Deep learning for opponent action prediction in robot soccer middle size league

van 't Klooster, Marjon E.

Award date:
2018

Technische Universiteit
Eindhoven
University of Technology

/ Department of
Mechanical Engineering

Deep Learning for Opponent Action Prediction in Robot Soccer Middle Size League

DC 2018.050 - M.E. van ´t Klooster

Master thesis

May 2018

Contents

Deep Learning for Opponent Action Prediction in Robot Soccer Middle Size League

Maria E. van 't Klooster, **prof. dr. Henk Nijmeijer** and **dr. Gijs Dubbelman**

Eindhoven University of Technology
Department of Mechanical Engineering
Master track Dynamics and Control

Abstract *In this research, we propose deep learning to predict future opponent actions for autonomous soccer robots and evaluate this in the context of the RoboCup Middle Size League. The ability to predict actions of opponents will be of key benefit for planning team strategy on positioning and team play, resulting in gaining faster ball possession and a stronger defense. We leverage advances in Convolutional Neural Networks by representing the game state, containing the positions of all elements on the field over time, as a temporal occupancy grid which can be represented and processed as an RGB image. Our approach is evaluated using measurement data that was recorded during nine international tournaments over the past four years which contains 4200 actions in total. The results demonstrate that 77% of the test-data can be correctly classified in three types of actions, , i.e. shot-on-goal, pass, and other actions, by our proposed network structure in real-time. This work shows promising results for opponent action prediction, potentially allowing for more strategic and intelligent team play in RoboCup.*

1 Introduction

The RoboCup Middle-Size League is a competition within RoboCup, where ten fully autonomous robots play soccer according to a subset of the FIFA rules. The objective of RoboCup is to stimulate development of robotics and innovation in Artificial Intelligence (AI) researches, with the aim to win the World championships against human by 2050 [1]. Since 2012, the robot soccer matches in the Middle-Size League are more directed to game play and strategies [2]. The robots react continuously on the actions of opponents and movement of the ball on real-time basis. The current software of Tech United, a team participating in the Middle Size League with already three world titles on their name, contains a rule-based system, where the strategic decisions are based on the current position and velocity of any object on the field, i.e. the ball, peers and opponents. The recognition of situations and prediction of opponent strategies and movements, which is the aim of our research, would be a significant improvement. The prediction of strategies and movements makes it possible to anticipate on future events. Moreover, with the knowledge of the next action of the opponent, the most promising counter action can be selected to gain possession of the ball.

The focus of this study is to investigate the possibility to predict the next action of the opponent in ball-possession by the use of a deep Convolutional Neural Networks (CNN) with as input temporal occupancy grid map which represents the robots' world state. The effectiveness of this method will be demonstrated on real game data of the past four years.

Fig. 1 Illustration of a match in the Middle Size League, where two teams of five autonomous robots play soccer on a field of 18x12m.

The research yields the following contributions:

- A representation of spatio-temporal information from the game state in an occupancy grid map.
- A Convolutional Neural Network design for predicting future opponent actions of soccer robots from these occupancy grid maps.
- Validation of the proposed representation and Convolutional Neural Network on real game play data.

This paper is divided in three main parts, Section 2 discusses related work on machine learning within RoboCup and the use of occupancy grid maps. Our method will be described in Section 3, explaining the world state representation as a temporal occupancy grid map and the structure of the CNN. In Section 4 and 5, the performance of the representation in combination with the CNN is evaluated, the reliability of the used data is determined, and the accuracy of the network to predict the opponent next action is verified. Section 6 lists the recommendations.

2 Related Work

Machine learning is already applied in the RoboCup environment for different purposes and in different leagues. An example, Gabel et al.[3] used a Markov decision process to select the action which can be performed by a Middle Size soccer robot. Reinforcement learning is used to solve the Markov decision processes and to learn individual skills of a single robot. An interesting discovery was the capability of the robots to perform higher level concepts, like passing to each other, with only the individual skills, e.g. shooting, driving. Copete et al.[4] proposed the prediction of the motion trajectory of players depending on ball and player positions without prior knowledge of the team's actions or strategies. Hereby, deep neural networks are used to represent player positions and a one-hidden-layer neural network for the estimation of the movement direction of the ball. This approach looks promising, the ball direction was estimated correctly for 84.1% and the robots' direction between 50.6% and 69.9%, but this has only been tested in the Soccer Simulation League. Briegel et al.[5] identified which action an opponent robot is going to execute without prior knowledge of a teams strategy. Using Support Vector Machines, the situations, generated in a simulator, were classified in a pass or shot on goal. The Support Vector Machine classifier achieved a correct prediction of 91%. The calculation could not be performed in real-time because the data required a considerable amount of pre-processing. In addition, only two classes are determined which will give many false positives in other situations. Currently, the functional performance of the methods above are insufficient to be used on real match data and during live game play of the Middle Size League. This limitation is addressed by our approach whereby opponents' future action is predicted in the Middle Size Leauge on real match data with the aid of Convolutional Neural Networks in real-time.

Movement and action prediction is not only an active research in robot soccer, a lot of research has been performed already in the automotive sector[6][7][8][9]. Byeoung Do et al.[10] proposed trajectory prediction via Recurrent Neural Network with long short term memory, representing the prediction on an occupancy grid map. This method performed better than a Kalman-based prediction model. Wagenaar et al. [11] describes a problem similar to the one posed in this work but then for human soccer. They proposed to predict goalscoring opportunities on a dataset containing images extracted from the interval of 10 seconds before shot on goal occurred by representing the world state in an occupancy map of 256 x 256 pixels. Here, Convolutional Neural Networks are used to categories the state in a promising or less promising goal opportunity. The best achieved result with this method is 67.1%. Compared to [11] and [5], our classification categories are more specific, instead of only two classes, our network classifies game situations in pass, shot on goal and other actions including e.g. scrum, intercept. Furthermore, our method will be applicable during the whole match instead of only 10 seconds before a goal-scoring opportunity will occur.

To summarize, the correct prediction rates of state-of-the-art methods give result in the range of 67.1% for human soccer data and 91% for simulated robot soccer data. These numbers are the baseline for the lower and upper performance bounds that we aim to achieve with our novel approach.

3 Method

The prediction of future opponent actions will be treated as a classification task, where the world state information of the soccer robots, until the current time step, is encoded in occupancy grid maps. These grid maps are the input to a classifier that classifies this world state with the next action of the opponent, i.e. pass, shot, and other actions. The advantages of using occupancy grid maps is the capability to capture both spatial and temporal information. Furthermore, occupancy grid maps have many similarities to images and can be processed by state-of-the art Convolutional Neural Networks, which have shown very promising results on image-related classification tasks [12][13][14]. A key property of images is the on spatial information relation between pixels in an image which decreases with distance. This segmentation information is comparable with the cell information of an occupancy grid map which contains the robot's trajectory. Because images and occupancy grid maps share these properties, CNNs are a promising approach to perform classification, detection, and prediction tasks on occupancy grid maps. In addition, CNNs can be deployed in real-time systems. The following subsections describe the occupancy grid map representation and the structure of two Convolutional Neural Networks, which are evaluated by experiments in Section 4.

3.1 Representation of World State

The two dimensional positions of opponent, peer, and the ball on the field are stored in three separated occupancy grid maps with size of 40x28 cells, where the cells size is 50x50 cm and approximately the size of a Middle Size robot. The temporal information is encoded in the occupied cells of the occupancy grid map. The highest value (255) represents the current position, the previous positions have progressively lower values. The value with which it decreases is equal to 255 divided by the number of included time steps. The cells of the occupancy grid map which are not occupied will have the value 0. For example, for the occupancy grid including 30 time steps representing the peer locations, a value of 230 in cell with location x=11, y=22, means that a peer was located at the field on x=6m, y=11m, two time steps back. The three occupancy grid maps (peer, opponent, ball) can be stacked and processed and visualized as RGB images.

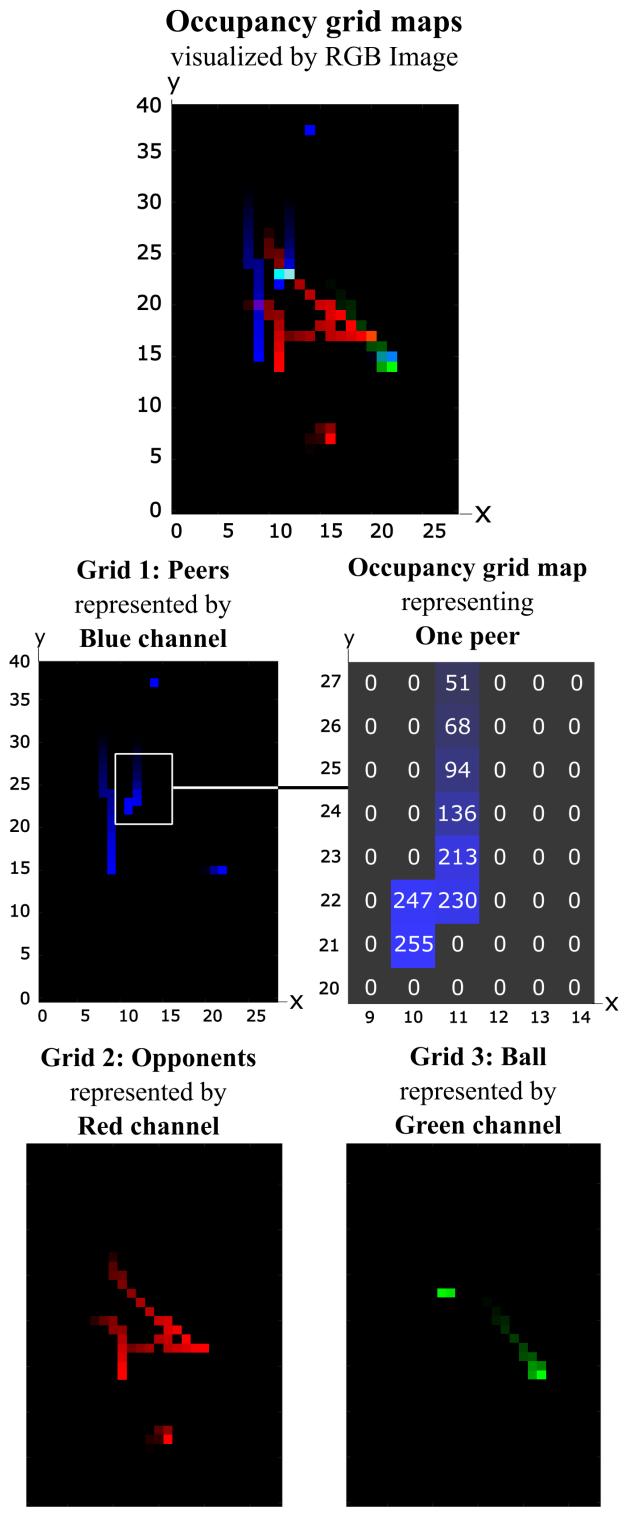

Fig. 2 World state representation as occupancy grid map. Game situation before shot at goal occur of Figure 5a.

Figure 2 shows an example of the world state before a shot on goal, every channel corresponds to one of the three occupancy grid maps. The temporal information is made visible through the increasing intensity of the color. Note, that our CNN approach is not limited to occupancy grids with three channels and can handle arbitrary number of auxiliary channels.

3.2 Convolutional Neural Network structure

Two networks are compared; The first one is an existing CNN for recognizing and classifying the CIFAR-10 dataset [15]. The CIFAR-10 dataset contains 60.000 32x32 colour images and is divided in ten classes (cat, dog, etc.). The structure of the existing CIFAR-10 network is shown in Figure 3 and achieves a test accuracy of 79-80% on the CIFAR-10 dataset for image classification [16].

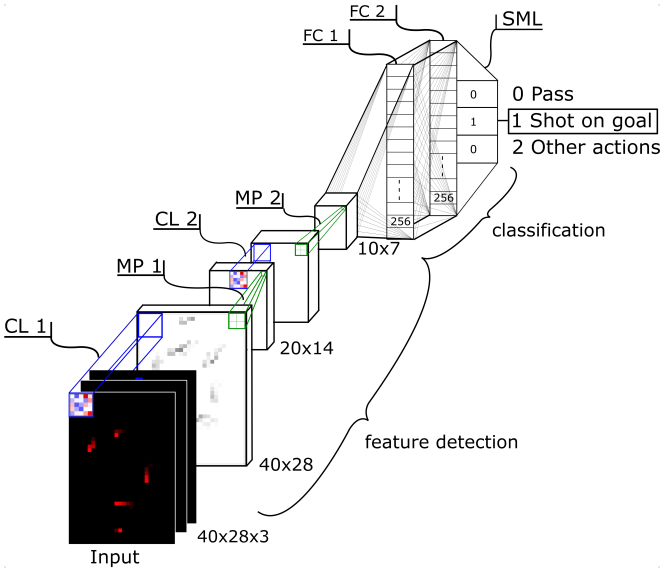

Fig. 3 Convolutional Neural Network Structure 1

The second CNN is an improvement of the first network [12][17]. Containing more convolutional layers for understanding higher-order features of the occupancy grid maps. Furthermore, more pooling layers are included, this affects the information of the input which the neuron receives in the last layers. One neuron of CNN 1 in the last layer, before the fully connected layer, receives information of 4x4 cells at once. However, one neuron of CNN 2, after the average pooling layer, receives information of 28x28 cells of the input. Therefore, CNN 2 classifies the occupancy grid maps using information at a larger spatial resolution. Figure 4 shows the structure of the CNN 2 which can be divided in two parts; feature detection and classification. Table 1 lists the parameters of the layers within the feature detection part of both CNNs.

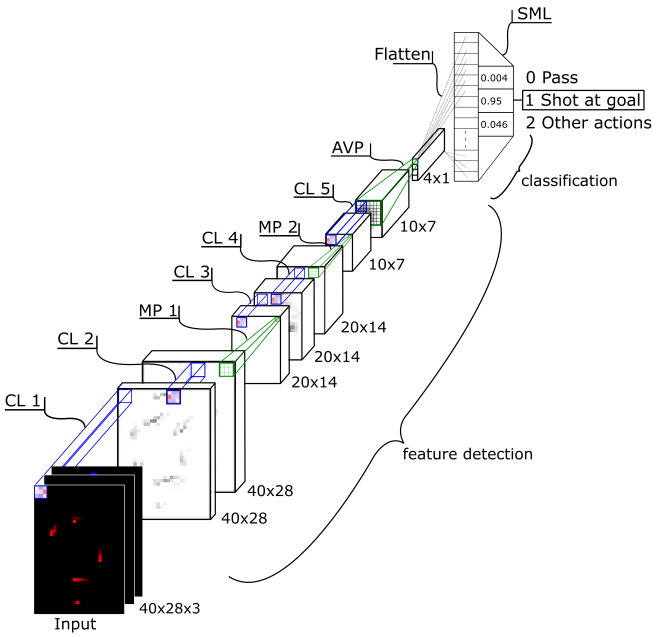

Fig. 4 Convolutional Neural Network Structure 2

Furthermore, the convolutional layers use Rectified Linear Units (ReLU) for the activation function of the output, meaning if the output of the neuron is negative it will be set to zero. The classification is done by flattening the last output matrix after the average pooling layer into one vector which is fully-connected to the softmax layer. The softmax layer use the softmax function:

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^{3} e^{a_k}}, \tag{1}$$

which takes as input $a_1 ... a_3$, which are the raw output of the fully connected layer for classes 1 to 3 (pass, shot on goal, and other actions), and converts them into normalized confidence values between 0 and 1. The output of the classifier is the class with the highest confidence.

Table 1 Parameters feature detection layers CNN

	Layer type	Filter size	Stride	#Filters
CNN 1	Convolutional (CL)	5 x 5	1	64
	Max Pooling (MP)	2 x 2	2	-
CNN 2	Convolutional (CL)	3 x 3	1	64
	Max Pooling (MP)	3 x 3	2	-
	Average Pooling (AVP)	7 x 7	1	-

*All layers use zero-padding

Training of the neuron activation weights is done by the Adam optimizer[18], using the cross entropy loss function, which computes adaptive learning rate for each parameter. The cross entropy loss function is the sum of the cross entropy loss for each individual training occupancy grid maps. This, per grid loss for training grid n with ground truth label $y_n(1,2,3)$, is

$$H_n = -log(p_{y_n}) \tag{2}$$

4 Experiments

The following experiments are performed to determine the accuracy of the temporal data and the achievable performance of the network on the proposed world state representation for opponent action prediction.

I *Performance validation of CNN and representation*: To determine whether the proposed representation is applicable in game situations of the MSL soccer robots, the achievable test accuracy and the computation time will be verified with classifying *referee tasks*. The achievable test accuracy of the CNN for the described world state representation is determined with a simple experiment, namely to recognize the referee tasks. The network will be trained with two easy referee situations, specifically Kickoff and Throw In, and should be classified with high accuracy by the CNN. These are straightforward because the position of the ball is obvious in these situations. In the second validation step of experiment 1, seven referee tasks are evaluated. The time which is needed for classification will be determined, to evaluate if the network is fast enough for the real-time strategy decisions of the soccer robots.

II *Correctness validation of processed data for opponent action classes*: The goal of the second experiment is to evaluate if the selected data is sufficient to predict a dynamic action. I.e., does the data contain enough detail to detect a pass or a shot on goal. The method to evaluate this is based on classifying *occurred actions*. A complete sequence of time steps where a pass is given is represented by a single occupancy grid and provided to the network. A measure for the performance of the network is the number of correctly detected passes or shots at goal. Figure 5 shows an example of a pass and shot on goal respectively. The remaining situations are allocated in a third class, which is of high importance when the system is used real-time in the software, because every situations must be classifiable.

III *Action prediction*: The final experiment is to define the performance of the network on the test-set during prediction of the *future action* of an opponent with the ball.

Hereby, an optimum for the included time period in the occupancy grid map will be determined.

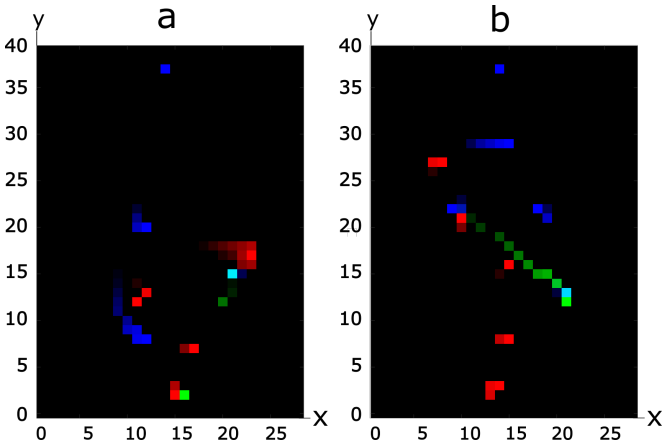

Fig. 5 Game situations represented by different number of time steps. (a) Pass (b) Shot at goal. For both, the cell size is 50 cm and respectively 15 and 12 past time steps are encoded in the occupancy grid.

4.1 Datasets and augmentation

The Middle Size soccer robots are equipped with sensors to observe the environment around them. The observed data and their own position on the field is communicated with each other at 40 Hz. This information is used to build a world model including velocity, movement direction etc. of peers, opponents and the ball on the field. Since 2010 match data is saved of World Championships and European tournaments. This historical data contains even more information besides the robots positions, such as status of some hardware parts and the communicated referee decisions. This data is stored with a sample rate of around 20 Hz for each robot of Tech United. As the games are more directed to game play and strategies due to innovation of robots and changes in rules [2], data of the last four years will be used. The data of World Championship 2017 will be used for the test-set. The training and test-set are in every case evenly distributed over the classes, this because research shows that balanced data-sets perform better[19][20]. To avoid over-fitting on the data by the Convolutional Neural Network, the occupancy grid maps of the training data will be mirrored on the horizontal and vertical axis and added to the training dataset.

The used datasets for the three experiments are described in the next paragraphs followed by three sections with the results.

I *Referee task-datasets*: The achievable test accuracy and the needed computation time will be determined with two datasets containing referee tasks, both with only the positions of the robots and ball at the field on one specific moment in time. The first dataset includes 2 types of referee tasks and the other data-set contains all types. The amount of game situations which is used for these experiments is listed in Table 2.

Table 2 *Referee task* dataset 1 and 2: Number of game situations in the training- and test-set.

	Data-set 1		Data-set 2	
Refbox type	**Training**	**Test**	**Training**	**Test**
Kickoff	1000	100	1000	50
Throw In	1000	100	1000	50
Freekick	-	-	1000	50
Corner	-	-	1000	50
Penalty	-	-	40	2
Goalkick	-	-	1000	50
Dropped ball	-	-	410	25
Total	2000	200	5450	277

II *Occurred action-datasets*: A dataset is generated whereby the actions pass, shot on goal and random other actions took place. The number of time steps included in the occupancy grid map differs. The time steps which are included for a pass is from the moment a pass starts till the ball is received by the peer. For shot on goal, the time steps between the start of a shot till the ball is nearby the backline are included. The third class contains random game-play situations other than pass and shot on goal with different number of time steps. Two data-sets are composed to compare the quality of information of the stored data of opponent and peers. The dataset containing opponent actions is based on observations of the Tech United robots. Table 3 lists the number of game situations with the ratio of *Peer* or *Opponent* for Dataset 3. Table 4 lists the amount of data with only *Peer* information.

Table 3 *Action* dataset 3: Number of game situations per class. With the ratio between *Peer*-data and *Opponent*-data.

	Training			Test		
Action	# Situations	Peer portion	Opponent portion	# Situations	Peer portion	Opponent portion
Pass	2500	38%	62%	100	45%	55%
Shot on goal	2500	98%	2%	100	91%	9%
Other Actions	2500			100		
Total	7500			300		

III *Future opponent action-datasets*: This dataset contains game situations which are taken backwards in time with different number of included time steps from the moment a robot receive the ball. An optimum number between 1 and 100 steps backwards in time will be determined for predicting an action. As mentioned earlier, data is stored with a frequency around 20Hz, hence a duration of maximal 5 seconds backwards in time will be included in the occupancy grid map. Figure 5 shows two different examples: before a pass, and before a shot on goal and both with another amount of included time steps. The *Action prediction*-datasets contains only *Peer*-data and the number of game situations is equal to Table 4.

Table 4 *Action* dataset 4: Number of game situations per class of only *Peer*-data

Action	Training	Test
Pass	1400	100
Shot on goal	1400	100
Other Actions	1400	100
Total	4200	300

4.2 Validation of CNN and representation results

The proposed world state representation where information of the opponents, peers and the ball is separated in the RGB channels is effective. As listed in Table 5, an accuracy of 98,5% is achieved on the first dataset with CNN 1. This high test accuracy was already achieved after one training batch of 400 game situations. Even with more classes, the network performs well. This can be substantiated with Table 7, which shows the confusion matrices of both CNNs after all game situations are trained. A confusion matrix lists the performance

Table 5 Experiment results on Test-set 1 after every training batch containing 400 game situations from *Referee task* dataset 1.

	Batch	Training time	Accuracy
CNN 1	0	-	50.0% (100/200)
	1	2:48 min	99.0% (198/200)
	2	2:52 min	98.5% (197/200)
	3	2:49 min	98.5% (197/200)
	4	2:39 min	98.5% (197/200)
	5	2:42 min	**98.5%** (197/200)
CNN 2	0	-	50.0% (100/200)
	1	6:32 min	98.0% (196/200)
	2	6:02 min	97.5% (195/200)
	3	6:50 min	97.0% (194/200)
	4	6:56 min	97.5 % (195/200)
	5	6:53 min	**98.0%** (196/200)

Table 6 Experiment results on Test-set 2 after every training batch containing 1090 game situations from *Referee task* dataset 2.

	Batch	Training time	Accuracy
CNN 1	0	-	18.1% (50/277)
	1	7:41 min	75.5% (209/277)
	2	7:45 min	77.3% (214/277)
	3	7:53 min	77.3% (214/277)
	4	8:11 min	78.4% (218/277)
	5	8:10 min	**77.6%** (215/277)
CNN 2	0	-	19.5% (54/277)
	1	11:33 min	69.3% (192/277)
	2	14:42 min	74.4% (206/277)
	3	17:49 min	76.2% (211/277)
	4	18:07 min	77.3% (214/277)
	5	16:56 min	**76.5%** (212/277)

on particular class and can only be read horizontally and illustrates the number of right and wrong predicted classifications, as can been seen on the third row of Table 7a. A Freekick is more often confused with a Goalkick than the other way around. Kickoff, Throw In, and Corner are easy to indicate. However, dynamic situations are harder to classify, with dynamic is meant the position where these referee tasks are taken.

Table 7 Confusion matrix of *Referee task* dataset 2

a. CNN 1

	Kickoff	Throw In	Freekick	Corner	Penalty	Goalkick	Dropped ball	Performance
Kickoff	49	1	0	0	0	0	0	98%
Throw In	1	43	4	1	0	0	1	86%
Freekick	0	11	30	2	0	7	0	60%
Corner	0	0	0	50	0	0	0	100%
Penalty	1	0	0	0	1	0	0	50%
Goalkick	1	4	6	1	0	38	0	76%
Dropped ball	3	4	11	1	0	2	4	16%

b. CNN 2

	Kickoff	Throw In	Freekick	Corner	Penalty	Goalkick	Dropped ball	Performance
Kickoff	50	0	0	0	0	0	0	100%
Throw In	1	43	2	3	0	0	1	86%
Freekick	1	10	30	2	0	7	0	60%
Corner	0	1	0	49	0	0	0	98%
Penalty	0	0	1	0	1	0	0	50%
Goalkick	0	9	10	1	0	30	0	60%
Dropped ball	3	3	9	1	0	0	9	36%

For example, the referee tasks Freekick and Dropped ball achieve low performance percentages namely 50% and 16% for CNN 1. The achieved results from CNN 1 and CNN 2 on dataset 1 and dataset 2 look promising, high test accuracy is possible when the condition of the class is well defined. E.g. a Kickoff is always taken at the center of the field.

Not only performance of the network on the datasets is of importance, also the duration to classify the game situations are significant, because eventually the network should be applied during a match. The training time values are listed after every batch in Table 5 and Table 6 column 2. Note; the training batches have a different size for the datasets. CNN 1 needs approximately 1.97 milliseconds and CNN 2 needs 7.61 milliseconds for classifying an game situation of the test-set. The strategic decisions of the robots are calculated on 1000 Hz and are communicated around 40 Hz with each other. A dedicated computation unit with similar computational power, has to be considered to execute this in real-time at the robot. Both Convolutional Neural Network structures can achieve high performance on the proposed representation in a time period in which strategic decisions must be taken.

4.3 Validation of action-dataset

Previous experiment confirms that the proposed representation is effective. In the following experiment, the measurement data for action prediction will be tested on reliability and the achievable performance will be determined. Figure 6 shows the test accuracy after every 500 trained game situations of the two networks trained on the *occurred action*-datasets. This procedure is repeated five times for the same training batches. The red and purple lines correspond to dataset 3 containing *opponent and peer* information. The blue and green lines belongs to dataset 4 containing only *peer*-information. The thick lines show the average of these experiments. Focusing only on the results of CNN 1, the two mean curves have the same shape only shifted along the vertical axis. This means that the *Peer*-dataset is more reliable than the dataset with *Opponent* information. This also applies for the results of CNN 2. The difference between those two data-sets are a result of wrong selected game situations from opponent data. It is not possible to track the opponent accurately. Moreover, as it is when an opponent moves outside the camera range and again inside, the label of the robot changes. This means that if the robot dribbles with the ball, a pass will be selected for the data-set because of the label change. This can be confirmed by comparing the two confusion matrices; Table 8b shows larger numbers of correct predicted passes and other actions in comparison with Table 8a without a large change in number of wrong predicted passes and other actions as shot on goal. The reliability is supported with the variance of the

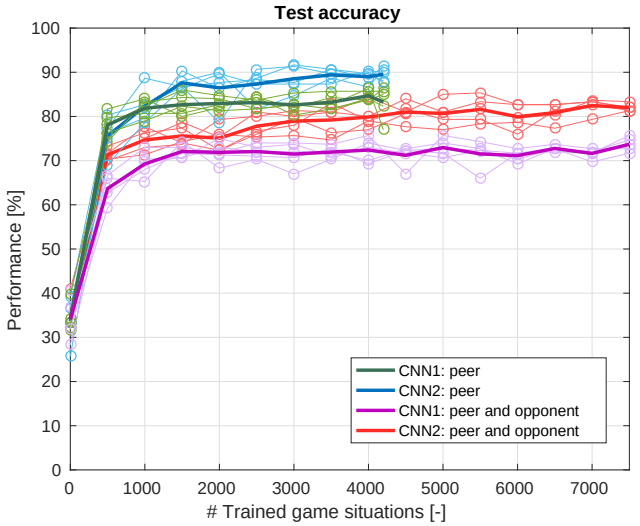

Fig. 6 Test performance on two data-sets containing occured actions

Table 8 Confusion Matrices related to occurred actions from CNN 1

a. Peer and Opponent data

Confusion matrix	Pass	Shot on goal	Other actions
Pass	*74*	4	22
Shot on goal	7	*80*	13
Other actions	23	10	*67*
Average performance			73.6% 221/300

b. Peer data

Confusion matrix	Pass	Shot on goal	Other actions
Pass	*92*	0	8
Shot on goal	9	*73*	18
Other actions	11	7	*82*
Average performance			82.3% 247/300

Table 9 Confusion Matrices related to occurred actions from CNN 2

a. Peer and Opponent data

Confusion matrix	Pass	Shot on goal	Other actions
Pass	*77*	2	21
Shot on goal	4	*91*	5
Other actions	15	7	*78*
Average performance			82% 246/300

b. Peer data

Confusion matrix	Pass	Shot on goal	Other actions
Pass	*93*	1	6
Shot on goal	2	*92*	6
Other actions	5	12	*84*
Average performance			89,7% 269/300

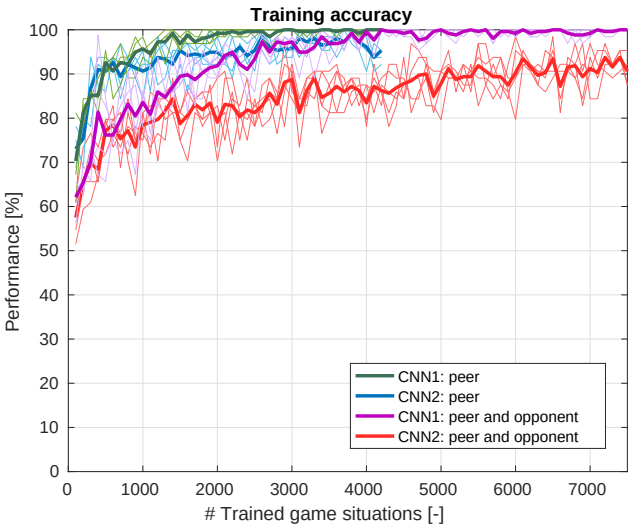

Fig. 7 Training performance on two data-sets containing occured actions

training process, which is in this case constant for both networks. After reproducing the experiments five times, the variance for CNN 1 for *Peer*-data is only $\sigma = 1.8\%$ and for CNN 2 is $\sigma = 2.2\%$. The test accuracy depends not only on the reliability but also on the amount of available data. The disadvantage of using only *Peer*-data is that less data is available through elimination of opponent data. However, the amount of training-data which is used seems sufficient for CNN 1, as can be concluded from the shape of the mean curve. The training-set would be too small if the curve is still increasing. This can also be confirmed in Figure 7 which shows the training accuracy, as can be seen the curves for CNN 1 both converge to 100%. This indicates that the performance cannot increase higher than 73.6% and 82.3% for the data-sets. In contrast to this, the curves of CNN 2 are still increasing, which means the performance can still increase when more train data would be available. However, the data-set for action prediction will only contain *Peer*-information because the reliability is more important than the amount of available data.

The achieved test accuracy of both networks on dataset 4 indicates a convenient prospect. The highest average after 4200 trained game situations is 82.3% for CNN 1 and 89.7% for CNN 2. The difference between those results can be found in the confusion matrices. Table 8 and Table 9 give average confusion matrices of the experiments from the last three measurement points. Taking a closer look to the performance on individual actions in the confusion matrix of CNN 1 shows that; The lower number of correct predicted shot on goals in comparison with passes could be a result of the velocity of the ball during these actions. A shot on goal is harder to classify because the ball has a higher velocity and is therefore visible in only a small amount of time steps, see Figure 5a for example. However, CNN 2 performs better on the class shot on goal, this can be a result of an understanding of higher-order features corresponding with the larger field-of-view of the kernel. Furthermore, CNN 2 performs well on the classes shot on goal and pass with the *Peer*-dataset. The class other actions score lower because of the variety of the situations. This experiment has been conducted to determine the correctly classified dynamic actions to define the upper bound performance for the next experiment. The goal of this next experiment is to predict future opponent actions by the Convolutional Neural Network instead of actions that have already occurred. The performance average of CNN 2 on the *Peer*-data is the target performance during the opponent action prediction experiments.

4.4 Opponent action prediction results

In the previous two sections the representation and the Convolutional Neural Network are validated and considered useful to recognize game situation in Robot Soccer. This section presents the results of the main experiment III; the prediction of opponent actions. The prediction of future opponent actions is determined through a classification task, where the following classes are defined; pass, shot on goal or other actions. The highest test accuracy average for action prediction is 76.9% achieved with CNN 2 on the dataset with 10 past time steps included. Figure 8 shows the performance of the CNNs on datasets with occupancy grid maps including a certain length of historical information. Every dataset is trained and tested five times, the mean values of these tests are listed in Table 10.

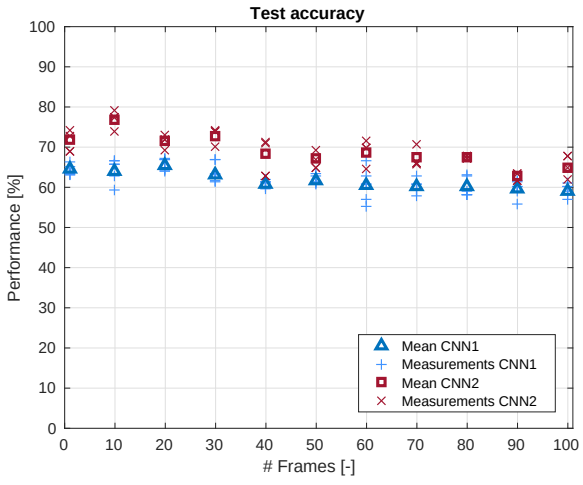

Fig. 8 Performance of CNN on different data-sets with increasing length of historical information

Table 10 Mean values corresponding to Figure 8

Historical time window	t	t-10	t-20	t-30	t-40	t-50
Mean [%] CNN 1	64.5	64.0	**65.5**	63.2	60.9	61.8
Mean [%] CNN 2	72.0	**76.9**	71.4	72.8	68.3	67.1

Historical time window	t-60	t-70	t-80	t-90	t-100
Mean [%] CNN 1	60.5	60.3	60.3	59.6	59.1
Mean [%] CNN 2	68.7	67.5	67.6	62.8	64.9

Table 11 Confusion Matrices related to the highest mean value

a. CNN 1

Confusion matrix	Pass	Shot on goal	Other actions
Pass	*76*	7	17
Shot on goal	19	*52*	29
Other actions	22	10	*68*
Average performance			65.3% 196/300

b. CNN 2

Confusion matrix	Pass	Shot on goal	Other actions
Pass	*74*	11	15
Shot on goal	18	*73*	9
Other actions	15	6	*79*
Average performance			75.3% 226/300

The average is only 1.6% higher than the average of the corresponding confusion matrix in Table 11, which is due to rounding of the overall test accuracy values. It is interesting to see in Table 10 that the next action of a soccer robot can be predicted using only position data of the robots on the field.

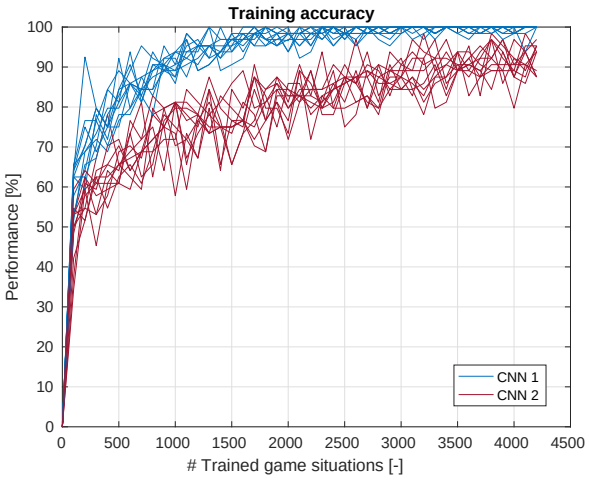

Fig. 9 Training batch accuracy of both CNNs

CNN 2 is able to predict the action for 72.0% with only the positions. The performance increases 4.9% with historical data in the occupancy grid maps, but the prediction is getting worse with more than 10 past time steps. A prediction accuracy of 76.9% is already a good performance but not as high as the aimed performance of 89.7% in the previous section. However, the limiting factor in the prediction for CNN 2 is the amount of training data, Figure 9 shows that the training accuracy is still converging to 100%, hence a higher performance could be possible.

5 Conclusions

In this research, measurement data from the past four years of the RoboCup Middle Size League soccer matches has been used to predict future actions of soccer robots using Convolutional Neural Networks. A representation is developed where current and past object positions are been represented in occupancy grid maps. We demonstrated the effectiveness of capturing the world state of the soccer robots in occupancy grid maps by classifying referee tasks with only information of a snapshot. In addition, we have verified that post analysis of game situations can be accurately classified for around 89.7% by including the robots trajectory in the occupancy grid map. Finally, we showed that Convolutional Neural Networks can learn to predict actions correctly for 76.9% by using occupancy grid maps filled with 0.5 seconds of historical data.

6 Recommendations

In future work, to gain better results, the dataset can be increased with simulation data and within several years, better data of the opponent can be obtained of shared data. Since a year, data is logged during the game of both teams, in a few years this could be enough data to use for this approach. In addition, the representation can be extended by including more information of the world state, e.g. the height of the ball could be integrated. This information could visualize the bouncing of the ball and might give a better insight for the neural network whether a pass is going to be received. Another example, for classifying a pass more accurately, velocity and movement direction could be included. Hence, the classes can then be divided in more detailed actions, e.g. pass versus intercept. However, using additional information has consequences for the neural network, which may need to be extended with more layers to extract higher-order features. Eventually, our approach will be integrated into the TU/e Middle Size League robots and evaluated during live game play, where, for example, the defense can be improved, because it is possible to anticipate on an upcoming pass or shot on goal. Instead of classifying a certain situation, a region of interest could be identified by the neural network, like the position where a pass is going to be received by the opponent, whereby the robot can intercept easier. Or the best position of the robots during a passive referee task could be determined by the neural network,

to gain faster ball possession. This research shows that the combination of multi-channel occupancy grad maps together with CNNs offer a powerful and extensible approach. It lays the foundation for many potential extensions to improve the cognitive capabilities of soccer robots.

References

[1] RoboCup Federation. Objective. http://robocup.org/objective, 2016.

[2] R. Molengraft van de R. Cunha B. Soetens. Robocup MSL - History, Accomplishments, Current Status and Challenges Ahead. *RoboCup 2014: Robot World Cup XVIII, Springer International Publishing, Cham, 624–635,*, 2015.

[3] T. Hafner R. Lange S. Lauer M. Riedmiller M. Gabel. Bridging the gap: Learning in the robocup simulation and midsize league. *In Proceedings of the 7th Portuguese Conference on Automatic Control (Controlo)*, 2006.

[4] J. Suzuki J. Wei Q. Iwaki R. Endo N. Mori H. Nagai Y. Asada M. Copete. Estimation of players' action in soccer matches based on deep autoencoder. *The 42th Annual Conference of Artificial Intelligence AI Challenge Study Group (SIG-Challenge) Procedure, JSAI Technical Report*, 2015.

[5] M. Briegel. Multi-agent action recognition and prediction using support vector machines, 2012.

[6] A. Fei X. Boucherche A. Magnano. Movement Prediction in Vehicular Networks. *Global Communications Conference (GLOBECOM), IEEE, San Diego,*, 2015.

[7] Martinson E. Chintalapudi V. Guo R. Olabiyi, O. Driver Action Prediction Using Deep (Bidirectional) Recurrent Neural Network. *Toyota InfoTechnology Center USA, California*, 2017.

[8] Misu T. Tawari A. Miranda A. Suga C. Fujimura K. Li, N. Driving Maneuver Prediction using Car Sensor and Driver Physiological Signals. *Proceedings of the 18th ACM International Conference on Multimodal Interaction, ACM, New York*, pages 108–112, 2016.

[9] Ohn-Bar E. Manubhai Trivedi M. Khosroshahi, A. Surround vehicles trajectory analysis with recurrent neural networks. *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), IEEE, Brazil*, 2016.

[10] Mook Kang C. Kim J. Hi Lee S. Choo Chung C. Won Choi J. Kim, B. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. *dblp computer science bibliography, CoRR*, 2017.

[11] Wiering M.A. Frencken W. Wagenaar, M. Using deep convolutional neural networks to predict goal-scoring opportunities in soccer. *Research Gate, Conference Paper, DOI: 10.5220/0006194804480455*, 2017.

[12] B. Graham. Fractional max-pooling.

[13] Meier U. Schmidhuber J. Ciresan, D. Multi-column deep neural network for image classification. *Technical Report No. IDSIA-04-12*, 2012.

[14] Chenyang Lu, Gijs Dubbelman, and Marinus Jacobus Gerardus van de Molengraft. Monocular Semantic Occupancy Grid Mapping with Convolutional Variational Auto-Encoders. *arXiv preprint, arXiv:1804.02176*, 2018.

[15] A. Krizhevsky. CIFAR-10 dataset. https://www.cs.toronto.edu/~kriz/cifar.html, 2009.

[16] M. R. Hvass Pedersen. CIFAR-10. https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/06_CIFAR-10.ipynb, 2016.

[17] Dosovitskiy A. Brox T. Riedmiller M. Springenberg, J.T. Striving for simplicity: The all convolutional net.

[18] S. Ruder. An overview of gradients descent optimization algorithms.

[19] Garcia E. He, H. Learning form imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9), September 2009.

[20] Obradovic Z. Vucetic, S. Classification on data with biased class distribution. *ECML 2001, LNAI 2167*.

[21] Bengio Y. Courville A. Goodfellow, I. *Deep Learning*. MIT Press, 2016. URL http://www.deeplearningbook.org.

[22] Kumar R. Rowen C. Hijazi, S. Using convolutional neural networks for image recognition. *Cadence Design Systems, Embedded Vision Alliance*, 2015.

[23] Grosse R. Ranganatch R. Y. Ng A. Lee, H. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th Annual International Conference on Machine Learning, ACM, New York*, pages 609–616, 2009.

[24] R. Benenson. Classification datasets results. http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html.

Appendix A. Convolutional Neural Network

Deep Neural Network is the umbrella name of neural networks with hidden layers. Examples are Recurrent Neural Network, Recursive Neural Network and Convolutional Neural Network. The last form is chosen for this research because of the efficiency of the network, less memory is required and the performance is better. Furthermore, Convolutional Neural Networks (CNN) are easier to tune and less time is needed to train the network. Finally, CNNs are especially successful in processing data which has a grid-structured topology. [21] [22]. Therefore, CNN is potentially well applicable for a soccer robot environment. An existing CNN for recognizing and classifying CIFAR-10 dataset is used, because good performance (79-80%) is achieved on this dataset. The CIFAR-10 dataset contains 60.000 32x32 colour images and is divided in 10 classifiers. This resolution is suitable for the data of the soccer robots.

A.1 Network structure

The first network used in this research is shown in Figure 10. It consist of two *Convolutional layers* (CL) and the output of these layers are down sampled by *Pooling layers* (PL). The output of the last pooling layer is connected to two *Fully-Connected layers* (FC) ending in a *Softmax layer* (SML). All the different layers has its own functionality. Table 12 gives an overview of all parameters which are adjustable in the layers.

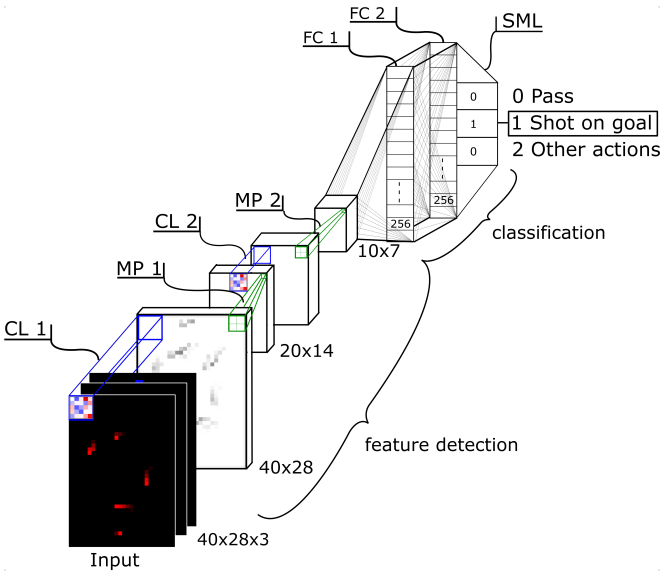

Fig. 10 Convolutional Neural Network Structure

A.2 Functionality of Neural Network components

The most important components of a Convolutional Neural Network are the convolutional, pooling and classification components;

Table 12 Adjustable parameters CNN.

Feature detection layers			
Convolutional layers (CL)			
Number of layers		2	[-]
Size of filters	FS	5 x 5	[pixel]
Number of filters	$\#F$	64	[-]
Zero-padding	ZP	✓	[-]
Stride	S	1	[pixel]
Activation		RELU	
Pooling layers			
Type		Max Pooling	
Size of filter	FS	2 x 2	[pixel]
Stride	S	2	[pixel]

Classification layers		
1st Fully-connected layer (FC)		
Vector size	256	
2nd Fully-connected layer		
Vector size	128	
2nd Softmax layer (SML)		
Vector size	3	

- *Convolution*; The Convolutional layer contains a number of certain sized filters. These are shifting with a step size along the input image matrix to search for features by calculating the dot product between the filter and input image. Figure 11 shows an image of 40x28 pixels with a filter of 5x5, which moves with a stride of 2 pixels along the image. As can be seen the filter doesn't come out exactly in the left image, which results in information loss around the edge of the image. To solve this problem a stroke of zeros will be added as demonstrated in the right picture. This is called zero-padding. The result of the dot product between the filters and the image is an *Activation Map* containing a feature, like edges, circles etc. More convolution layers can be coupled to find more complicated features [23], the input of the second layer is the activation map of the first layer
- *Pooling*; After extracting the activation map the image dimensions are reduced by spatial pooling. Comparable with the procedure of a Convolution layer, a matrix is moved along the image. The region which the matrix covers could be averaged, summed or the maximal value could be extracted. This value is stored in a smaller feature map. Some benefits of pooling is reduction of computations, less over-fitting, less sensitive for transformations, distortions and translations.
- *Classification*; The classification part consist of fully-connected layers and a Softmax layer. The Fully-

Connected layers ensures a relation between all the features of the last Convolutional layer and the Softmax layer. The Softmax layer produces the outcome of the neural network, which is a vector containing probability values for the different classifiers. These probability values together has a value of 1.

The Convolutional layers and Fully-Connected layers are build up out of neurons. The manner in which neurons are connected determines the property of these layers.

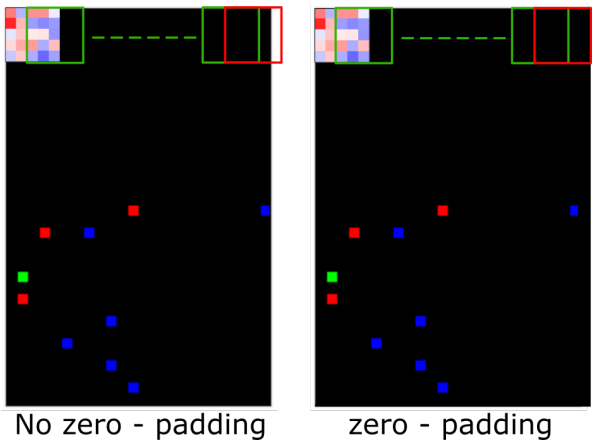

Fig. 11 A 40x28 pixel image in combination with a 5x5 filter with a stepsize of 2 pixels. The left image is without zero-padding and the right with zero-padding.

A.3 One single neuron

A Neural Network is build up from many neurons. These are connected to each other with certain weight factors, which are determined during the training process. A large dataset of input images is labeled with the corresponding output response and is used for training the neurons. The values of the input images are multiplied with weight factors. All connections linked to a neuron are added together including a bias value for an offset, which is also adjusted during training phase. Then, the summation goes through an activation function to introduce non-linearity in the system before it leaves the neuron. There are several types of activation functions which can be used; Sigmoid, Tanh or in the chosen network a Rectified Linear Unit (ReLU). The training process with ReLU functions is faster as Sigmoid or Tanh functions. The downside of ReLU functions is the possiblity to over-train the network which results in non-active neurons.

Appendix B. CNN Improvement

An existing CNN with only two convolutional layers is chosen for predicting actions of the soccer robots, this CNN will be compared with another CNN which will be chosen in this appendix. A benchmark is set-up for the CIFAR-10 dataset containing 32x32 images[24]. This size is comparable with the input representation of the world state of the soccer robots. Therefore, a network structure of the benchmark will be used as starting point which will be improved for the specific representation. CNN 1 in Table 14 is the second best performing CNN on the CIFAR-10 dataset[17]. Preference is given to this structure because of the few layers.

B.1 Experiment

Table 14 and Table 15 shows slightly different CNN structures. By comparing the performance of the CNNs on the same data-sets the most optimal CNN could be composed. Table 13 gives an overview of parameters which can be determined by comparing specific CNNs. The parameters of the Max-pooling layers haven't been changed because previous researches shows those settings are commonly used for CNN architectures.[12][17]

Table 13 CNN comparison

CNN *X* w.r.t. CNN *Y*	Parameter	Result
CNN 2 - CNN 3	Type of classification layers	1%
CNN 3 - CNN 4	Increase # filters per layer	-1%
CNN 2* - CNN 5	Increase in filter size	-4.6%
CNN 2* - CNN 7	Zero - padding last layer	-
CNN 6 - CNN 8	Extra pooling layer	6.4%
CNN 2* - CNN 6	Increase # convolutional layers	-5%

B.1.1 Data-sets

The action prediction data-set with only *Peer*-information with 20 frames from the past included is used for the CNNs of Table 14. This data-set is enlarged with passes from earlier years and with the remaining data from the last World championship for the CNNs of Table 15.

B.1.2 Results

The results of the CNNs are listed in the last 3 rows of Table 14 and Table 15. Table 13 third column shows the test accuracy difference between the compared CNNs. Figure 12 shows the training accuracy of the CNNs.

B.2 Conclusion

Several CNNs structures are compared with each other by testing the performance on the same dataset. Figure 12 gives an indication on which CNNs are still learning. CNN 3 and CNN 4 are after 2000 trained images already converged to 100%, the difference between those two and the other CNNs

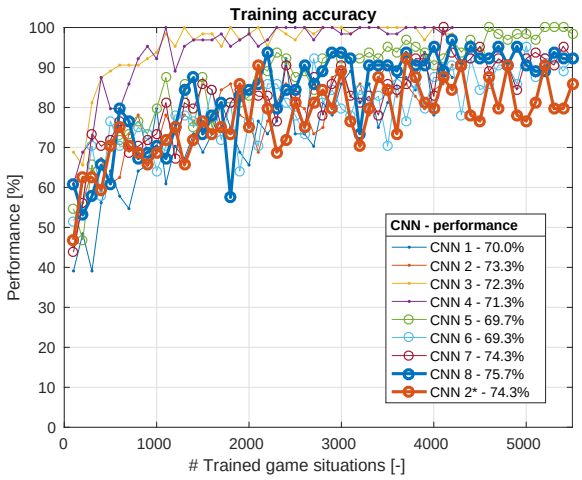

Fig. 12 Training accuracy CNNs

are the classification layers. In combination with the comparison of CNN 2 with CNN 3, as described in Table 13, it can be concluded that average pooling can be better used than fully-connected layers. Furthermore, the performance difference between CNN 3 and CNN 4 is only 1%, so more filters in a convolutional layer doesn't improve the recognition of more features, but increases the training time enormously. The size of a filter and number of convolutional layers has also influence on the training time and feature recognition; an increase in those two parameters doesn't improve the network in this case. This because the data-set is not large enough for learning higher-order features. For further research CNN 7 will be used, this because the training time isn't increased much with zero-padding comparing with CNN 2 and, the training accuracy converge slower to 100% as CNN 8 which CNN 2 has the capability to perform better with a larger training dataset.

Table 14 CNN structures and results on training data-set 1

	CNN 1				CNN 2	
		FS	*S*	*#F*		*FS*
Structure	*CL 1*	3 x 3	1	64	CL 1	3 x 3
	CL 2	3 x 3	1	64	CL 2	3 x 3
	[HTML]C0C0C0MP 1	[HTML]C0C0C03 x 3	[HTML]C0C0C02	[HTML]C0C0C0	[HTML]C0C0C0MP 1	[HTML]C0C0C03 x 3
	CL 3	3 x 3	1	64	CL 3	3 x 3
	CL 4	3 x 3	1	64	CL 4	3 x 3
	[HTML]C0C0C0MP 2	[HTML]C0C0C03 x 3	[HTML]C0C0C02	[HTML]C0C0C0	[HTML]C0C0C0MP 2	[HTML]C0C0C03 x 3
	CL 5	3 x 3	1	64	CL 5	3 x 3
	CL 6	1 x 1	1	64		
	CL 7	1 x 1	1	3		
	AP	7 x 7	1	No ZP	AP	7 x 7
	SML	3			SML	3
Input						28 x 28
Training time			44 min			43 min (58 mi
Test accuracy			70%			73.3% (74.3%

*Performance on data-set 2

Note; all layers use zero-padding (ZP) unless otherwise is indicated.

Table 15 CNN structures and results on training data-set 2

	CNN 5				CNN 6	
		FS	*S*	*#F*		*FS*
Structure	CL 1	5 x 5	1	64	CL 1	3 x 3
	CL 2	5 x 5	1	64	CL 2	3 x 3
	[HTML]C0C0C0MP 1	[HTML]C0C0C03 x 3	[HTML]C0C0C02	[HTML]C0C0C0	[HTML]C0C0C0MP 1	[HTML]C0C0C03 x 3
	CL 3	5 x 5	1	64	CL 3	3 x 3
	CL 4	5 x 5	1	64	CL 4	3 x 3
	[HTML]C0C0C0MP 2	[HTML]C0C0C03 x 3	[HTML]C0C0C02	[HTML]C0C0C0	[HTML]C0C0C0MP 2	[HTML]C0C0C03 x 3
	CL 5	5 x 5	1	64	CL 5	3 x 3
					CL 6	3 x 3
	AP	7 x 7	1	No ZP	AP	7 x 7
	SML	3			SML	3
Input			28 x 28			28 x 28
Training time			2h 10min			1h 1min
Test accuracy			69.7%			69.3%

Note; all layers use zero-padding unless otherwise is indicated.

Appendix C. Data generation

This appendix explains the selecting procedure of the classifiers from measurement data.

C.1 Filtering data for referee situation prediction

The Greenfield data contains information of the robots such as position, velocity, status of some hardware parts etc. but also the communicated information from the Referee box (refbox). For *validation of CNN and representation* the referee tasks are used in the data-sets. The frame which is selected for the data-set is taken at the moment when the robots are positioned to restart the game after a referee task is given.

C.2 Filtering data for action prediction

As mentioned before Greenfield data contains much information about the status of the match. The situations *shot on goal* and *pass* can be filtered by different methods. The paragraph below describes which method is used for selecting data for the data-sets.

- *Shot on goal*: The first step is to find the frame when refbox tasks; *goalkick* and *kickoff*, are given. These refbox situations are a result of a *shot on goal*. Next step is to select the moment when the ball is last possessed by a robot before a stop signal is given, corresponding with the mentioned refbox tasks. The frame is being checked on two points to ensure the selected data is a *shot on goal*. First criteria is if the robot is in a range of 0.5 meters from the ball, and second if the shot really is towards the goal by determine position dependent boundaries in which the ball moves. Figure 13 shows an example; The two boundaries start from the robot and crosses the intersection of the goalarea and the backline on both sides of the goal. The ball position should be higher than the left boundary but lower than the right boundary. Note that the conditions changes depending on the position on the field.
- *Pass*: Information about whether an opponent or peer is controlling the ball and in particular which robot is possessing the ball is stored in Greenfield data. This information is used to determine the passes. During a *pass* or *intercept* the ball possession changes between the robots. In case of a *pass* there is an ID change between two opponents or two peers. If a change occurs between opponent and peer than an *intercept* took place.
- *Other actions*: The third classifier contains different types of situations; *scrum, intercept, dribble*. These situations are selected randomly but during gameplay and the filtered frames for *shot on goal* and *pass* are eliminated.

C.3 Data augmentation

A large amount of data is needed to achieve proper output of neural networks [21]. To avoid over-fitting, the training data will be mirrored on the horizontal and vertical axis for all experiments. Furthermore, the training and test-set is evenly distributed over the classifiers, this because researches shows that balanced data-sets perform better. [19][20] Therefore, the amount of data for the training-set with referee situations is increased by including extra frames before the start signal. This is the situation when robots are almost positioned for the refbox moment, but gives a small variation of the robots location in comparison with the unique image. Table 16 second column shows the amount of unique refbox situations and the third column shows the extra frames per unique image. Table 17 column 2 lists the number of available data for action prediction. Hereby, the classifier *shot on goal* is increased by using the view of different robots. Every robot observes the environment and capture the positions of the objects on the field, which gives small variation in data.

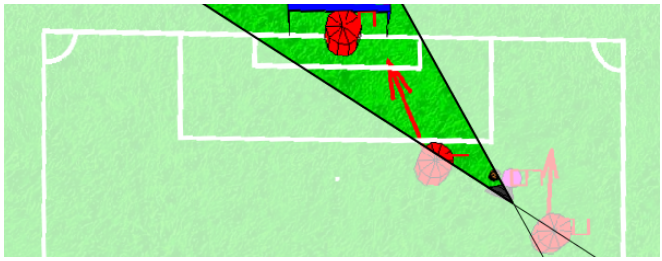

Fig. 13 Shown situation is a *shot on goal* during a match. Black lines are boundaries in which the ball should stay inside.

Table 16 Number of unique refbox situations

Refbox type	Unique	Extra frames
Kickoff	831	1
Throw In	1181	0
Freekick	869	1
Corner	259	5
Penalty	6	9
Goalkick	406	2
Dropped ball	66	9
Total	3618	

Table 17 Number of actions

		Training		Test	
		Peer	Opponent	Peer	Opponent
Pass		1449	2396	358	432
Shot on goal	Unique	877	33	111	7
	Extra	1625	7		
Other actions		5423		1053	

Technische Universiteit
Eindhoven
University of Technology

/ Department of
Mechanical Engineering

P.O. Box 513
5600 MB Eindhoven
The Netherlands