Klassifikation von Fahrzeugen aus RGB- und LiDAR-Daten mit Convolutional Neural Networks

ROBIN NIESSNER[1], HENDRIK SCHILLING[2], STEFAN HINZ[1] & BORIS JUTZI[1]

Zusammenfassung: In den letzten Jahren wurde ein signifikanter Fortschritt in der Detektion, Identifikation und Klassifikation von Objekten und Bildern durch Convolutional Neural Networks (CNNs) gemacht. In dem vorliegenden Beitrag wird die Eignung von CNNs zur Analyse von RGB und LiDAR Fernerkundungsdaten anhand des Beispiels der Fahrzeugklassifikation untersucht. Hierfür werden drei unterschiedliche Methoden zum Trainieren von CNNs analysiert, die es ermöglichen, CNNs auf Datensätzen mit wenigen hunderten Trainingsbeispielen zu trainieren. Zwei der vorgestellten Methoden basieren auf dem Konzept des Transferlernens. Merkmale, die mit Hilfe eines vortrainierten CNNs erstellt werden, werden bei der ersten Methode mit einer Support Vector Machine (SVM) klassifiziert. Bei der zweiten Methode wird ein CNN' auf einen anderen Datensatz angepasst und ein Teil des CNN wird neu trainiert. Bei der dritten Methode werden eigene flache neuronale Netze designt und trainiert. Die Evaluation der Methoden erfolgt auf einem von der IEEE Geoscience and Remote Sensing Society (GRSS) bereitgestellt Datensatz aus kombinierten RGB und LiDAR Aufnahmen eines urbanen Bereiches. Es wird gezeigt, dass mit diesen Methoden sowohl auf RGB als auch auf LiDAR Daten Klassifikationsergebnisse von hoher Genauigkeit erzielt warden können. Anhand der Ergebnisse wird deutlich, dass durch Methoden des Transferlernens auf RGB Daten trainierte Merkmale auf LiDAR Daten übertragbar sind. Mit Hilfe dieser Merkmale werden auf LiDAR Daten genauere Klassifikationsergebnisse erzielt als auf RGB Daten. Die besten Klassifikationsergebnisse werden mit den selbst designten und trainierten neuronalen Netzen erreicht, welche aus deutlich weniger Ebenen aufgebaut sind als gängige tiefe neuronale Netze. Auch hier werden mit LiDAR Daten höhere Klassifikationsgenauigkeiten erreicht als mit RGB Daten.

1 Einleitung

Verkehrsbezogene Daten sind das zentrale Thema für die Überwachung und das Monitoring urbaner Regionen. Daher ist die automatisierte Analyse dieser Daten, um eine eine parametrisierte Charakterisierung abzuleiten, unerlässlich. Typische Parameter sind beispielsweise die Fahrzeugpositionen, die Anzahl der Fahrzeuge, die Verkehrsdichte und der Verkehrsfluss. Zur Erkennung und Klassifizierung von Fahrzeugen kann der Gradient zwischen Fahrzeug und Hintergrund hilfreich sein, da er häufig eine starke Kennlinie aufweist. Daher werden auf Gradienten basierende Algorithmen verwendet, um Fahrzeuge in Bildern zu bestimmen, z.B. durch Histogram of Oriented Gradients (DALAL & TRIGGS 2005), Local Binary (OJALA et al., 1994) oder SIFT Descriptors (LOWE 1999). Um die Klassifikationsleistung zu erhöhen und die Falschalarmrate zu reduzieren, können zusätzliche Daten betrachtet werden (HINZ & STILLA 2006; TÜRMER et al., 2013).

[1] Karlsruher Institut für Technologie, Institut für Photogrammetrie und Fernerkundung, Englerstr. 6, D-76131 Karlsruhe, E-Mail: [robin.niessner, stefan.hinz, boris.jutzi]@kit.edu
[2] Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung, Gutleuthausstr. 1, D-76275 Ettlingen, E-Mail: hendrik.schilling@iosb.fraunhofer.de

Das Kombinieren von Daten gewonnen aus verschiedenen Arten von Sensoren für die Analyse ist eine Strategie zur Steigerung der Klassifikationsleistung, insbesondere wenn die verwendeten Daten komplementär sind. Daher können radiometrische Daten in Form von RGB-Bildern und die mit einem LiDAR-Sensor gemessenen geometrischen Daten für die Fahrzeugklassifizierung vorteilhaft sein. Zur Einschränkung des Suchraums für die Fahrzeugerkennung sind geometrische Daten prinzipiell von Vorteil. Die LiDAR-Daten können zum Extrahieren von Fahrzeugen genutzt werden. Weiterhin können Merkmale für die Objektklassifizierung anhand der LiDAR-Daten gewonnen werden (JUTZI & GROSS 2009; WEINMANN et al., 2015). Um Fahrzeughypothesen abzuleiten, können in diesem Zusammenhang verschiedene Merkmale berechnet werden, indem radiometrische (optische) und geometrische (Elevations-) Daten zur Klassifizierung verwendet werden. Die Fusion der Daten erfolgt meist durch einen (adaptierten) State-of-the-Art-Klassifikator (SCHILLING et al., 2018).
Aktuelle Arbeiten verwenden Convolutional Neural Networks (CNNs). Diese mehrschichtigen neuronalen Netze sind so ausgelegt, dass sie optimale Merkmale aus Trainingsdaten für ein gegebenes Klassifikationsproblem lernen und vielversprechende Ergebnisse für Erkennungs- und Klassifikationsaufgaben zeigen.
In diesem Beitrag wird das Potential von Convolutional Neural Networks speziell für die Fahrzeugklassifikation untersucht. Daher werden drei Ansätze angegangen, um Klassifikatoren basierend auf Convolutional Neural Networks zu trainieren. Der Hauptbeitrag ist:

- CNNs basierend auf RGB- und LiDAR-Daten führen zu Klassifikationsergebnissen mit hoher Genauigkeit
- Merkmale, die von RGB-Daten abgeleitet werden und durch Transferlernen in LiDAR-Daten übertragen werden, führen zu besseren Klassifikationsergebnissen als die alleinige Verwendung von RGB-Daten
- Neuronale Netzwerke mit weniger Ebenen als weit verbreitete neuronale Netzwerke mit vielen Ebenen können bei einfachen Klassifikationsaufgaben mit wenigen Trainingsdaten zu besseren Klassifikationsergebnissen führen

2 Problemstellung - Klassifikation von Fahrzeugen

In diesem Beitrag werden drei auf CNNs basierende Ansätze vorgestellt, um Fahrzeuge in RGB und LiDAR zu klassifizieren. In Abschnitt 2.2.1 und Abschnitt 2.2.2 wird ein vortrainiertes CNN verwendet. Diese Ideen basieren auf dem Konzept des Transferlernens. In Abschnitt 2.3 wird ein CNN von Grund auf neu entworfen und trainiert. Durch diese Ansätze können die normalerweise benötigten enormen Mengen an Trainingsdaten reduziert werden, um CNNs zu trainieren.

2.1 RGB und LiDAR Datensatz

Zum Training und Evaluation der Experimente wird ein Datensatz verwendet, welcher vom Image Analysis and Data Fusion Technical Committee (IADF TC) of the IEEE Geoscience and Remote Sensing Society (GRSS) bereitgestellt wird. Die Daten bestehen aus einem RGB- und LiDAR-Datensatz. Der Datensatz wurde mit einem flugzeuggetragenem System über dem belgischen Hafen Zeebrügge aufgenommen. Die RGB-Daten sind Orthophotos mit einer

Auflösung von 5 cm. Die LiDAR-Daten werden als digitales Oberflächenmodell (DSM) mit einem Punktabstand von 10 cm bereitgestellt. Die Auflösung der RGB-Daten wird auf 10 cm reduziert, um mit der GSD der LiDAR-Daten zu übereinzustimmen. Da nur ein Datensatz verfügbar ist, werden die Trainings- und Validierungsdaten lokal getrennt, um die Korrelation zwischen den beiden Sätzen zu minimieren. Zu Trainingszwecken werden die Daten erweitert, indem sie um 90°, 180° und 270° gedreht werden, um mehr Trainingdaten zu erstellen und den Klassifikator robuster gegenüber Rotationsvarianzen zu machen. Weiterhin wird, um große Bias in den Aktivierungsfunktionen der CNNs zu vermeiden, von allen Kanälen der Bilder der Mittelwert subtrahiert.

2.2 Transferlernen

Die folgenden zwei Ansätze basieren auf dem Konzept des Transferlernens. Die Idee hinter diesen Ansätzen ist, dass CNNs, die auf einen Datensatz trainiert sind, übertragen werden können, um einen anderen Datensatz zu klassifizieren.
Verwendet wird das VGGNet (SIMONYAN & ZISSERMAN, 2014), ein CNN basierend auf der AlexNet-Architektur von (KRIZHEVSKY et al., 2012) welches auf dem ImageNet-Datensatz vortrainiert wurde (DENG et al., 2009). Die Merkmale, die das CNN in den ersten Ebenen erlernt, werden als allgemeine Merkmale betrachtet. Die folgenden Ebenen in der Architektur des CNN enthalten abstraktere Merkmale. Abbildung 1 zeigt die vortrainierten Merkmale der ersten Ebene des VGGNet.
Die Merkmale erscheinen wie eine Art von Gabor- und Blob-Merkmalen. Diese allgemeinen Merkmale sind für viele 2D-Bilderkennungsaufgaben geeignet.

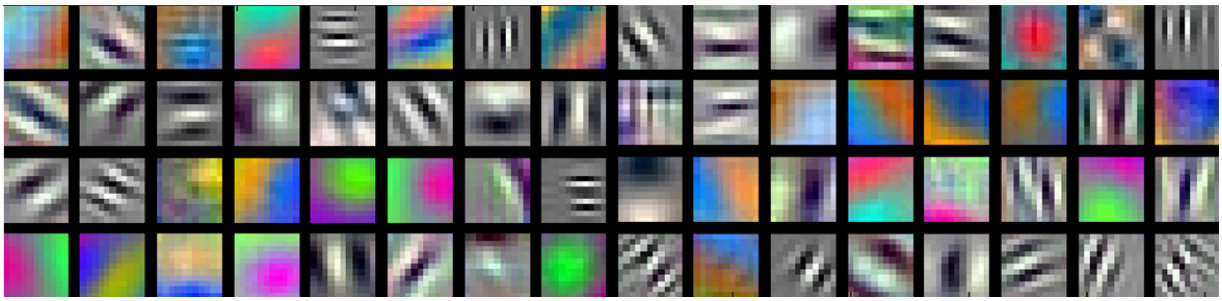


Abb. 1: Merkmale des vortrainierten VGGNet der ersten Ebene berechnet in *MATLAB*. Das Aussehen der Filtermasken ist ähnlich wie bei Gabor- und Blob-Filtern

2.2.1 CNN Merkmalsvektor

In diesem Ansatz wird das CNN lediglich als Merkmalsextraktor verwendet. Um dies zu erreichen, wird die letzte vollständig verbundene (fully connected) Ebene des CNNs entfernt, welche normalerweise die Klassifikationsebene ist. Wie in Abbildung 2a dargestellt, entspricht der resultierende Merkmalsvektor der Ausgabe der Faltung der vorletzten Ebene. Die Größe des Merkmalsvektors hängt von der CNN-Architektur ab, in diesem Falle ist die Größe *4096 x 1*.
Die extrahierten Features werden verwendet, um einen Standardklassifikator, eine lineare Support Vector Machine (SVM), zu trainieren. Um die RGB-, LiDAR-Höhen- und Intensitätsdaten zu fusionieren, werden die Merkmalsvektoren für das SVM-Training verkettet. Die Dimensionen der resultierenden Vektoren sind *4096 x 1, 8192 x 1* und *12288 x 1*.

2.2.2 Fine-Tuning

In diesem Ansatz wird das vortrainierte CNN als Merkmalsextraktor und Klassifikator verwendet. Das Trainieren eines vortrainierten CNN auf einem anderen Datensatz wird Fine-Tinung genannt. Auch hier müssen kleine Anpassungen am CNN vorgenommen werden. Theoretisch gibt es keine Einschränkungen, wie viele Änderungen der Architektur des CNN durchgeführt werden. Da nur eine kleine Menge von Trainingsdaten zur Verfügung steht, soll die Anzahl der Parameter, die neu trainiert werden müssen, so klein wie möglich gehalten werden. Das VGGNet wurde auf dem ImageNet-Datensatz trainiert. Somit besteht die Klassifikations-Ausgabeebene aus 1000 Elementen für die 1000 Klassen des ImageNet. Da nur die Hintergrundklasse und die Fahrzeugklasse getrennt werden sollen, muss die Klassifizierungsebene in eine Zwei-Elemente-Ebene geändert werden. Die grün hervorgehobenen Elemente in Abbildung 2b ersetzen die vorherige Klassifizierungsebene. Der Rest des CNN bleibt gleich. Die Gewichte zwischen der vorletzten *fully-connected* Ebene und der Klassifizierungsebene müssen bei diesem Ansatz neu trainiert werden. Hierfür werden sie zufällig initialisiert und mit Hilfe des stochastischen Gradientenverfahrens optimiert.

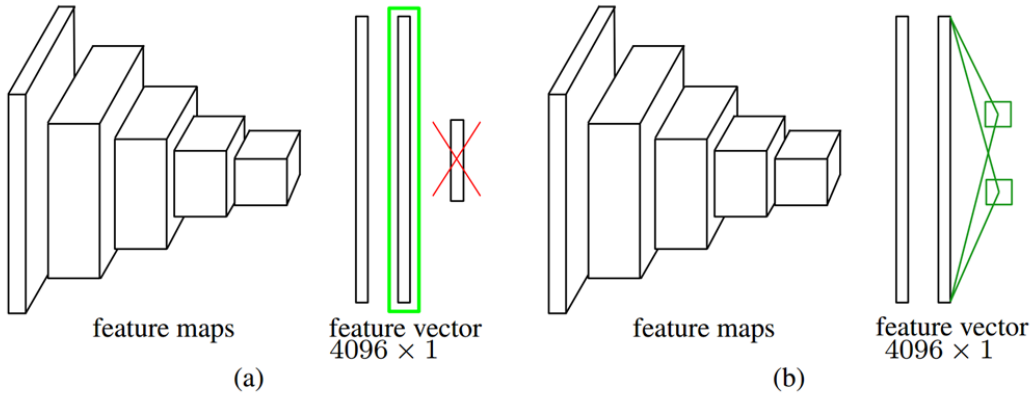


Abb. 2: (a) Änderungen im VGGNet, um die Ausgabe der letzten Ebene als Merkmalsvektor zu verwenden. Die Klassifikationsebene wird entfernt und die vorletzte Ebene mit der Größe *4096 x 1* wird als Merkmalsvektor verwendet. (b) Änderungen zur Feinabstimmung des VGGNet auf die Klassifizierungsaufgabe. Die Klassifizierungsebene und die entsprechenden Gewichte werden durch eine Zwei-Elemente-Klassifizierungsebene mit Gewichtsmatrizen der entsprechenden Größe ersetzt.

Ein Nachteil dieses Ansatzes ist, dass die Dimensionalität der Eingabedaten nicht geändert werden kann, da die vortrainierten Gewichtungen in der ersten Ebene des CNN nicht geändert werden sollen. Die Anzahl der Kanäle des Eingabebildes ist auf die Größe der Daten beschränkt, mit denen das CNN ursprünglich trainiert wurde. Dies impliziert, dass die RGB- und LiDAR-Daten nicht auf der Eingangsebene des CNN fusioniert werden können.

2.3 Selbsterstellte CNNs

Dieser Ansatz basiert auf dem Entwerfen und vollständigen Trainieren adaptierter CNN Architekturen. Aktuelle CNNs bestehen aus Millionen von Parametern und erfordern eine große Menge an Trainingsdaten. Da ein CNN für ein binäres Klassifizierungsproblem trainiert wird, erscheint ein CNN mit wenigen Parametern für diese Problemstellung ausreichend.

In Tabelle 1 sind die Architekturen der zwei leistungsstärksten CNNs dieser Studie aufgeführt. *S* ist die Schrittweite *stride* und *P* das *padding* der Filter der jeweiligen Faltungsebenen. Als Pooling-Methode wird das Max-Pooling verwendet und während des Trainings wird ein Dropout auf der ersten vollständig verbundenen (fully connected) Ebene angewandt, um ein Überanpassung (Overfitting) zu verhindern. Die entworfenen flachen CNNs bestehen nur aus einem Bruchteil von Parametern im Vergleich zu tiefen CNNs wie AlexNet. Die Größe des Filters in der ersten Schicht des Medium-CNN wurde mit der gleichen Größe wie die des Filters aus der AlexNet-Architektur gewählt. Die Größe der Filter des Large-CNN wurde so gewählt, dass sie auf der Skala der Objekte in dem Bild liegt, das klassifiziert werden soll.

Tab. 1: Die Architektur des von Grund auf neu konzipiert und trainierten CNNs.

	Medium-CNN
Conv1	*16x8x8, S=3,P=0 x3 pool*
Conv2	*32x5x5, S=1, P=0, x2 pool*
Conv3	*64x4x4, S=1, P=0*
Full4	*128, dropout*
Full5	2, softmax
Parameter	180.000

Im Gegensatz zum Ansatz von Abschnitt 2.2.2 können die RGB-, LiDAR-Höhen- und Intensitätsdaten in der Eingabeebene fusioniert werden. Hierfür werden die Kanäle der Bilder verkettet.

3 Ergebnisse und Evaluierung

Das Training der CNNs wurde an einem ausgeglichenen Datensatz mit je 399 Trainingsbeispielen für die Hintergrund- und Fahrzeugklasse vor der Datenerweiterung durchgeführt. Da es in einem realistischen Szenario mehr Hintergrund als Fahrzeuge gibt, werden 426 Fahrzeug- und 6017 Hintergrundmuster für die Validierung des Trainings verwendet, die mit einer Watershed-basierten Segmentierungsmethode erzeugt wurden (SCHILLING & BULATOV, 2016). Da die Datenverteilung für die Evaluation sehr unausgeglichen ist, ist die Gesamtgenauigkeit nicht geeignet, die Qualität der Klassifikation darzustellen. Der F-Score, das harmonische Mittel der Präzision und des Recalls, eignet sich besser für diese Aufgabe. Die Trainingsergebnisse werden mit einem Klassifizierungsansatz basierend auf den Histogramm von orientierten Gradienten (HOG)-Merkmalen mit einer *Random Forest* Klassifikation verglichen.

3.1 Klassifikationsergebnis

Das Training wird unter Verwendung des *MatConvNet* Frameworks in *MATLAB* (VEDALDI & LENC, 2015) durchgeführt. Für die Klassifizierungsebene wird ein Softmax-Klassifikator mit einer Cross-Entropy-Loss-Funktion verwendet, um die Gewichte zu aktualisieren. Als Optimierungsfunktion wird das Stochastische Gradientenverfahren (SGD) gewählt. Das CNN-Training wird mit Lernraten (LR) von *0,01*, *0,001* und *0,0001* durchgeführt. Die folgenden Beispiele des CNN-Trainings zeigen nur die besten Kombinationen von LR und Datentypen.

3.1.1 CNN Merkmalsvektor

Das Training der SVM mit den CNN-Merkmalen erfolgt durch eine 10-fache Kreuzvalidierung. Die SVM mit dem besten Ergebnis wird verwendet, um die Validierungsdaten zu klassifizieren. Wie in Abbildung 3 gezeigt, erreicht die Klassifikation mit den RGB-Daten den niedrigsten F-Score (*0,794*), obwohl die CNN-Merkmale ursprünglich in einem RGB-Datensatz trainiert wurden. Das beste Klassifikationsergebnis wird durch die Fusion des Elevations- und Intensitätsdatensatzes erreicht (F-Score = *0,929*). Das Hinzufügen der RGB-Daten zur Fusion des LiDAR-Datensatzes übertrifft noch immer die Ergebnisse mit RGB-Daten und liefert gute Ergebnisse, verbessert jedoch nicht die Ergebnisse der Fusion der LiDAR-Daten.

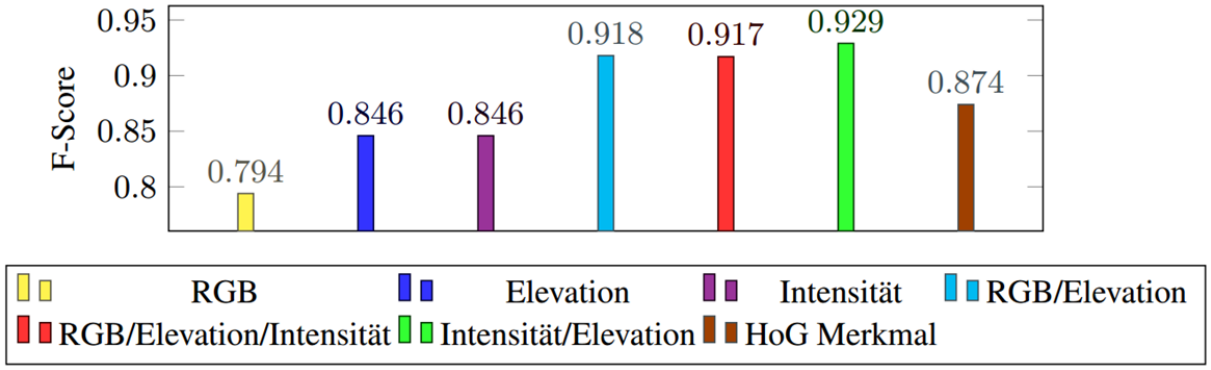


Abb. 3: *F-Scores* für das CNN Merkmalstraining mit einer SVM.

3.1.2 Fine-Tuning

Für das Fine-Tuning wird das gleiche vortrainierte CNN (VGGNet) verwendet, wie es bereits beim CNN-Merkmalsvektor Ansatz verwendet wurde. Für diesen Ansatz stehen nur Klassifikationsergebnisse für jeden Datentyp, nicht die fusionierten Daten zur Verfügung. Wie in Abbildung 4 gezeigt, übertrifft die Klassifikation auf der LiDAR-Elevation (F-Score: *0,903*) und Intensität (F-Score: *0,902*) wieder die Ergebnisse der RGB-Daten (F-Score: *0,867*). Im Vergleich zu den Ergebnissen aus Abbildung 3 zeigt die Klassifikation mit einzelnen Sensordaten verbesserte Ergebnisse für alle Datensätze.

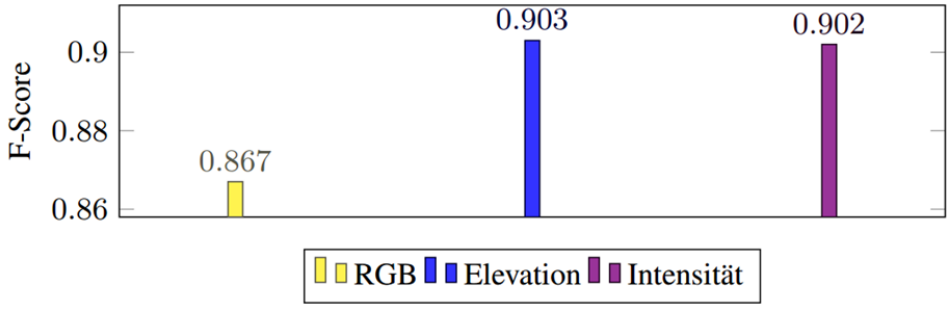


Abb. 4: *F-Scores* für das Fine-Tuning des VGGNet.

3.1.3 Selbsterstellte CNNs

Für die von Grund auf neu entwickelt und trainierten CNNs wird das Medium-CNN betrachtet. Beim Testen mehrerer Architekturen von CNNs zeigte dieses die beste Leistung sowohl bei der

Dauer des Trainings als auch bei den Klassifikationsergebnissen. Wie in Abbildung 5 und Abbildung 6 gezeigt, kann das Training mit der Medium-CNN Architektur im RGB-Datensatz (F-Score: *0,803*, gelbe Linie mit Dreiecken) die Ergebnisse, die durch das Training mit den LiDAR-Daten erreicht werden, nicht erreichen. Das beste Klassifikationsergebnis wird durch die CNN-Datenfusion von Elevation und Intensität erreicht (F-Score: *0,955*, grüne Linie mit Punkten).
Verglichen wird auch das Ergebnis des Medium-CNN-Trainings mit einer HoG-Feature-Klassifikation (F-Score: *0,874*, braune Linie mit Quadraten), die auf fusionierten RGB- und LiDAR-Datensätzen trainiert wurde.
Während dieser Klassifizierungsansatz bessere Ergebnisse liefert als die Klassifikation von RGB-Daten mit dem Medium-CNN, liegt er innerhalb des gleichen Bereichs wie das auf dem Intensitätsdatensatz trainierte Medium-CNN und die Fusion von RGB- und Elevationsdaten. Für die Elevationsdaten, die fusionierten Daten des LiDAR und die fusionierten RGB- und LiDAR-Datensätze übertrifft das Medium-CNN die HoG-Feature-Klassifikation.

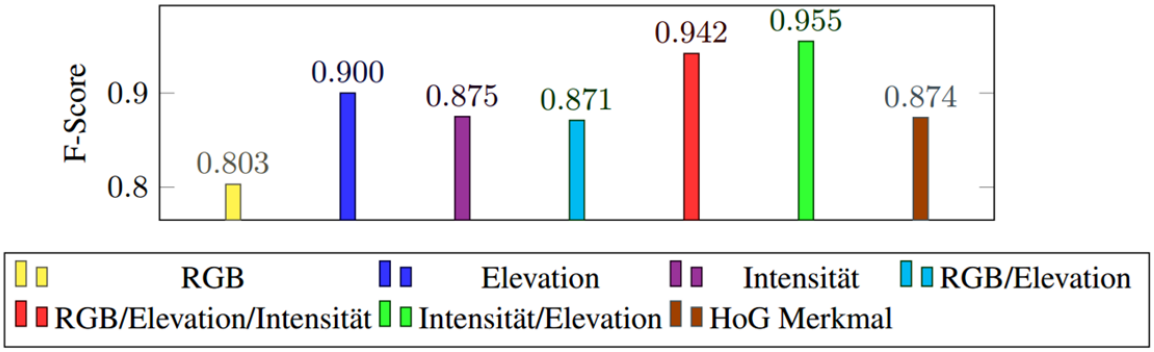


Abb. 5: *F-Scores* für die Trainingsergebnisse, die von verschiedenen Daten des Medium-CNN abgeleitet wurden, mit dem HoG-Merkmal-Klassifizierungsergebnis zum Vergleich.

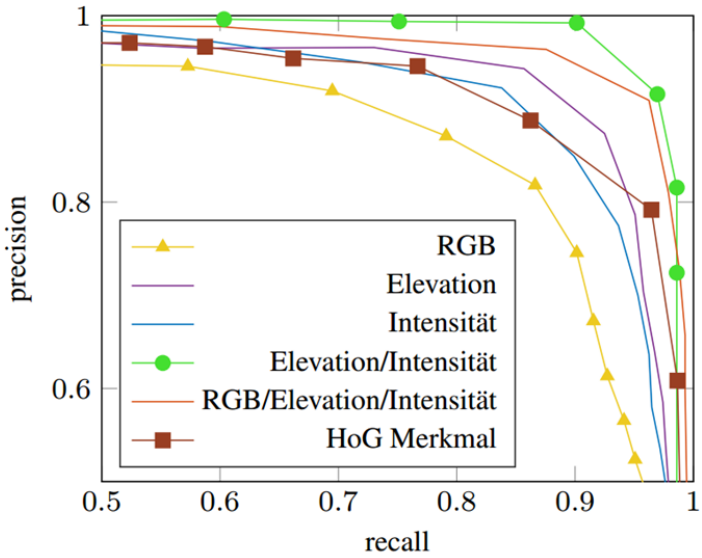


Abb. 6: PR-Kurven der Klassifikationsergebnisse, die aus verschiedenen Daten für das Medium-CNN abgeleitet wurden, mit dem HoG-Feature-Klassifizierungsergebnis als Vergleich.

3.2 Sliding-Window Klassifikation

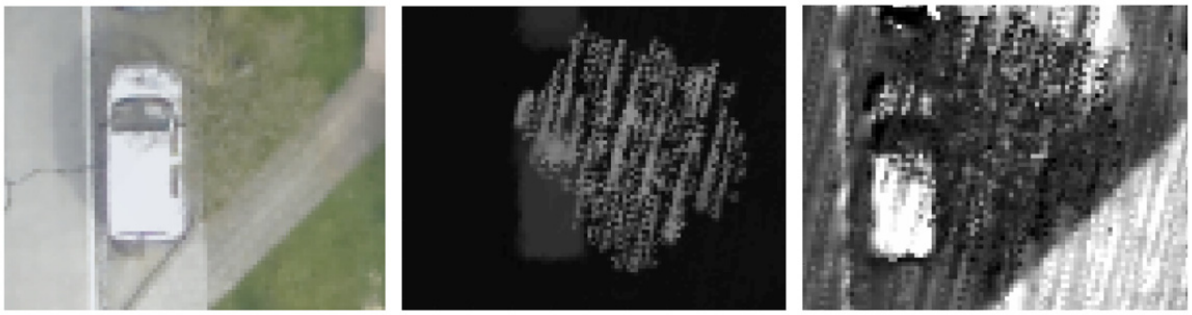
Schließlich soll ein Detektionsergebnis des CNN-basierten Klassifikationsansatzes auf einem gegebenen Datensatz gezeigt werden, um die Leistungsfähigkeit des Klassifikators zu untersuchen. Das Medium-CNN wird für eine Sliding-Window Klassifikation auf der Fusion von LiDAR-Daten gewählt, da es die beste Leistung in Abschnitt 3.1 zeigt. Die Detektion erfolgt über ein gleitendes Fenster mit einer Größe von *80 x 80* Pixeln. Die Abtastrate ist mit 10 Pixel vorgegeben, so dass sichergestellt ist, dass jedes Fahrzeug enthalten ist und die Anzahl der zu berechnenden Bilder minimiert wird. Für einen Testbereich der Größe *5000 x 5000* Pixel werden *243.049* Bilder berechnet. Abbildung 7 zeigt Beispiele aus dem klassifizierten Bereich. Zur Visualisierung wird das RGB-Bild als Hintergrund verwendet, obwohl die Daten auf der Fusion von LiDAR-Daten klassifiziert wurden. Die Klassifikationsergebnisse werden unter Verwendung einer Heatmap dargestellt, bei der die Klassifizierungswerte der klassifizierten Pixel dargestellt werden. Auf einem großen offenen Feld, bei denen die Fahrzeuge verteilt sind, hat der Klassifikator keine Probleme, alle Fahrzeuge mit geringer Fehlerrate (falsch positiv) zu erkennen.

Eine anspruchsvollere Aufgabe ist die Detektion von Fahrzeugen, die nahe beieinander stehen oder sich im Hinterhof befinden. In Abbildung 7c wird ein Beispiel gezeigt, bei dem Fahrzeuge nebeneinander geparkt sind, mit einigen Ausnahmen kann der Klassifikator die Fahrzeuge noch trennen. Abbildung 7d zeigt einen interessanten Fall, bei dem die Fahrzeuge völlig unterschiedliche Größen haben. Der Klassifikator kann kleine Fahrzeuge bis hin zu großen Wohnwagen erkennen, obwohl die Anzahl der Trainingsbeispiele für Wohnwagen sehr gering ist.

Abb. 7: Abbildungen (a) und (b) zeigen Beispiele aus dem gesamten klassifizierten Bereich. Abbildung (c) zeigt ausschnittsweise ein Beispiel für die Klassifikation auf Parkplätzen und Abbildung (d) zeigt, dass sich der Klassifikator auch für Hinterhöfe mit Fahr Fahrzeugen unterschiedlicher Größen eignet. Es ist zu beachten, dass die Klassifizierung der Fahrzeuge anhand der Fusion von LiDAR-Daten durchgeführt wird, jedoch werden die Ergebnisse unter Verwendung der RGB-Daten als Hintergrund angezeigt, um die Sichtbarkeit zu erhöhen. Das Heatmap-Overlay verwendet die vom Medium-CNN berechneten Klassifizierungswerte.

Eine herausfordernde Aufgabe ist gegeben, wenn die Fahrzeuge von Vegetation oder ähnlichen Objekten bedeckt sind. Obwohl das Fahrzeug in der RGB-Darstellung in Abbildung 8a vollständig sichtbar ist, hat unser Klassifikator es nicht als Fahrzeug klassifiziert. In den Elevationsdaten in Abbildung 8b ist das Fahrzeug kaum sichtbar, obwohl der letzten Puls des LiDAR-Signals verwendet wird, um die 2D-Bilder darzustellen. Auch in der

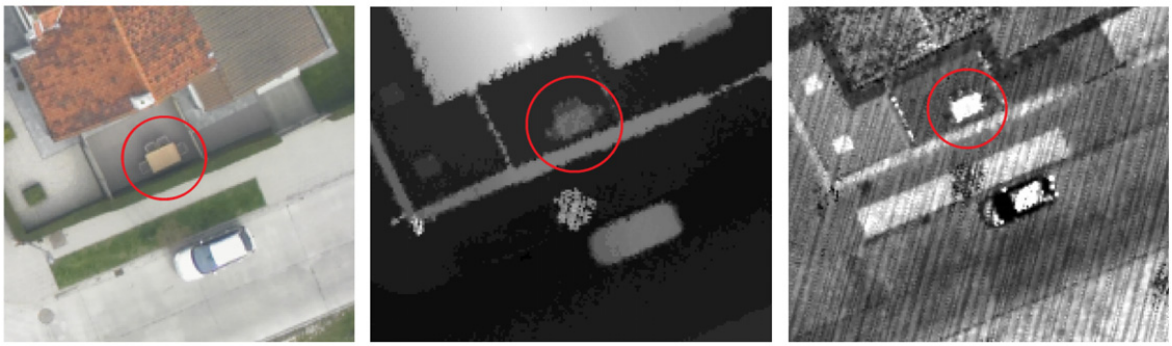
Intensitätsdarstellung in Abbildung 8c sind große Bereiche des Fahrzeugs nicht sichtbar. Da die Merkmale, die das CNN in den ersten Ebenen erlernt, meistens Kantendetektoren sind, kann es das Fahrzeug nicht korrekt klassifizieren.



(a) (b) (c)

Abb. 8: Fehlklassifizierung eines von einem Baum bedeckten Fahrzeugs. (a) zeigt das Fahrzeug in den RGB-Daten. (b) das Fahrzeuge ist kaum sichtbar. (c) Im Intensitätsbild ist die Front des Fahrzeugs größtenteils von einem Baum bedeckt.

Andere Hindernisse sind Objekte, die Fahrzeugen in ihrer Geometrie sehr ähnlich sind. Die Abbildung 9a zeigt Gartenmöbel, die in der RGB-Darstellung bedingt anders aussehen als ein Fahrzeug. Betrachtet man die Darstellung von Elevation (9b) und Intensität (9c), ähneln die Möbel einem kleinen Fahrzeug oder dem Dach eines Fahrzeugs. Der Detektor muss für verschiedene Größen empfindlich sein, da das Ziel darin besteht, alles von einem kleinen Fahrzeug bis zu einem Wohnwagen zu erfassen.



(a) (b) (c)

Abb. 9: Fehlklassifizierung eines Objekts, das bedingt einem Fahrzeug ähnelt. RGB-Bild (a) zeigt Gartenmöbel, die fälschlicherweise als Fahrzeug bestimmt wurden. Im Elevationsbild (b) und im Intensitätsbild (c) haben die Möbel die gleiche Art von Kanten wie ein Fahrzeugdach, was zu einer Fehlklassifizierung führt.

4 Diskussion und Fazit

In Tabelle 2 wird einen Überblick über alle erzielten Klassifizierungsergebnisse gegeben.
Die besten Ergebnisse wurden durch das Medium-CNN erzielt, das auf die Fusion der LiDAR-Daten ausgelegt und trainiert wurde, dicht gefolgt von der Fusion von RGB- und LiDAR-

Datensätzen. Für 5 von 6 Untersuchungen konnte zudem die Klassifikationsleistung durch die Fusion der Sensordaten im Vergleich zur Verwendung der einzelnen Daten erhöht werden. Die einzige Ausnahme ist die Fusion von RGB- und Elevationsdaten für die selbsterstellten CNNs. Der Fine-Tuning-Ansatz konnte die besten Klassifikationsergebnisse für die einzelnen RGB- und LiDAR-Daten erzielen. Betrachtet man die Gesamtleistung, liefert die LiDAR-basierte Klassifizierung bessere Ergebnisse als die RGB-basierte Klassifizierung.

Tab. 2: Übersicht der erreichten F-Scores für alle Trainings für jeden Datentyp und jede Kombination.

Daten	CNN Merkmalstraining	Fine-Tuning	Selbsterst. CNNs
RGB	0,794	0,867	0,803
Elevation	0,846	0,903	0,900
Intensität	0,846	0,902	0,875
RGB/Elevation	0,918	-	0,871
Elevation/Intensität	0,929	-	0,955
RGB/Elevation/Intensität	0,917	-	0,942

Dies ist ziemlich überraschend, da die CNNs, die für die Transferlernansätze (CNN Feature, Fine-Tuning) verwendet wurden, mit einem RGB-Datensatz vortrainiert wurden. Dies führt zu der Schlussfolgerung, dass in RGB-Daten gelernte Merkmale in LiDAR-Daten übertragbar sind. Die LiDAR-Daten scheinen für diese Klassifizierung besser geeignet zu sein. Da nur wenige Hundert Trainingsbeispiele zur Verfügung stehen, kann die Einheitlichkeit der Geometrie der Fahrzeuge zu stabileren geometrischen Merkmalen als die Radiometrie aus den RGB-Daten führen, da Fahrzeuge unterschiedlicher Farben in den LiDAR-Daten meist das gleiche Aussehen haben. Ein Grund für die abnehmende Leistung durch Hinzufügen der RGB-Daten zur Fusion der LiDAR-Daten könnte eine ungenaue Koregistrierung der 3D-LiDAR-Daten auf den 2D-RGB-Daten sein. Dies führt zu unscharfen Kanten. Da die Filter in der ersten Schicht der CNNs hauptsächlich Kantendetektoren sind, kann sich das Ergebnis hierdurch verschlechtern. Weiterhin zeigt die Klassifikationsleistung des Medium-CNN mit nur 180.000 Parametern, dass tiefe CNNs für binäre Klassifizierungsaufgaben in dieser Fernerkundungsanwendung nicht erforderlich sind.
Bei zukünftigen Untersuchungen wollen wir weitere CNN-Architekturen wie siamesische CNNs testen, um das Problem einer unpräzisen Koregistrierung anzugehen und mehr Aufmerksamkeit auf die Eigenschaften jedes Eingangskanals zu richten. Darüber hinaus beabsichtigen wir, CNN-Klassifizierungen für Hyperspektraldaten und Kombinationen von RGB-, Hyperspektral- und LiDAR-Datensätzen zu testen.

5 Literaturverzeichnis

DALAL, N. & TRIGGS, B., 2005: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, **1**, IEEE, 886–893.

DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K & FEI-FEI, L., 2009: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, IEEE, 248–255.

HINZ, S. & STILLA, U., 2006: Car detection in aerial thermal images by local and global evidence accumulation. Pattern Recognition Letters, **27**(4), 308–315.

JUTZI, B. & GROSS, H., 2009: Nearest neighbour classification on laser point clouds to gain object structures from buildings. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, **38**(Part 1), 4–7.

KRIZSHEVSKY, A., SUTSKEVER, I. & HINTEN, G. E., 2012: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, 1097–1105.

LOWE, D. G., 1999: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, **2**, 1150–1157.

OJALA, T., PIETIKAINEN, M. & HARWOOD, D., 1994: Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition, **1**, 582–585.

SCHILLING, H. & BULATOV, D., 2016: Segmentation methods for detection of stationary vehicles in combined elevation and optical data. In: International Conference on Pattern Recognition (ICPR), International Society for Optics and Photonics, 592–597.

SCHILLING, H., BULATOV, D. & MIDDELMANN, W., 2018: Object-based detection of vehicles using combined optical and elevation data. ISPRS Journal of Journal of Photogrammetry and Remote Sensing, **6**(6), 85-105.

SIMONYAN, K. & ZISSERMAN, A. 2014: Very deep convolutional networks for large-scale image recognition. CoRR.

TÜRMER, S., KURZ, F., REINARTZ, P & STILLA, U., 2013: Airborne vehicle detection in dense urban areas using hog features and disparity maps. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, **6**(6), 2327–2337.

VEDALDI, A. & LENC, L., 2015: Matconvnet: Convolutional neural networks for Matlab. In: Proceedings of the 23rd ACM international conference on Multimedia, ACM, 689–692.

WEINMANN, M., SCHMIDT, A., MALLET, C., HINZ, S., ROTTENSTEINER, F. & JUTZI, B., 2015: Contextual classification of point cloud data by exploiting individual 3d neighbourhoods. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences **2**(3), 271.