

Forschungsprojekt und Seminar

Computer Science

# Handgestenerkennung mit Entscheidungsbäumen

von

Tom Dymel

Februar 2021

Betreut von

Dr. Venzke

Institute of Telematics, Hamburg University of Technology

Erstprüfer

Prof. Dr. Volker Turau

Institute of Telematics  
Hamburg University of Technology

Zweitprüfer

Dr. Venzke

Institute of Telematics  
Hamburg University of Technology



# Inhaltsverzeichnis

<b>Verwendete Symbole</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Techniques and Methods</b>	<b>3</b>
2.1 Decision Trees . . . . .	3
2.2 Ensemble . . . . .	3
2.2.1 Voting . . . . .	3
2.2.2 Bagging . . . . .	3
2.2.3 Random Forest . . . . .	3
2.2.4 Boosting? . . . . .	3
2.3 Generation . . . . .	3
2.3.1 Greedy approximation (CARD?) / (Boosting?) . . . . .	3
2.3.2 Cherry Picking? . . . . .	3
2.3.3 Scikit-Learn? . . . . .	3
<b>3 State of the Art</b>	<b>5</b>
3.1 Gesture candidate . . . . .	5
3.2 Feed Forward Neural Network . . . . .	5
3.2.1 Performance . . . . .	5
3.2.2 Resource usage . . . . .	5
3.2.3 Optimization . . . . .	5
3.3 Recurrent Neural Network . . . . .	5
3.4 Related Work . . . . .	5
<b>4 Achievements</b>	<b>7</b>
4.1 Infrastructure . . . . .	7
4.1.1 Parsing gestures . . . . .	7
4.1.2 Feature extraction . . . . .	7
4.1.3 Training . . . . .	7
4.1.4 Arduino . . . . .	7
4.1.5 Utility Tools . . . . .	7
4.1.6 Workflow? . . . . .	7
4.2 DymelData . . . . .	7
4.2.1 Configurations . . . . .	8
4.2.2 NullGestures . . . . .	8
4.2.3 Synthetic Brightness Dataset . . . . .	8
4.3 Features . . . . .	8
4.3.1 Requirements . . . . .	8
4.3.2 Curse of dimensionality . . . . .	8
4.3.3 Unpromising features . . . . .	8

## INHALTSVERZEICHNIS

4.3.4	Brightness distribution . . . . .	8
4.3.5	Motion History . . . . .	8
4.3.6	Center of Gravity Distribution . . . . .	8
4.4	Performance Evaluation . . . . .	9
4.4.1	Testsets . . . . .	9
4.4.2	Feasible solution . . . . .	9
4.4.3	Considering NullGestures . . . . .	9
4.4.4	Brightness Distribution . . . . .	9
4.4.5	Motion History . . . . .	9
4.4.6	Brightness Distribution and Motion History . . . . .	10
4.4.7	Center of Gravity Distribution Float Ansatz . . . . .	10
4.4.8	Center of Gravity Distribution Integer Ansatz . . . . .	10
4.4.9	Center of Gravity Distribution Float and Integer Ansatz . . . . .	10
4.4.10	Comparison to previous work . . . . .	10
4.5	Execution Time Evaluation . . . . .	10
4.5.1	WCEP and WCET . . . . .	10
4.5.2	AVR Compiler for AtMega328p . . . . .	10
4.5.3	Optimization Level . . . . .	10
4.5.4	Feature extraction . . . . .	11
4.5.5	Tree Evaluation . . . . .	11
4.5.6	Forest Evaluation . . . . .	11
4.5.7	Total Execution Time . . . . .	11
4.6	Size Evaluation . . . . .	11
4.6.1	CCP (TODO: Abrev.) . . . . .	11
4.6.2	Minimum Leaf Sample Size . . . . .	11
<b>5</b>	<b>Conclusion</b>	<b>13</b>
<b>A</b>	<b>Content of the DVD</b>	<b>15</b>

## Verwendete Symbole

$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$	Regular sets of numbers
$\mathcal{NP}, \mathcal{P}$	Complexity classes
$\mathcal{V} = \{v_0, \dots, v_{N-1}\}$	Set of $N$ nodes $v_i$ belonging to a network with sink $v_0$
$\varrho$	Node density, i.e., the average number of nodes within another node's communication range
$(v_i, v_j) \in \mathcal{E}$	Set of bidirectional communication links in the network
$G = (\mathcal{V}, \mathcal{E})$	Graph representation of a wireless sensor network
$\mathcal{N}_i = \{v_j \in \mathcal{V} \mid i \neq j \wedge (v_i, v_j) \in \mathcal{E}\}$	The set of bidirectional communication partners of node $v_i$
$\mathcal{T} \subseteq \mathcal{E}, \quad  \mathcal{T}  = N-1$	Routing tree rooted in the sink
$\mathcal{T}_i$	Subtree rooted in (and including) node $v_i$
$\mathcal{C}_i, \quad \mathcal{C}_i =  \mathcal{C}_i $	The set and number of children of node $v_i$ in $\mathcal{T}$
$\mathcal{F} = \{v_i \in \mathcal{V} \mid \mathcal{C}_i = \emptyset\}$	Set of leafs in $\mathcal{T}$ and the number of leafs
$\mathcal{F}_i$	Set of leafs in the subtree $\mathcal{T}_i$ of $\mathcal{T}$

## VERWENDETE SYMBOLE

# Introduction

## 1 INTRODUCTION



# Techniques and Methods

## 2.1 Decision Trees

## 2.2 Ensemble

### 2.2.1 Voting

### 2.2.2 Bagging

### 2.2.3 Random Forest

### 2.2.4 Boosting?

## 2.3 Generation

Talk about NP-Complete etc.

### 2.3.1 Greedy approximation (CARD?) / (Boosting?)

### 2.3.2 Cherry Picking?

### 2.3.3 Scikit-Learn?

## 2 TECHNIQUES AND METHODS

## **State of the Art**

### **3.1 Gesture candidate**

### **3.2 Feed Forward Neural Network**

#### **3.2.1 Performance**

#### **3.2.2 Resource usage**

#### **3.2.3 Optimization**

#### **Compression**

#### **Pruning**

### **3.3 Recurrent Neural Network**

### **3.4 Related Work**

TODO!

### 3 STATE OF THE ART

# Achievements

## 4.1 Infrastructure

### 4.1.1 Parsing gestures

### 4.1.2 Feature extraction

### 4.1.3 Training

Scikit-Learn

Ensemble

### 4.1.4 Arduino

Export Tree as C code

Feature extraction

### 4.1.5 Utility Tools

Gesture Recorder

Playground

Gesture Simulation

### 4.1.6 Workflow?

## 4.2 DymelData

TODO: Motivation @ consistent real world evaluation

## 4 ACHIEVEMENTS

### 4.2.1 Configurations

### 4.2.2 NullGestures

Explain what it is in general

**Corner**

**Same side in and out**

**Rotations**

### 4.2.3 Synthetic Brightness Dataset

## 4.3 Features

### 4.3.1 Requirements

**Invariance to velocity**

**Development over time**

**Sense of direction**

**Sense of position**

### 4.3.2 Curse of dimensionality

### 4.3.3 Unpromising features

Explain that they had been tried out and did work out. Explain why

### 4.3.4 Brightness distribution

Explain what it is. Note this as "discretized" position. Explain what requirements it fulfills. Explain different variants.

### 4.3.5 Motion History

Explain origin and how it works. Explain what requirements it fulfills and why.

**Implementation**

### 4.3.6 Center of Gravity Distribution

Motivate.?

**Center of Gravity**

**Integer variant**

**Adding development over time**

**Window size**

Also talk about the squishing?

### 4.4 Performance Evaluation

Talk about what was searched for, i.e. not all sizes had been tried out. Talk about different testset sizes? Talk about the greedy nature of the training algorithm? (Was that done in Cherry Picking?)

#### 4.4.1 Testsets

Motivate. Explain what they are used for. Explain whats in there?

#### Metrics

Talk about used metrics

**Klisch**

**DymelNull**

**DymelGesture**

#### 4.4.2 Feasible solution

#### 4.4.3 Considering NullGestures

#### 4.4.4 Brightness Distribution

Show graphs about: Best solution, Best feasible solution, With and WithOUT considering null gestures. Talk when it starts to generalize more poorly(?)

#### 4.4.5 Motion History

Show graphs about: Best solution, Best feasible solution, With and WithOUT considering null gestures. Talk when it starts to generalize more poorly(?)

## 4 ACHIEVEMENTS

### 4.4.6 Brightness Distribution and Motion History

Show graphs about: Best solution, Best feasible solution, With and WithOUT considering null gestures. Talk when it starts to generalize more poorly(?)

### 4.4.7 Center of Gravity Distribution Float Ansatz

Show graphs about: Best solution, Best feasible solution, With and WithOUT considering null gestures. Talk when it starts to generalize more poorly(?)

Talk about brightness distribution!!

### 4.4.8 Center of Gravity Distribution Integer Ansatz

Show graphs about: Best solution, Best feasible solution, With and WithOUT considering null gestures. Talk when it starts to generalize more poorly(?)

Talk about brightness distribution!!

### 4.4.9 Center of Gravity Distribution Float and Integer Ansatz

Show graphs about: Best solution, Best feasible solution, With and WithOUT considering null gestures. Talk when it starts to generalize more poorly(?)

Here state what was expected and state how it worked out! Talk about brightness distribution!! ==> Alternative approach ==> Stacking (?)

### 4.4.10 Comparison to previous work

## 4.5 Execution Time Evaluation

Talk about constrained micro controller. Say that more frames potentially give more insight about fast gestures or that a longer battery life can be achieved. Say that we only consider COCD here, because that has the best feasible solution.

### 4.5.1 WCEP and WCET

### 4.5.2 AVR Compiler for AtMega328p

### 4.5.3 Optimization Level

Say that we considered O2 since the feasible solution can be compiled in O2. And why its a good Kompromiss between O3 and Os



### 4.5.4 Feature extraction

### 4.5.5 Tree Evaluation

### 4.5.6 Forest Evaluation

Here mainly talk about voting.

### 4.5.7 Total Execution Time

## 4.6 Size Evaluation

Explain that scikit learn provides some hyperparameters that can be changed. Elaborate which exist. Say that I considered here two parameters because all ensemble methods supported that. Maybe other reasons.

### 4.6.1 CCP (TODO: Abrev.)

### 4.6.2 Minimum Leaf Sample Size

## 4 ACHIEVEMENTS

## Conclusion

Do a summary. Should be brief but also name everything and its result more or less. Say the improvement compared to the old approach. (Or downsides).

Mention future work. => Stacking

## 5 CONCLUSION

## Content of the DVD

In this chapter, you should explain the content of your DVD.