Please cite this paper as:
@inproceedings{nitsch2019object,
title = ”Object Classification Based on Unsupervised Learned Multi-Modal Features For Overcoming Sensor Failures”,
author = ”Nitsch, Julia and Nieto, Juan and Schmitd, Max and Siegwart, Roland and Cadena, Cesar”,
booktitle = ”2019 IEEE International Conference on Robotics and Automation ({ICRA})”,
year = 2019;
}

Object Classification Based on Unsupervised Learned Multi-Modal Features For Overcoming Sensor Failures

Julia Nitsch[1,2], Juan Nieto[2], Roland Siegwart[2], Max Schmidt[1] and Cesar Cadena[2]

Abstract— **For autonomous driving applications it is critical to know which type of road users and road side infrastructure are present to plan driving manoeuvres accordingly. Therefore autonomous cars are equipped with different sensor modalities to robustly perceive its environment. However, for classification modules based on machine learning techniques it is challenging to overcome unseen sensor noise. This work presents an object classification module operating on unsupervised learned multi-modal features with the ability to overcome gradual or total sensor failure. A two stage approach composed of an unsupervised feature training and a uni-modal and multi-modal classifiers training is presented. We propose a simple but effective decision module switching between uni-modal and multi-modal classifiers based on the closeness in the feature space to the training data. Evaluations on the ModelNet 40 data set show that the proposed approach has a 14% accuracy gain compared to a late fusion approach operating on a noisy point cloud data and a 6% accuracy gain when operating on noisy image data.**

I. INTRODUCTION

For autonomous driving applications it is important to know which type of road users surround the ego vehicle and which kind of road side infrastructure is present in the current traffic scene. Using a multi-modal sensory setup for classifying road users and road side infrastructure is a reasonable choice: firstly different modalities capture complementary information which is beneficial for classification and secondly, multiple sensors allows for robustifying classification against modality losses.

State of the art research test vehicles are usually equipped with multiple sensors like lidar, radar, camera and ultrasonic sensors. Data retrieved from these sensors are often processed in a sequential pipeline for object recognition: first the strength of a single modality in detecting the object are exploited and afterwards it is classified within another modality [7], [8], [20]. However if a single modality fails the whole pipeline fails. So different fusion techniques or redundant processing units are used to deliver results even if one modality fails to support the idea of robustifying the perception system.

One could argue that modality losses could easily be detected on system level and be tackled in simply switching between single modality classifiers. Compared to complete modality loss it is nearly impossible to detect noisy data already on system level. Therefore sensor setup choices for autonomous driving follow the idea that different modalities overcome challenging scenarios in a complementary way due to different sensing technologies. So if one sensing technology is prune to errors in one scenario another modality is usually added to the system which is not disturbed in this specific scenario. For example a camera could capture noisy images under difficult lighting conditions whereas a lidar is not affected due to its sensing technology. However if the classification module is not exploiting the benefit of this behaviour through detecting it, the benefit of complementary sensing technologies is lost in terms of robustness.

Classifiers utilizing machine learning techniques achieve astonishing results within the image domain and recently also on 3D data. One drawback on most of these techniques is that they have to be exposed to specific sensor noise during training time in order to overcome it as suggested in [2], [10]. From an implementation point of view it is unfeasible to model realistically all kind of sensor noise. Furthermore it is also impractical to record all type of sensor noise due to the enormous time consumption, and even though there is no guarantee for whole coverage. Thus, it is important that classifiers overcome also noisy input data which is not present in the training set.

Within this work we propose a multi-modal architecture for classification, which overcomes gradual or total sensor failure. First, multi-modal features are learned in an unsupervised manner. Then, these learned features are further used as input for a supervised classifier which explicitly handles noise not seen during training stage. We evaluate the proposed architecture with a scenario including lidar and camera sensors while the architecture can be easily extended with other modalities. In summary, the contributions of this work are:

- An unsupervised multi-modal feature extraction from image and 3d-point cloud data
- Supervised classification using multi-modal features with overcoming gradual or total sensor failure

II. RELATED WORK

In this section related work for multi-modal fusion approaches in automotive applications is discussed in Sec. II-A and for other research fields in Sec. II-B. Finally within Sec. II-C literature about point cloud feature extraction and

This research has been partially funded by the EU H2020 research project under grant agreement No 688652, the Swiss State Secretariat for Education, Research and Innovation (SERI) No 15.0284, and by the Swiss National Science Foundation through the National Center of Competence in Research Robotics (NCCR).

[1] Ibeo Automotive Systems GmbH, {julia.nitsch, max.schmidt}@ibeo-as.com

[2] Autonomous Systems Lab, ETH Zurich, {jnieto, rsiegwart, cesarc}@ethz.ch

reconstruction is discussed[1].

A. Multi-modal fusion approaches in automotive applications

Some learning approaches follow a sequential processing pipeline detect objects within one modality first and then classify them within another modality [8], [20], [7]. However if one of the modalities fail, the whole classification task fails. Another sequential processing approach is to first semantically label images and use this information together with computed disparity map from stereo cameras to obtain the labeled object proposals [25]. A late fusion approach is applied in an end-to-end learning fasshion in [16]. The authors introduce a sensor drop strategy during training in order to overcome modality losses. They report negligible policy drops when evaluating on modality losses and noisy input. Following their approach in training a late fusion classifier on our data set resulted in serious accuracy drops as reported in Sec. IV. Except for [16], none of the fusion approaches report the ability to overcome modality losses.

B. Multi-modal fusion in other research fields

Multi-modal features can be learned unsupervised using Autoencoder (AE) architectures as proposed in [17] and [5]. During training the AE is exposed to all possible input combinations including modality losses. Thus, it is able to overcome modality losses. Compared to ours, these works are trained on data containing modality losses which our classifier is explicitly not. AE architectures without specialized feature extractions for each modality use fully connected neural networks for processing each input separately [18], [5]. Early fusion, late fusion and temporal fusion strategies for multi-modal input are evaluated in means of accuracy for a gesture detection application in [18], which leads to similar accuracies for all fusion approaches. This finding is in line with suggestions in [23] and [1] that there is no clear preferable strategy yet when to fuse modalities. A late fusion scheme with fusing modality specific features for video and audio input is presented [26] to compute interestingness for video streams. Similar modality specific features are also applied for semantic segmentation tasks as in [28]. Here, late fusion weights are computed for each class based on early feature layers and so they are able to adapt to different challenging conditions. Whereas the class weights are learned in a data driven approach in [28], authors in [4] propose a late fusion based on a statistical fusion approach taking the performance of each single modality classifier into account. However, compared to our method, these approaches trained their classifiers with containing known challenging conditions already in their training set. Except [17] and [5] whose architectures are trained on modality losses none of the other architectures reports overcoming total sensor failures.

C. Point cloud handling

Point clouds are processed with 2D convolutions after projecting the point cloud as presented in [27]. The drawback of 2D projections is losing 3D information to some extend but on the other hand well known convolutions from image processing can be applied. Point clouds could also be transferred to 3D voxel grids and convolved with 3D kernels as suggested in [14], [13]. Alternatively, combinations of 3D convolutions and 2D convolutions are applied as suggested in [22]. 3D convolution implementations are very limited due to high memory consumption. Architectures for object classification, or respectively detection and classification, operating directly on the point cloud data structure are presented in [21], [31].

Object point clouds are reconstructed from image features in [12]. Therefore image features are extracted utilizing convolutional neural network (CNN) architecture and fully connected layers for reconstructing the point clouds. We follow their approach in extracting image features for reconstructing point clouds. Similar to our work, unsupervised feature learning strategies for point cloud features are also proposed in [11], [30], with the intention for further usage in supervised classification tasks. Compared to ours, which uses the raw point cloud structure, in [11] the point cloud is first voxelized before 3D convolutions and respectively deconvolutions are applied. For processing the raw point cloud a graph based encoder which computes features taking neighboring points into account is used in [30]. Whereas our point cloud processing only computes per point functions following [21]. Although we propose a multi-modal feature approach we compare our point cloud feature extraction performance to [30] in Sec. IV-B. Furthermore, we follow the suggestion using the Chamfer distance as loss between reconstructed and ground truth point cloud as it is proposed in [12] and [30].

III. Method

Within this section first the architecture visualized in Fig. 1 for learning unsupervised multi-modal features is presented. We start with discussing the image processing pipeline in Sec. III-A and continue in Sec. III-B with the point cloud processing pipeline. Then the sensor drop module is described in Sec. III-C. Afterwards the supervised classifier (see Fig. 2) operating on the previously unsupervised learned features is described in Sec. III-D. Finally the decision module is described in Sec. III-E which enables robustness against sensor failures.

A. Image Encoder / Decoder

The image encoding and decoding block within the architecture is highlighted with a dashed green rectangle in Fig. 1. The encoding part follows the image feature extraction from [12]. This feature extraction architecture stacks multiple convolutional layers and extracts a 3x4x128 feature code per image. In comparison to [12] batch normalization is applied for each convolutional block.

[1] Please note that the focus of this work lies on learned features and learned multi-modal data fusion for sensor failures. We are aware of other works using traditional approaches for feature extraction and fusion but they are not discussed here due to space limitations.

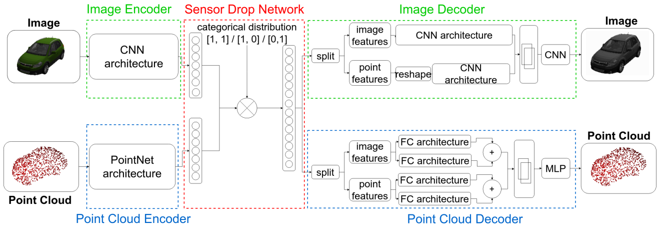


Fig. 1. The architecture for unsupervised feature extraction is visualized. Processing blocks for image encoding and decoding are shown in green. Within the *Image Encoder* image features are extracted following [12]. The processing blocks for point cloud encoding and decoding are visualized in blue. For extracting point cloud features we utilized the PointNet [21] architecture within the *Point Cloud Encoder*. The *Sensor Drop Network* following [16] supports the network in becoming invariant to sensor losses and is shown in red.

For decoding the stacked feature code is splitted in point cloud features and image features again. The code representing image features is processed through a CNN using convolutional filters and transposed convolutions. The code representing point cloud features is first reshaped to two dimensional structure of shape 48x32x1. Afterwards convolutional layers and transposed convolutions are applied to the point cloud features. Once both features are processed by individual CNNs the output is concatenated and a convolution is applied on the fused output. The mean squared error loss on pixel values is used for comparing the generated output image to the gray scale version of the input image.

B. Point Cloud Encoder / Decoder

The point cloud encoding and decoding block within the architecture is highlighted with a dashed blue rectangle in Fig. 1. The encoding block uses the global feature extraction from the PointNet architecture [21]. We extract 1536 global point cloud features per input cloud to compute a balanced amount of point cloud and image features.

Similar to the image decoder the code gets first splitted into point cloud features and image features within the decoder. Both features are separately processed by a fully connected network inspired by the decoder presented in [12]. These networks consist of parallel applied fully connected layers which are then summed up. Finally the output of the image feature processing and the point cloud feature processing is concatenated and further processed by a multi layer perceptron (MLP) to fuse both processing streams. In comparison to [12] no short cuts between encoder and decoder are used because the main objective is learning descriptive features. The loss for comparing the generated output point cloud to the input pointcloud is based on the Chamfer distance as suggested in [12] and [30].

C. Sensor Drop

The aim of the sensor drop network is supporting the presented unsupervised architecture in becoming invariant to sensor losses following [16]. It is highlighted with a dashed red rectangle in Fig. 1. During training time either all point cloud features or all image features or no features are set to zero. The decision if and which features are set to 0 is drawn from a categorical distribution. This is one possibility to ensure that at least one modality is presented to the network. In comparison to [16] the features are activated with a hyperbolic tangent (tanh) nonlinearity before concatenation with the aim of scaling features accordingly.

D. Classifier

Weights within the encoders get fixed after the unsupervised training step. The now extracted features are forwarded to classification module highlighted with a dashed orange rectangle in Fig. 2. The classifier follows the idea of splitting the code into image and point cloud features like both decoders do. Both features are processed by a fully connected architecture separately and single classifiers are trained for each modality in a pretraining step. Furthermore, a mean feature representation is stored and updated for each single class per modality during the classifier training. Afterwards, a late fusion module scales and adds up outputs from the single modality classifier. The late fusion classifier is trained following the sensor drop strategy with the intention to overcome modelity losses. However, empirical results (see Tab. II and Fig. 11) show that the late fusion classifier works best if both modalities produce good results or both modalities have weaknesses. Compared to uni-modal classifiers the accuracy of the late fusion classifier also drops significantly if one of the modalities is noised.

E. Decision Module

Therefore we propose a simple but effective decision module (see Fig. 2) to overcome this accuracy drop by the late fusion classifier. This decision modules switches between uni-modal classifiers and the late-fusion classifier on basis of feature closeness to the training data set. Therefore, the cosine similarity between the stored mean feature vector and the predicted class feature vector is computed. Based upon a simple threshold comparison the decision module chooses either a uni-model or the multi-modal classifier. Please note that training data does not contain noise or modality losses since we want to show the ability of the decision module overcoming unseen noise or failures.

IV. Evaluations

Before starting with discussing the experiments conducted on the proposed unsupervised learning architecture and the supervised classifier, the used data sets are presented in Sec. IV-A. Following the evaluation on the quality of unsupervised learned features is presented in Sec. IV-B. Finally in Sec. IV-C the performance of the classifier is evaluated on noise patterns and modality losses not present in the training data.

A. Data Set for Experiments

For training unsupervised features we use 3D object models and corresponding textures from the ShapeNet Models Core v2 [6]. This data set covers 55 common object classes with approximiatly 51300 different object models. The classifier is trained on the ModelNet 40 data set presented in [29]. This data set contains 3D models from 40 common

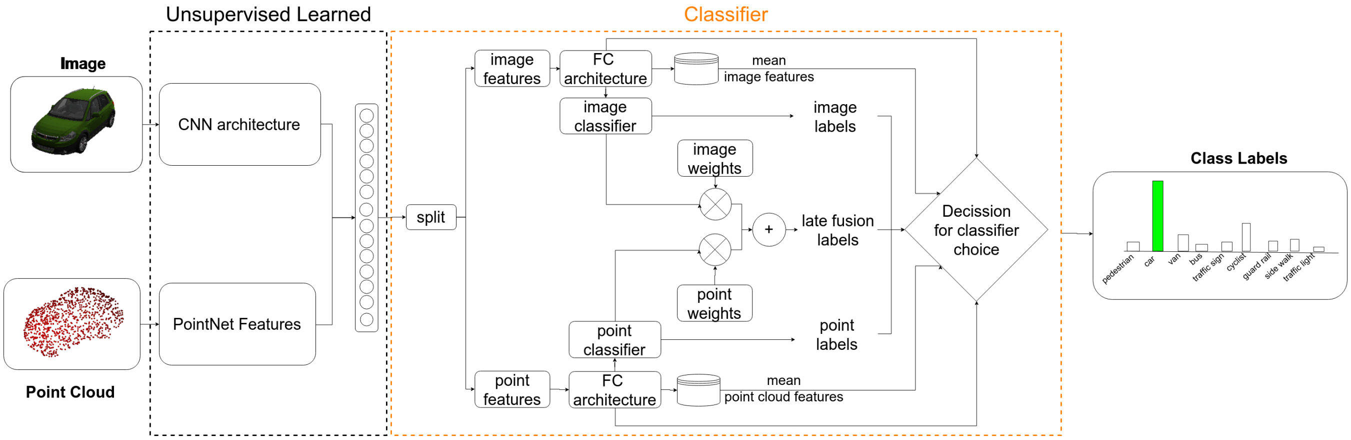


Fig. 2. The classifier operating on previously learned features is shown. It switches between uni-modal and multi-modal classifiers based on the closeness in feature space to the training data to overcome senor failures and noisy data which are not present in the training data set.

object categories (e.g: car, airplane, person, laptop ecetera). In order to retrieve a full model point cloud we utilize the *pyntcloud* python library [9] and random sample 2048 points from the objects. Following the preprocessing steps in [21] all objects get centered and normalized to units sphere before sampling. For capturing images we utilize Blender [3]. Except where explicitly stated these data is used throughout all experiments.

B. Evaluation on Quality of Unsupervised Learned Features

1) Quantitative Results: We are following [30] to evaluate the performance of the AE for learning descriptive features in first performing an unsupervised training on the ShapeNet data set and then reporting the accuracy of a linear support vector machine (SVM) on the ModelNet 40 data set. We utilize the implementation from [19] for the SVM and use the same training and testing split is used as in the ModelNet 40 benchmark [29]. We evaluate three different models of our proposed architecture which are visualized in Fig. 3. In order to compare the purely point cloud based unsupervised architectures to results stated in [30] we run *Model 1* additionally on the same data set as used in [30]. Furthermore, to compare the effect of *Sensor Drop Network* we run experiments on *Model 2* and *Model 3* with and without *Sensor Drop Network* enabled. All models are trained for 50 epochs before the SVM classifier is trained. The SVM results on uni-modal and multi-modal features are stated in Tab. I. For *Model 2* and *Model 3* results reported on uni-modal features are based on simply splitting the multi-modal feature vector accordingly to point cloud and image features.

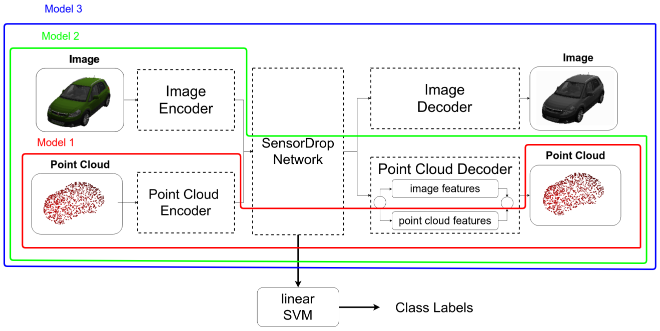


Fig. 3. The point cloud to point cloud AE is highlighted in red and referred as *Model 1*, the point cloud and image to point cloud AE is highlighted in green and referred as *Model 2* and the presented multi-modal AE is highlighted in blue and referred as *Model 3*. After training all models on the ShapeNet dataset, their learned feature performance is evaluated on the ModelNet 40 dataset using a linear SVM as classifier.

As it can be seen from the first experiment the proposed point AE achieves 86.7% which would score second best reported by the results in [30]. One could argue that our feature vector is unnecessarily large compared to [30]. The reason for this is that our AE architecture follows idea from [12] for the image encoder in reconstructing point clouds from images. For using a balanced amount of features between point clouds and images we enlarge the point cloud feature size. Furthermore we are not using a graph based point cloud encoder like [30] and are so not exploiting point neighbourhood relations while still achieving a reasonable accuracy. The obvious benefit for not using graph based encoder is that a graph has not to be build up for every point cloud which results in a faster execution time. From the results stated in Tab. I a clear benefit for using multi-modal features cannot be seen. We hypothesize that fusing modalities does not lead to an accuracy improvement of the classifier if one of the modalities performs already well. However, we observe a small drop when enabling sensor drop during training. We hypothesize that if sensor drop is enabled the architecture must be trained for more epochs to achieve the same accuracy as without sensor drop since it is not seeing the same amount of data in the decoders.

2) Qualitative Results: We also show the ability of the decoders to reconstruct full object point clouds computed from images and depth sensor input. For this evaluation the same training and validation data set as in [12] is used. Within Fig. 4 the reconstruction of a car is visualized using both modalities as input as well as only image and only point cloud features when enabling the sensor drop switch.

C. Evaluation on Supervised Classifier and Decision Module

Within this section we present figures on uni-modal, late fusion and the proposed classifier accuracy and additionally the classifier choice ratio for the proposed method. The

TABLE I

EVALUATION OF FEATURE QUALITY

Model	SVM acc. Point Cloud & Image	SVM acc. Point Cloud	SVM acc. Image
Model 1 data from [30]	-	86.7%	-
Model 1	-	86.3%	-
Model 2 wo sensor drop	85.87%	86.97%	78.36 %
Model 2 w sensor drop	85.47%	87.33%	78.85 %
Model 3 wo sensor drop	86.2%	87.05%	78.3 %
Model 3 w sensor drop	84.7%	85.9%	78.6 %

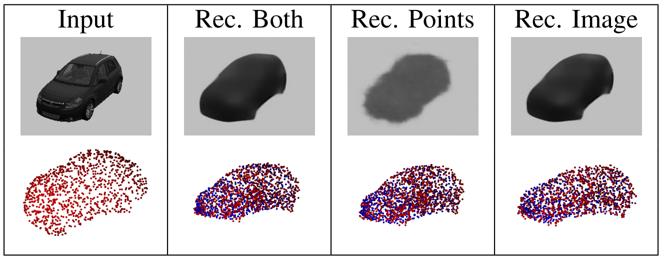
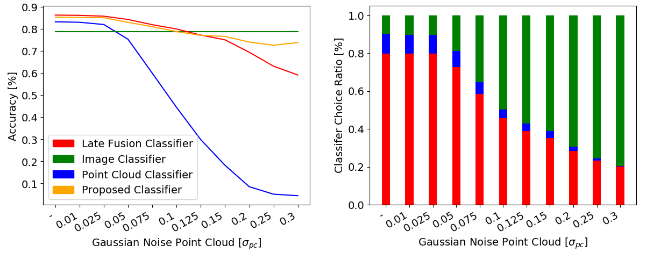


Fig. 4. We trained the network on the same training and validation data which is used in [12]. We show a car reconstruction from the validation set given both modality features, only point cloud features and only image features. Within this scenario the network is trained to reconstruct the whole object point cloud using the point cloud generated from the depth image as input. The ground truth is shown in red and the reconstructed point cloud in blue.

classifier choice ratio states the ratio of how often the uni-modal and respectively the late fusion classifier are chosen by the proposed decision module. Due to assumption of having different sensing technologies we especially focus on results disturbing one modality while not noising the other modality. Nevertheless, we also state figures when noising both modalities on the example of Gaussian noise. Please note that the first entry in each of the following figure represents the results on undisturbed data to compare the effects of adding noise.

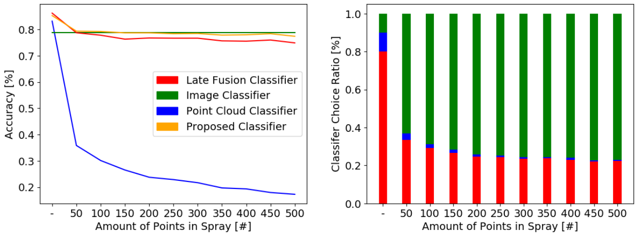
First, Gaussian noise is added to the point cloud with increasing σ_{pc}. As it can be seen in Fig. 5 with increasing noise level our decision module prefers the uni-module image classifier. The noisiest level is also stated in Tab. II where it can bee seen that our proposed decision module outperforms the late fusion classifier by more than 14% and is only 5% less accurate than the uni-modal image classifier.



(a) Classification Accuracy

(b) Classifier Choice Ratio

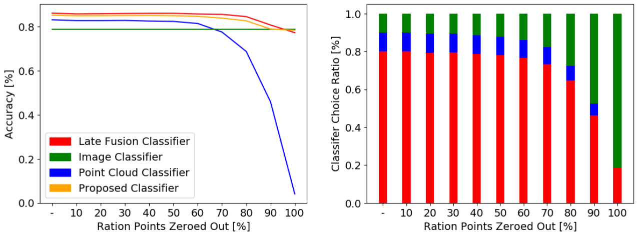
Fig. 5. Gaussian noise is added to the point cloud with increasing σ_{pc}. The proposed classifier prefers the image classifier when the point cloud becomes too noisy.



(a) Classification Accuracy

(b) Classifier Choice Ratio

Fig. 6. Objects are augmented with an increasing simulated spray cloud. The point cloud classifier is not robust against this noise and drops immediately so the proposed classifier mainly chooses the image classifier which performs better than the late fusion classifier.



(a) Classification Accuracy

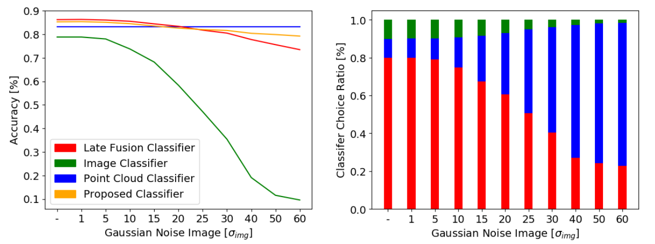
(b) Classifier Choice Ratio

Fig. 7. An increasing amount of points within the point cloud are set to zero which simulates different sensor measurement failures. The point cloud feature extraction as well as the point cloud classifier are quite robust against this noise and therefore the late fusion performs in most of the experiments better than the uni-modal image classifier. So the late fusion classifier is mainly chosen by the proposed classifier.

The second noise pattern applied to the point cloud is the *spray* noise. Here, point clouds with increasing size following a normal distribution with $\sigma = 0.25$ are randomly attached to objects. These kind of additive point clouds are a common issue within automotive perception, especially in rainy environment due to the spray cars produce on wet street or exhaust clouds. We can observe an immediate drop within the uni-modal classifier accuracy in Fig. 6. This is due to the chosen point cloud encoder. Since the encoder especially focuses on corner points and edges which are furthest away from object center it gets fooled by these additional point clouds. However, it can also be seen that the proposed classifier follows the image classifier and produces nearly as good classification results and exceeds the late fusion classifier.

The next evaluation is run on densed out point clouds where an increasing amount of random points are set to the coordinates (0,0,0) to simulate sensor failures. The results for this experiment are visualized in Fig. 7. We observe that the point cloud encoder can handle this noise type much better than the *spray* noise. We hypothesize this is due to the fact that important corner points are still present (or its neighbors) which is not changing the overall appearance of the object. Within this experiment we observe good results from the late fusion classifier which results in preferring this classifier over the image classifier most of the time. However, if all points are zeroed, our proposed classifier exceed the late fusion classifier by 1.1%.

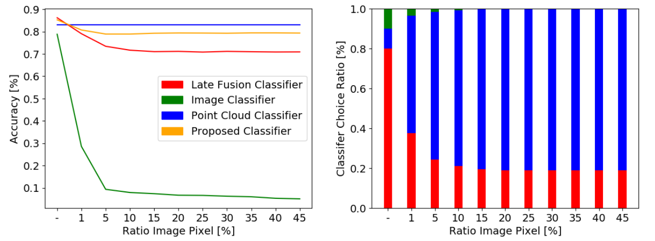
The first experiment conducted on images adds Gaussian noise (see Fig. 8). In the corner case $\sigma_{img} = 60$ the proposed



(a) Classification Accuracy

(b) Classifier Choice Ratio

Fig. 8. Gaussian noise is added to the image with increasing σ_{img}. The proposed classifier prefers the point cloud classifier with increasing σ_{img}.



(a) Classification Accuracy

(b) Classifier Choice Ratio

Fig. 9. Salt and pepper noise is applied to an increasing amount of image pixels. The proposed classifier immediately relies on the point cloud classifier and achieves throughout a better accuracy than the late fusion classier.

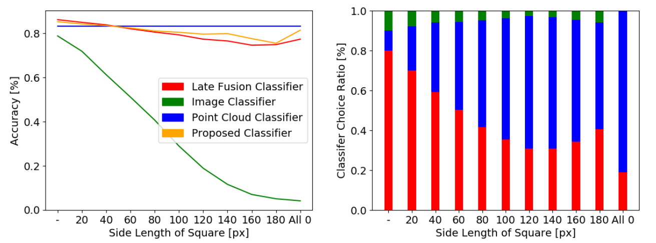
classifier exceeds the late fusion classifier by more than 5% as stated in Tab. II. As it is shown in [15] and [24] that salt-and-pepper noise has strong influence on CNN performance we also performed tests on this impulsive noise and on shot noise to evaluate the image classifier performance on them. The results are plotted in Fig. 9 for salt-and-pepper noise. The image classifier accuracy drops immediately and the proposed classifier mainly chooses the output of the uni-modal point cloud classifier. Results for shot noise follow a similar pattern as for salt-and-pepper noise. In Fig. 10 the image is overlayed with randomly rotated black squares with increasing size until a complete image failure is simulated in setting all pixel values to 0. It can be seen that the proposed method prefers the uni-modal point cloud classifier again and achieves more than 4% better accuracy compared to the late fusion approach within the complete image failure case.

Finally, we apply Gaussian noise to both input modalities and report the results in Fig. 11. It can be seen that the late fusion classifier outperforms the uni-modal classifiers and is therefore preferred by our proposed classifier.

V. Conclusion and Future Work

Within this work we present an architecture for learning multi-modal features in an unsupervised way. Quantitative results on the extracted feature quality are presented on the example of linear SVM classification results. Furthermore, qualitative results for reconstructing both modalities when enabling the sensor drop switch are shown. Even though the sensor drop generalization technique gives promising results for feature extraction in supporting the reconstruction of other modalities, it does not support the late fusion approach to overcome sensor failures on our data set, especially in the case of noisy input data. Thus, we introduce a classifier which uses a simple but effective decision module switching between different uni-modal and multi-modal classifiers based on the closeness in the feature space to the training data. Hence, it overcomes sensor failures and noise which are not present in the training set.

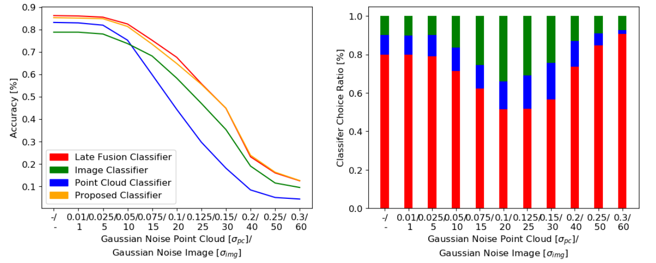
In future work we will further investigate the effect of sensor drop generalization to different fusion techniques like early or deep fusion for unsupervised feature extraction. Furthermore we will investigate the *closeness in feature space* idea from a data driven approach to see if other metrics can be found to reason about the extracted feature quality compared to features extracted from the training set. Beside overcoming sensor failure this could help in detecting unknown noise patterns in sensor data. Additionally we will investigate the extension of the network to radar sensors which are commonly used in automotive setups. Since radar sensors provide a point cloud with velocity information we could utilize similar encoding structures as for depth point clouds and extend the decoders accordingly.



(a) Classification Accuracy

(b) Classifier Choice Ratio

Fig. 10. The image is disturbed with a black square applied to random rotations and increasing square size. Please not that in the last experiment 'All 0' all pixel values are set to 0. With increasing square size the proposed classifier outperforms the late fusion approach.



(a) Classification Accuracy

(b) Classifier Choice Ratio

Fig. 11. Gaussian noise is added to both input modalities. The proposed classifier prefers throughout the late fusion classifier.

TABLE II
Results of Corner Cases

input data	Point Cloud Classifier	Image Classifier	Late Fusion Classifier	Proposed Classifier
undisturbed img undisturbed pc	83.16%	78.81%	86.20%	85.27%
failure img undisturbed pc	83.16%	4.06%	77.35%	81.41%
undisturbed img failure pc	4.06 %	78.81%	77.31%	78.41%
img Gauss noise (σ_{img} = 60) undisturbed pc	83.16%	9.58%	73.46%	79.22%
undisturbed img pc Gauss noise (σ_{pc} = 0.3)	4.42%	78.81%	59.09%	73.74%

REFERENCES

[1] T. Baltruaitis, C. Ahuja, and L. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, Feb 2019.

[2] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[3] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2018.

[4] H. Blum, A. Gawel, R. Siegwart, and C. Cadena. Modular sensor fusion for semantic segmentation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[5] C. Cadena, A. R. Dick, and I. D. Reid. Multi-modal auto-encoders as joint estimators for robotics scene understanding. In *Robotics: Science and Systems*, 2016.

[6] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015.

[7] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1259–1272, May 2018.

[8] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *IEEE CVPR*, 2017.

[9] D. de la Iglesia Castro. pyntcloud library. `https://github.com/daavoo/pyntcloud`, 2016–2018.

[10] J. Ding, B. Chen, H. Liu, and M. Huang. Convolutional neural network with data augmentation for sar target recognition. *IEEE Geoscience and Remote Sensing Letters*, 13(3):364–368, March 2016.

[11] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena. SegMap: 3D Segment Mapping using Data-Driven Descriptors. In *Robotics: Science and Systems*, 2018.

[12] H. Fan, H. Su, and L. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 38, 2017.

[13] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016.

[14] V. Hegde and R. Zadeh. Fusionnet: 3d object classification using multiple data representations. *arXiv preprint arXiv:1607.05695*, 2016.

[15] H. Hosseini, B. Xiao, and R. Poovendran. Google's cloud vision api is not robust to noise. In *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*, pages 101–105. IEEE, 2017.

[16] G.-H. Liu, A. Siravuru, S. Prabhakar, M. Veloso, and G. Kantor. Learning end-to-end multimodal sensor policies for autonomous navigation. In S. Levine, V. Vanhoucke, and K. Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 249–261. PMLR, 13–15 Nov 2017.

[17] 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)*, pages 689–696, 2011.

[18] N. Nishida and H. Nakayama. Multimodal gesture recognition using multi-stream recurrent neural network. In *Pacific-Rim Symposium on Image and Video Technology*, pages 682–694. Springer, 2015.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2018.

[21] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

[22] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.

[23] D. Ramachandram and G. W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017.

[24] P. Roy, S. Ghosh, S. Bhattacharya, and U. Pal. Effects of degradations on deep neural network architectures. *arXiv preprint arXiv:1807.10108*, 2018.

[25] L. Schneider, M. Cordts, T. Rehfeld, D. Pfeiffer, M. Enzweiler, U. Franke, M. Pollefeys, and S. Roth. Semantic stixels: Depth is not enough. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, pages 110–117. IEEE, 2016.

[26] Y. Shen, C.-H. Demarty, and N. Q. Duong. Deep learning for multimodal-based video interestingness prediction. In *Multimedia and Expo (ICME), 2017 IEEE International Conference on*, pages 1003–1008. IEEE, 2017.

[27] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[28] A. Valada, J. Vertens, A. Dhall, and W. Burgard. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4644–4651. IEEE, 2017.

[29] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[30] Y. Yang, C. Feng, Y. Shen, and D. Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2018.

[31] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.