

Freie Universität Berlin  
Fachbereich für Mathematik und Informatik  
Institut für Informatik  
Biorobotics Lab

# **Erkennung von Schwänzeltänzen mit tiefen neuronalen Netzen**

Bachelorarbeit

Mehmed Halilovic  
Matrikelnummer: 4663111

Gutachter: Dr. Tim Landgraf

Zweitgutachter: Prof. Raul Rojas

März 2015

# Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben. Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Datum

Unterschrift

## **Abstract**

Honigbienen benutzen den Schwänzeltanz, um anderen Bienen mitzuteilen wo Futterstellen sind. Diesen Tanz automatisiert zu erkennen ist eine schwere Aufgabe für die es bereits einige Systeme gibt. Eines dieser Systeme, der Waggle Dance Detector, bietet eine automatisierte Echtzeiterkennung von Schwänzeltänzen. In dieser Arbeit wird die Erkennungsrate des Waggle Dance Detectors mit Hilfe eines Convolutional Neuronal Networks verbessert. Dazu werden die Ergebnisse des Waggle Dance Detectors im Zuge eines Nachbearbeitung-Schrittes nochmals klassifiziert, um falsch erkannte Tänze heraus zu filtern.

# Inhaltsverzeichnis

<b>Abstract</b>	<b>i</b>
<b>1 Einleitung und Motivation</b>	<b>1</b>
<b>2 Stand der Forschung</b>	<b>2</b>
2.1 Schwänzeltanz . . . . .	2
2.2 Systeme zur Erkennung vom Schwänzeltanz . . . . .	3
2.2.1 Tracking von Honigbienen - Kimura . . . . .	3
2.2.2 Erkennung des Verhaltens der Bienen - Balch . . . . .	3
2.2.3 Waggle Dance Detector - Rau . . . . .	4
2.3 Zusammenfassung . . . . .	5
<b>3 Implementierung</b>	<b>6</b>
3.1 Anforderungen . . . . .	6
3.2 Struktur der Lösung . . . . .	7
3.3 Ground-Truth-Daten erstellen . . . . .	7
3.4 Modell des neuronalen Netzwerks . . . . .	9

3.4.1	Neuronale Netzwerke . . . . .	9
3.4.2	Modell zur Erkennung von Schwänzeltänzen . . . . .	10
3.5	Training des Modells . . . . .	12
3.6	Tanzerkennung . . . . .	12
<b>4</b>	<b>Evaluation</b>	<b>14</b>
4.1	Training ohne Gewichtung 2015er-Daten . . . . .	15
4.2	Training mit Gewichtung 2015er-Daten . . . . .	16
4.3	Training mit Gewichtung 2014er-Daten . . . . .	17
4.4	Lokalisierung von Futterstellen . . . . .	19
4.5	Zusammenfassung . . . . .	21
<b>5</b>	<b>Diskussion und Ausblick</b>	<b>22</b>
5.1	Auswertung der Evaluation . . . . .	22
5.2	Diskussion . . . . .	24
5.3	Ausblick . . . . .	25
	<b>Bibliography</b>	<b>25</b>

# Kapitel 1

## Einleitung und Motivation

Honigbienen sind aufgrund ihrer Bedeutung für die Umwelt und ihrer sozialen Strukturen innerhalb einer Kolonie oft zu einem Forschungsthema geworden [9] [14] [18]. Eines der Verhalten von Interesse ist der Schwänzeltanz der Bienen. Mit diesem können sie anderen Bienen den Ort von Futterstellen durch einen geregelten Bewegungsablauf mitteilen [16]. Um diesen automatisiert zu erkennen wurden bereits verschiedene Systeme entwickelt [2] [3] [8]. Allerdings sind diese Systeme hinsichtlich ihrer Genauigkeit noch nicht voll ausgereift.

Trotz aktueller Erfolge mit künstlichen neuronalen Netzwerken [11], gibt es noch keine Systeme die diese für die Erkennung von Bientänzen einsetzen.

Das Ziel dieser Arbeit ist es eine Lösung mit Hilfe von neuronalen Netzen zu implementieren. Dafür wird der Waggle Dance Detector von Rau [8] als Grundlage verwendet, um dann mit Hilfe von neuronalen Netzen falsch klassifizierte Tänze heraus zu filtern und die Ergebnisse zu verbessern.

# Kapitel 2

## Stand der Forschung

In diesem Abschnitt werden bestehende Lösungsansätze für die Erkennung von Schwänzeltänzen vorgestellt. Dafür wird zunächst der Ablauf des Schwänzeltanzes erklärt. Anschließend werden drei verschiedene Arbeiten vorgestellt. Es wird genauer auf die Arbeit von Rau eingegangen, da diese als Grundlage für diese Arbeit dient. Besonderer Wert wird dabei auf die Erkennungsraten von Schwänzeltänzen gelegt.

### 2.1 Schwänzeltanz

Wie schon in der Einleitung beschrieben teilen Bienen mit Hilfe des Schwänzeltanzes anderen Bienen den Ort einer Futterstelle mit. Dabei läuft die Biene einen Pfad der wie eine 8 aussieht, so wie in Abbildung 2.1 angedeutet. Sie fängt mit einer gerade gerichteten Bewegung an. Währenddessen macht sie eine schwingende Bewegung mit ihrem Unterleib. Nach der geraden Bewegung läuft sie abwechselnd nach links oder rechts in der Form eines Halbkreises wieder zum Anfang und wiederholt diese Bewegung einige Male. Während dieses Vorgangs wird sie von anderen Bienen beobachtet, die mit dieser Information die Futterstelle finden können. [16]



Abbildung 2.1: Der Pfad einer Biene während des Schwänzeltanzes.  
Die weißen Pfeile deuten den Pfad der Biene an. Quelle des Bildes: Chittka [1]

## 2.2 Systeme zur Erkennung vom Schwänzeltanz

Alle im folgenden vorgestellten Systeme zur Erkennung vom Schwänzeltanz basieren darauf, dass Bienen mit einer Kamera in ihrem Bienenstock aufgenommen werden und das dieses Videomaterial dann analysiert wird.

### 2.2.1 Tracking von Honigbienen - Kimura

Ein System welches entwickelt wurde um mehrere Bienen gleichzeitig zu verfolgen, wurde von Kimura et al. [3] vorgestellt. Hauptaugenmerk wurde dabei auf die Erkennung von Bienen gelegt. Benutzt wurde eine Vektorquantisierungs-Methode mit der man es schafft 75% der Bienen in einer Kolonie zu erkennen. Zudem wird beschrieben, dass mit Hilfe dieses Systems der Schwänzeltanz erkannt werden kann. Wie genau dies funktioniert ist aus der vorliegenden Arbeit nicht ersichtlich. Auch ist nicht klar ob die Erkennung voll automatisiert und in Echtzeit möglich ist.

### 2.2.2 Erkennung des Verhaltens der Bienen - Balch

Ein System zur Erkennung der Verhalten von Bienen wurde von Balch et al. [2] vorgestellt. Dort wurde versucht das Verhalten von Bienen mit Hilfe einer Kombination aus Kernel-Regression

und einem Hidden Markov Model zu lösen. Dafür wird eine Tracking-Software genutzt mit der Features von Bienen extrahiert werden können. Mit diesen Features werden, mit Hilfe eines dafür trainierten Regression-Klassifikators, Bewegungen klassifiziert. Anschließend wird mit einem Hidden-Markov-Modells aus den Bewegungsabläufen auf das Verhalten der Biene geschlossen. Die Genauigkeit der Verhaltenserkennung von dieser Methode wird mit 79% angegeben. Aus der Erkennungsrate für das Verhalten kann man nicht direkt auf die Erkennungsrate für den Schwänzeltanz schließen.

### 2.2.3 Waggle Dance Detector - Rau

Sowohl bei der Arbeit von Kimura, als auch bei Balch ist nicht klar, ob die Erkennung voll automatisiert und in Echtzeit stattfinden kann.

Rau [8] hat sich bei seinem System das Ziel gesetzt voll automatisiert in Echtzeit Bientänze zu erkennen und gleichzeitig die Richtung des Tanzes zu bestimmen. Die Grundlage für das System bietet ein Dot-Detector. Dieser reagiert für jeden Pixel auf Änderungen, die bestimmte Frequenzen haben, mit einem Signal. Diese Signale werden für einige Zeit zwischengespeichert. Anschließend werden die Signale zu Clustern gruppiert. Dann werden Tänze gefunden, indem man die Cluster zeitlich verbindet. Falls es sich um einen Tanz handelt berechnet Rau zusätzlich die Richtung des Tanzes. Die Methode zur Bestimmung der Richtung wurde von Mielke [6] in ihrer Arbeit nochmals verbessert. Für jeden erkannten Tanz werden die Bildausschnitte indem sich der Tanz befindet gespeichert.

Rau testet die Methode mit drei verschiedenen Datenmengen, wobei die sich in ihrer Größe und Auflösung unterscheiden. Die sogenannten BeeHouse und BeesBook Datenmengen zeichnen sich dadurch aus, das sie beide mit einer niedrigen Auflösung von 0.42 bis 0.43 dpmm und 102HZ aufgenommen werden. Sie unterscheiden sich vor allem in ihrer Belichtung. Außerdem gibt es in den BeeHouse Aufnahmen viele Motten die die Detektion stören. Die dritte RoboBee Datenmenge besitzt eine deutlich größere Auflösung als die anderen beiden.

Die Wiedererkennungsrates von Schwänzeltänzen wird bei den RoboBee Daten mit 96,4% an-

gegeben, wobei mit den falsch klassifizierten Tänzen eine Erkennungsrate von 89,5% erreicht wird.

Das Verhältnis von positiven, d.h. richtig klassifizierten, und negativen, d.h. falsch klassifizierten, Tänzen ist bei Beehouse mit 78,6% zu 21,4% angegeben. Bei BeesBook wird ein Verhältnis von 67,4% zu 32,6% angegeben. Über die tatsächliche Wiedererkennungsrate von Schwänzeltänzen wird keine Aussage getroffen, da die Detektion live auf den Kamera Daten durchgeführt wurde und nur die Ergebnisse ausgewertet wurden.

## 2.3 Zusammenfassung

In diesem Abschnitt wurden drei Arbeiten vorgestellt, die sich mit der Erkennung von Schwänzeltänzen auseinandersetzen. Es wurden die Lösungsansätze vorgestellt die dazu dienen Schwänzeltänze automatisiert zu finden. Allerdings ging nur aus der Arbeit von Rau hervor, das die Lösung in der Lage ist Schwänzeltänze in Echtzeit zu erkennen. Das größte Problem, welches sich alle drei Arbeiten teilen, ist das die Erkennungsrate bei keiner der Lösungen nahe 100% gekommen ist. Das bedeutet das es noch Verbesserungspotenzial für solche Systeme gibt.

# Kapitel 3

## Implementierung

Keine der Lösungen für die automatisierte Detektierung von Schwänzeltänzen war hinsichtlich der Erkennungsrate zufriedenstellend. Der Waggle Dance Detector von Rau zeichnete sich durch seine Echtzeiterkennung und gleichzeitige Bestimmung der Tanzrichtung aus. Dies sorgt dafür das man nicht die gesamten Filmaufnahmen speichern muss, sondern nur die Erkennungen des Waggle Dance Detectors. Ein Problem des Waggle Dance Detectors ist die hohe Rate an falsch erkannten Schwänzeltänzen. In dieser Arbeit wird versucht dieses Problem mit neuronalen Netzen zu lösen, indem die erkannten Schwänzeltänze nochmals gefiltert werden.

### 3.1 Anforderungen

Ziel dieser Arbeit ist es die Ergebnisse von Rau's Waggle Dance Detector nochmals zu klassifizieren. Das System soll möglichst alle falsch erkannten Tänze herausfiltern. Wichtiger jedoch ist dabei die Erhaltung aller richtig erkannten Tänze. Die Erkennung muss außerdem dazu in der Lage sein die WDD-Ordnerstruktur durchzugehen, um jeden Tanz einzeln zu klassifizieren. Anforderungen an die Laufzeit werden hier nicht gestellt, da die Klassifizierung mit den neuronalen Netzen als Nachbearbeitungsschritt durchgeführt werden kann.

## 3.2 Struktur der Lösung

Um zu entscheiden, ob es sich bei einer Erkennung um einen Tanz handelt, wird ein neuronales Netz modelliert, welches mit Hilfe der Aufnahme eines Tanzes sagen kann mit welcher Wahrscheinlichkeit es sich um einen Tanz handelt.

Da unterschiedlich viele Frames in einer Aufnahme sein können wird die Aufnahme in gleich große Fenster von 30 Frames geteilt. Das Netz wird dazu genutzt zu entscheiden, ob ein Fenster von 30 Frames zu einem Tanz gehört oder nicht zu einem Tanz gehört.

Anschließend wird mit Hilfe der Ergebnisse aller Fenster eines Tanzes entschieden, ob der Tanz tatsächlich ein Tanz ist oder nicht.

Für das Training des neuronalen Netzes werden Trainingsdaten benötigt. Für die Erstellung von Ground-Truth-Daten werden die Ergebnisse des Waggle Dance Detectors aus den Jahren 2014 und 2015 genommen. Gleichzeitig werden auch Daten zur Validierung erstellt um die neuronalen Netze später testen zu können.

## 3.3 Ground-Truth-Daten erstellen

In diesem Abschnitt wird erklärt wie die Ground-Truth-Daten für das Training und die Validierung erstellt werden.

Die Ergebnisse des Waggle Dance Detector aus dem Jahr 2014 und 2015 liegen in einer speziellen Ordnerstruktur, im folgenden WDD-Ordnerstruktur genannt, vor. Die Ordnerstruktur ist wie folgt aufgebaut:

```
--<Datum> (Beispiel: "20150822")
```

```
---- <Datum>-<Uhrzeit>-<Kamera> (Beispiel: "20150822-1134-0")
```

```
----- <Nummer> (Zu jeder Zeit kann es mehrere Detektionen geben. Diese werden  
aufsteigend durchnummeriert. Beispiel: "1")
```

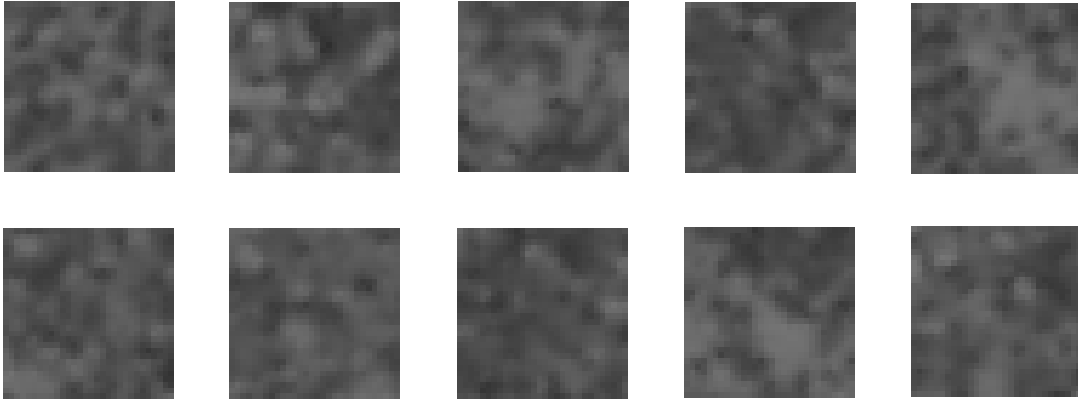


Abbildung 3.1: Zehn zufällige Frames aus den Tänzen aus den 2015-Daten. Aufgrund der niedrigen Auflösung ist es schwer Bienen auf diesen Bildern zu erkennen. Diese werden erst durch die Bewegung gut sichtbar, wenn die Bilder eines Tanzes als Folge betrachtet werden.

In den durchnummerierten Ordnern gibt es Metadaten die in einer CSV-Datei gespeichert sind. Darin stehen Daten wie die genaue Zeit, die Länge, die Position und die Orientierung des Tanzes. Außerdem sind alle zu dem Tanz gehörigen Frames durchnummeriert als Bild im PNG-Format im Ordner gespeichert. Dabei sind die Bilder in den 2014er-Daten 20x20 Pixel groß und die Bilder in den 2015er-Daten 30x30 Pixel groß.

Selbst für einen Menschen kann es schwer sein einen Schwänzeltanz zu erkennen. Dies liegt vor allem an der niedrigen Auflösung die für die Aufnahmen verwendet werden, wie man in Abbildung 3.1 sehen kann. Für die Entscheidung ob es sich um einen Tanz handelt ist die schwingende Bewegung, während der gerade gerichteten Bewegung, der Biene ausschlaggebend.

Für die Erstellung der Ground-Truth-Daten wurde Matlab benutzt. Aufbauend auf der Lösung, die Mielke in Ihrer Arbeit [6] für die Erstellung der Ground-Truth-Daten benutzt hat, wurde ein Matlab Programm geschrieben mit dem man als Benutzer entscheiden kann, ob es sich bei einer Erkennung um einen Tanz handelt oder nicht. Das Programm ermöglicht es die WDD-Ordnerstruktur durchzugehen und jeden Tanz einzeln zu betrachten. Dabei kann man die Frames eines Tanzes durchgehen und den Tanz mit Hilfe eines Tastendrucks als positiv oder negativ klassifizieren. Das Ergebnis wird in einer CSV-Datei im Ordner, der zum Tanz gehört, gespeichert.

Aus den 2014-Daten wurden ca. 2350 Tänze betrachtet, von denen ca. 990 positive Tänze waren.

Aus den 2015-Daten wurden ca. 3000 Tänze betrachtet, von denen ca. 850 als positive Tänze gekennzeichnet wurden.

An dieser Verteilung erkennt man, dass es viele falsch erkannte Tänze gibt und dass eine Filterung einen großen Mehrwert bietet.

Um das neuronale Netz trainieren zu können werden die Ground-Truth-Daten für das Training in Matrizen verpackt. Eine Matrix besteht dabei aus allen Samples und die zweite Matrix enthält den dazugehörige Ground-Truth-Wert. Die Samples bestehen aus jeweils 30 aufeinanderfolgenden Frames eines Tanzes. Für die Erstellung wurde ein MATLAB-Programm geschrieben welches eine WDD Ordnerstruktur durchgeht und jeden gefunden Tanz zu dem Ground-Truth-Daten existieren in die Matrizen einfügt. Dabei werden immer alle möglichen Fenster von 30 aufeinanderfolgenden Frames eines Tanzes in die Matrix eingefügt, so dass ein Tanz mehrere Samples liefert. Auf diese Art und Weise bekommt man aus 2250 Tänzen bereits ca. 90.000 Samples, jedes Sample ist dabei 30 Frames lang, um das Netz zu trainieren. Dabei werden die Trainings-Daten nochmals in eine Trainings- und Validierungsmenge unterteilt, um den Fortschritt des Trainings beurteilen zu können. Es wird darauf geachtet dass alle Samples aus einem Tanz entweder in der Trainingsmenge oder in der Validierungsmenge landen. Ansonsten könnte es dazu führen, dass die Validierung verfälscht wird, da sich Samples aus einem Tanz sehr ähnlich sehen können. Außerdem wird die Reihenfolge der Samples in der Trainings-Matrix am Ende zufällig gemischt.

## 3.4 Modell des neuronalen Netzwerks

### 3.4.1 Neuronale Netzwerke

Künstliche neuronale Netze bestehen aus vielen einfachen Neuronen, die miteinander verbunden sind. Jedes Neuron ist eine einfache Funktion die zu einer Eingabe eine Ausgabe liefert. Die Input-Neuronen erhalten als Eingabe die Werte aus der realen Welt. Andere Neuronen bekommen als Eingabe eine gewichtete Ausgabe anderer Neuronen. In der Regel sind Neuronen in

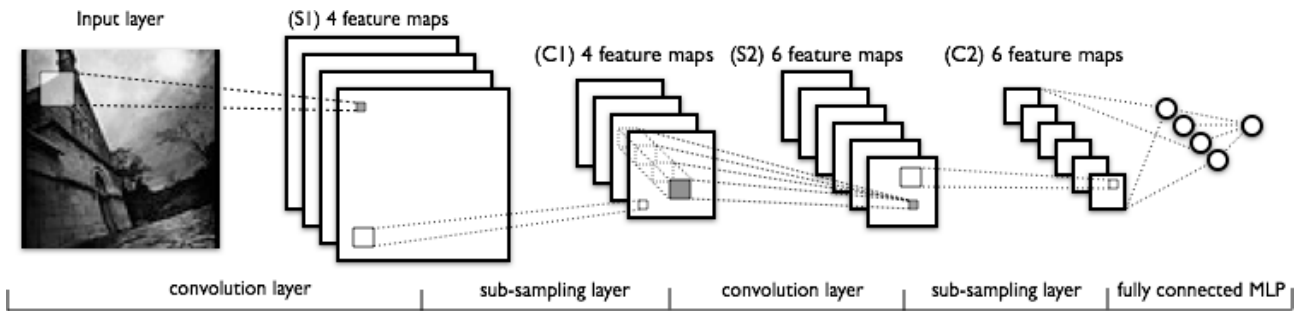


Abbildung 3.2: Struktur eines Convolutional Neural Networks.  
Quelle des Bildes: [19]

Schichten miteinander verbunden. Bei tiefen neuronalen Netzen gibt es viele solcher Schichten. Beim Lernen eines Netzes versucht man die Gewichte so zu bestimmen, dass sie für ein gegebenes Problem das richtige Ergebnis liefern. [13]

Für das Erkennen von Schwänzeltänzen wurden in dieser Arbeit Convolutional Neural Networks benutzt. Dies sind neuronale Netzwerke bei denen die Neuronen so miteinander verbunden sind, dass ein Neuron nur die Ausgaben von Neuronen erhält, die in den Ursprungsdaten nah beieinander liegen. Bei einem Bild heißt das, dass ein Neuron aus der ersten Schicht als Eingabe einen kleinen Bereich von nah beieinander liegenden Pixeln als Eingabe erhält. Dies wird auch Filter oder Kernel genannt. Meistens trainiert man mehrere solcher Filter um bessere Ergebnisse zu erhalten. Diese erste Schicht wird als eine Convolutional Schicht (englisch Convolutional Layer) bezeichnet.

Eine typische Struktur für ein Convolutional Neural Network sieht man in Abbildung 3.2. Wie man sieht kann es mehr als nur einen Convolutional Layer geben. Am Ende folgt eine Reihe von voll verbundenen Schichten.

Für weitere Informationen über künstliche neuronale Netzwerke, die über diese Erklärung hinaus gehen, wird auf [5] [7] [10] [12] verwiesen.

### 3.4.2 Modell zur Erkennung von Schwänzeltänzen

Das neuronale Netz zur Erkennung des Schwänzeltanzes soll in der Lage sein zu sagen, ob eine Folge von 30 Frames zu einem Tanz gehört.

Es wurde sich entschieden das neuronale Netz in Python zu schreiben und Keras zu benutzen. Keras ist eine Deep Learning Bibliothek mit der man schnell neuronale Netze modellieren kann. [4]

Die Entscheidung Convolutional Neural Networks zu benutzen wurde getroffen, da sie einfach zu implementieren waren. Außerdem kann man den zeitlichen Verlauf mit Hilfe der Filter trainieren.

Das Modell des neuronalen Netzwerks für die Schwänzeltanz-Erkennung besteht aus drei 2D-Convolutional Schichten, gefolgt von mehreren voll verbundenen Schichten. Der Aufbau ist ähnlich zu dem in Abbildung 3.2. Die drei 2D-Convolutional Schichten benutzen in folgender Reihenfolge jeweils 16, 8 und 8 Filter der Größe 5x5, 3x3, 5x5. Nach den Convolutional Schichten gibt es eine MaxPooling-Schicht der Größe 2x2. Anschließend folgen acht voll verbundene Schichten mit jeweils 2048, 128, 32, 16, 7 und 2 Neuronen. Alle Neuronen im Netz bis auf die Neuronen in der letzten Schicht benutzen die ReLu-Funktion. Die Neuronen in der letzten Schicht benutzen die Softmax-Funktion.

Um Überanpassung entgegenzuwirken wurde an einigen Stellen Dropout eingefügt. Dieser sorgt dafür, dass während des Trainings immer nur ein zufällig ausgewählter Teil der Neuronen benutzt wird [15]. Dieser ist nach den drei Convolutional Schichten mit einem Wert von 0.25 eingefügt, d.h. 25% alle Neuronen werden nicht benutzt. Dann ist Dropout nochmal nach der ersten voll verbundenen Schicht mit 0.5 und nach der zweiten voll verbundenen Schicht mit 0.25 eingefügt.

Die 30 Frames werden dem Netz in einer Matrix übergeben, die so aufgebaut ist wie ein Bild mit 30-dimensionalem Farbraum, daher (#Frames, Höhe, Breite). Die Convolutional Schicht trainiert so mehrere Filter auf allen Frames. Ein Filter verbindet auf diese Art und Weise alle Frames miteinander, so das eine zeitliche Verbindung trainiert werden kann. Anschließend werden die Ergebnisse der Filter mit Hilfe der voll verbundenen Schichten verknüpft.

## 3.5 Training des Modells

Beim anfänglichen Training der Netze, bei denen teilweise noch kein Dropout benutzt wurde, kam es merkbar zu Überanpassung (englisch Overfitting). Das heißt das neuronale Netz ist auf den Trainingsdaten gut, aber es generalisiert nicht auf neue unbekannte Daten. Dies fällt dadurch auf, dass die Ergebnisse auf den Validierungsdaten wesentlich schlechter sind als die auf den Trainingsdaten.

Deshalb werden die Daten künstlich vervielfacht. Dabei werden sie während des Trainings in zufälliger Kombination gespiegelt, um bis zu 180 Grad rotiert oder um einige Pixel verschoben. Es gibt weit über 100 verschiedene Möglichkeiten ein Sample auf diese Art zu verändern, allerdings wurde jedes Sample während des Trainings nur ca. 50 mal benutzt. Durch die künstliche Vervielfachung und das Einfügen von Dropout trat keine merkbare Überanpassung mehr auf.

Da positive Tänze wichtiger sind als negative Tänze sind, wurden die Netze auch mit einer Klassengewichtung zu Gunsten von positiven Tänzen trainiert. Das bedeutet das falsch klassifizierte positive Tänze mit einer höheren Gewichtung in die Kostenfunktion eingehen als falsch klassifizierte negative Tänze.

Am Ende des Trainings wurden die Gewichte des Netzwerks gespeichert und konnten für die Tanzerkennung eingesetzt werden.

Aufgrund der unterschiedlichen Größe der Aufnahmen in den 2014er-Daten wurde ein separates Netz für die 2014er Daten trainiert. Der Aufbau des Netzes ist gleich, nur die Convolutional Layer waren kleiner, da die Aufnahmen nur 20x20 Pixel groß waren. Außerdem wurden weniger Tänze für das Training verwendet.

## 3.6 Tanzerkennung

Es wurde ein Python-Programm geschrieben, welches das neuronale Netzwerk mit den trainierten Gewichten benutzt. Dieses Programm kann die WDD-Ordnerstruktur durchgehen und alle

Ausschnitte, in denen sich eventuelle Tänze befinden, mit dem neuronalen Netz klassifizieren.

Da das Netzwerk nur mit Eingaben von 30 Frames arbeitet, aber eine Tanz mehr als 100 Frames lang sein kann, wird das Netzwerk wie ein Sliding Window über den gesamten Tanz angewendet.

Um zu entscheiden, ob es sich um einen Tanz handelt oder nicht, wird der Mittelwert über alle Ergebnisse eines Ausschnittes berechnet. Ist der Mittelwert über einem festgelegten Schwellwert, so wird entschieden das es sich um einen Tanz handelt. Anhand einer ROC-Kurve kann man erkennen wie dieser Schwellwert gewählt werden sollte. Da es wichtiger ist positive Tänzen nicht herauszufiltern, kann dieser Schwellwert zu Gunsten der positiven Tänze angepasst werden. Die Erstellung der ROC-Kurve wird in der Evaluation erklärt.

# Kapitel 4

## Evaluation

In diesem Abschnitt werden die Ergebnisse der Tanzerkennung vorgestellt. Dazu werden einmal die Ergebnisse des neuronalen Netzwerkes in Form einer Confusion Matrix und die Ergebnisse der Tanzerkennung mit Hilfe einer ROC-Kurve vorgestellt.

Außerdem wird das Netzwerk für die 2014er-Daten anschließend genutzt um die Daten eines gesamten Tages zu filtern. Mit den Daten des Tages wird dann aus den Tänzen auf die Orte der Futterstellen geschlossen. Anschließend wird das gleiche mit den gefilterten Daten getan, um dann die Ergebnisse zu vergleichen.

Für die Evaluation wurden die 2015er-Daten, bestehend aus ca. 3000 Tänzen und die 2014er-Daten bestehend aus ca. 2350 Tänzen benutzt. Nach der im Kapitel 3 beschriebenen Methode werden die Daten in Trainingsmenge und Validierungsmenge geteilt. Bei den 2015er-Daten wurde 25% der Daten zur Validierung benutzt. Bei den 2014er-Daten wurden nur ca. 20% für die Validierung benutzt.

Das neuronale Netzwerk wird zunächst mit den Trainingsdaten trainiert. Anschließend wird es mit den Validierungsdaten getestet, indem die Confusion-Matrix und die ROC-Kurve erstellt werden.

Es wird ein Training ohne Gewichtung und ein Training mit Gewichtung für die 2015er-Daten durchgeführt. Mit den 2014er-Daten wird nur ein Training mit Gewichtung durchgeführt.

		Realität	
		Tanz	Kein Tanz
Vorhergesagte Klasse	Tanz	13256 (36%)	4127 (11,2%)
	Kein Tanz	324 (0,9%)	19124 (51,9%)

TP-Rate 0.9761    FP-Rate 0.1775

Tabelle 4.1: Confusion-Matrix vom neuronalen Netzwerk ohne Gewichtung (2015-Daten)  
Die Confusion-Matrix wurde mit Hilfe der Tänze aus den Validierungsdaten erstellt. Dabei wurden alle möglichen Fenster von 30-Frames eines Tanzes benutzt und getestet wie gut das neuronale Netzwerk ein solches Fenster als Ausschnitt eines Tanzes erkennt.

## 4.1 Training ohne Gewichtung 2015er-Daten

Beim Training ohne Klassengewichtung wurde das neuronale Netzwerk, wie in Kapitel 3 beschrieben, trainiert. Die Confusion-Matrix wurde mit Hilfe aller möglichen 30-Frame-Fenster aus den Validierungsdaten erstellt. Das Ergebnis kann man in Tabelle 4.1 sehen.

In der Tabelle kann man ablesen für wie viele Fenster von 30 Frames richtig klassifiziert wurde, ob das Fenster zu einem Tanz gehört oder nicht. Außerdem kann man sehen wie gut die Erkennung für die einzelnen Klassen ist. Aus der Tabelle kann man die TP-Rate und die FP-Rate ablesen. Die TP-Rate und FP-Rate berechnet sich wie in den Gleichungen 4.1 und 4.2. Ein hoher Wert für die TP-Rate ist gut, da das heißt das viele positive Tänze richtig klassifiziert wurden. Bei der FP-Rate ist ein niedriger Wert gut, da das heißt das viele negative Tänze richtig klassifiziert wurden.

$$TP - Rate = \frac{\text{Richtig klassifizierte positive Taenze}}{\text{Gesamtanzahl der positiven Taenze}} \quad (4.1)$$

$$FP - Rate = \frac{\text{Falsch klassifizierte negative Taenze}}{\text{Gesamtanzahl der negativen Taenze}} \quad (4.2)$$

Mit Hilfe der Tabelle kann man die Erkennungsrate wie in Gleichung 4.3 berechnen. Damit ergibt sich aus der Tabelle eine Erkennungsrate von 87.91%. Die TP-Rate liegt bei 97.61% und die FP-Rate bei 17,75%.

$$\text{Erkennungsrate} = \frac{\text{Richtig klassifizierte Taenze}}{\text{Gesamtanzahl der Taenze}} \quad (4.3)$$

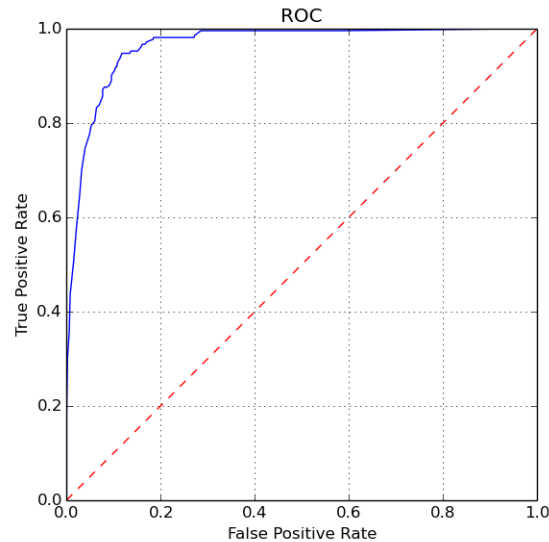


Abbildung 4.1: ROC-Kurve für das Training ohne Gewichtung der 2015-Daten  
Die ROC-Kurve wurde mit Hilfe des trainierten Netzwerkes und den Validierungsdaten erstellt.

Für die Tanzerkennung wurden die Ergebnisse für alle Fenster eines Tanzes gemittelt. Anschließend wurden 100 Schwellwerte von 0 bis 1 im Abstand von 0.01 getestet. Wenn die Wahrscheinlichkeit dafür, dass es ein Tanz war, über dem Schwellwert war, wurde er als positiver Tanz klassifiziert. Wenn man die TP-Rate und die FP-Rate für alle Schwellwerte graphisch darstellt, ergibt sich die ROC-Kurve aus Abbildung 4.1. Mit Hilfe der ROC-Kurve kann man erkennen, wie das Verhältnis von TP-Rate und FP-Rate sich für verschiedene Schwellwerte verhält. Optimal wäre es, wenn man mit einem der Schwellwerte den optimalen Punkt, ganz oben links in der Ecke, erreichen könnte. Als guter Kompromiss zwischen der Erhaltung von positiven Tänzen und Filterung von negativen Tänzen bieten sich Werte aus der oberen linken Region an.

## 4.2 Training mit Gewichtung 2015er-Daten

Beim Training mit Gewichtung wurde eine Gewichtung von 1 zu 5 zu Gunsten der positiven Tänze benutzt. Die Confusion-Matrix wurde wie zuvor bestimmt. In Tabelle 4.2 kann man sehen, dass sich die Werte etwas zu Gunsten der positiven Tänze verschoben haben. Die TP-Rate ist um fast 2% besser geworden, allerdings wurde die FP-Rate um 9% schlechter.

		Realität	
		Tanz	Kein Tanz
Vorhergesagte Klasse	Tanz	13478 (36,6%)	6227 (16,9%)
	Kein Tanz	102 (0,3%)	17024 (46,2%)

TP-Rate 0.9925    FP-Rate 0.2678

Tabelle 4.2: Confusion-Matrix vom neuronalen Netzwerk mit Gewichtung (2015-Daten)  
 Die Confusion-Matrix wurde mit Hilfe der Tänze aus den Validierungsdaten erstellt. Dabei wurden alle möglichen Fenster von 30-Frames eines Tanzes benutzt und getestet wie gut das neuronale Netzwerk ein solches Fenster als Ausschnitt eines Tanzes erkennt.

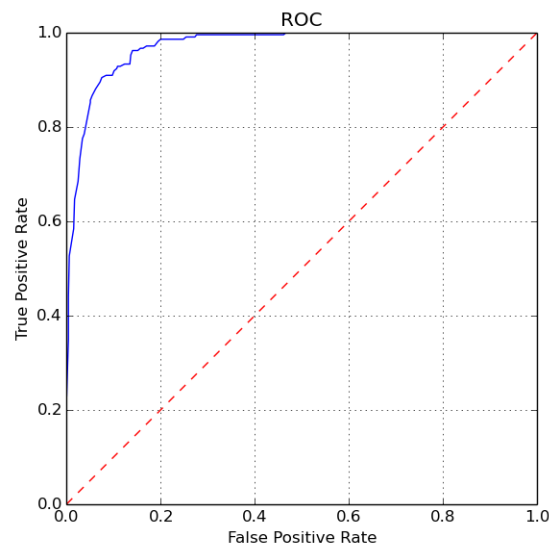


Abbildung 4.2: ROC-Kurve für das Training mit Gewichtung der 2015-Daten  
 Die ROC-Kurve wurde mit Hilfe des trainierten Netzwerkes und den Validierungsdaten erstellt.

Auch die ROC-Kurve wurde wie zuvor erstellt. Diese kann man in Abbildung 4.2 sehen. Wie erwartet hat sich die Kurve an einigen Stellen zu Gunsten der TP-Rate leicht nach oben verschoben.

### 4.3 Training mit Gewichtung 2014er-Daten

Beim Training mit den 2014-er Daten wurde das Netzwerk wieder mit einer Gewichtung von 1 zu 5 zu Gunsten der positiven Tänze trainiert. Im Gegensatz zu den 2015-er Daten standen hier weniger Tänze für das Training und die Validierung zur Verfügung.

		Realität	
		Tanz	Kein Tanz
Vorhergesagte Klasse	Tanz	6986 (39,8%)	2203 (12,5%)
	Kein Tanz	437 (2,5%)	7930 (45,2%)

TP-Rate 0.9411    FP-Rate 0.2174

Tabelle 4.3: Confusion-Matrix vom neuronalen Netzwerk ohne Gewichtung (2014-Daten)  
 Die Confusion-Matrix wurde mit Hilfe der Tänze aus den Validierungsdaten erstellt. Dabei wurden alle möglichen Fenster von 30-Frames eines Tanzes benutzt und getestet wie gut das neuronale Netzwerk ein solches Fenster als Ausschnitt eines Tanzes erkennt.

Für die 2014er-Daten ergibt sich die Confusion-Matrix in Tabelle 4.3. Man sieht das die Werte schlechter sind als bei den 2015er-Daten.

Die ROC-Kurve sieht man in Abbildung 4.3. Auch hier erkennt man das die ROC-Kurve schlechter ist als bei den 2015er-Daten. Dies zeigt sich zum Beispiel dann, wenn man über 99% der richtigen Tänze erhalten will. Mit dem 2014er-Netzwerk kann man dann ca. 45% der falsch erkannten Tänze herausfiltern. Bei den 2015er-Daten konnte man noch über 70% herausfiltern.

Um zu untersuchen welche Art von Erkennungen, bei denen es sich nicht um einen Tanz handelt, als richtige Tänze klassifiziert werden, wurden 55 solcher falsch klassifizierten Erkennungen betrachtet. Dabei traten vor allem Bewegungen von Motten oder Bienen auf, die nicht der Schwänzeltanz waren. Bei 20 dieser Erkennungen liefen oder flogen Motten durch das Bild, die für einen Menschen sehr leicht als solche zu erkennen sind. Beim Rest der Erkennungen handelte es sich fast nur um Bienen die nicht den Schwänzeltanz vorführten, sondern andere Bewegung machten. Einige dieser Bewegungen sahen dem Schwänzeltanz ähnlich. Allerdings handelte es sich nur bei einer dieser Erkennungen um den Schwänzeltanz.

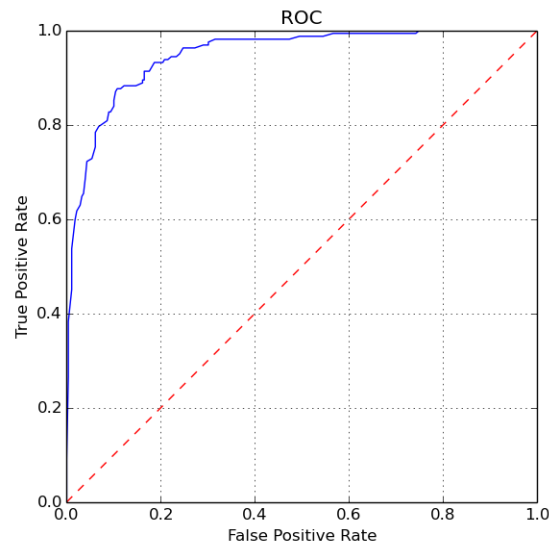


Abbildung 4.3: ROC-Kurve für das Training mit Gewichtung der 2014-Daten. Die ROC-Kurve wurde mit Hilfe des trainierten Netzwerkes und den Validierungsdaten erstellt.

## 4.4 Lokalisierung von Futterstellen

In dem Paper von Wario et al. [17] wurde mit Hilfe der Daten eines Tages aus den 2014er-Daten auf die Positionen von Futterstellen geschlossen. An diesem Tag wurde den Bienen eine künstliche Futterstelle gegeben. Dafür wurden die Ergebnisse des WDD benutzt, um mit der Orientierung und der Länge des Tanzes auf die Position in der Karte zu schließen. Mit von Tim Landgraf zur Verfügung gestelltem Code konnte ein sehr ähnliches Ergebnis für den Tag rekonstruiert werden, welches in Abbildung 4.4 zu sehen ist.

Die Daten für den Tag wurden anschließend mit dem 2014er-Netzwerk gefiltert. Aufgrund der Tatsache, dass das 2014er-Netzwerk etwas schlechter ist musste der Schwellwert so gewählt werden das auch einige Tänze verloren gehen um ein signifikant anderes Ergebnis zu erhalten. So wurde ein Schwellwert gewählt bei dem die TP-Rate bei ca. 90% und die FP-Rate bei ca. 21% liegt.

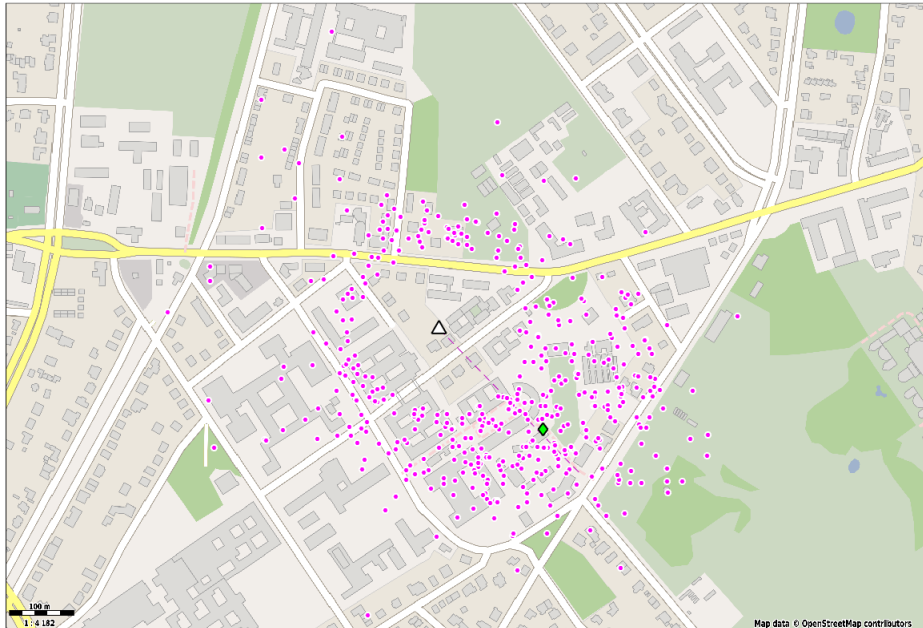


Abbildung 4.4: Karte mit den, aufgrund der Daten eines Tages, lokalisierten Futterstellen eingezeichnet. Beim Erstellen wurden die Ergebnisse des WDD benutzt. Die künstlich zur Verfügung gestellte Futterstelle ist als grüne Raute eingezeichnet.

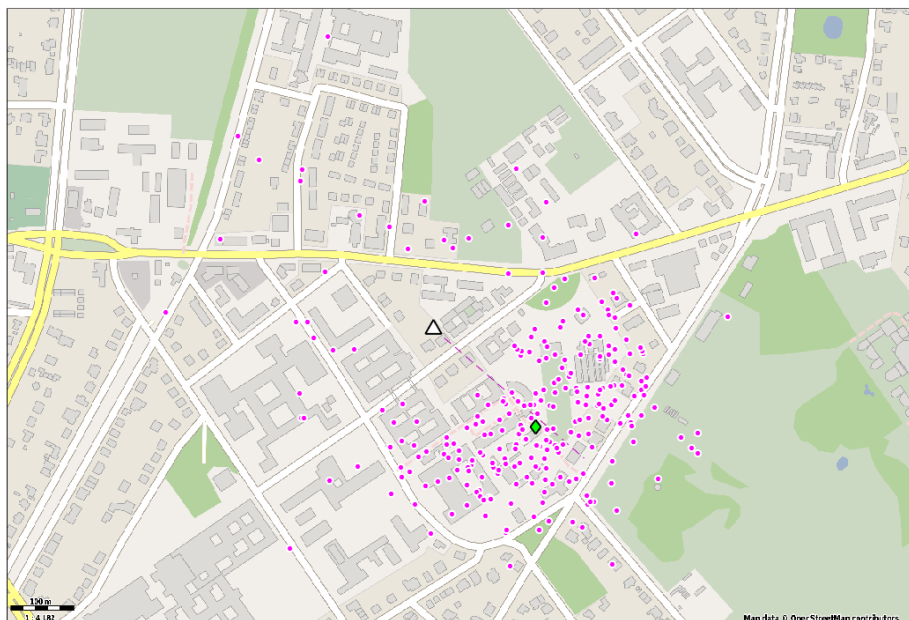


Abbildung 4.5: Karte mit den, aufgrund der gefilterten Daten eines Tages, lokalisierten Futterstellen eingezeichnet. Beim Erstellen wurden die gefilterten Ergebnisse des WDD benutzt. Die künstlich zur Verfügung gestellte Futterstelle ist als grüne Raute eingezeichnet.

Bei der Filterungen der Daten für diesen Tag sind von den 5713 Tänzen 2711 heraus gefiltert worden. In Abbildung 4.5 sind die mit den gefilterten Daten lokalisierten Futterstellen eingezeichnet. Wie man sieht ist ein großer Teil des Rings, der ursprünglich auf der Karte in Abbildung 4.4 zu sehen war, verschwunden.

## 4.5 Zusammenfassung

In diesem Abschnitt wurden die Ergebnisse von drei verschiedenen Trainingssetups vorgestellt.

Frühere Trainingssetups mit den 2014-Daten haben gezeigt das für das vorliegende neuronale Netzwerk eine Trainingsmenge mit 750 Tänzen zu wenig ist, da sich Überanpassung trotz Vervielfältigung bemerkbar macht. Das Training mit mehr Daten hat die Ergebnisse deutlich verbessert. Beim Training mit Gewichtung war im Vergleich zum Training ohne Gewichtung die TP-Rate etwas besser.

Bei der Lokalisierung der Futterstellen sind viele Lokalisierungen verschwunden. Übrig geblieben sind vor allem Lokalisierungen die in Richtung der künstlichen Futterstelle liegen, was auf ein gutes Ergebnis hindeutet.

# Kapitel 5

## Diskussion und Ausblick

In diesem Abschnitt werden zunächst die Ergebnisse der Evaluation ausgewertet, um zu entscheiden ob die in Kapitel 3 gesetzten Anforderungen erfüllt worden sind. Anschließend wird über die Vor- und Nachteile der in dieser Arbeit vorgestellten Lösung diskutiert und es wird ein Ausblick darüber gegeben wie die Schwänzeltanzerkennung weiter verbessert werden kann.

### 5.1 Auswertung der Evaluation

Das Ziel dieser Arbeit war es die Ergebnisse des Waggle Dance Detectors zu klassifizieren und möglichst viele negative Tänze, daher falsch erkannte Tänze, herauszufiltern, während möglichst alle positiven Tänze, daher richtig erkannte Tänze, erhalten bleiben sollten. Mit den neuronalen Netzwerken ist es möglich die Ausgabe des Waggle Dance Detectors zu klassifizieren.

Bei der Tanzerkennung ohne Klassengewichtung können mit entsprechendem Schwellwert mehr als ca. 96.50% der positiven Tänze erhalten bleiben. Zudem werden dann ca. 83% der negativen Tänze herausgefiltert. Durch Klassengewichtung konnte dieses Ergebnis für die Erhaltung von positiven Tänzen nochmal verbessert werden, so dass 98.08% der Tänze erhalten bleiben während 80.6% aller negativen Tänze herausgefiltert werden. Auch wenn dies noch ca. 20% von einem perfektes Ergebnis entfernt ist, werden die Ergebnisse des Waggle Dance Detectors

		Realität	
		Tanz	Kein Tanz
Vorhergesagte Klasse	Tanz	205 (27,5%)	104 (14%)
	Kein Tanz	4 (0,5%)	432 (58%)

TP-Rate 0.9808    FP-Rate 0.1940

Tabelle 5.1: Confusion-Matrix der Tanzerkennung mit einem Schwellwert von 0.62. Die Confusion-Matrix wurde mit Hilfe der Tänze aus den Validierungsdaten erstellt. Dabei wurde jeder Tanz mit Hilfe des Schwellwerts 0.62 klassifiziert.

damit deutlich besser. Die Tabelle 5.1 wurde mit einem Schwellwert von 0.62 erstellt und zeigt das entsprechende Resultat für die 2015-er Validierungsdaten. Die Verteilung von positiven und negativen Tänzen war in den Daten zuvor 1:2.5, d.h es gab 2.5 Mal mehr negative Tänze als positive Tänze. Nach der Filterung ergibt sich ein Verhältnis von 2:1, d.h es gibt nun doppelt so viele positive Tänze wie negative Tänze. Falls man eine weitere Verbesserung für die TP-Rate haben möchte kann man den Schwellwert zu Gunsten der positiven Tänze verändern.

Bei den 2014er-Daten waren die Ergebnisse etwas schlechter. Meine Vermutung ist, dass durch die geringere Anzahl an Trainingsdaten, es dem Netzwerk nicht möglich war gut zu generalisieren. Während es bei den 2015er-Daten gereicht hat die Daten vor jedem Lauf zufällig vor zu verarbeiten, war dies aufgrund der geringen Anzahl an Daten bei den 2014er-Daten nicht genug um gleiche Ergebnisse zu erzielen. Weitere Tests wären nötig um genauer zu untersuchen wie viele Tänze für optimale Ergebnisse nötig sind und welche andere Faktoren eine Rolle dabei spielen.

Dennoch war es mit dem 2014er-Netzwerk möglich ein gutes Ergebnis für die Lokalisierung der Futterstellen zu erzielen. Viele Futterstellen sind verschwunden und es sind größtenteils Tänze in Richtung der künstlichen Futterstelle übrig geblieben. Um dieses Ergebnis besser bewerten zu können sind weitere Untersuchungen erforderlich.

## 5.2 Diskussion

Die in Kapitel 3 gesetzten Anforderungen wurden hinreichend erfüllt. Obwohl die Erkennungsrate nicht perfekt ist, werden die Ergebnisse des WDD deutlich verbessert, da bei den 2015er-Daten 80% der negativen Tänze herausgefiltert werden können, bei Erhaltung von über 98% der positiven Tänze. Ich denke, es wurde gezeigt das neuronale Netzwerke in der Lage sind den Schwänzeltanz zu erkennen. Die niedrige Auflösung der Frames, die es auch für den Menschen schwer macht einige Tänze zu erkennen, sorgt möglicherweise auch bei dem neuronalen Netzwerk für Probleme.

Ein weiteres Problem ist die Notwendigkeit der Erstellung von Ground-Truth-Daten. Das hier trainierte Netzwerk wurde mit 2250 Tänzen trainiert, die händisch erstellt werden müssen. Ein Ansatz bei dem keine oder deutlich weniger Ground-Truth-Daten erstellt werden müssten wäre wünschenswert.

Außerdem kommt es aufgrund der niedrigen Auflösung der Bilder zu teilweise falschen Ground-Truth-Daten, da es selbst für einen Menschen schwer sein kann einen Tanz zu klassifizieren. Auch das kann zu Problemen beim Training führen.

Auch die Struktur des hier verwendeten neuronalen Netzwerkes ist sicherlich nicht optimal. 2D-Convolutional Schichten werden in der Regel für normale Bildererkennung verwendet. Um die zeitliche Verbindung von Frames besser zu modellieren, könnten 3D-Convolutional Schichten hergenommen werden.

Ein weiteres Problem dieser Arbeit ist, dass die Erkennung von Schwänzeltänzen mit neuronalen Netzwerken, auf der Grundlage des Waggle Dance Detectors stattfindet. Vor allem ist nicht klar wie viele der positiven Tänze während der Echtzeiterkennung des Waggle Dance Detectors verloren gehen. Mit dem gewählten Ansatz in dieser Arbeit ist es nicht möglich diese Tänze zu erkennen. Eine Verbesserung könnte erzielt werden, wenn das neuronale Netzwerk in Echtzeit ohne oder als Ergänzung zu dem Waggle Dance Detectors läuft. Dadurch könnten auch Tänze erkannt werden, die aktuell vom Waggle Dance Detector übersehen werden. Die Leistungssteigerung von Grafikkarten und die Tatsache, das das Netzwerk 30 Frames auf einmal verarbeiten

kann, könnten das möglich machen.

## 5.3 Ausblick

Das Ziel dieser Arbeit eine gute Schwänzeltanz-Erkennung zu implementieren, ist auf Grundlage des Waggle Dance Detectors von Rau gelungen. Die Ergebnisse des Waggle Dance Detectors wurden in dieser Arbeit mit Hilfe von neuronalen Netzwerken verbessert. Allerdings ist es auch in dieser Arbeit nicht gelungen eine Schwänzeltanz-Erkennungsrate von nahe 100% zu erreichen. Um den Waggle Dance Detector mittels neuronalen Netzwerken weiter zu verbessern sollten verschiedene Netzwerk-Architekturen analysiert und erprobt werden. Neue Ansätze die weniger Ground-Truth-Daten benötigen wären ebenfalls wünschenswert, da es weiterhin eine aufwändige Arbeit bleibt Bientänze manuell zu finden. Neue Ansätze mit unüberwachtem Lernen könnten einen wichtigen Beitrag leisten, da sehr viele nicht markierte Daten verfügbar sind.

Eine weitere Verbesserung könnte erzielt werden, indem man die Echtzeiterkennung mit einer größeren Auflösung durchführt, so dass es für das neuronale Netzwerk leichter ist Bienen zu erkennen. Auch wäre es interessant ein System, ohne den Waggle Dance Detector von Rau, zu entwickeln, welches ausschließlich für die Erkennung mit neuronalen Netzwerken entwickelt ist.

Bisher gibt es nur den Waggle Dance Detector von Rau, um Schwänzeltänze automatisiert in Echtzeit zu erkennen. Zusammen mit der hier vorgestellten Verbesserung sind eine Vielzahl von Anwendungen möglich. Zum Beispiel kann man mit den erkannten Tänzen untersuchen welche Art von Futterquelle von Bienen bevorzugt wird. Mit Hilfe der Erkennungen kann man auch die Anzahl von Schwänzeltänzen innerhalb einer Kolonie bestimmen. Dies könnte Rückschlüsse über die Gesundheit einer Kolonie geben.

# Literaturverzeichnis

- [1] Lars Chittka, 2004, Dances as Windows into Insect Perception, <http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.0020216>
- [2] Adam Feldman und Tucker Balch, 2004, Representing Honey Bee Behavior for Recognition Using Human Trainable Models, <http://www.cc.gatech.edu/~tucker/Papers/feldman04.pdf>
- [3] Toshifumi Kimura, Mizue Ohashi, Ryuichi Okada, und Hidetoshi Ikeno, 2011, A new approach for the simultaneous tracking of multiple honeybees for analysis of hive behavior, *Apidologie*, 42(5):607–617, 2011. ISSN 0044-8435, doi: 10.1007/s13592-011-0060-6, <http://dx.doi.org/10.1007/s13592-011-0060-6>
- [4] Keras Documentation, <http://keras.io>, Abfragedatum 01.03.2015
- [5] David Kriesel, 2007, A Brief Introduction to Neural Networks, available at <http://www.dkriesel.com>
- [6] Laura Mielke, 2015, Improvement of the Waggle Dance Detector
- [7] Andrew Ng, Jiquan Ngiam, Chuan Yu Foo, Yifan Mai, Caroline Suen, Adam Coates, Andrew Maas, Awni Hannun, Brody Huval, Tao Wang, Sameep Tandon, UFLDL Tutorial, Deep Learning tutorial, <http://ufldl.stanford.edu/tutorial/>
- [8] Alexander Rau, 2014, Realtime Honey Bee Waggle Dance Decoding System, <http://www.mi.fu-berlin.de/inf/groups/ag-ki/Theses/Completed-theses/Master-Diploma-theses/2014/Rau/index.html>

- [9] Riley, J. R., Greggers, U., Smith, A. D., Reynolds, D. R., and Menzel, R., 2005, The flight paths of honeybees recruited by the waggle dance. *Nature* 435, 205–207. doi: 10.1038/nature03526
- [10] Raul Rojas, 1996, Springer-Verlag, *Neural Networks - A Systematic Introduction*, available at <https://page.mi.fu-berlin.de/rojas/neural/>
- [11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg und Li Fei-Fei, 2015, ImageNet Large Scale Visual Recognition Challenge, <http://arxiv.org/pdf/1409.0575.pdf>
- [12] Stuart J. Russell and Peter Norvig, 2013, Prentice Hall; Auflage: New international edition, ISBN-13: 978-1292024202
- [13] Jürgen Schmidhuber, 2014, *Deep Learning in Neural Networks: An Overview Technical Report*, <http://arxiv.org/pdf/1404.7828v4.pdf>
- [14] Seeley T., 1995, *The Wisdom of the Hive*, Cambridge, MA: Harvard University Press.
- [15] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever und Ruslan Salakhutdinov, 2014, Dropout: A Simple Way to Prevent Neural Networks from Overfitting, <https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf>
- [16] David R. Tarpy und North Carolina Cooperative Extension Service, 2004, *The Honey Bee Dance Language*, <https://www.cals.ncsu.edu/entomology/apiculture/pdfs/1.11\%20copy.pdf>
- [17] Fernando Wario, Benjamin Wild, Margaret J. Couvillon, Raúl Rojas and Tim Landgraf, 2015, Automatic methods for long-term tracking and the detection and decoding of communication dances in honeybees, <http://dx.doi.org/10.3389/fevo.2015.00103>
- [18] Weidenmüller A., Seeley T. D., 1999, Imprecision in waggle dances of the honeybee (*Apis mellifera*) for nearby food sources: error or adaptation? *Behav. Ecol. Sociobiol.* 46, 190–199. doi: 10.1007/s0026500 50609

- [19] Deeplearning Documentation, <http://deeplearning.net/tutorial/lenet.html>, Abfragedatum 01.03.2015
- [20] Receiver Operating Characteristic, Abfragedatum 01.03.2016, Wikipedia: The Free Encyclopedia. Wikimedia Foundation, [https://en.wikipedia.org/wiki/Receiver\\_operating\\_characteristic](https://en.wikipedia.org/wiki/Receiver_operating_characteristic)