Klassifizierung von Werkzeugen

(freie Aufgabe)

Praktikumsbericht Computervision

Sommersemester 2019

Moritz Klein

Frank Neumann

# Beschreibung der freien Aufgabe

Wir versuchen drei Werkzeuge (Hammer, Zange und Hobel) mit Hilfe eines neuronalen Netzes zu klassifizieren. Hierzu wurde uns von einem Kommilitonen das vortrainierte Netzwerk VGG16 [1] empfohlen. Für die Erkennung der eigenen Werkzeuge wird nur der letzte Layer mit neuen Bildern trainiert. Bei der abschließenden Klassifizierung verfolgen wir zwei Ansätze:

Im ersten Ansatz wird in einem Bild mit mehreren Werkzeugen ein Werkzeug eines bestimmten Typs gesucht und anschließend im Bild markiert. Hierzu wird das Bild mit einer Maske abgetastet und dieser maskierte Bereich an das neuronale Netz zur Klassifikation übergeben. Dabei werden für die einzelnen Bereiche die Wahrscheinlichkeiten gespeichert und am Ende der Bereich mit der höchsten Wahrscheinlichkeit markiert.

Im zweiten Ansatz werden die von uns trainierten Werkzeuge in einem Bild unter Zuhilfenahme des OpenCV-Frameworks markiert, und diese Bildausschnitte an das neuronale Netz zur Klassifikation übergeben.

# Ausarbeitung

## Datenbeschaffung

Zunächst wurde im Internet nach entsprechenden Datensätzen recherchiert. Die Trainingsdatensätze wurden aus dem ImageNet [2] Datensatz mit Hilfe des imagenetscraper command-line utility beschafft [3].

$ imagenetscraper n03481172 data/training/hammer --size 224,224 # hammer

$ imagenetscraper n02680754 data/training/wrench --size 224,224 # wrench

$ imagenetscraper n03954731 data/training/plane --size 224,224 # plane

Damit das Netzwerk nicht durch markante Bildausschnitte falsche Klassifikationen lernt, wurden zusätzlich Bilder mit Texturen gesucht [4] und das Netzwerk entsprechend trainiert. Diese Texturen wurden vor dem Training mit einem Shell-Script auf die gleiche Größe gebracht:

# set same size for UIUC images

mkdir $folder\_unsorted/background

for file in $folder\_path/uiuc\_texture/\*.jpg; do

fileName=$(basename "$file")

convert $file -resize $sizepx\x$sizepx! $folder\_training/background/$fileName

done

### Aufteilung der Datensätze für Training, Validierung und Test

Die Datensätze für Training, Validierung und Test wurden folgendermaßen aufgeteilt:

Training: 3667 Bilder (80%)

Validierung: 468 Bilder (10%)

Test: 470 Bilder (10%)

## Kurze Vorstellung des neuronalen Netzwerks VGG16

Das Convolutional Neuronal Network VGG16 [1] wurde von K. Simonyan und A. Zisserman von der University of Oxford im Paper „Very Deep Convolutional Networks for Large-Scale Image Recognition“ vorgestellt. Es gehört mit 92,5% Test-Genauigkeit zu den TOP 5 Netzwerken beim Vergleich des ImageNet Datensatzes, welcher aus über 14 Millionen Bildern und 1000 Klassen besteht [2].

Um das letzte Layer für unseren Fall zu trainieren wurden die Basislayer eingefroren und der Final Layer neu definiert und entsprechend unserer gewünschten Klassifikation trainiert.

## Werkzeuge finden

Beim ersten Ansatz, bei dem ein bestimmtes Werkzeug im Bild gesucht wurde, erhielten auch Bildausschnitte ein False-Positive, wenn nur ein kleiner Ausschnitt des Werkzeuges zu sehen war. Das war auch dann der Fall, wenn eine hohe Konfidenz angestrebt und die Sigmoid-Funktion verwendet wurde (Abb. 1).

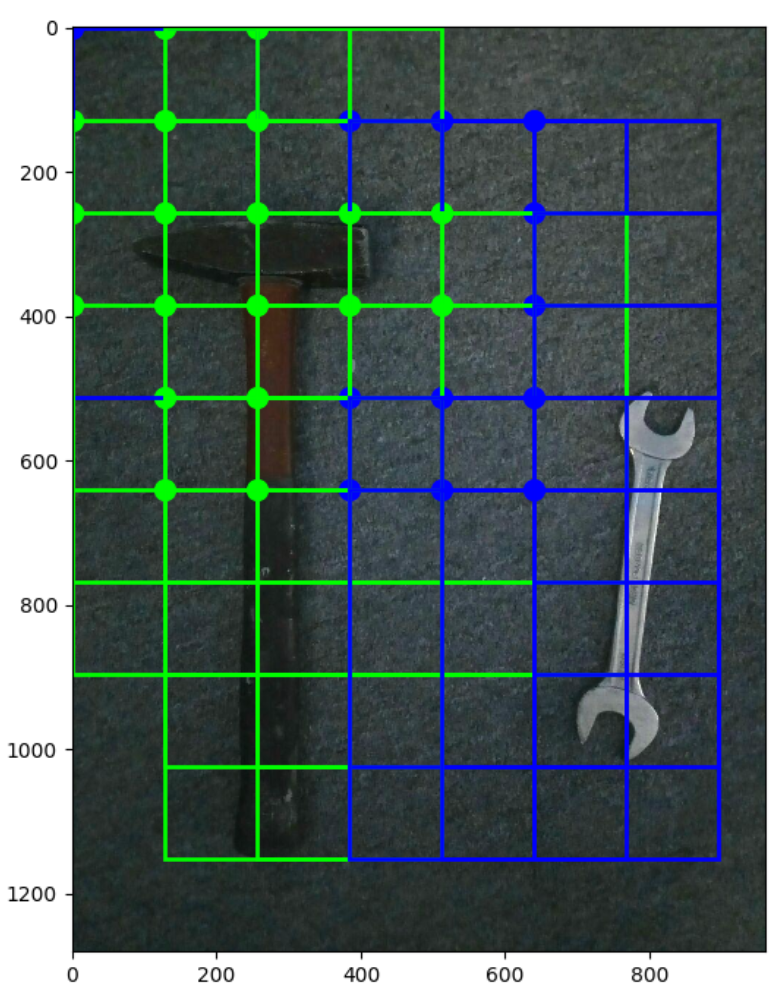


Abbildung – Gerastertes Bild

Im zweiten Ansatz wurden Boundingboxen mit Hilfe von OpenCV-Algorithmen zur Kanten- bzw. Konturenerkennung um die Objekte gezeichnet. Diese Bereiche wurden anschließend an das neuronale Netz zur Klassifizierung zu übergeben (Abb. 2).



Abbildung – Gelabeltes Bild

## Ergebnis

# Referenzen

[1] <https://neurohive.io/en/popular-networks/vgg16/>

[2] <http://image-net.org>

[3] <https://github.com/spinda/imagenetscraper>

[4] <http://slazebni.cs.illinois.edu/research/uiuc_texture_dataset.zip>