

Aufgaben zu DLM7

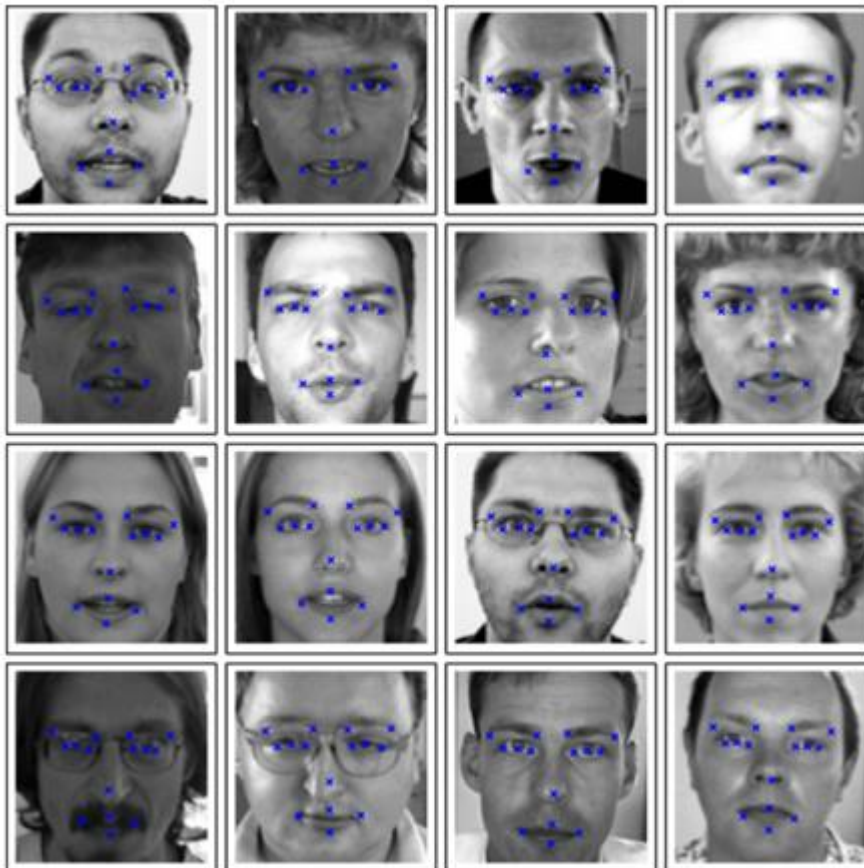
Im heutigen Labor werden Sie zentrale Punkte in Gesichtern (Facial Keypoints) mit Hilfe von Neuronalen Netzwerken bestimmen.

Insgesamt stehen 7048 verschiedene Bilder zur Verfügung. Da jedoch nicht jedes dieser Bilder alle 30 Keypoints als Target beinhaltet, werden nur diejenigen Bilder zum Training genutzt, die die vollständigen Target Labels haben.

Zum Laden der Daten benutzen wir die Python Bibliothek Pandas.

Da die Testdaten keine targets enthalten, müssen wir den Datensatz eigenständig in einen Trainings- und einen Validations-Datensatz teilen.

Die Testdaten sollten jedoch zur visuellen Begutachtung Ihres Netzwerkes in Betracht gezogen werden.



1) Laden Sie die Daten mit Hilfe der *load2d* Methode. Gehen Sie die Funktion Schritt für Schritt durch und versuchen Sie jeden Schritt nachzuvollziehen.

2) Warum kann es vorteilhaft sein, den Target-Vektor y vor dem Training zu skalieren?

Ist eine Methode der anderen vorzuziehen? $[0, 1]$ vs $[-1, 1]$

Vergleichen Sie dazu im Weiteren die Performance Ihres Netzwerks bezüglich beider Skalierungen.

3) Warum skalieren wir alle Bilddaten vor dem Training $[0, 1]$? Konvergiert Ihr Netzwerk auch ohne Skalierung, d.h., wenn die Pixelintensitäten im Intervall $[0, 255]$ liegen?

4) Trainieren Sie ein einfaches ConvNet. Visualisieren Sie die berechneten Keypoints in Ihren Testdaten. Wie schätzen Sie die Performance Ihres Modells? Achten sie auf die Varianz der Keypoints in y_{pred} .

5) Erweitern Sie das ConvNet um DropoutLayer. Trainieren Sie Ihr Netzwerk und visualisieren Sie das Ergebnis erneut.

Tipp: Versuchen Sie erstmal einen DropoutLayer zwischen die letzten beiden Fully-connected layers hinzuzufügen. Als Standard-Wert können Sie $p=0.5$ nutzen.

Fügen Sie anschließend mehr Dropout-Layer hinzu. Experimentieren Sie mit der probability p .

6) Versuchen Sie Ihr Netz mit Data Augmentation weiter zu trainieren. (flip_ratio im Generator auf z.B. 0.5 setzen) Was können Sie bezüglich des Trainings erkennen? Fällt der Loss schneller?

