Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung -Anwendungen 1 WS2011/2012

Vitalij Stepanov

Analyse komplexer Szenen mit Hilfe von Convolutional Neural Networks

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Vitalij Stepanov

Analyse komplexer Szenen mit Hilfe von Convolutional Neural Networks

Ausarbeitung -Anwendungen 1 WS2011/2012 eingereicht im Rahmen der Masterprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Gunter Klemke; Prof. Dr. Kai Von Luck

Eingereicht am: 29.02.2012

Vitalij Stepanov

Thema der Arbeit
Analyse komplexer Szenen mit Hilfe von Convolutional Neural Networks

Stichworte
CNN, Convolutional Network, SIFT, NN, Neuronale Netze, Klassifkation

Kurzzusammenfassung
Die Bildverarbeitung gewinnt immer mehr an Bedeutung nicht nur im industriellen, als auch im Privaten Bereich. Der technologische Vorschritt fordert Maschinen immer mehr mit der Umgebung zu interagieren. Im Bereich der Robotertechnik sind gestellten Aufgaben beispielsweise die Lokalisierung bzw. Bewegung von Objekten, die Lokalisierung und gegebenenfalls die Interaktion mit Personen, aber auch das Erkennen bzw. Ausweichen von Hindernissen. Das Sterben nach mehr Sicherheit bringt Autohersteller immer neuere Lösungen wie Fahrspurassistent, Assistent zur erkennung der Blickrichtung oder auch Erkennung von Fußgänger. Diese Arbeit untersucht die Anwendung für diese Zwecke von Convolutional Neural Networks, analysiert damit verbundene Probleme und Risiken.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einführung

Bereits von der Geburt an lernt der Mensch die Information aus der Umgebung zu extrahieren. Objekt für Objekt trainiert er sein Gehirn die Umgebung wahr zu nehmen. Die Erkennung von Objekten ist eine Notwendigkeit nicht nur für Menschen, sondern auch für die Maschinen, die den Menschen assistieren. Das, was der Mensch langsam mit der Zeit gelernt hat, muss den Maschinen erst beigebracht werden. Daher gewinnt die Bildverarbeitung immer mehr Bedeutung sowohl in der Industrie als auch im privaten Bereich.

1.1 Motivation

Der technologische Vorschritt fordert Maschinen immer mehr mit der Umgebung zu interagieren. Die Problemstellung bei der industriellen Verarbeitung stellt die Automatisierung der optischen Qualitätskontrolle bzw. die Vermessung der Erzeugnisse dar. Auch im privaten Bereich wächst die Anfrage auf autonome Maschinen, wie beispielsweise selbstständige Staubsauger, stätig. Die Pfandautomaten gehören seit langer Zeit zu unserem täglichen Leben. Dabei wird nicht nur die Form der Flasche untersucht, sondern auch festgestellt, ob es sich um einer Pfandflasche handelt. Durch die Verformungen von Plastikflaschen wird diese Aufgabe deutlich erschwert. Hersteller streben nach Maschinen, die sich in der Umgebung zurecht finden und bestimmte Aufgaben selbständig erledigen.

Im Bereich der Robotertechnik umfasst die Intelligenz der Maschinen in Bezug auf die Interaktion mit der Umgebung bzw. autonome Navigation ein sehr umfangreiches Gebiet. Die gestellten Aufgaben sing beispielsweise die Lokalisierung bzw. Bewegung von Objekten, die Lokalisierung und gegebenenfalls die Interaktion mit Personen, aber auch das Erkennen bzw. Ausweichen von Hindernissen.

Der Wunsch nach mehr Sicherheit bring die Autohersteller zu den neuen Ideen, die den Komfort bzw. die Sicherheit von den Insassen und auch anderen Verkehrsteilnehmer erhöhen. Ein deutlicher Beispiel dafür ist in Sermanet und LeCun (2011) und Peemen u. a. (2011) beschriebene Assistenten zur Erkennung von Verkehrszeichen. Die Systeme zum Erkennen von Fußgänger können den Fahrer bei unerwarteten Gefahren warnen bzw. eingreifen und damit die Verletzung von Fußgänger vermeiden. Eine Neuentwicklung im Bereich Fahrerkomfort

stellt der Fahrerassistent zur Analyse von Wahrnehmfähigkeiten des Fahrers dar Fraunhofer Institut für Digitale Medientechnologie (2010). Dabei wird das Eye Tracking Verfahren verwendet, der die Augen des Fahrers jederzeit beobachtet und bei sehr langsam bewegten bzw. geschlossenen Augenlider davon ausgeht, dass der Fahrer übermüdet ist. Eine akustische Warnung weckt ihn wieder aus dem Sekundenschlaf.

1.2 Problemstellung und Zielsetzung

Die Detektion von Objekten ist mit unterschiedlichen Mittel wie beispielsweise Laserscanner oder Radar möglich. Dabei kann man eine auf der Fahrbahn legende Plastiktüte bzw. die Zeitung von dem Stein oder einem liegenden Mensch schlecht unterscheiden. Ein wegen der vom Wind wehender Zeitung eingeleiteter Ausweichmanöver könnte zu dem unerwarteten Ausgang führen. Daher ist der Einsatz einer Kamera meistens unverzichtbar. Die Extraktion der Information aus der Umgebung setzt eine sehr große Generalisierungsfähigkeiten voraus. In der reellen Umgebung werden Objekte meistens von den anderen Objekten verdeckt. Die Invariation der Mustern, unterschiedliche Lichtverhältnisse oder perspektivische Veränderungen machen diese Aufgaben besonders anspruchsvoll. Auch bei verrauschten, verzerrten (affine Transformation) Aufnahmen muss die Klassifikation einwandfrei funktionieren. Die Ressourcen der Geräte sind meistens begrenzt, daher stellt die Verarbeitung der hochauflösender Bilder in Echtzeit eine besondere Herausforderung dar. Wenn die Verarbeitung der Daten zu lange dauert, kann es Konsequenzen auf die Reaktionszeit haben. Betrachtet man beispielsweise eine Applikation zur Navigation von Roboter, so kann nicht abgewartet werden, bis die Verarbeitung der Daten abgeschlossen ist. Während dessen kann der Roboter ein Hindernis erreicht haben und hat keine Möglichkeit mehr auszuweichen. Daher werden immer effizienteren Methoden entwickelt. Motiviert von der Aufbau eines menschlichen Gehirns, wurde Von Yan LeCun ein Verfahren mit dem Namen Convolutional Neural Netzwerk entwickelt. Dieser Verfahren kombiniert die Erfahrungen aus den Neuronalen Netzen und Faltungsmasken zu einem sehr effizienten Werkzeug Huang und LeCun (2006). Diese Arbeit soll untersuchen wie gut ist dieses Verfahren für diese Problematik einsetzbar ist. Weiterhin soll untersucht werden wie stark die Anzahl der Objekte bzw die Varianz auf die Ergebnisse der Erkennung einwirkt.

2 Lösungsansätze

In diesem Kapitel werden die gängigen Verfahren zur Bildverarbeitung vorgestellt, deren Vorteile bzw. Nachteile beschrieben und miteinander vergleichen. Der Abschnitt 2.1 beschreibt SIFT Verfahren. In dem Abschnitt 2.2 wird ein häufig gebräuchliches Verfahren zur Klassifikation mittels neuronalen Netzen vorgestellt. Dabei wird versucht das menschliche Gehirn zu nachzuahmen. Der Abschnitt 2.3 erläutert eine Weiterentwicklung der neuronalen Netzen. Die Erweiterung besteht darin, dass die Bildgröße stufenweise reduziert wird. Mittels der Faltung wird die Information der benachbarten Pixel an die unteren Schichten mitgegeben.

2.1 SIFT

Unter dem Akronym SIFT verbirgt sich Scale-Invariant Feature Transform Lowe (2004). Um bestimmte Objekte in der Umgebung zu lokalisieren, wird das Bild nach den invarianten Merkmalen untersucht. Dieses Verfahren ist unempfindlich gegen Koordinatentranslation, robust gegen Belichtungsvariation, Bildrauschen und geringen geometrischen Deformationen. Die markanten Punkte werden in vier aufeinanderfolgenden schritten lokalisiert, gefiltert und in einem 128-dimensionalen Vektor gespeichert Fries (2011). Anhand dieses Vektors kann das Bild durchsucht werden, um festzustellen ob das Objekt enthalten ist.

2.1.1 Ermittlung von Merkmalspunkte

Im ersten Schritt werden die Kandidaten als Merkmale gesucht. Dafür wird aus dem Quellbild mittels Gauß-Filters Bilddimension reduziert und in die in der Abbildung 2.1 dargestellten Gauß Pyramide eingeordnet. Für die Reduktion der Größe wird jeder zweite Pixel in x- und y-Richtung genommen. Aufgrund der Gaußfilterung enthält jeder Pixel die Information der Nachbarn. Die benachbarten Stufen der Pyramide werden voneinander subtrahiert und bilden damit eine weitere Pyramide, Difference of Gaussian (DoG). Anschließen wird nach Extrempunkten gesucht. Dafür werden die 8 direkten Nachbarn und 9 Nachbarn der benachbarten Schichten der DoG Pyramide untersucht. Ist der Wert größer bzw. kleiner, als alle Nachbarn, so ist ein Extrema gefunden.

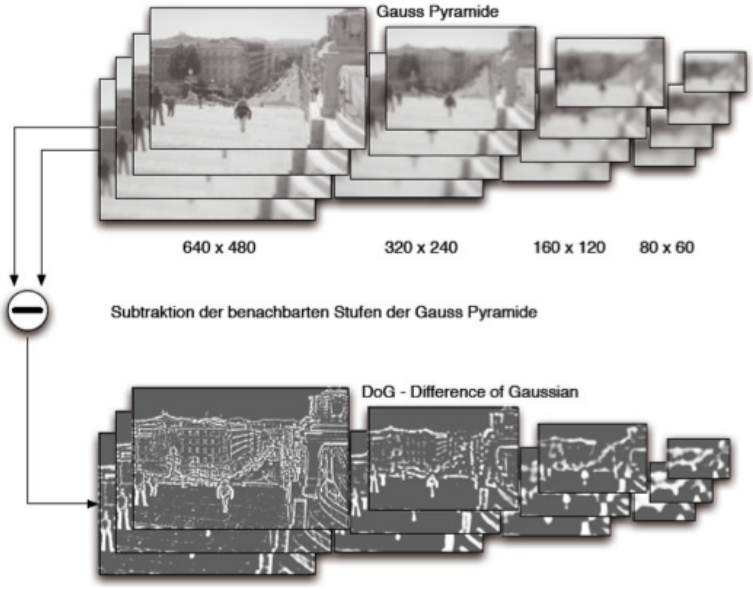


Abbildung 2.1: Gauß Pyramide zur Bestimmung der Merkmalpunkte (Quelle: Fries, 2011)

2.1.2 Filterung der Merkmalspunkte

Nicht jeder gefundene Extrempunkt ist als markanter Punkt geeignet. Aus der Menge der gefundenen Punkte werden nur stabile Punkte gefiltert. Stabile Bereiche sind beispielsweise kontrastreiche Punkte, Ecken, Kanten oder homogene Bereiche.

2.1.3 Bestimmung der Hauptorientierungen

Für die Bestimmung der Hauptorientierung eines Merkmals müssen zunächst die Gradienten seiner näheren Umgebung bestimmt werden. Dafür wird die Helligkeit der einzelnen Punkte untersucht. Der Gradient wird als Vektor mit der Länge und Orientierung angegeben. Anschließend werden die Werte anhand der Orientierung in 36 Winkelbereiche zusammengefasst und die Längen des Bereiches akkumuliert. Aus dem damit erzeugten Orientierungshistogramm wird mittels der größten Gradientenlänge die Hauptorientierung des Merkmals bestimmt.

2.1.4 Erzeugen eines 128-dimensionaler Merkmalvektors

Ein nach Lowe (2004) definierte 128-dimensionaler Deskriptor beschreibt eindeutig ein Merkmalspunkt. Er besteht aus seiner Bildposition, seiner Hauptorientierung und aus dem Merkmalsvektor. Der Deskriptor besteht aus 4x4 = 16 Orientierungshistogrammen. Die Histogramme werden, wie im Abschnitt 2.1.3 beschrieben, in 8 Winkelbereiche je 45° aufgeteilt. Anhand des

Merkmalsvektors kann beim Vergleich von Merkmalen die Invarianz bezüglich Helligkeits-Kontrastveränderungen und Rotation erreicht werden.

2.1.5 Beurteilung

Mithilfe des SIFT Verfahrens können rotierte, skalierte Objekte in unterschiedlichen Lichtverhältnissen und Teilweiser affiner Verzerrung erkannt werden. Dafür müssen alle zu erkennenden Objekte in einer Datenbank gespeichert werden. Kommt ein neues Objekt dazu, muss das System eingelernt werden. Aufgrund der fehlenden Fähigkeit zur Generalisierung kann dieses Verfahren nicht in allen Bereichen verwendet werden, weil oftmals nicht möglich ist zusätzliche Objekte Ad-hoc einzulernen.

2.2 Künstliche neuronale Netze

Die Idee von künstlichen neuronalen Netzen beruht auf dem menschlichen Gehirn, der aus Neuronen, deren Verbindungen und Synapsen besteht. Wird eine Verbindung wiederholt verwendet, verstärkt sich diese. Dieses Prinzip wird abstrakt für die Informationsverarbeitung angepasst (vgl. Wikipedia (2012)). Neuronale Netze werden oft für Bildverarbeitung verwendet um eine Klassifizierungsfunktion zu approximieren. Ein so genanntes Multilayer-Perzeptron Netz besteht aus den Neuronen, die zu mehreren Schichten eingeordnet sind. Man unterscheidet zwischen einer Eingangsschicht, mehreren verdeckten Schichten und einer Ausgangsschicht. In der Abbildung 2.2 ist ein beispielhaftes neuronales Netz dargestellt. Die

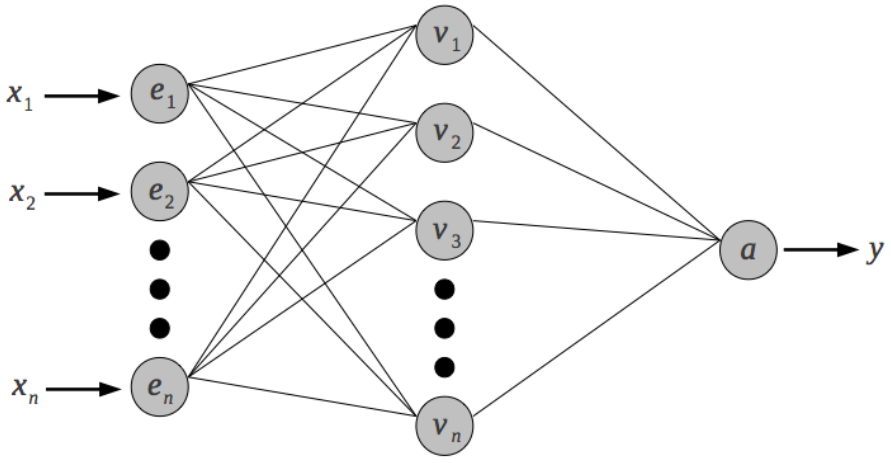


Abbildung 2.2: Neuronalse Netz (Quelle: Meisel, 2011)

Neuronen der Eingangsschicht sind mit den Neuronen der verdeckten Schicht voll vermascht. Weiterhin ist jede versteckte Schicht mit der nachfolgenden Schicht bzw. Ausgangsschicht voll vermascht. Über die Ausgangsschicht wird das Ergebnis der Klassifikation mitgeteilt.

2.2.1 Das Neuron

Ein in der Abbildung 2.3 dargestelltes Perzeptron Neuron bildet eine kleinste Einheit eines neuronalen Netzes und besteht aus einer oder mehreren Eingängen. Jeder der Eingänge wird mit dem entsprechenden Gewicht multipliziert. Die Summe der gewichteten Eingängen und dem Schwellwert (bias) bildet einen Wert, der mittels einer Aktivierungsfunktion zu einem Ausgangswert führt. Dabei kann der Ausgang nur eine 1 oder 0 als Wert annehmen. So kann

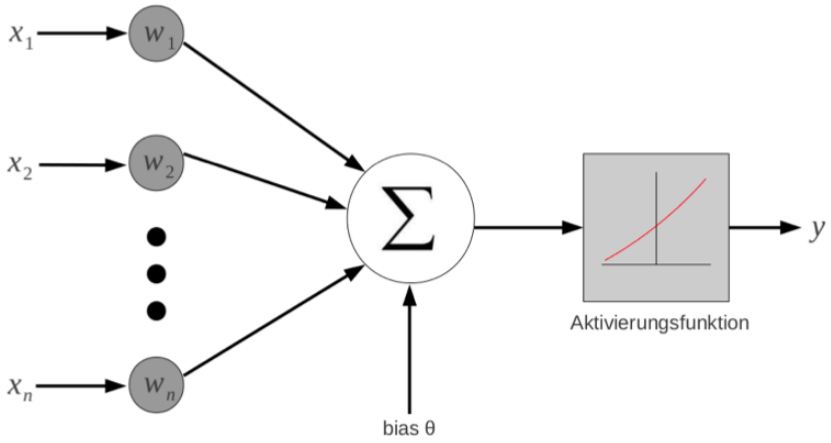
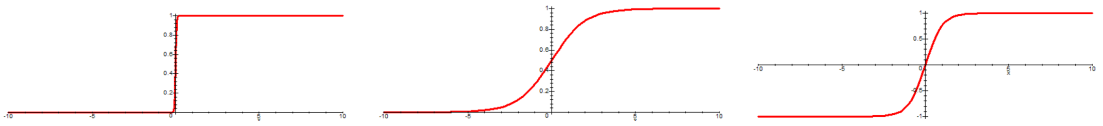


Abbildung 2.3: Aufbau eines Perzeptron Neurons (Quelle: Meisel, 2011)

festgestellt werden, ob ein Muster zu einer bestimmten Klasse gehört. Als Aktivierungsfunktionen werden, je nach Anwendung, in der Abbildung 2.4 Schritt- bzw. Sigmoidfunktionen verwendet. Der Vorteil einer Sigmoidfunktion liegt daran, dass sie differenzierbar ist und daher für den im Abschnitt 2.2.2 beschriebenen Lernalgorythmus am besten geeignet ist.



(a) Schrittfunktion

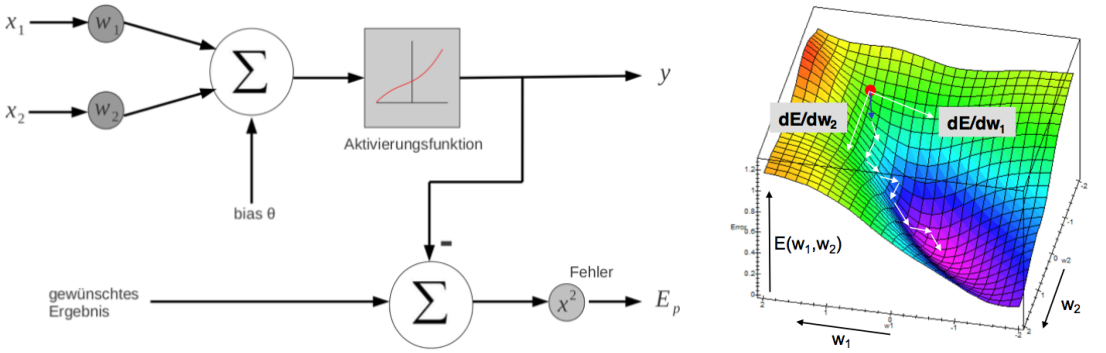
(b) Logistische Funktion (sigmoid) $f_{log} = \frac{1}{1+e^{-x}}$

(c) Tangens hyperbolicus (sigmoid) $f_{log} = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Abbildung 2.4: Arten von Aktivierungsfunktionen (Quelle: Meisel, 2011)

2.2.2 Einlernen von neuronalen Netzen

Das Ziel des Trainierens ist es, die Gewichte bzw. den Schwellwert der einzelnen Neuronen so einzustellen, dass der Fehler der Klassifizierung minimiert wird. Dazu wird am Eingang ein Trainingsmuster eingelegt (siehe Abbildung 2.5 links) und mit dem gewünschten Ergebnis verglichen. Anschließen wird ein Differenzvektor gebildet, der für die Stimmung der einzelnen Neuronen benutzt wird. Das Ziel ist es in der Abbildung 2.5 dargestelltes Tal zu erreichen, wobei die Schrittweite des Abstiegs im jeweiligen Zyklus vom Differenzvektor abhängt. Das



(a) Lernen eines neuronales Netzes (b) Backpropagation Lernalgorythmus

Abbildung 2.5: Schematische Darstellung eines Lernalprozesses (Quelle: Meisel, 2011)

Trainieren der Gewichte wird von der Ausgabeschicht rückwärts durchgeführt. Wurden alle Trainingsdaten einmal verwendet, so spricht man von einer *Epoche*. Als Abbruchbedingung für das Lernalgorythmus kann entweder eine bestimmte Fehlertoleranz oder eine bestimmte Anzahl der Epochen angegeben werden.

2.3 Convolutional Neural Networks

Die Convolutional Neural Networks stellen eine Weiterentwicklung von neuronalen Netzen dar. Um eine bessere Generalisierung zu erreichen, werden anstatt Gewichte die Faltungsmasken benutzt. Dabei werden die Informationen der benachbarten Pixel gebündelt an die nächste Schicht weitergegeben. Auch hier besteht das Netz aus Eingangsschicht, mehreren aufeinanderfolgenden Schichten und der Ausgangschicht. Vom Eingang zum Ausgang wird die Größe des Bildes stark reduziert, sodass die Pixel in der letzten Schicht die Information des Gesamtbildes enthalten.

2.3.1 Faltung

Mit einer Faltung wird das Quellbild nach bestimmten Kriterien wie beispielsweise Kantenglättung oder bezüglich der Schärfe gefiltert. Dabei wird mittels eines Faltungskerns ähnlich der in der Abbildung 2.1 Gauß Pyramide ein neues Bild erzeugt. Die korrespondierenden Pixel des Quellbildes werden mit den Koeffizienten des in der Abbildung 2.6 dargestellten Faltungskerns

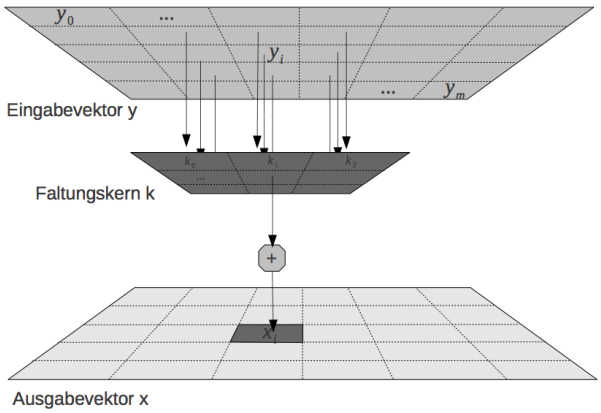


Abbildung 2.6: Grundprinzip der Faltung (Quelle: Hassenklöver, 2012)

multipliziert und aus der Summe ein neues Pixelwert im Zielbild gebildet. Nach der Anwendung auf ein komplettes Bild resultiert ein gefiltertes Zielbild. Mittels der Koeffizienten des Faltungskerns wird das entsprechende Filterungseffekt erreicht. Die Größe des Faltungskerns kann variabel gestaltet werden. Für die einfachere Rechnung werden die ungeraden Längen bzw. Höhen gewählt. Auch nicht quadratische Faltungskerne sind erlaubt.

2.3.2 Das Netzwerk und die Schichten

Ähnlich den in dem Abschnitt 2.2 beschriebenen Netzwerken besteht ein Convolutional Netzwerk aus einer Eingangsschicht, mehreren sich abwechselnden Convolutional und Subsampling Schichten und einer Ausgangschicht. Daher wird das Netzwerk logisch in Eingangsvector, *Feature Extraktion* und Klassifikationsbereich unterteilt. Der *Feature Extraktion* Bereich ist rot gestrichelt umrandet und der Klassifikationsbereich blau gestrichelt. In der Abbildung 2.7 ist ein beispielhaftes Netzwerk zur Klassifikation von Zahlen dargestellt. Der Eingangsvektor hat die Dimension 32x32 Pixel. Die Convolutionschicht stellt das Ergebnis der Faltung dar. Eine Subsamplingsschicht reduziert meistens die Bilddimension um Faktor vier und führt gleichzeitig Faltung durch. Im ersten Schritt wird das Eingangsbild mittels einer Convolutionschicht 5x5 gefaltet. Nach diesem Vorgang entstehen mehrere unabhängige Bilder, die in LeCun u. a.

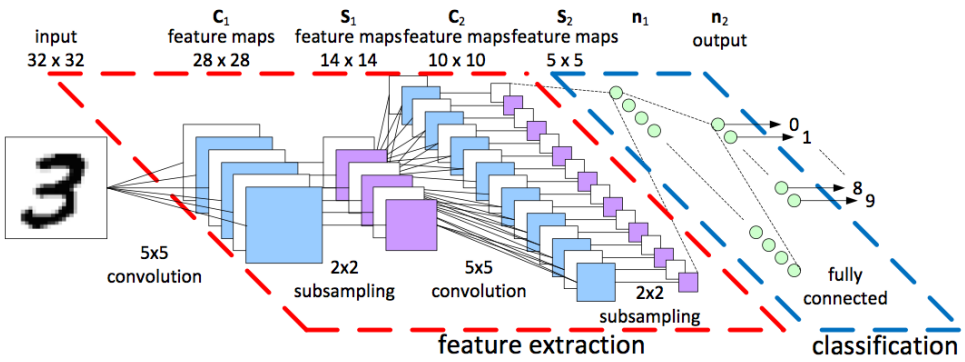


Abbildung 2.7: Ein beispielhafter Aufbau eines Convolutional Netzwerks zur Klassifikation von Zahlen (Quelle Peemen u. a., 2011)

(2001) als *feature maps* genannt werden. Jede Verbindungslinie steht für einen Faltungskern. Die Bestimmung der Gewichte wird beim Lernprozess bestimmt. Somit erhält man mehrere parallele Faltungen, die für Generalisierung sorgen. Die nachfolgende Subsamplingsschicht reduziert die Dimension um Faktor Vier. Jede nachfolgende Schicht kann wiederum mehrere *feauter maps* haben. Damit kann die Anzahl der *feature maps* beliebig wachsen und somit für eine hochgradige Generalisierung sorgen. Die letzte Convolution- bzw. Subsamplingsschicht wird mit den Perzeptron Neuronen der Klassifikationsschicht voll vermascht. Das bedeutet, dass jeder Perzeptron Neuron mit jedem *feature maps* der letzten feature extraction Schicht verbunden ist. Die Ausgangsschicht stellt dann mittels eines im Bereich von 0 bis 1.0 *Threshold* Wertes ob ein Treffer erzielt wurde Hassenklöver (2012).

2.3.3 Lernvorgang

Im Gegensatz zu den im Abschnitt 2.2 vorgestellten Multi-Perzeprton Netzwerken werden anstatt der Gewichte der Neuronen die Gewichte bzw. Koeffizienten der Faltungskerne bestimmt. Dafür wird im Abschnitt 2.2.2 vorgestellte Backpropagation-Lernalgorythmus angewendet. Bei der Wahl der Trainingsdaten sind einigen Regel zu beachten. Das Verhältnis der Repräsentanten der Objekte und der Abbildungen der Umgebung ist von besonderer Bedeutung. Um ein besseres Ergebnis zu erzielen, soll nach Hassenklöver (2012) das Verhältnis 3:2 betragen. Stimmt das Verhältnis nicht, so ist mit dem größeren Fehler zu rechnen. Die Anzahl der Epochen muss so gewählt werden, dass die Erkennungsrate maximiert wird. Es ist darauf zu achten, dass sich mehrere *feature maps* Filterkerne teilen können Duffner (2011). Die Subsamplingsschicht unterscheidet sich von der Convolutionschicht daran, dass sie für jede Verbindung nur ein zu trainierendes Gewicht enthält.

3 Related Works

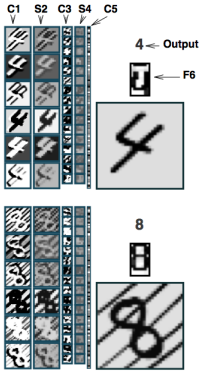
Dieses Kapitel stellt einigen mittels Convolutional Neural Networks realisierte Lösungen vor und vergleicht die erreichten Ergebnisse. Zu einem der führenden Forscher im Bereich von Faltungsnetzwerken gehört Yan LeCun. In Computational and Biological Learning Lab der Universität von New York betreut er das Projekt EBlearn++ Sermanet u. a. (2009b). Die dabei entstandene frei verfügbare Bibliothek verwendet er in seinen Forschungsarbeiten. Zu diesen Arbeiten gehören sowohl die Erkennung von 2-dimensionalen Objekten, wie beispielsweise Schriftzeichen LeCun u. a. (2001), als auch 3-dimensionalen Objekten mit perspektivischen Variationen und unterschiedlichen Lichtverhältnissen Huang und LeCun (2006).

3.1 Schrifterkennung

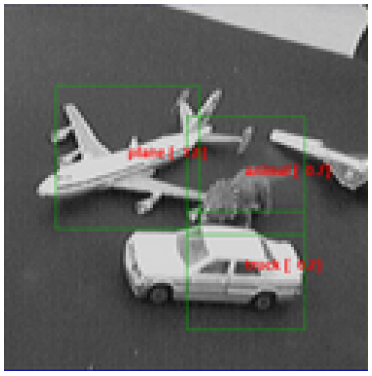
Mit dem Netzwerk LeNet 5 zur Erkennung von handgeschriebenen Zeichen hat LeCun u. a. (2001) gezeigt, dass Faltungsnetzwerke auch bei rotierten, stark verrauschten und belichtungsvarianten Eingangsdaten sehr gut funktionieren. In der Abbildung 3.1 links sind die Eingangsdaten, sowie die jeweiligen Faltungsschichten und das Ergebnis der Klassifikation dargestellt. Das Netzwerk wird auf die Bilder mit der Auflösung von 32x32 Pixel angewendet. Die dabei resultierende Erkennungsrate beträgt 99,0%.

3.2 Erkennung von hochvarianten Objekten

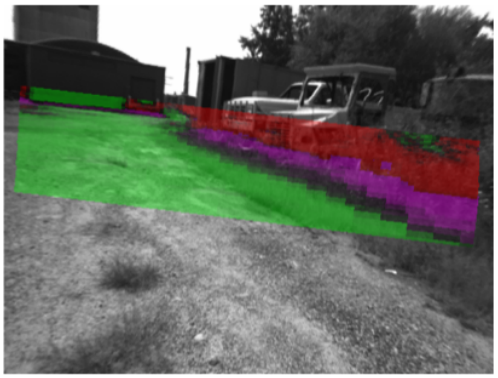
In der Abbildung 3.1 in der Mitte ist ein LeNet 7 dargestellt, das die hochvariante Objekte in unterschiedlichen Perspektiven und Lichtverhältnissen erkennt LeCun u. a. (2004b). Für diese Zwecke wurde ein Netzwerk namens LeNet 7 entwickelt. Dabei wurden als Objekte Menschen, Flugzeuge, Autos und Vierbeiner benutzt. Trotzt der hohen Varianz von Objektklassen, perspektivischer Rotationen und unterschiedlicher Lichtverhältnisse, wurden bei der Auflösung von 96x96 die Erkennungsrate von 93,8% erreicht. Weiterhin hat er gezeigt, dass für diese Berechnungen 4.66 Mio. Multiplikationsakkumulationen (MAK) benötigt werden. Ein Bild mit der Auflösung von 240x240, in 169 Teilbilder geteilt, erfordert 47.5 Mio. MAK. Ein nicht CNN



(a) Erkennung von handgeschriebenen Schriftzeichen



(b) Erkennung von hochvarianten Objekten mit wechselnden Lichtverhältnissen und Perspektive



(c) Erkennung der Strassenlage für Navigation eines Roboters im Gelände

Abbildung 3.1: CNN Beispielanwendungen (Quelle: LeCun u. a., 2001, 2004a; Sermanet, 2009)

basierte Klassifizierer erfordert dagegen 788 Mio. MAK (Siehe LeCun u. a. (2004a)), wobei fast 20 mal weniger Berechnungen benötigt werden.

3.3 Navigation im Gelände

Eine weitere Entwicklung dieser Forschungsgruppe ist in der Abbildung 3.1 rechts dargestellt. Es handelt sich um einer Architektur zur Planung von Routen für Roboter im Gelände. Der mit dem Faltungsnetzwerk lokalisierte Bereich wird in der Abbildung grün dargestellt. Die lila und rote Farbe markieren die Hindernisse Sermanet u. a. (2009a).

3.4 Erkennung von Verkehrszeichen

Mit einem weiteren Faltungsnetzwerk wurde von Sermanet und LeCun (2011) ein Fahrerassistent zum Erkennen der Verkehrsschilder entworfen. Alle Trainingsdaten Im Bereich von 15x15 bis 250x250 wurden auf eine gemeinsame Auflösung von 32x32 Pixel hoch- bzw. runterskaliert. In der Abbildung 3.2 sind einigen Beispiele dargestellt. Die dabei resultierende Erkennungsrate ergab 99,17%. Im Gegensatz dazu betrug die Erkennungsrate eines Menschen 98,81% (vgl. Sermanet und LeCun (2011)). Aufgrund der unterschiedlichen Klassifikationsverfahren kann vermutet werden, dass die Fehler bei unterschiedlichen Repräsentanten vorkommen. Daher bring diese Lösung eine enorme Sicherheit durch Redundanz.

Abbildung 3.2: Beispiele für Varianz von Verkehrszeichen (Quelle Sermanet und LeCun, 2011)

3.5 Erkennung von Verkehrsschilder zur Geschwindigkeitsbegrenzung

Das von dem Eindhoven University of Technology entwickelte CNN zur Erkennung von geschwindigkeitsbegrenzenden Verkehrsschilder kann in Echtzeit Bilder mit einer Auflösung von 1280x720 und einer Wiederholrate von 35,7 Bilder/s erkennen Peemen u. a. (2011). Dabei beträgt die Erkennungsrate 99,81%. Ein Wesentlicher Nachteil dieser Lösung ist der enorme Ressourcenverbrauch. Um diesen Anforderungen gerecht zu werden, wurde ein Cluster aus vier Rechnern mit jeweils vier leistungsfähigen Grafikkarten Aufgebaut. Für die Berechnung der Faltungsnetzwerke wurden die GPUs in Anspruch genommen.

3.6 Gesichtserkennung

In der Hochschule für Angewandten Wissenschaften Hamburg wurde von Hassenklöver (2012) ein Faltungsnetzwerk zur Erkennung von Gesichter realisiert. Dabei wurden mehrere Anwendungsfälle untersucht:

- das Bild enthält ein Gesicht bzw. kein Gesicht
- das Bild enthält ein bestimmtes Gesicht bzw. ein anderes Gesicht

Als Trainingsdaten kamen die Bilder mit der Auflösung von 32x32 Pixel in Frage. Die Erkennungsrate von irgendeinem Gesicht ergab 96,4%. Das Erkennen von einem bestimmten Gesicht betrug 53,8%.

4 Fazit

In dieser Arbeit wurde die Problematik der Bildverarbeitung in Bezug auf Objekterkennung deutlich gemacht. Zudem wurden die unterschiedlichen Verfahren zur Erkennung von Objekten in der Umgebung vorgestellt und analysiert. Weiterhin wurden die Ergebnisse der gleichartigen Arbeiten untersucht.

4.1 Risikoabschätzung

In diesem Abschnitt wird versucht die Risiken zu lokalisieren und deren Eintrittswahrscheinlichkeit bzw. resultierende Schaden abzuschätzen. Dafür werden alle Risiken in drei Stufen gering, mittel oder hoch eingestuft. Zu einem der wichtigsten Risiken gehören die eskalierenden Ressourcenanforderungen mit der steigenden Bildgröße (vgl. Peemen u. a. (2011)). Ein weiteres Problem stellt die explodierende Menge der Trainingsdaten bei einer großen Anzahl der zu erkennenden Objekten dar. Ein sehr wichtiger Aspekt, der direkt auf die Ergebnisse einwirkt, ist das richtige Verhältnis von Trainingsdaten. Die Trainingsdaten sollen nicht nur die Repräsentanten sondern auch Hintergrunddaten beinhalten. Wird beispielsweise zu geringer Anteil an Hintergrunddaten gewählt, so wächst damit der Fehler bei der Erkennung. Ein weiteres Risiko stellt Overlearning dar. Bei sehr oft wiederholten Lernvorgängen werden die Trainingsdaten auswendig gelernt, was die Möglichkeit zur Abstraktion und Generalisierung beeinträchtigt. Da jede Art der neuronalen Netzen nur einen Annäherung an die Zielfunktion darstellt, und diese nur mit heuristischen Methoden erreichbar ist, kann mit diesem Verfahren keine optimale Lösung garantiert werden. Bei den Aufgaben mit der polynomialen bzw. exponentiellen Komplexität wir das gerne in Kauf genommen, weil die Verbesserung oftmals empirisch nicht möglich ist. Eine Beobachtung dieser Risiken während des Projektverlaufs ist daher unerlässlich.

4.2 Chancen

Wie im Abschnitt 3.2 erwähnt, erfordern Faltungsnetzwerke ca. 20 mal so wenig Multiplikationsakkumulationen (siehe Huang und LeCun (2006)) wie die nicht CNN-basierte Klassifikato-

ren. Eine derartig große Ressourcenersparnis kann eine realistische Chance bei den Lösungen von komplexen Problemen der Bildverarbeitung darstellen. Zudem verspricht der hohe Grad der Generalisierung bzw. Abstraktion die Anwendung der Faltungsnetzwerke in bisher sehr schwer realisierbaren Anwendungen.

4.3 Ausblick auf weiteres Vorgehen

Auf der Grundlagen von der Bachhelorarbeit Hassenklöver (2012) soll zuerst untersucht werden wie die Anzahl der Objektklassen auf die Ekennungsrate einwirkt. Weiterhin soll untersucht werden, wie der Aufbau eines Faltungsnetzwerks in einem bestimmten Aufgabenbereich die Erkennungsrate der Objekte verbessern kann. Zunächst soll das Lernverhalten untersucht werden. Die Methoden zur Vorbereitung der Training bzw. Testdaten sollen entwickelt werden. Das erfolgreiche Lernen des Netzes hängt von der Qualität und der Varianz der Trainingsdaten bzw. der Testdaten. Daher soll eine optimale Konfiguration aus den Trainingsdaten, der Anzahl der Trainingszyklen gefunden werden. Mit dem Training eines geeigneten Faltungsnetzwerkes soll dann die Implementierung eines Produktes abgeschlossen werden.

Literaturverzeichnis

[Duffner 2011] Duffner, Stefan: Face Image analysis with convolutional neural networks. (2011)

[Fraunhofer Institut für Digitale Medientechnologie 2010] Fraunhofer Institut für Digitale Medientechnologie: Eye Tracking. (2010). – URL http://www.fraunhofer.de/de/presse/presseinformationen/2010/10/eye-tracker-sekundenschlaf-blickrichtungserkennung.html

[Fries 2011] Fries, Carsten: Kamerabasierte Identifizierung und Lokalisierung von Gegenständen für flexible Roboter. (2011)

[Hassenklöver 2012] Hassenklöver, Tobias: Klassifikation hochvarianter Muster mit Faltungsnetzwerken. (2012)

[Huang und LeCun 2006] Huang, Fu-Jie ; LeCun, Yann: Large-Scale Learning with SVM and Convolutional Nets for Generic Object Categorization. In: *Proc. Computer Vision and Pattern Recognition Conference (CVPR'06)*, IEEE Press, 2006

[LeCun u. a. 2004a] LeCun ; Huang ; Bottou: *Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting*. 2004. – URL http://www.cs.nyu.edu/~yann/research/norb/index.html

[LeCun u. a. 2001] LeCun, Y. ; Bottou, L. ; Bengio, Y. ; Haffner, P.: Gradient-Based Learning Applied to Document Recognition. In: *Intelligent Signal Processing*, IEEE Press, 2001, S. 306–351

[LeCun u. a. 2004b] LeCun, Yann ; Huang, Fu-Jie ; Bottou, Leon: Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. In: *Proceedings of CVPR'04*, IEEE Press, 2004

[Lowe 2004] Lowe, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision* (2004)

[Meisel 2011] Meisel, Andreas: Robot Vision Vorlesung. (2011)

[Peemen u. a. 2011] Peemen ; Maurice ; Mesman ; Bart ; Henk: Speed Sign detection and Recognition by Convolutional Neural Networks. (2011). – URL http://parse.ele.tue.nl/research/projects

[Sermanet 2009] Sermanet: *A Multi-Range Architecture for Collision-Free Off-Road Robot Navigation*. 2009. – URL http://www.cs.nyu.edu/~yann/research/lagr/index.html

[Sermanet u. a. 2009a] Sermanet, Pierre ; Hadsell, Raia ; Scoffier, Marco ; Grimes, Matt ; Ben, Jan ; Erkan, Ayse ; Crudele, Chris ; Muller, Urs ; LeCun, Yann: A Multi-Range Architecture for Collision-Free Off-Road Robot Navigation. In: *Journal of Field Robotics* 26 (2009), January, Nr. 1, S. 58–87

[Sermanet u. a. 2009b] Sermanet, Pierre ; Kavukcuoglu, Koray ; LeCun, Yann: EBLearn: Open-Source Energy-Based Learning in C++. In: *Proc. International Conference on Tools with Artificial Intelligence (ICTAI'09)*, IEEE, 2009

[Sermanet und LeCun 2011] Sermanet, Pierre ; LeCun, Yann: Traffic Sign Recognition with Multi-Scale Convolutional Networks. In: *Proceedings of International Joint Conference on Neural Networks (IJCNN'11)*, 2011

[Wikipedia 2012] Wikipedia: Künstliches neuronales Netz. (2012). – URL http://de.wikipedia.org/wiki/K%C3%BCnstliches_neuronales_Netz

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 29.02.2012 Vitalij Stepanov