DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg
Campus Friedrichshafen

Funktion zur Erkennung und Darstellung von Bildern mit geringer Auflösung

Report Deep Learning

Studiengang Elektrotechnik

Studienrichtung Fahrzeugelektronik

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

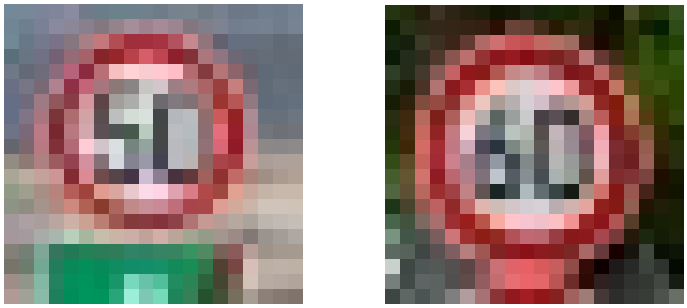
von

Max Rössler

Abgabedatum:	13.06.2022
Matrikelnummer:	2017444
Kurs:	TFE19-2
Ausbildungsfirma:	ZF Friedrichshafen AG
Gutachter der Dualen Hochschule:	Mark Schutera

1 Problemstellung

Um ein neuronales Netz zu validieren muss ein geeignetes Set aus Validierungsdaten vorliegen. Dabei gilt es herauszufinden, wie gut das neuronale Netz auf bestimmte Daten reagiert. Im Falle des Netzes zur Verkehrszeichenerkennung der Vorlesung Deep Learning haben die Validierungsdateien sehr viele Defizite. Bei der Überprüfung dieser Dateien wurde festgestellt, dass einige Bilder sehr kleine Auflösungen aufweisen. Diese Tatsache resultiert in sehr geringen Pixeldichten, wodurch nur bedingt eine sichere Aussage über die Klassifizierung des Bildes erfolgen kann. Am Beispiel in Abbildung 1.1 lässt sich die Auswirkung einer geringen Pixeldichte an zwei Vergleichsbildern erkennen. Hier sind Verkehrszeichen zur Mindestgeschwindigkeit von 50 km/h (1.1a) und 60 km/h (1.1b) mit einer Auflösung von 20x20 px dargestellt.



(a) **(b)**

Abbildung 1.1: Vergleich der Schilder 50 km/h und 60 km/h mit einer Auflösung von 20x20 px.

Durch diese geringen Pixeldichten erscheinen die Schilder auf den Bildern verschwommen und es sind keine scharfen Kanten mehr zu erkennen.

Das neuronale Netz kategorisiert die Verkehrsschilder anhand deren individuellen Eigenschaften. Wenn die Bilder jedoch durch eine zu geringe Auflösung wichtige Details verlieren, sinkt dadurch die Qualität der Klassifizierung. Auch für einen Menschen kann es schwierig sein, Bilder mit geringer Pixeldichte richtig zuzuordnen. Besonders Verkehrsschilder, die sich stark ähneln, lassen sich mit abnehmender Auflösung immer schlechter einordnen.

2 Lösungsansatz

Um dieses Problem bei der Validierung des neuronalen Netzes zu berücksichtigen, müssen Bilder mit geringer Auflösung hervorgehoben werden. Dazu wird zunächst festgelegt, mit welcher Sicherheit ein Bild mindestens klassifiziert werden muss, um einen Klassifizierungsfehler aufgrund der Auflösung ausschließen zu können.

Anhand von diesem Schwellwert sollen Bilder mit geringer Auflösung detektiert werden. Um mit diesen Daten arbeiten zu können wird eine grafische Oberfläche generiert, welche sowohl die Bilder, als auch die zugehörigen Daten visualisiert. Zu diesen Daten gehört der Titel der Datei, die Auflösung des Bildes, die Wahrscheinlichkeit der Klassifizierung dieses Bildes und das Ergebnis der tatsächlichen Klassifizierung. Anhand dieser Daten kann der Nutzer ableiten, ob die dargestellten Bilder aufgrund Ihrer Bildgröße falsch klassifiziert wurden.

Ein weiteres Problem liegt darin, dass das Verkehrsschild auf jedem Bild keine festgelegte Größe aufweist. Es kann also vorkommen, dass ein Verkehrsschild nur eine sehr kleine Fläche des Bildes einnimmt und dadurch an Qualität verliert. Tritt dieses Problem in Kombination mit einer geringen Auflösung des Bildes auf, fällt die Klassifizierung des Verkehrsschildes denkbar schwer.

Auch dieses Problem kann in der grafischen Oberfläche vom Nutzer überprüft werden. Wenn das Verkehrszeichen auf einem Bild, zusätzlich zur geringen Auflösung, sehr klein ist, kann möglicherweise keine aussagekräftige Klassifizierung vorgenommen werden.

Eine Schlussfolgerung des Nutzers könnte im Falle von falsch klassifizierten Bildern z.B. eine Erweiterung der Trainingsdaten dieser Klasse sein.

3 Funktionen

3.1 Einzelwahrscheinlichkeit der Klassifizierung eines Bildes

Um zu entscheiden, ob ein Bild mit ausreichender Sicherheit klassifiziert wurde wird die Wahrscheinlichkeit des neuronalen Netzes benötigt, mit der das Bild dieser Klasse zugeordnet wurde. Für diese Funktion wird die Operation Softmax des Tensorflow-Moduls „Primitive Neural Net Operations“ (NN) verwendet. Diese Funktion kann für Klassifizierungsaufgaben mit mehreren verfügbaren Klassen verwendet werden, um die Wahrscheinlichkeit der Klassifizierung für jede Klasse zu erhalten. Die Summe aller dieser Klassenwahrscheinlichkeiten für ein Bild beträgt eins.

Das neuronale Netz weist jedem Bild eine Wahrscheinlichkeit zu. Diese sagt aus, wie wahrscheinlich das Bild dieser Klasse zugeordnet werden kann. Nachdem alle Werte für ein Bild berechnet wurden, wird dieses in die Klasse mit der höchsten Wahrscheinlichkeit eingeordnet. [vgl. Ten22]

Die Softmax-Funktion muss aufgerufen werden, nachdem die Validierungsdaten klassifiziert wurden. Der Rückgabewert der Funktion wird auf die Variable softmax geschrieben, welche in eine weitere Funktion calc_prediction_probability übergeben wird. Da der Rückgabewert der Softmax-Funktion in Form eines Tensors vorliegt, wird dieser zunächst in ein Numpy-Array umgewandelt. Dadurch lassen sich die Werte besser weiterverarbeiten, da ein Tensor den Datentyp dtype hat. [vgl. Ten22]

Der nächste Schritt besteht darin, für jedes Bild den Maximalwert der Klassenwahrscheinlichkeiten zu finden. Dazu wird die Numpy-Funktion amax verwendet. Der Maximalwert für jedes Bild wird schließlich in eine Liste geschrieben, welche ebenso der Rückgabewert dieser Funktion ist.

3.2 Schwellwertberechnung der Auflösung

Um Bilder mit einer geringen Auflösung ausgeben zu können muss ein Schwellwert ermittelt werden, ab dem die Auflösung eines Bildes als kritisch angesehen wird. Dieser Schwellwert kann entweder anhand von komplexen Versuchen festgelegt werden, oder individuell für jedes Set von Validierungsdaten berechnet werden.

Dafür werden zunächst alle Bilder der Validierungsdaten mithilfe der Funktion os.walk durchlaufen. Dabei wird die jeweilige Position in der Liste der Einzelwahrscheinlichkeiten mit dem festgelegten Schwellwert verglichen. Liegt die Wahrscheinlichkeit unter dem Schwellwert, wird die Auflösung dieses Bildes in zwei Listen für horizontale und vertikale Pixelanzahl gespeichert.

Dadurch liegen nun alle Auflösungen vor, welche unter dem Schwellwert der Einzelwahrscheinlichkeiten liegen. Anhand von diesen Daten wird durch die Bildung der Mittelwerte von beiden Achsen zwei Schwellwerte für die Auflösung generiert. Diese werden als Rückgabewerte übergeben.

3.3 Bilder mit geringer Auflösung finden

Anschließend können alle Validierungsdateien durchlaufen werden, um kleine Bilder auf Basis der errechneten Schwellwerte zu finden. Dafür wird eine Klasse critical _image angelegt, welche die Attribute Dateipfad, Dateinamen, Auflösungswerte, Einzelwahrscheinlichkeit, Klassifizierung und korrekten Klasse enthält. Beim Durchlaufen der Dateien wird sowohl die horizontale, als auch die vertikale Anzahl an Pixeln mit den Schwellwerten verglichen. Fällt ein Bild unter diese, wird davon ein Objekt mit allen Attributen erzeugt und in die Liste critical _images geschrieben. Diese Liste dient gleichzeitig als Rückgabewert der Funktion.

3.4 Grafische Oberfläche

Wie in Kapitel 2 beschrieben soll der Nutzer über alle Validierungsdateien informiert werden, welche eine geringe Auflösung aufweisen. Dazu wird eine grafische Oberfläche implementiert, die mit der Bibliothek tkinter umgesetzt wird. [vgl. tki22]

Als Orientierung für einen scrollbaren Container wird [You22] verwendet. Zu Beginn wird ein Hauptfenster erstellt, welches als Basis für die GUI dient. In diesem Hauptfenster wird anschließend ein canvas erzeugt. welches die volle Größe des Hauptfensters einnimmt. Diesem Canvas wird ein Scrollbar−Objekt hinzugefügt, welches dafür sorgt, dass die Liste an Bildern nicht auf den festgelegten Bereich in der GUI beschränkt ist. Durch eine vertikale, scrollbare Liste wird die Benutzerfreundlichkeit erhöht.[vgl. use17]

Für die Anzeige des eigentlichen Inhalts wird im Canvas ein weiterer Frame erzeugt. Dieser ist notwendig, da in einem Canvas keine direkten Inhalte platziert werden können. Durch eine Schleife werden alle Bilder in der Liste critical _images untereinander im Frame Platziert. Dies wird durch das grid−Attribut eines Labels durchgeführt. Als Reihenindex wird der Iterator der Schleife verwendet. Neben den Bildern werden zusätzlich die Attribute Dateiname, Auflösung, Wahrscheinlichkeit und Klassifizierung dargestellt. Durch ein Message-Objekt werden diese untereinander und linksbündig gelistet.

4 Zusammenfassung und Ausblick

Durch die grafische Oberfläche und die strukturierte Auflistung von potentiell fehlerhaften Bildern kann der Nutzer schnell identifizieren, auf welchen Klassen Probleme entstehen können, wenn die zugehörigen Bilder eine zu geringe Auflösung aufweisen. Durch die Anpassung des Schwellwerts für die Einzelwahrscheinlichkeit zur Klassifizierung kann bestimmt werden, wie hoch die Anzahl an Bildern ist, aus denen der Schwellwert für die Auflösung berechnet wird.

Diese Berechnung wurde unter der Annahme implementiert, dass die Wahrscheinlichkeit zur Klassifizierung mit sinkender Auflösung kleiner wird. In den Versuchen wurde jedoch ersichtlich, dass die Validierungsdaten zusätzlich zur Auflösung weitere Defizite aufweisen, welche ebenfalls Einfluss auf die Wahrscheinlichkeit haben. Deshalb könnte eine weitere Variante der Funktion sowohl einen festgelegten Schwellwert für die Einzelwahrscheinlichkeit, als auch für die Auflösung haben. Dadurch könnte beim detektieren der kritischen Bilder eine weitere If-Abfrage implementiert werden, welche nur Bilder sucht, die eine geringe Wahrscheinlichkeit und eine geringe Auflösung haben. Diese Variante der Funktion mit einem festgelegten Schwellwert für die Auflösung wurde zu Versuchszwecken zusätzlich im Quellcode implementiert (siehe Funktion get_small_images, Alternative function with fixed threshold_shape).

Eine exakte Berechnung für einen Schwellwert der Auflösung wäre durch umfangreiche Validierungsdaten möglich, in denen Bilder ohne sonstige Defizite in unterschiedlichen Größen vorkommen. Dadurch könnten sehr genaue Schwellwerte berechnet werden, ab denen ein Bild mit sinkender Wahrscheinlichkeit klassifiziert wird. Vermutlich würde die Kennlinie der Einzelwahrscheinlichkeit, aufgetragen über die Pixeldichte, linear, oder sogar exponentiell sinken.

Literatur

[Ten22] TensorFlow. *tf.nn.softmax | TensorFlow Core v2.9.1*. 24.05.2022. URL: https://www.tensorflow.org/api_docs/python/tf/nn/softmax (besucht am 11. 06. 2022).

[tki22] tkinter. *tkinter — Python interface to Tcl/Tk — Python 3.10.5 documentation*. 11.06.2022. URL: https://docs.python.org/3/library/tkinter.html (besucht am 11. 06. 2022).

[use17] user3300676. *tkinter Canvas Scrollbar with Grid?* 2017. URL: https://stackoverflow.com/questions/43731784/tkinter-canvas-scrollbar-with-grid (besucht am 09. 06. 2022).

[You22] YouTube. *Adding a Full Screen ScrollBar - Python Tkinter GUI Tutorial #96*. 9.06.2022. URL: https://www.youtube.com/watch?v=0WafQCaok6g (besucht am 09. 06. 2022).