Learning human motion features and trajectory predictions in large changing environments

Robin Eberhard, Stefan Dörr

***Abstract*—This study present a new approach to detecting humans in service environments. Making use of the latest advances in machine learning, fast scalable applications can be modeled without having knowledge about the inner workings. We present a leg detection tool, that operates on laser scan data, classifying each point as human or non-human. The results are further used to learn to identify people, allowing for precise operation of an autonomous robot in the service industry. This also includes prediction of trajectories, which will be used for movement in crowded scenes.**

***Keywords*—IEEEtran, journal, LaTeX, paper, template.**

I. Introduction

Autonomous robots are making their way from the production industry to the service industry. This change introduces a new set of problems, that deal with interactions of people and robots. These include for example following and approaching customers, and navigating through a crowded environment.

To solve these problems, the first step is the detection of people, which is here done using a safety laser detector on leg-height [***ref laser scanner***]. In the past, this has already been approached with model-based solutions, that require prior knowledge about shapes and behaviour of legs [***ref arras and subsequent paper***]. We present a new approach, where a neural network learns characteristics on its own and places unique identifiers on persons, making it easy to track them over a long period of time.

Having the information and history of positions of people, we can learn trajectories, behaviour and intentions when the robot is serving customers. We provide a way to model the trajectories using Long short-term memory (LSTM) cells in combination with a Mixture density layer, which outputs a set of normal distributions, similar to *paper from graves*.

In this paper, we present the models that are used to train the program, as well as some benchmarking and comparisons with similar projects.

II. Model architecture

The detection and identification of legs and therefore people based on laser scans is separated into three steps. First, the laser scans are classified by using a set of convolutional and pooling layers. The points labeled as belonging to legs are sorted out and typically clustered based on euclidian distance, in order to identify persons. This approach does not work well, when people are close together or even when they are moving, as the program would then either cluster too many legs together or see two legs as separate people respectively.

This can be countered by supplying the clustering algorithm with additional information. During the convolution step, the network learns features which are unique to legs in different situations. Those can be extracted and passed as additional parameters to the clustering algorithm, which not only allows higher accuracy, but also labeling [***better word***] a pair of legs with a unique identifier.

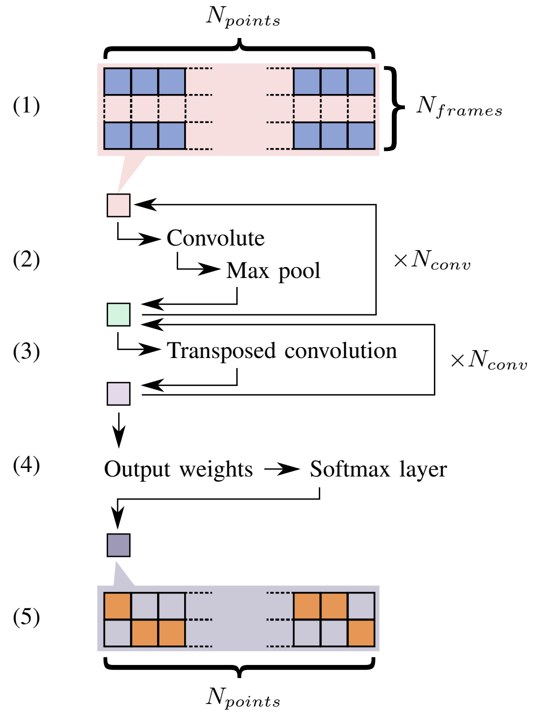

Fig. 1. **Laser scan classification.** (1) The input is a 2D array of $N_{points} \times N_{frames}$ containing distances from a laser scan. It is fed into (2), where features are extracted using convolutions and downscaled using a pooling layer. After N_{conv} iterations of (2), the now one-dimensional output is upscaled in (3) into the shape N_{points}. (4) multiplies the output and applies the softmax function, resulting in probabilities of each point belonging to class human or non-human in (5).

Having identifiers and positions, they can then be fed into the third and last step of the setup, which aims to predict trajectories of individual people. The network takes a set of positions, that are run through a LSTM cell, which implicitly works as a memory for the program and holds the information necessary to build future trajectories.

A. Classification of the laser scan

In order to classify a laser scan, a snapshot of the current scan is taken in equal timesteps. Each snapshot contains one dimensional arrays of length N_{points} with distances r_i to the closest intersections from the laser beams.

From the one-dimensional array as input, the network would only find characteristics in the objective shapes of the laser scans. To also account for movement, N_{frames} snapshots are composed together. The desired outputs are probabilities for each point in the most current snapshot to be in either class leg or non-leg.

Therefore, the mathematical description of the model can be described by a function F that maps an input of size $[N_{frames}, N_{points}]$ to an output of size $[2, N_{points}]$.

The output is modelled from the input as outlined in Figure 2 and described in more detail in the following. When convolutional layers are used for classification, a network will train the weights such that they represent characteristics for a class. In this way, the characteristics are first of geometrical nature and become more abstract the more layers there are.

The input is therefore fed into a convolutional layer and then downsampled using a max pooling layer. This is repeated N_{conv} times, after which the output is a one dimensional array of length smaller than N_{points}.

To gain a classification for each point in the most current laser scan, the output is upscaled using transpose convolutions, again extracting features. Multiplication by output weights and applying the softmax function results in the desired shape $[2, N_{points}]$, where each entry is the probability of the point belonging to class leg and non-leg.

B. Clustering of leg points

In order to gain information about the current position of people, the classified points are clustered together. This usually poses two major problems:

- The clustering has a high margin of error when legs of a single person are too far away or when legs of different people are together and
- tracking of a person has to be handled externally.

With the following approach, we find a high accuracy and are also able to track people on certain features.

As the inputs are convoluted, the network finds characteristics, such as shapes and movement. Some of these features are more important than others, this is why we implemented a training in order to find weights w^{feats}, such that parameters of people in different situations are easily separable. The ith convoluted layer has its weights w_i^{conv} multiplied by w_i^{feats} and the resulting cluster parameter p_i is then found by the sum of all weights:

$$p_i = w_i^{feats} * \sum_j w_{i,j}^{conv} \tag{1}$$

This introduces apart from the euclidian distance $2 * N_{convs}$ additional parameters to the clustering algorithm, highly increasing accuracy. As the parameters have been chosen such that people in different situations are easily separable and given the unlikeliness of two people being in the same position in the same situation, we can additionally use the parameters to track the people, which allows for example to follow customers around.

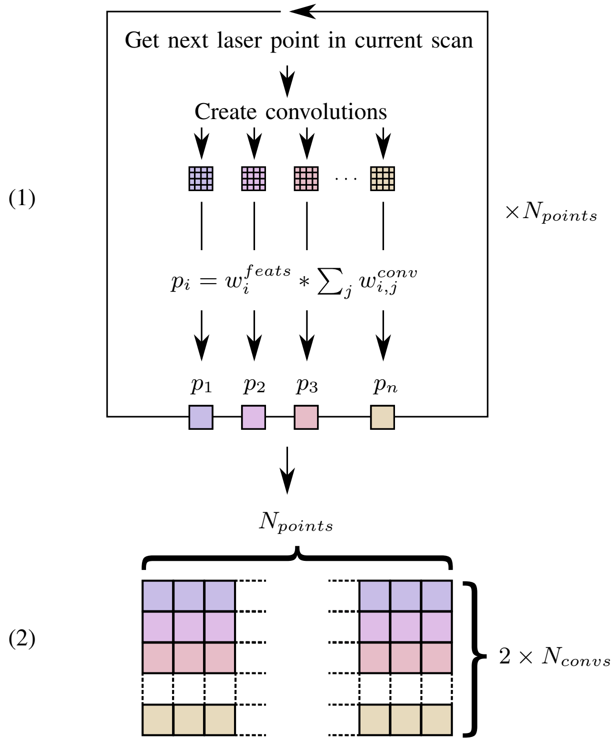

Fig. 2. **Laser scan classification.** (1) The input is a 2D array of $N_{points} \times N_{frames}$ containing distances from a laser scan. It is fed into (2), where features are extracted using convolutions and downscaled using a pooling layer. After N_{conv} iterations of (2), the now one-dimensional output is upscaled in (3) into the shape N_{points}. (4) multiplies the output and applies the softmax function, resulting in probabilities of each point belonging to class human or non-human in (5).

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.