
PROJEKT ZUR ERKENNUNG VON VERKEHRSZEICHEN

Aybüke Bayramıç
Informatik
Türkisch-Deutsche Universität
e160503111@stud.tau.edu.tr

Abstrakt

In diesem Projekt werden wir “Deep Neural Networks und Convolutional Neural Networks” verwenden, um Verkehrszeichen zu klassifizieren. Mit dem deutschen Verkehrszeichendatensatz werden wir ein Modell trainieren, um Verkehrszeichen in natürlichen Bildern zu dekodieren. Nachdem das Modell trainiert wurde, testen wir unser Modellprogramm an neuen Bildern von Verkehrszeichen, die wir im Internet finden.

1 Einführung

Verkehrszeichen sind ein wesentlicher Bestandteil unserer Straßeninfrastruktur. Ohne solche nützlichen Schilder wären wir höchstwahrscheinlich mit mehr Unfällen konfrontiert, da die Fahrer keine Rückmeldung darüber erhalten würden, wie schnell sie sicher fahren könnten, oder über Straßenarbeiten, scharfe Kurven und bevorstehende Schulübergänge informiert würden. In unserer heutigen Zeit sterben jedes Jahr rund 1,3 Millionen Menschen auf Straßen. Diese Zahl wäre ohne unsere Verkehrszeichen viel höher. Autonome Fahrzeuge müssen natürlich auch die Straßenverkehrsordnung einhalten und daher Verkehrszeichen erkennen und verstehen.

Traditionell wurden Standard-Computer-Vision-Methoden verwendet, um Verkehrszeichen zu erkennen und zu klassifizieren. Diese erforderten jedoch erhebliche und zeitaufwändige manuelle Arbeit, um wichtige Merkmale in Bildern von Hand herzustellen. Stattdessen erstellen wir durch Anwendung von Deep Learning auf dieses Problem ein Modell, das Verkehrszeichen zuverlässig klassifiziert und lernt, die für dieses Problem am besten geeigneten Merkmale selbst zu identifizieren.

2 Data

Der Datensatz [1], den wir zum Trainieren unseres eigenen benutzerdefinierten Verkehrszeichen-klassifikators verwenden, ist der Deutsche Verkehrszeichenerkennungs-Benchmark (GTSRB). Der GTSRB-Datensatz besteht aus 43 Verkehrszeichenklassen und fast 50.000 Bildern.

Trainingsbilder werden nach Tracks gruppiert. Jede Spur enthält 30 Bilder eines einzelnen physischen Verkehrszeichens. Die Bilder enthalten jeweils ein Verkehrszeichen.

Die Bildgrößen variieren zwischen 15 x 15 und 250 x 250 Pixel. Bilder sind nicht unbedingt quadratisch und das tatsächliche Verkehrszeichen muss nicht unbedingt im Bild zentriert sein. Dies gilt für Bilder, die sich im vollständigen Kamerabild nahe am Bildrand befanden.

Wenn ich detailliertere Informationen gebe, enthält dieser Datensatz 39209 Trainings- und 12630 Testbeispiele. Alle Bildgrößen sind zur Vereinfachung beim Schreiben des Codes auf $32 * 32 * 3$ eingestellt.

Es gibt Fotos von 43 verschiedenen Verkehrszeichen. Diese erstellen die Klassen. Es gibt Unterschiede in der Anzahl der Instanzen in diese Klassen. Zum Beispiel enthält der Trainingsdatensatz, weniger als 500 Instanzen der 42. Klasse, während die 1. Klasse mehr als 2000 enthält. Wir werden über Lösungen dafür sprechen.

Der Aufnahmewinkel, die Qualität und die Farbtöne aller Bilder unterscheiden sich voneinander. Alle werden in ein ähnliches Format gebracht, damit dies beim Training unseres Modells keine Schwierigkeiten verursacht.

Hier können Sie einige Beispiele von der Daten sehen.



3 Methodische Vorgehensweise

Ich verwende Jupyter Notebook in der Fortschritts- und Lernphase dieses Projekts. Ich gehe ähnliche Beispiele mit demselben Datensatz durch. CNN wird im Projekt verwendet. Es ist erwiesen, dass CNNs in der Lage sind, Probleme mit wirklich hoher Genauigkeit im Vergleich zur menschlichen Leistung zu lösen. Es wurden viele Arbeiten vorgeschlagen, um die Klassifizierung von Verkehrszeichen durch verschiedene Methoden des maschinellen Lernens zu behandeln, bei denen CNNs die anderen übertreffen. Da die Klassifizierung von Verkehrszeichen in der Automobilindustrie sehr gefragt ist, wurden große Anstrengungen unternommen, um eine Klassifizierung in Echtzeit zu erreichen. Also, was ist das CNN?

Beim Deep Learning ist ein Faltungs-Neuronales Netzwerk (CNN oder ConvNet) eine Klasse von Deep-Neural-Netzwerken, die am häufigsten zur Analyse visueller Bilder verwendet werden. Sie haben Anwendungen in der Bild- und Videoerkennung, in Empfehlungssystemen, in der Bildklassifizierung, in der medizinischen Bildanalyse, in der Verarbeitung natürlicher Sprache und in finanziellen Zeitreihen. [2]

CNNs sind regulierte Versionen von mehrschichtigen Perzeptronen. Mehrschichtige Perzeptrone bedeuten normalerweise vollständig verbundene Netzwerke, dh jedes Neuron in einer Schicht ist mit allen Neuronen in der nächsten Schicht verbunden. Die "vollständige Vernetzung" dieser Netzwerke macht sie anfällig für eine Überanpassung von Daten. [2]

Um mit dem Projekt fortzufahren, müssen die Daten untersucht und analysiert werden. Dazu laden wir zuerst die Daten auf unseren Computer herunter und lesen sie. Wir untersuchen die Informationen wie Training und Testset, Bildgröße im Detail. Visualisierung der Datenexploration können Sie finden und werden im Notizbuch angezeigt. Anzahl der Instanzen (von Klassen) können gesehen werden.

Wir stellen fest, dass der Datensatz ein ziemlich großes Klassenungleichgewicht aufweist. Einige Klassen haben weniger als 500 Beispiele, während andere mehr als 2000 haben. Nach der Visualisierung einer zufälligen Stichprobe von Bildern pro Klasse des Datensatzes finden wir die folgenden Eigenschaften in den Bildern.

Wie ich bereits erwähnte, die Größe der Zeichen variiert innerhalb derselben Klasse, es gibt eine leichte Drehung der Zeichen in den Bildern, vielleicht scheinen nur wenige Grad Bilder der Zeichen in leicht unterschiedlichen Winkeln aufgenommen zu werden. Zeichen scheinen hauptsächlich in der Nähe des Zentrums zu liegen des Bildes Die Qualität des Bildes auf der Innenseite des Zeichens variiert erheblich in Bezug auf die Qualität. Manchmal ist es schwierig zu sagen, was sich im Bild befindet. Die Farbe der Zeichen scheint in diesem Datensatz kein wesentlicher Faktor zu sein, jedes Zeichen scheint anhand der Form des Schilds und der Formen innerhalb des Schildes unterscheidbar zu sein. Dabei werden die Daten zuerst verarbeitet und dann in Graustufen konvertiert.

4 Verwandte Arbeit

Das Verkehrszeichenerkennungssystem kann allgemein in zwei Teile unterteilt werden: Erkennung und Erkennung. Die Verkehrszeichen haben eine charakteristische Farbe und spezifische Formen, die vom Fahrer leicht beobachtet werden können. Die Erkennung von Verkehrszeichen basiert normalerweise auf den inhärenten Merkmalen von Verkehrszeichen (wie Farbe und Form).

GonzalezReyna [3] schlug eine Methode vor, die auf einer Richtungsgradientenkarte und einer Karhunen-Loeve-Transformation basiert und deren Klassifizierungsgenauigkeit unter Verwendung des GTSDDB-Datensatzes (German Traffic Sign Detection Benchmark) 95,9% beträgt. In den letzten Jahren haben einige Wissenschaftler viele auf maschinellem Lernen basierende Methoden zur Erkennung von Verkehrszeichen vorgeschlagen [4], wie beispielsweise das Aggregate Channel Feature (ACF) und den Integrated Channel Feature Fusion Detector [5]. In [6] wird ein Verkehrszeichenerkennungsverfahren vorgeschlagen, das die evolutionäre Adaboost-Erkennung und die Wald-ECOC-Klassifizierung verwendet.

5 Verweise

- [1] <http://benchmark.ini.rub.de/?section=gtsrb&subsection=news>
- [2] https://en.wikipedia.org/wiki/Convolutional_neural_network
- [3] Gonzalez-Reyna S E, Avina-Cervantes J G, Ledesma-Orozco S E, et al. Eigen-gradients for traffic sign recognition[J]. Mathematical Problems in Engineering, 2013, 364305: 1-6.
<https://doi.org/10.1155/2013/364305>
- [4] Stallkamp J, Schlipsing M, Salmen J, et al. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition[J]. Neural networks, 2012, 32: 323-332.
<https://doi.org/10.1016/j.neunet.2012.02.016>
- [5] Møgelmoose A, Liu D, Trivedi M M. Detection of US traffic signs[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(6): 3116-3125.
<https://ieeexplore.ieee.org/document/7116530>
- [6] Baró X, Escalera S, Vitrià J, et al. Traffic sign recognition using evolutionary adaboost detection and forest-ECOC classification[J]. IEEE Transactions on Intelligent Transportation Systems, 2009, 10(1): 113-126.
<https://doi.org/10.1109/TITS.2008.2011702>

6 GitHub Link

<https://github.com/aybukeb/Deep-Learning/blob/master/pythonCodes.ipynb>