Griechenland, Budapest, USA	2003 - 2010	Straßenansichten, rektifiziert	Objektbezogen
5	eTRIMS [38]	60	Basel, Bonn, Berlin, Heidelberg, Karlsruhe, UK, München, Prag, Hamburg	2009	Straßenansichten	Semantisch
6	ECP [85]	104	Paris	2010	Straßenansichten, rektifiziert	Semantisch
7	ICG Graz50 [77]	50	Graz	2012	Straßenansichten, rektifiziert	Semantisch

8	ETHZ CVL RueMonge 2014 [76]	434	Paris	2014	Straßenansichten	Semantisch
9	ParisArt-Deco [28]	79	Paris	2014	Straßenansichten, rektifiziert	Semantisch

Anhang E

Beispielbilder der Gebäude des Hamburg Image Datasets



Abbildung 50: Fassade der Mengestraße 6



Abbildung 51: Fassade der Weimarer Straße 43

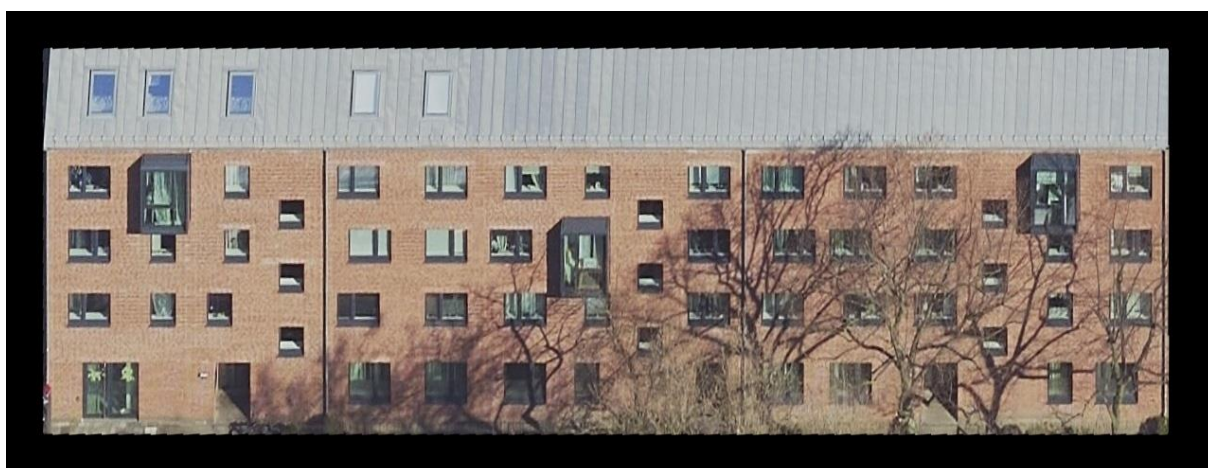


Abbildung 52: Fassade der Weimarer Straße 49



Abbildung 53: Fassade der Fährstraße 73



Abbildung 54: Fassade der Veringstraße 149



Abbildung 55: Fassade der Georg-Wilhelm-Straße 141

Plakette zu Zerstörung und Wiederaufbau



Abbildung 56: Plakette mit Daten zur Zerstörung und dem Wiederaufbau des Gebäudes Weimarer Straße 43

Sturmflutgebiet Hamburg 1962

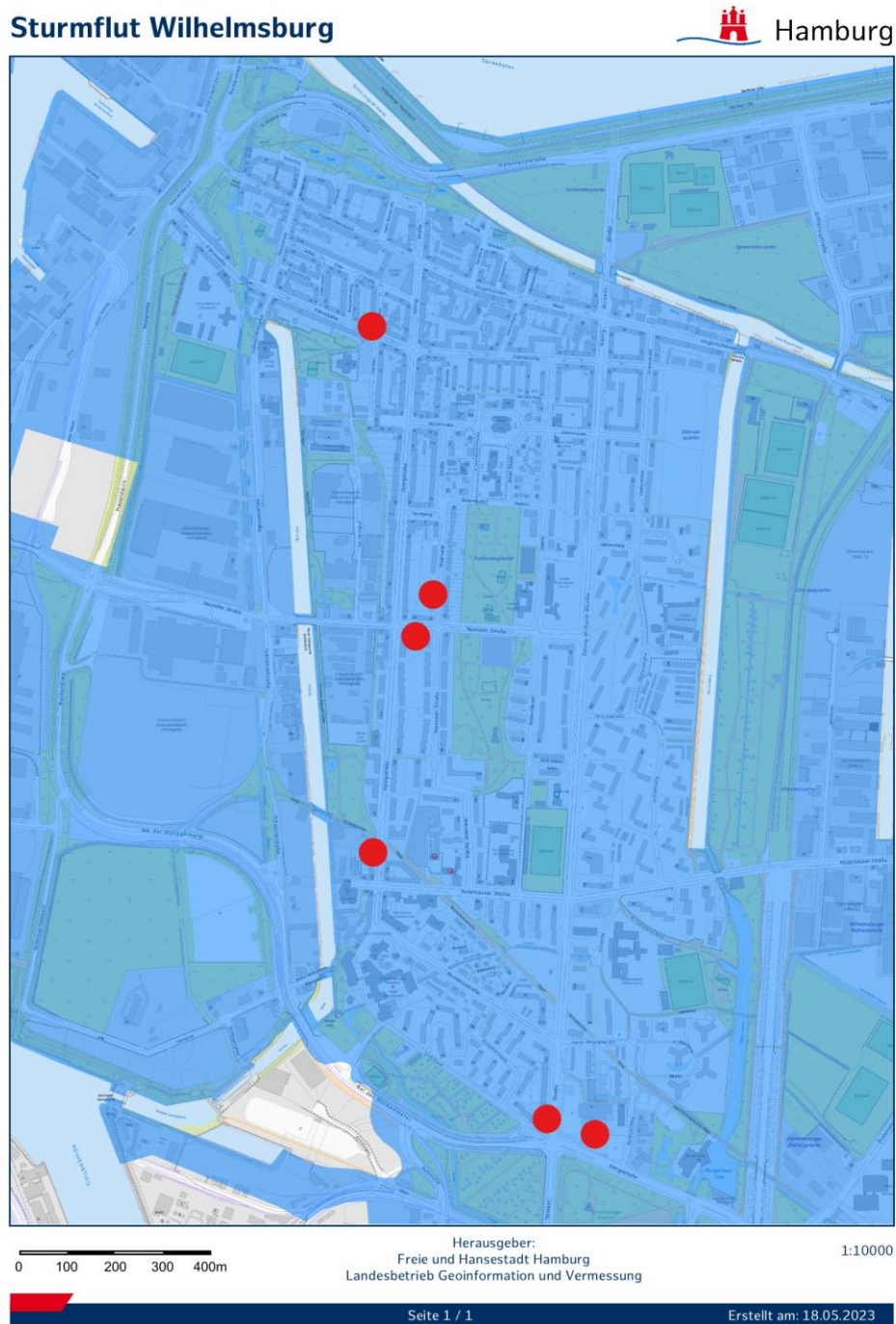


Abbildung 57: Von der Sturmflut 1962 betroffenes Gebiet im Hamburger Stadtteil Wilhelmsburg. Das überflutete Gebiet ist blau hinterlegt. Die Positionen der 6 Gebäude des Hamburg Image Datasets sind durch rote Punkte markiert. Erstellt auf der Internetseite der Stadt Hamburg [32].

Gebäude Mengestraße 6 und Hermann-Westphal-Straße 2a-c



Abbildung 58: Luftbild der Gebäude Mengestraße 6 (links) und Hermann-Westphal-Straße 2a-c (rechts) [25].

Anhang F

Einfluss des Score-Grenzwertes auf mAP

In Tabelle 8 wird deutlich, dass das mAP mit zunehmendem Score-Grenzwert abnimmt. Dieses Verhalten scheint sonderbar, da erwartet werden könnte, dass das maximale mAP bis zu einem Score-Grenzwert zu- und danach abnimmt. Bei genauerer Betrachtung der in Kapitel 2.4.3 beschriebenen Berechnung von mAP wird jedoch deutlich, dass der Score-Grenzwert nur die Anzahl an Vorhersagen, die in die Berechnung mit einbezogen werden, variiert. Dies wird in Abbildung 59 deutlich, in der die PR-Kurve für unterschiedliche Score-Grenzwerte für das beste Modell aus Tabelle 6 dargestellt ist.

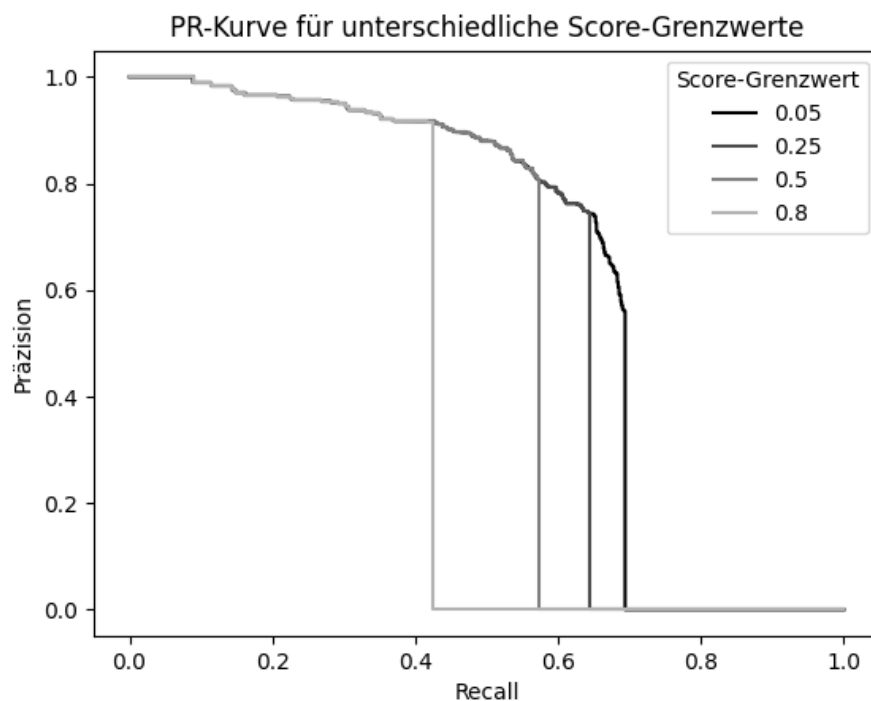


Abbildung 59: PR-Kurve für unterschiedliche Score-Grenzwerte für das beste Modell aus Tabelle 6.

Je höher der Score-Grenzwert ist, desto weniger vorhergesagte Fenster werden berücksichtigt. Da die Vorhersagen des ML-Modells anhand ihres Scores absteigend sortiert sind, sind dementsprechend fast alle als Fenster erkannten Objekte tatsächlich Fenster und die Präzision ist sehr hoch. Viele der tatsächlichen Fenster werden jedoch auch übersehen, was zu einem geringen Recall führt. In Abbildung 59 bricht die PR-Kurve daher für hohe Score-Grenzwerte schon bei niedrigen Recall-Werten auf 0 ab. Die Fläche unter der PR-Kurve, die mAP repräsentiert, ist dementsprechend niedrig. Je kleiner der Score-Grenzwert wird, desto mehr vorhergesagte Fenster werden mit in die PR-Kurve einbezogen. Die Präzision sinkt, da bei geringerem Score-Grenzwert mehr Fenster, die als solche erkannt wurden, in Wahrheit keine Fenster sind.

Gleichzeitig steigt der Recall, weil immer mehr der tatsächlich vorhandenen Fenster erkannt werden. Insgesamt wird die Fläche unter der PR-Kurve größer, was ein steigendes mAP zur Folge hat.

Ergebnisse für $\Delta R_{\text{Gebäude, mean}}$

Tabelle 14: Ergebnisse für $\Delta R_{\text{Gebäude, mean}}$ und σ auf Basis der vom ML-Modell erkannten Fenster mit und ohne Nachbearbeitung für unterschiedliche Score-Grenzwerte sowie auf Basis der IWU-Gebäudetypologie.

Ergebnisse auf Basis der Vorhersagen des ML-Modells

	Mit Nachbearbeitung		Ohne Nachbearbeitung	
Score-Grenzwert	$\Delta R_{\text{Gebäude, mean}}$ [%]	σ [%]	$\Delta R_{\text{Gebäude, mean}}$ [%]	σ [%]
0,05	-137,80	225,75	-341,24	384,84
0,25	1,21	20,24	-8,83	21,04
0,5	16,89	21,27	14,64	21,08
0,8	43,18	16,07	43,13	15,73

Ergebnisse auf Basis der IWU-Gebäudetypologie

	$\Delta R_{\text{Gebäude, mean}}$ [%]	σ [%]
IWU	-84,04	71,07