

Digitale Bildverarbeitung und Mustererkennung

Dokumentation des Programmentwurfs

Studiengang Elektrotechnik

Studienrichtung Fahrzeugelektronik

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Paul Straubmeier

Erstellungsdatum:	27. Dezember 2023
Bearbeitungszeitraum:	25.10.2023 - 05.01.2024
Matrikelnummer:	9062584
Kurs:	TFE21-2
Gutachter der Dualen Hochschule:	Mark Schutera

1 Ziele

Die vorliegende Dokumentation ist eine Ergänzung zum Programmentwurf. Sie soll Ideen bei der Verbesserung des Modells „Marvin“ darstellen und geht auf verwendete Experimente, Methoden und Ergebnisse ein. Bevor mit der Umsetzung der Software begonnen wird, sind im Folgenden die festgelegten Ziele zu sehen. Der beiliegende Entwurf versucht eine gute Balance zwischen einer hohen Genauigkeit und wenigen Parametern und Labeln zu finden.

Accuracy	Parameter	Label
> 98 %	< 5000	< 5000

2 Umsetzung

Architektur

Meine Architektur zur Ziffererkennung besteht aus sieben Layern. Als Netztyp wurde ein Convolutional Neural Network (CNN) gewählt, da dieser fähig ist automatisch komplexe Merkmale aus Rohbilddaten zu extrahieren und zu erlernen. Er eignet sich damit besonders für die Mustererkennung beim MNIST-Datensatz. [Ana23] Die Hyperparameter der einzelnen Layer wurden, aufgrund der geringen Erfahrung mit neuronalen Netzen, meist durch Experimentieren bestimmt.

1. Das Modell beginnt mit einem Convolutional-Layer („Conv2D“), um Muster und Merkmale des Labels zu erkennen. Diese Schicht hat 16 Filter der Größe 3x3. Die genutzte Aktivierungsfunktion ist „ReLU (Rectified Linear Unit)“.
2. Es folgt ein „Max Pooling“-Layer mit einem Fenster der Größe 2x2. Dies reduziert die Dimensionen und somit die Parameter. Außerdem konzentriert sich das Modell so nur auf die relevantesten Informationen.
3. Um die Genauigkeit der Ergebnisse zu verbessern, wird der gleiche Ablauf nochmal ver-

wendet. Diesmal hat der „Conv2D“-Layer 8 Filter der Größe 3x3. Die genutzte Aktivierungsfunktion ist wieder „ReLU“.

4. Es folgt ein weiteres mal ein „Max Pooling“-Layer mit einem Fenster der Größe 2x2.

5. Im nächsten Schritt wird ein „Dropout“-Layer eingebaut. Bei diesem werden zufällig 50 % der Neuronen während des Trainings deaktiviert, um Overfitting zu reduzieren.

6. An dieser Stelle werden die Daten im „Flatten“-Layer von einer Feature Map in einen eindimensionalen Vektor umgewandelt, um sie als Eingabe für den „Dense“-Layer verwenden zu können.

7. In diesem „Dense“-Layer werden die Bilder klassifiziert. Die Ziffer mit der höchsten Wahrscheinlichkeit wird dann ausgegeben.

Training

Trainiert wurde zu Beginn nur mit 30 Epochen, um kurzfristige Folgen verschiedener Hyperparameter und das Einbauen weiterer Layer zu testen. Dadurch spart man Rechen- und Wartezeit und kann schnell auf Veränderungen reagieren. Bei der Verfeinerung bis zum finalen Modells trainierte ich dann bis zu 300 Epochen, um zu sehen, wie sich das Netz in höheren Epochen verhält und wie lange es sich noch verbessern kann bis es einen Grenzwert erreicht. Falls das Modell diesen erreicht hat wird zur Vermeidung von unnötiger Rechenzeit der „Early Stopping Callback“ mit einer Patience von 30 benutzt. Wenn sich die Accuracy folglich 30 mal nicht verbessert hat, wurde das Training beendet. Mit „Shuffle“ wurde sichergestellt, dass die Trainingsdaten vor jeder Epoche gemischt werden. Dies hilft bei der Generalisierung des Modells und verhindert, dass die Reihenfolge der Daten auswendig gelernt wird. Außerdem wurde mit „Validation Split“ ein Teil des Datensatzes als Validierungsdatsatz verwendet, um bereits während des Trainings das Risiko von Overfitting zu verringern. Für die „Batch Size“ wurde 32 festgelegt, da es üblich ist.

3 Ergebnisse

Das Ergebnis meiner Anstrengungen zeigen folgende Werte.

Accuracy	Parameter	Label
98,19 %	3330	5000

In allen drei Kategorien wurden die Ziele erreicht. Dennoch macht das Modell ein paar kleinere Fehler. Die größten Probleme liegen bei der Ziffer 7. Marvin erkennt sie häufig als 2. Auch die Ziffer 9 wird häufig als 4 erkannt. Am besten erkennt das Modell die 0 und die 1. In 10.000 Testlabels wurde diese Ziffern nur 11 mal nicht erkannt. Die folgende Matrix fasst die Ergebnisse zusammen.

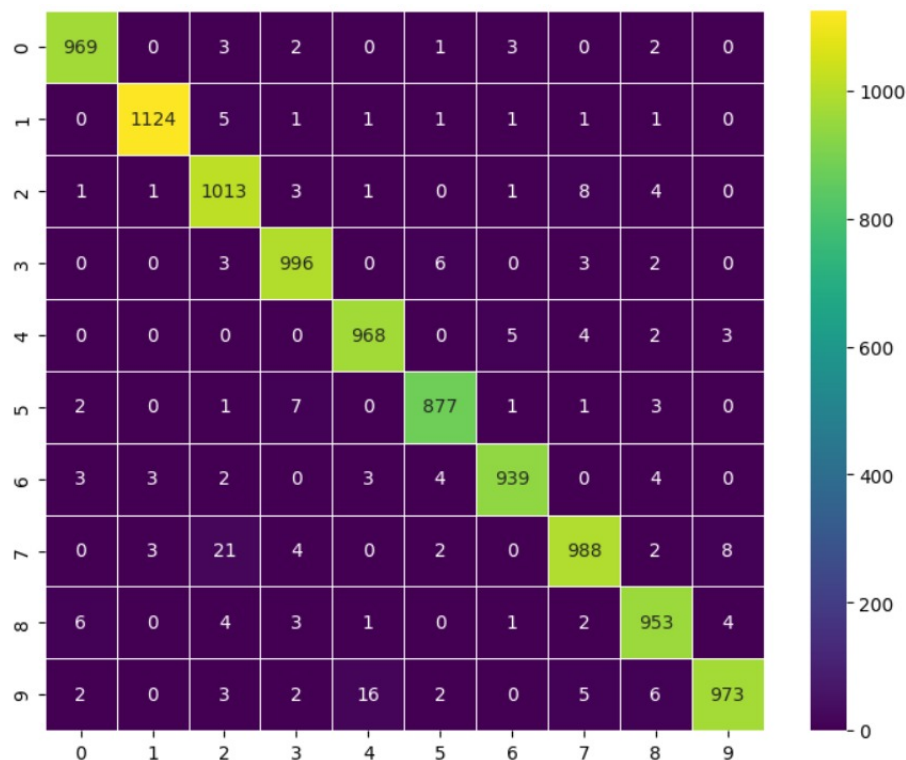


Abbildung 1: Darstellung der richtigen/falschen Zifferbestimmungen des Modells

Der Grund für die Schwierigkeiten von Marvin bei den Ziffern 4 und 7 bzw. 9 und 4 ist die Ähnlichkeit der Ziffern in der Handschrift. Für eine bessere Unterscheidung könnte man mehr Labels zum Trainieren verwenden oder ein komplexeres Modell mit mehr Layern und Parametern erschaffen.

Literatur

- [Ana23] Vidhya Analytics. *Image Classification Using CNN. Why CNN for Image Classification?* [Online; accessed 27-December-2023]. 2023.