Survey of neural networks in autonomous driving

Article · July 2017

CITATIONS
0

READS
4,461

1 author:

Gustav von Zitzewitz
Technische Universität München
2 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Survey of neural networks in autonomous driving

Gustav von Zitzewitz

***Abstract*—For the last five years neural networks are in the center of attention when talking about the reachability of autonomous driving. After briefly introducing the theoretical background of deep neural networks this report gives an overview of current applications using state-of-the-art system architectures like recurrent and convolutional neural networks. Furthermore, it examines the advantages and hindrances that are connected to the use of deep learning. It raises and tries to answer the questions that a vehicle needs to be able to answer to achieve autonomy: *Where am I? Where is everybody else? How do I get from A to B? What is the driver up to?* Finally, it discusses the state of research and gives a hint about future possibilities for further improvement.**

I. Introduction to Deep Neural Networks

DEEP neural networks (DNN) are at the moment the field of research in data science where the most effort and as a result progress is made. Every year new network architectures and improvements to existing systems are presented by a wide range of scientists all over the globe. Big tech companies such as Google and Facebook as well as carmakers such as BMW and Tesla invest billions in the research of deep learning, which pushes the development to not yet seen levels. As the objective of this report is to survey neural networks in autonomous driving, the report is structured as follows: First there is a brief introduction on the topic of DNNs. Therefore the main characteristics as well as the most important network architectures for autonomous driving approaches, recurrent neural networks (RNN) and convolutional neural networks (CNN) are introduced. Moreover, different learning techniques like supervised, unsupervised and reinforcement learning and various types of fundamental tasks for DNNs, such as classification, regression, object detection and segmentation are presented. Since there are plenty of available sources, such as Goodfellow *et al.* [1] concerning deep learning (DL) and neural nets (NN), the mathematical basics are not discussed because it would exceed the scope of this report and as they can be quickly acquired if necessary. Second current applications including localization and mapping, scene understanding, movement planning and driver state are presented. Third advantages and hindrances of DL are weighted against each other to finally discuss the results, followed by a possible outlook for the future.

A. Characteristics

Neural networks contain three types of layers: Input, hidden and output, each consisting of one to many nodes whereas each node is represented by a set of weights and one bias. This leads to the question, what makes a NN deep? It is the fact that multiple hidden layers are connected between the input and output layer whereas several mathematical functions are combined in each of them. The essential part of each hidden layer is the activation function to introduce non-linearity into the network, which makes it able to solve non-linear equations. DL works with increasing the complexity of those combined functions with layer depth. After the information is passed through the net it is most often reassembled. This can be done for example with fully connected layers to feed an N-way softmax function that transfers weights to possibilities of N possible outcomes. In contrast to that, DNNs can also be trained in an end-to-end fashion where the input is maped directly to a control signal as output. Finally, there are two different states of a network: First the training process, where a calculated error is minimized by backpropagation, where the weights and bias are adjusted according to the negative gradient, in combination with various optimization possibilitest like adaptive learningrate or momentum [2], [3]. Second inference, where training is completed and the network operates on it's specific task on unseen data.

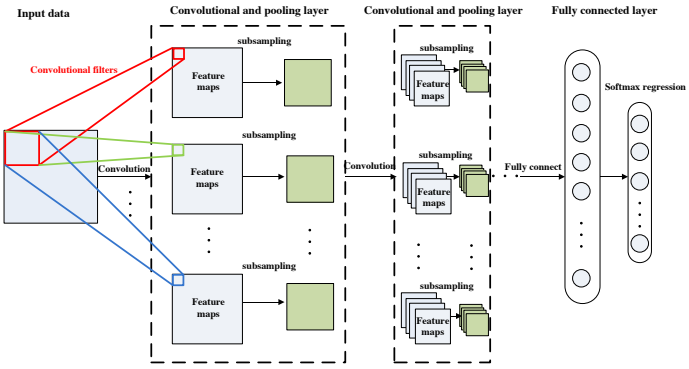


Figure 1. Exemplary architecture of a CNN with multiple hidden layers [4]: Next to the Input there are several layers containing convolutional and pooling stages. After the final fully connected layer the softmax regression is computed as output. The activation function stages are not explicitly shown in the picture but they are applied after each convolutional stage.

B. Architecture

There are multiple architectures of DNNs [5], [6] that can handle all kind of input data for different tasks, but they are basically separable in two main types: Feedforward (FFNN) and recurrent neural networks.

1) Convolutional Neural Networks: For computer vision tasks the current benchmark are CNNs. They typically contain three stages in most hidden layers as shown in figure 1. Next to the activation functions, which are most likely rectified linear units (ReLU) [7], convolutional filter stages are applied

Authors: Gustav von Zitzewitz (3636797, Gustav.Zitzewitz@tum.de), **Course**: Advanced Seminar Summer Semester 2017 **Submitted**: July 13, 2017 **Supervisor**: Dipl.-Math. Florian Mirusa. Neuroscientific System Theory (Prof. Dr. Jörg Conradt), Technische Universität München, Arcisstraße 21, 80333 München, Germany.

to recognize local connections of features from previous layers and to store them in a map that serves as input for the next layer [2]. The third stage is pooling [8], which is a form of non-linear down sampling. It is a regularization technique used to merge semantically similar features into one to reduce the dimensions of the calculated matrices to finally increase computation speed. Furthermore, it creates invariance to shifts and distortion. For example 2x2 maxpooling takes a quadratic four pixel window and decreases it to one, where the new pixel value is the maximum of the previous used.

2) Recurrent Neural Networks: In contrast to FFNs where there are no loops at all in the graph, single nodes of RNNs can be connected to themselves or previous nodes [2]. RNNs are used for processing sequential data. They can be able to grow in width/unfold with each time step and store memory about previous states, which makes them very dynamic but problematic to train due to vanishing/exploding gradients. In combination with long short term memory (LSTM) modules, that are able to learn long term dependencies, RNNs are capable to make predictions about future events. Therefore they are the current benchmark for planning and prediction tasks. Olah [9] gives a very detailed explanation about how LSTMs work.

C. Learning

There are in particular three different main learning techniques: Supervised, unsupervised and reinforcement learning. In supervised learning the network is trained with labelled data to be finally able to predict the label on unknown test data. In unsupervised learning the training data is not labelled at all and the network learns to cluster the data in different groups to distinguish for example a car from a pedestrian. In reinforcement learning the network is rewarded when taking actions that lead to states with a high revenue, therefore it learns a policy of actions to maximize the reward in each situation [10]. Next to those three basic forms of learning there are multiple other techniques: In Multimodal Deep Learning networks are trained with more than one input data type, like audio plus video [11]. Transfer learning is used to decrease the amount of necessary training data and to be able to build on already developed networks to decrease time necessary for training. For example by taking a pre-trained CNN and freezing the weights of the lower hidden layers as they contain more simple feature detectors like edges or colors, which nearly every fundamental network task needs. Training is then only done with the high level layers on a particular data set [12]. Finally, there are also ways to artificially expand the training data and to make the network robust against changes to the input data while inference. Data augmentation slightly manipulates the training data, like rotations and shifts if used on images, while the label is kept the same.

D. Tasks

There are various tasks that can be solved by DNNs that are useful for autonomous driving, but the four fundamental tasks are: Classification, detection, segmentation and regression. Other more advanced tasks like scene understanding or path planning build up on those basic four. Classification networks [13], [14] identify and categorize objects. A vision classifier network for example categorizes objects in a picture frame. Networks with detection tasks [5], [15], [16] in contrast are able to recognize and mark certain objects in a frame. Networks with segmentation tasks [17] partition pictures into sets of pixels (segments) to locate boundaries of objects. For this task special CNNs with Encoder-Decoder architectures are usable [18]. Finally regression tasks are often solved in the last layer of a network to map a continuous inputs to continuous outputs.

This section's topic was to give a short introduction into the wide area of neural networks and deep learning. It provided a selection of the latest references for the interested reader to dig deeper. In the following section some state-of-the-art applications for autonomous driving are presented.

II. Current Applications of Deep Learning in Autonomous Driving

There are in general four questions a car needs to be able to answer to achieve the final goal of autonomy as described by Friedman *et al.* [3] in their MIT course about deep learning in autonomous driving.

1) Where am I? → *Localization and Mapping*
2) Where is everybody else? → *Scene Understanding*
3) How do I get from A to B? → *Movement Planning*
4) What's the driver up to? → *Driver State*

Answering those questions can be realized in two different ways. One way is via semantic abstraction where each task is executed in a seperate network and afterwards combined with classical control & decision-making algorithms [19]. The other approach is called end-to-end, where a single DNN takes all the car's inputs and computes a final control command as output. It is important to notice that some applications can not be assigned to only one specific task. Therefore some of the following applications overlap in their topics.

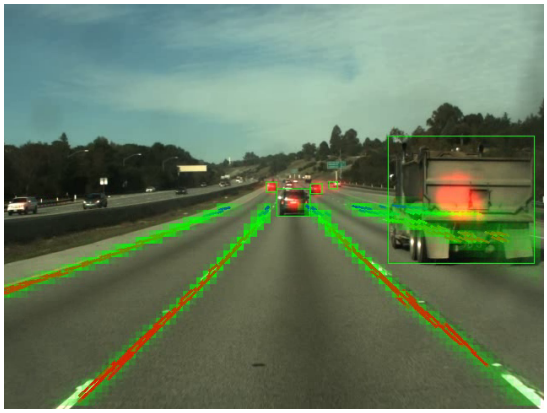


Figure 2. Sample output of Huval's *et al.* [20] CNN, capable of lane prediction and vehicle detection including bounding boxes to localize them in each picture frame. Green color indicates where the network is trained to fire and red color shows the network's output in inference

A. Detection and Classification

One of the first autonomous driving tasks mastered by DNNs was traffic sign recognition. In fact CNNs are since 2012 better than humans on recognizing street signs with an accuracy of $99,46\%$ [13]. Related topics like line, traffic light and vehicle detection have accuracies on a similar level when applied on state-of-art CNN architectures as seen in figure 2. An example of a state-of-the-art CNN for detection and localization tasks, developed by Redmond and Farhadi [21] is YOLO Darknet v2. It can detect more than 9000 Objects in real-time at 40-70 fps with a mean accuracy of nearly 80%, which makes it capable of detecting everything necessary for automotive tasks in a video or an onboard-camera.

B. Scene Understanding

Semantic segmentation is an technique used for road scene understanding. Badrinarayanan *et al.* [18] use a special CNN encoder-decoder architecture as explained in figure 3. After the input image is processed through the network a pixel wise classification is computed to identify each pixel to the belonging object. It achieves a prediction accuracy of around 88% for cars and 96% for roads. Although it struggles with pedestrians, the achieved accuracy of 62% still outperforms all other tested algorithmic methods by over 10%.

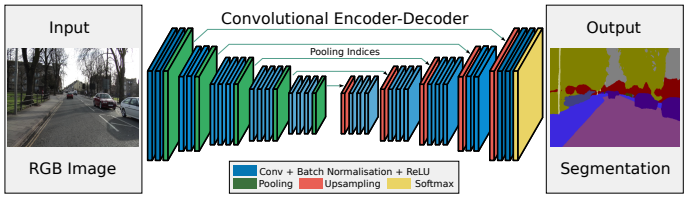
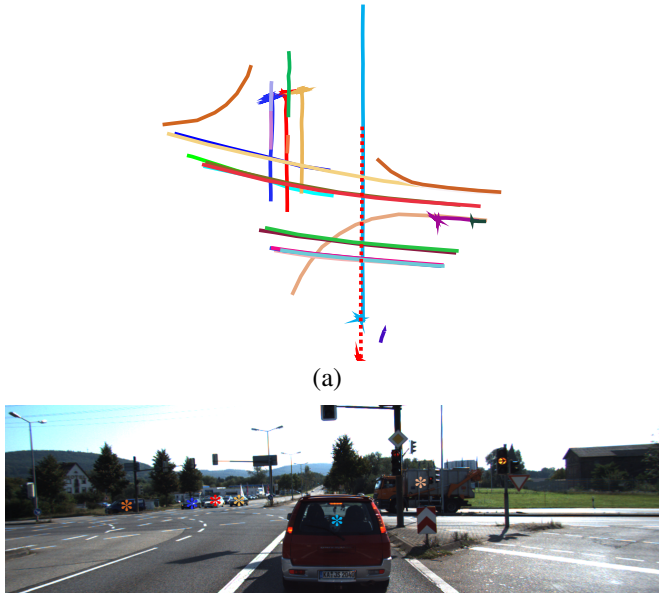


Figure 3. Badrinarayanan's *et al.* [18] SegNet architecture: The Encoder down-samples the input with 2x2 maxpooling stages while the pooling indices are stored for later up-sampling.The features are extracted through convolutional stages combined with batch normalization and ReLUs as activation functions. The decoder up-samples the features again before they are fed into a multi-class softmax regression layer for pixel wise classification.

Surround Vehicle Trajectory Analysis (SVTA) is using Long Short Term Memory (LSTM) in RNNs aswell as 3D trajectory cues. LSTM approaches are the state-of-the-art systems for speech recognition and real-time speech translation. They are capable of learning long term dependencies like remembering how a sentence started right before it is finished to get the context right. The same problem is faced when future predictions want to be made about what other road users are up to do. In a recent paper Khosroshahi *et al.* [22] used the KITTI benchmark, which is a cooperation between Karlsruhe Institut of Technology and Toyota [24], including their camera, GPS and lidar sensors to train their SVTA framework. The sensor signals are fed into a RNN-LSTM network to predict the trajectories of surrounding vehicles as shown in figure 4. The paper concludes that their system was able to make good predictions for coarse labels such as turning versus going straight but predicting a finer activity label space with more output options was problematic.

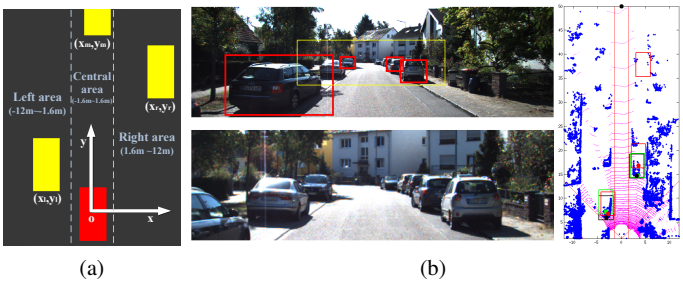


(a) (b)

Figure 4. Khosroshahi's *et al.* [22] SVTA framework: (a) It maps the surrounding vehicle trajectories gathered by the KITTI dataset on the road surface while the dotted line refers to the ego car's path. (b) Shows the corresponding image of the studied four way intersection.

C. Localization and Mapping

Using the camera signal to get accurate bounding box locations around pixels of detected objects also the distance and relative speed is obtainable by matching with the radar signal [20]. Besides 2D, also 3D object detection is possible from single monocular images with the assumption made from Xiaozhi Chen *et al.* [15] that objects recognized by the vehicle's sensors should be on the ground plane (zero height). Chenyi Chen *et al.* [23] used this assumption to estimate car distances as explained in figure 5. Like the SVTA sytsm, the camera and lidar signals of the KITTI dataset served as input. For this approach a two CCN system was used. One for close range (2-25m) and one for far range (15-55m) object detection due to the low resolution of the ipnput images. For the final distance projection the output of both CNNs are combined.



(a) (b)

Figure 5. Chenyi Chen's *et al* [23] distance estimation concept: (a) The ground plane is divided into three areas. With The coordinate system defined relative to the car at zero height the objective is to estimate the coordinates of the closest car in each area. (b) The bounding boxes output of two different perception approaches (red and green) are compared, whereas the central crop (yellow box) is sent to the far range CNN. Finally, the distance projections are displayed in the lidar visualisation where the black boxes correspond to the ground truth.

D. Movement Planning

Another Application is movement planning on small scales like finding a way around obstacles using short range sensors like camera, lidar, sonar and radar and navigation on the bigger scale with long range sensors like GPS where finding the fastest or most efficient route is important. Huang *et al.* [25] developed a framework visual path prediction. It consists of two CNNs that separately model the spatial and temporal context as shown in figure 6.

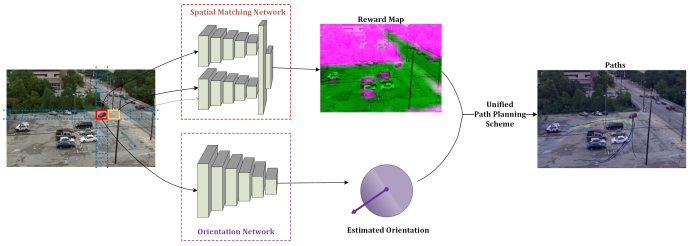


Figure 6. Overview of Huang's *et al.* [25] framework. First the object of interest is cropped out with a bounding box (red). The rest of the image is then divided into local environment patches of the same size as the object (dotted blue). The spatial matching network gets fed with the object plus an environment patch (yellow) at each time to generate a reward map of the scene. The orientation networks estimates the object's facing orientation to indicate the object's preferred movement direction for future steps. They both serve as input for the unified path planning scheme. Finally, the path predictions are displayed on the input image

For training 8.5 hours of the VIRAT Video Dataset containing 11 different outdoor scenes and for testing the smaller KIT AIS dataset consisting of 9 different outdoor scenes was used. The results were compared to state-of-the-art nearest neighbour and mid-level element algorithmic approaches, which both were outperformed on all scenes by accuracies around 30%. Drive.ai [26] let's their small fleet of four autonomous Audis even take one further step. Their cars do decision making and motion planing on difficult situations like the American four way stop, where the first come first serve rule is applied, or even turning on red, which is allowed in most intersections. The small tech-startup claims to be able to build level 4 autonomous systems in the next years, which means a self-driving system that doesn't require human intervention in most scenarios.

E. Driver State

The Driver State surveillance task is to predict the driver's future actions. Jain's *et al.* [27] developed a system called Driver activity anticipation (DAA) which uses a similar network architecture as SVTA. The system fuses the sensor inputs of a cockpit camera that tracks the driver's face movement and the sensors that understand the outside context like GPS and outside cameras. In order to anticipate upcoming manoeuvres the Network learns to predict the future driver manoeuvres given only a partial temporal context. This is possible through the special RNN structure with LSTM units, that are trained to map all sequences of partial observations to the future event as shown in figure 7. With this system a precision of 90.5% and a recall of 87.4% is achievable on a training set of 1180 miles natural driving.

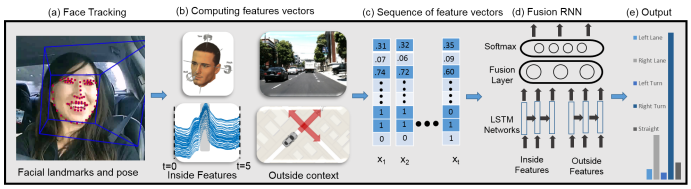


Figure 7. Jain's *et al.* [27] system architecture for driver activity anticipation: First the face is tracked (a) in the next step the feature vectors for inside and outside context are computed (b), which are then fed into the RNN (c+d) to finally get the manoeuvre prediction (e).

F. End-to-End

The previously mentioned End-to-End approach has recently received a lot of attention as Bojarski *et al.* of Nvidia [28], [29] introduced their later called PilotNet system. PilotNet is a CNN consisting of nine layers in total including a normalization layer, five convolutional layers and three fully connected layers. The whole system architecture for training is shown in figure 8. Finally the trained system is able to drive on the road with just one centred camera as input signal.

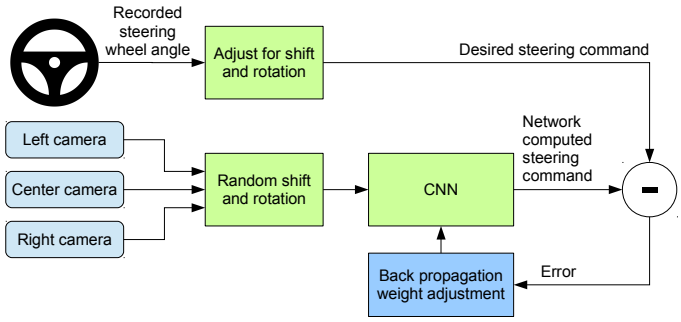


Figure 8. Nvidia's [28] end-to-end training set-up: Training data is collected via the video signals of three behind the wind shield mounted cameras combined with the steering wheel angle received by the car's CAN-Bus. The CNN computation happens in a Nvidia Drive Px, which is the predecessor model of the already mentioned GPU system. The output is a steering command $1/r$, where r is the turning radius in meters, which is constantly compared to the real steering angle to adjust the weights via backpropagation. An important part of their training process was the already mentioned data augmentation technique because training with data only from an human driver is not sufficient. They randomly altered images in the training process where the car was shifted or rotated on the road.

The most interesting part is that the car needs no other DL system or classical control unit for autonomy, everything happens inside PilotNet. It learns how to steer the vehicle by being trained on the data collected from a human pilot drive along different roads for 72 hours. If looked at the activation maps of different layers inside PilotNet as shown in figure 9 it becomes clear that it learned to detect useful road features like road lines without ever been taught what a road looks like. Besides highway driving it also manages to steer safely on off-road ways or through cones on parking lots, as it recognizes the outlines of what are meant to be roads in completely different scenarios.

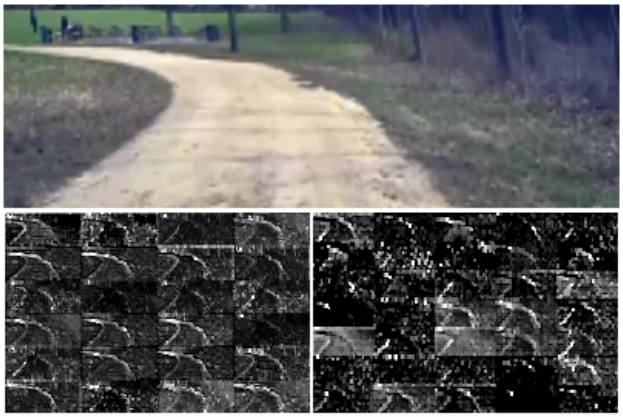


Figure 9. PilotNet's [28] first and second layer outputs for an input image of an off-road pathway. The activation maps recognize the road's features, visible by the white lines standing out of the black background.

This section took a glance at recent research projects of neural networks and deep learning in the field of autonomous driving. Before the discussion to what extend the four given questions can already be answered, the next section takes a closer look at the strengths and weaknesses of deep learning to be finally able to give a outlook about future possibilities.

III. Advantages and hindrances in the use of Deep Learning

Deep Learning has several convincing advantages over classical algorithmic or robotical control approaches. Nevertheless, it also has to overcome quite a few significant technical hurdles. In this section the advantages and hindrances of DL are examined.

A. The Bright Side

1) Improvement through Learning: One major argument why deep learning is the driving force to achieve autonomous driving is because it makes machines steadily improve through learning. Andrew Ng, chief scientist of Baidu [30] claims DNNs keep improving with the amount of data available for learning where other machine learning algorithms already reach their limit as shown in figure 10. Moreover, Reinforcement learning enables learning from experience instead of having programmed every step and turn by man. In contrast to other methodologies, deep learning does not need hand crafted features. It learns simple features in the lower shallow layers as well as complex ones with increasing layer depth on its own. Learning in this case means that the DNN is fed huge amounts of input data to gradually improve the output prediction by minimizing the cost function through iterativley adjustment of the Nets' weights in the hidden layers. DL reacts better in unseen scenarios than rule-based approaches as it can adapt into complex situations where classical rule based control fails [31]. If a difficult situation is encountered, the data is collected and the deep learning brain is trained on it so it knows how to react or decide the next time a similar situation occurs [26].

Figure 10. Andrew Ng's [30] graph shows how deep learning is said to outperforms traditional learning algorithms with increasing amount of available training data.

2) Computation Power: Although NNs have been known for decades, their development stuck until ten years ago as hardware was not advanced enough to collect, store and process the massive amount of data necessary to properly train DNNs. This changed with the availability of powerful Graphic Processing Units (GPU), which perform many calculations at once aswell as data storages in huge sizes at very low cost. They are able to shorten the time necessary for training a big network from days or weeks down to hours, hence the computing task is no barrier any more claims Shapiro, Senior Director of Automotive at Nvidia [32], [33]. Nvidia is one of the leading developers for GPUs that are mastering the complex computation tasks while training and enabling real-time decision making while driving. They developed an all-in-one system especially for online computation in autonomous cars named Nvidia Drive Px 2 [34]. With each two mobile procesessors and GPUs It can operate up to 24 trillion mathematical operations per secound and can be stacked parallel for further performance increase. As confirmed by Nvidea [35] Tesla is using the Drive Px 2 System for its autopilot systems in all three models since late 2016 as it has 40 times more computing power than their old system.

3) Knowledge and Software Availability: Setting up a theoretical complex DNN, like a CNN described in the introduction, is easy to handle with now available resources. With open source programming libraries such as Tensorflow of Google or Caffe of Berkley University it is possible to build up and train DNNs as well as to validate the results afterwards with minimum programming knowledge. The open source documentation about DL via blogs, tutorials or youtube videos is massively increasing since 2015 as it is a field of research that attracts a lot of young data scientists. Compared to classical rule based decision making just a few lines of code and thereby expert knowledge is necessary to get started working on the topic.

4) Performance: Finally, the most important reason why DL is the future of autonomous driving is performance. DNNs had their breakthrough around 2012 with AlexNet winning the ImageNet competition, which is basically the "annual olympics of computer vision", by dropping the actual best error rate record

from 26% to 16% [14]. This led to dropping classification errors every year through various systems as shown in figure 11. Since then DNNs, in particular CNNs outperform any other form of machine learning approach by orders of magnitude especially at Computer Vision applications such as object detection and classification.

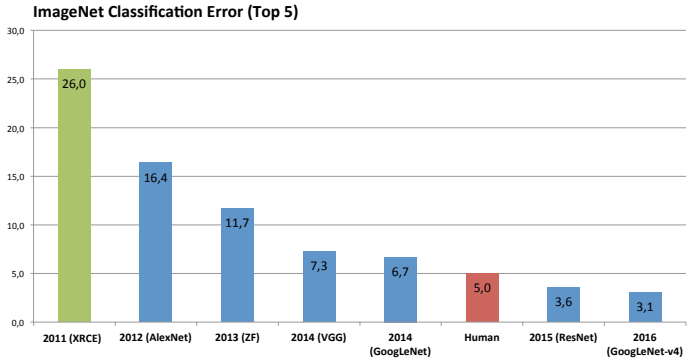


Figure 11. Winner results of the ImageNet large scale visual recognition challenge (LSVRC) of the past years on the top-5 classification task: The green bar indicates the best computer vision approach, whereas the blue bars are all deep neural network architectures. The human score is represented as the red bar. (The values for the diagram are compiled from the image-net homepage [36]).

B. The Dark Side

1) Labelled Data: Most DNNs are trained with supervised learning that requires massive amounts of labelled training and testing data. Having enough labeled data is important, as DNNS tend to overfit. An indication for overfitting, due to too little training data, is when the error on the training dataset is already small but the error on an unseen test set is still high. To overcome this problem companies employ thousands of people to label their data. It is approximately 800 hours of work labelling collected driving footage to recieve one hour of usable data [26] as every video frame is basically one image, which makes $108,000$ images to be labelled at 30 frames per secound (fps). Therefore Amazon hosted a platform called Mechanical Turk [37]. It is a service to outsource the labelling of data to companies and schools in low wage countries. Although it provides reasonable results, supporting it is questionable.

2) Sensor Fusion: Another big challenge is sensor fusion as it is especially difficult to filter unneeded information and to fuse the different kind of data provided by sensors working on an autonomous car. Possible sensor signals include cameras, lidar, radar, GPS, ultrasonic, Infra-red and audio, moreover driver signals such as steering angle, brake- and acceleration pedal position and finally vehicle dynamic signals such as speed, yaw rate, damper force and wheel slip. To refine this wast amount of information collected by sensors to make them suitable for training is one major hindrance to autonomous driving because it still needs lots of man working power.

3) Blackbox Problem: DNNs are often seen as Blackboxes [26], this is especially problematic in end-to-end realizations. The fact that it is not really visible what happens inside the hidden layers between in- and output and how it finally gets to its solution is the reason a lot of companies hesistate to use DNNs. Furthermore, troubleshooting and debugging can sometimes be a hard task the deeper a network gets. Especially in applications concerning the security of human life it is absolutely necessary to know exactly what the system does. It is one major hindrance to first collect enough training data of so called ”corner cases”, situations in autonomous driving that are extremely rare but dangerous at the same time, and later validate the behaviour of the network in those cases [19].

4) Manipulability: Although DNNs perform robust in various conditions, they are not completely secure against manipulation. In 2013, a Group of Scientists from Google, Facebook and some major US Universities published a paper where they were able to spoof at that time state-of-the-art DNNs such as AlexNet with putting filters or minimal distortions over test pictures that a human eye can’t even recognize but led the DNNs to drop their accuracy by a factor of ten. This Problem is not limited to camera signals, also other sensors like Lidar are spoofable with similar tricks [38]. To overcome this problem, techniques like the previously mentioned data augmentation or batch normalization, introduced by Ioffe *et al.* [39] to reduce covariate shift, are used.

5) Safety Validation: The final challenge is to make security and safety validation of DNNs in autonomous driving meet auto industy standards as they calculate stochastic results that are likely to make false positive and false negative decisions [19]. The human level safety is said to be near a mean time between failure of 10^9 hours. It would take a fleet of million cars that operates one hour a day thousand days of testing to see one catastrophic failure. In fact, you must repeat testing several times to achieve statistically signifcant statements. Furthermore, it is impossible to test every corner situation that may occur due to exponentially exploding combination possibilities of different kind of impacts such as sensor or system failure, weather conditions, unusual behaviour of other road users and many more. To cope with these blind spots in the safety validation of DNNs a relatively mature testing technology could be fault injection. The reason why very high Automotive Safety Integrity Levels (ASIL) [40] must be achieved with full autonomy is that there might be critical situations where the driver does not have the ability to take corrections, instead the computer system must handle any fault or risk occurance. An approach to lessen this Problem is to use ASIL-decomposition where a monitor/actuator architecture is used. The monitor is designed with high level ASIL while the actuator can work on a lower level. In combination with heterogeneous redundancy the ASIL level is further increased and the system is able to do a saftey fail-stop if necessary.

This section examined the advantages and hindrances

connected to the use of deep learning. It showed that former hindrances like computation power are already solved, whereas recent hurdles like safety validation are already in the progress. In the subsequent sections the main results of this report are summarized and discussed.

IV. Discussion and Future Outlook

This final section contains the discussion about how far research actually is and which approaches seem to be most promising. This is followed by a brief future outlook about interesting methodologies before the eventual conclusion.

A. Discussion

Summarizing the presented applications it is safe to say that besides object detection as well as classification and localization applications using CNNs, which already perform on series or close to series level, most Deep Learning applications for autonomous driving still need further research. This also applies to applications which already work well on their tested scenarios like Nvidia's End-to-End approach. But validating those systems for auto industry standards is a completely different story as they face the blackbox problem. A possibility to get these applications closer to series would be if they are not supposed to work on full autonomy but customized to fulfil lower level driver assistance tasks where the driver is still kept in charge. The most promising research fields are thereby recurrent neural networks in combination with long short term memory approaches as they are able to learn long term dependencies to predict future actions, which is substantial to achieve autonomous driving. Eventually, being able to handle networks that use sensor fusion is the key to achieve high safety levels, for example to use radar signal to detect false negatives in object detection by the camera system. Finally, state-of-the-art methods of deep learning are at least partially able to answer the previous raised questions concerning localization, mapping, scene understanding, movement planing and driver state.

B. Future Outlook

While gathering information for this report the author came across some interesting methodological approaches where unfortunately not enough documentation was available. As most networks are trained on supervised learning a promising strategy could be to use unsupervised learning to pre-label training data to save huge amount of time as only the label validation would remain as a human task. Furthermore, the combination of CNNs and RNNs that use reinforcement learning to create networks that are able to do high performance vision based motion planning tasks. Another application is to survey the driver's mental condition to timely recognize if he faces health problems or is likely to fall asleep to be able to give warn signals or to take over control. Moreover, simulating training data especially for corner cases is a promising approach as the validation of Deep Learning systems is still a not yet reached goal. Finally, a drive style to active chassis application could be interesting, where a Deep Learning system is trained to recognize different driving styles, to be able to adapt the car's dynamics behaviour if the driver changes between smooth or sporty driving to maximize driving pleasure as well as safety and comfort.

V. Conclusion

This report gave a brief introduction into the wide area of deep learning. Convolutional and recurrent neural networks have been presented as the benchmark for their respective application fields, object detection and long term planning. Furthermore, current state-of-the-art applications including amongst others, End-to-End approaches, driver state and scene understanding have been illuminated. Moreover, the advantages and hindrances connected to the use of deep learning have been examined. With the result that many hurdles, like safety validation or the necessary amount of training data are already in the progress of being solved. Eventually the decisive argument why deep neural networks will prevail is their high performance enhanced due to the availability of powerful computation components. Ultimately, should not be underestimated that even if research is making great progress, there is still a long way to achieve autonomous driving.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] L. Fridman, "Learning to drive: Convolutional neural networks and end-to-end learning of the full driving tasks", MIT, Course 6.S094, Tech. Rep., 2017, lecture 3.

[4] L. Jing, T. Wang, M. Zhao, and P. Wang, "An adaptive multi-sensor data fusion method based on deep convolutional neural networks for fault diagnosis of planetary gearbox", *Sensors*, vol. 17, no. 2, p. 414, 2017.

[5] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, "Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving", *ArXiv preprint arXiv:1612.01051*, 2016.

[6] J. Morton, T. A. Wheeler, and M. J. Kochenderfer, "Analysis of recurrent neural networks for probabilistic modeling of driver behavior", *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1289–1298, 2017.

[7] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines", in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[8] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks", *ArXiv preprint arXiv:1301.3557*, 2013.

[9] (26-06-2017). Understanding lstm networks, [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[10] L. Fridman, "Learning to move: Deep reinforcement learning for motion planning", MIT, Course 6.S094, Tech. Rep., 2017, lecture 2.

[11] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning", in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?", in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[13] D. CireşAn, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification", *Neural Networks*, vol. 32, pp. 333–338, 2012.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[15] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2147–2156.

[16] B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection", in *Communication Software and Networks (ICCSN), 2016 8th IEEE International Conference on*, IEEE, 2016, pp. 581–585.

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation", *ArXiv preprint arXiv:1511.00561*, 2015.

[19] T. Dubner, P. Edin, B. Lakshman, and Y. Suganuma, "I see. i think. i drive. (i learn)", *KPMG*, 2016.

[20] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, *et al.*, "An empirical evaluation of deep learning on highway driving", *ArXiv preprint arXiv:1504.01716*, 2015.

[21] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger", *ArXiv preprint arXiv:1612.08242*, 2016.

[22] A. Khosroshahi, E. Ohn-Bar, and M. M. Trivedi, "Surround vehicles trajectory analysis with recurrent neural networks", in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, IEEE, 2016, pp. 2267–2272.

[23] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2722–2730.

[24] (26-06-2017). Welcome to the kitti vision benchmark suite!, [Online]. Available: http://www.cvlibs.net/datasets/kitti/.

[25] S. Huang, X. Li, Z. Zhang, Z. He, F. Wu, W. Liu, J. Tang, and Y. Zhuang, "Deep learning driven visual path prediction from a single image", *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5892–5904, 2016.

[26] E. Ackermann, "How drive.ai is mastering autonomous driving with deep learning", *IEEE Spectrum*, 2017.

[27] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture", in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, IEEE, 2016, pp. 3118–3125.

[28] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars", *ArXiv preprint arXiv:1604.07316*, 2016.

[29] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car", *ArXiv preprint arXiv:1704.07911*, 2017.

[30] (25-06-2017). Andrew ng, chief scientist of baidu, [Online]. Available: http://www.andrewng.org.

[31] M. Copeland. (24-06-2017). What's the difference between artificial intelligence, machine learning, and deep learning?, [Online]. Available: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/.

[32] B. Lewis. (24-06-2017). Educating the auto with deep learning for active safety systems, [Online]. Available: http://embedded-computing.com/articles/educating-the-auto-with-deep-learning-for-active-safety-systems/.

[33] D. Shapiro. (24-06-2017). Here's how deep learning will accelerate self-driving cars, [Online]. Available: https://blogs.nvidia.com/blog/2015/02/24/deep-learning-drive/.

[34] (25-06-2017). Neueste version der nvidia drive px 2, [Online]. Available: http://www.nvidia.de/object/drive-px-de.html.

[35] (25-06-2017). Tesla and nvidia, [Online]. Available: http://www.nvidia.com/object/tesla-and-nvidia.html.

[36] (28-06-2017). Imagenet lsvrc challenge, [Online]. Available: http://image-net.org/challenges/LSVRC.

[37] (24-06-2017). Amazon mechanical turk, [Online]. Available: https://www.mturk.com/mturk/welcome.

[38] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks", *ArXiv preprint arXiv:1312.6199*, 2013.

[39] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *International Conference on Machine Learning*, 2015, pp. 448–456.

[40] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation", *SAE International Journal of Transportation Safety*, vol. 4, no. 2016-01-0128, pp. 15–24, 2016.