



WESTFÄLISCHE
WILHELMS-UNIVERSITÄT
MÜNSTER

RotEqNet

Rotierte Kernel zur Bestimmung der Blickrichtung von Mistkäfern

Gliederung

- ▶ Motivation
- ▶ FIM - Datensatz
- ▶ Künstliche Neuronale Netze
- ▶ Metriken
- ▶ Autoencoder
- ▶ Deconvolution Neural Network
- ▶ Architekturen zur Bildsegmentierung
- ▶ Denoising Autoencoder

Motivation

Bei RotEqNet (Rotation equivariant vector field network) handelt es sich um eine Convolutional-Netzarchitektur, welche sich durch folgende Punkte auszeichnet:

- ▶ zusätzliche Rotation der Kernel beim Bewegen über den Input
- ▶ sehr kompakte Modelle mit wenig Parametern
- ▶ vergleichbare Leistung zu Netzen mit deutlich größeren Modellen

RotEqNet - Layers

RotConv

Forwardpass

- ▶ Rotation eines kanonischen Grundfilters
- ▶ Anwendung der rotierten Filter auf den Input
- ▶ zusätzliche Durchpropagierung des Winkels

Backwardpass

- ▶ Rückrotation und Summation aller Gradienten
- ▶ Anpassung der Gewichte des kanonischen Filters

RotEqNet - Layers

Orientation Pooling

- ▶ pixelweise Bestimmung der größten Magnitude
- ▶ Bestimmung des zugehörigen Winkels
- ▶ Anwendung von ReLu auf die Magnitude
- ▶ Umwandlung der Darstellung in kartesische Koordinaten
⇒ Vectorfield

RotEqNet - Layers

Spatial Pooling

- ▶ angepasste Version des Max-Pooling für Vectorfields

VectorBatchNorm

- ▶ angepasste Version von BatchNorm für Vectorfields
- ▶ BatchNorm wird lediglich auf die Magnitude angewendet

Netzarchitektur

Für alle RotConv Layer gelten folgende Parameter:

Padding: 9/2

R: 17

Layer	Parameter
RotConv	(1, 4, [9x9])
Orientation Pooling	
VectorBatchNorm	(4)
Spatial Pooling	Factor: 2
RotConv	(4, 8, [9x9])
Orientation Pooling	
VectorBatchNorm	(8)
Spatial Pooling	Factor: 2
RotConv	(8, 4, [9x9])
Orientation Pooling	
VectorBatchNorm	(4)
VectorUpsample	Factor: 2

RotConv	(4, 2, [9x9])
Orientation Pooling	
VectorBatchNorm	(2)
VectorUpsample	Factor: 2
RotConv	(2, 1, [9x9])
Orientation Pooling	
RotConv	(1, 1, [9x9])
Orientation Pooling	
Vector2Magnitude	

Entwicklung

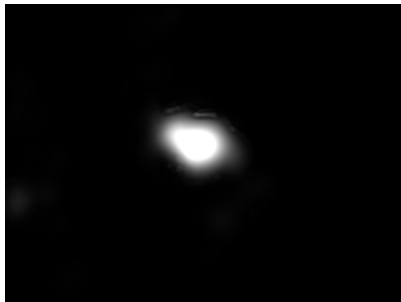
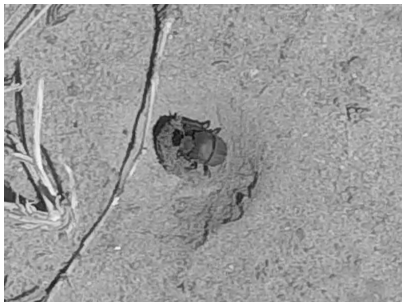
Projekt geforked von Anders Waldeland

- ▶ Fix für Python 3 Kompatibilität
- ▶ Parser für .grndr Dateien
- ▶ Modularisierung des Codes und CleanUp
- ▶ Fix der mathematischen Berechnungen (nach Paper)
- ▶ Visualisierung des Vectorfields eines Bildes
- ▶ Training und Netzoptimierung anhand künstlichen Datensatzes

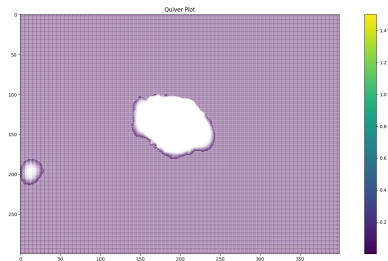
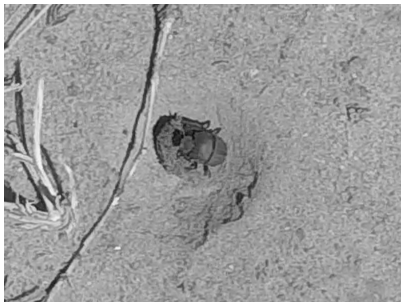
Entwicklung

- ▶ Test verschiedener Netzarchitekturen
 - ▶ Object Detection (nur auf Magnitude)
 - ▶ Klassifikation nach Winkeln
 - ▶ komplettes Vectorfield (Magnitude und Winkel)
- ▶ Test verschiedener Lossfunktionen
 - ▶ BinaryCrossEntropy
 - ▶ F1 Loss
 - ▶ L1
 - ▶ Smooth L1
 - ▶ CategoricalCrossEntropy

Beispiel - Detection



Beispiel - Vectorfield



Beispiel - Vectorfield

