Classification of Streetsigns Using Gaussian Process Latent Variable Models

Wilfried Wöber
University of Natural Resources and Life Science & UAS Technikum Wien
Vienna, Austria
woeber@technikum-wien.at

Mohamed Aburaia
Department Industrial Engineering UAS Technikum Wien
Vienna, Austria
aburaia@technikum-wien.at

Cristina Olaverri-Monreal
Chair Sustainable Transport Logistics 4.0 Johannes Kepler University
Linz, Austria
cristina.olaverri-monreal@jku.at

***Abstract*—Since the rise of deep artificial neuronal nets, object detection and classification became an autonomous procedure, where both, feature extraction and feature processing (e.g.: classification) is done using an architecture based on artificial neurons. The shortcomings of deep neuronal nets are mainly based the black box models and the architecture of the networks, which cannot be estimated. Unknown behavior and over-fitting is still an unsolved problem. Thus, human-made parameters like the number of neurons or the definition of activation functions must be set. This work presents a non-parametric and non-linear approach for image processing using latent variable models. We used Gaussian process latent variable models for street sign feature extraction, where a latent representation is estimated without prior knowledge such as class label. Based on the latent representation, we visualizes the features and use state-of-the-art classifier for street sign classification. Our results proves, that our approach extracts useful features for classification. Our approach has still shortcomings, such as computational time, which are current areas of research.**

***Index Terms*—Gaussian Process Lantent Variable Models, Image Classification, Feature Extraction**

I. Introduction

The detection of streetsigns is a critical procedure in intelligent transport systems [1], [2], where information from images must be extracted and processed, e.g. classified. Different methods, such as Boosting [3], [4] or support vector machines [5] can be used. The rise of convolutional neuronal nets [6] in 2012 [7] and extensions for detection [8], [9] contributed to object detection in autonomous driving [10]–[13] significantly by automate both, the information extraction and processing.

On the one hand, high dynamical environment, such as lightening condition and weather [1], [2], leads to uncertainty and thus machine/statistical learning is required. On the other hand, [14], [15] showed the restrictions of neuronal nets in terms of robustness. Further, the layer architecture including number of neurons and activation function must be set manually. Different alternative algorithms such as deep random forests [16], deep exponential families [17] or deep Gaussian processes [18] were developed to overcome limitations and simultaneously overcome general neuronal net problems such as black-box modelling. Still, hyperparameters such as number of features to extract must be set manually.

In this work we contribute to the state of the art by extracting information regarding classification of traffic signs through Gaussian process latent variable models (GPLVM) [19], which results in a fully non-linear and non-parametric approach.

We then visualize the latent space using an optimization criterion based on the information loss of the latent space estimation itself for further analysis and classification. Further, classification based on the extracted information is performed using state-of-the-art algorithms.

Thus, the contribution of this work is a fully explicable methodology for unsupervised feature extraction for classification using GPLVMs. To implement this, we discuss (i) the extraction of a latent representation of a street sign dataset using Gaussian process latent variable models, (ii) the estimation of the latent dimension using the reconstruction error, (iii) the visualization of the results in terms of regions of interest and finally (iv) a classification approach based on the extracted latent representation.

This work is structured as follows: chapter II gives an overview of the current state in object classification and detection. Chapter III introduces the used methods. The used dataset and experimental results are discussed in chapter IV. Finally, chapter V discusses the results before finally, chapter VI summarizes this work and outlines further work.

II. State of the Art

Since the publication of AlexNet [7] in 2012, image classification and object detection became a fully automated task, where both, feature extraction and classification were done using a neuronal net [8], [9], [20]. Convolutional neuronal nets (CNN) [6], [21] are the backbone of the classification and detection, where a deep architecture of artificial neurons is used. In addition to object detection, movement estimation [22] or segmentation (e.g.: pixel-wise weed detection [23]) are typical applications of CNNs.

Since deep neuronal nets tends to become black-box models and overfit using limited data [24], alternative deep architecture rises [16]–[18], where the deep architecture are created using alternative models. In [16], the authors stacked

 DOI: 10.1109/ICCVE45908.2019.8964883.

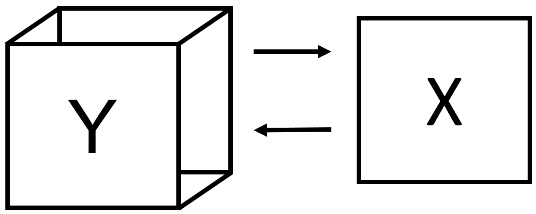

Fig. 1: Visualization of the processing idea using Gaussian process latent variable models. The data $\mathbf{Y}$ is compressed to the latent representation $\mathbf{X}$.

random forests to create a new deep architecture. [25] used support vector machines to create a deep architecture for regression problems.

Those alternative architectures try to overcome limitations of neuronal nets. Gaussian processes [26, Ch. 6.4] were used in [19], [27] to estimate a latent representation of a dataset. In [18], [28], this idea was extended by stacking models based on [29], where the limitations and similarities of neuronal nets in contrast to Gaussian processes were discussed.

Independent from the algorithm itself, a latent representation of the data is estimated during the training process. Using neuronal nets or other deep architectures, this is typically done using a supervised learning procedure, thus a class label is used.

Different to those approaches, Gaussian process latent variable models (GPLVM) [19], [30] is an unsupervised learning approach, where no prior knowledge such as the class label is needed. A latent representation of the data is estimated, which "compresses" the data using model capabilities and data. By estimating the latent representation, the compression results in a lower dimensional space, where the most important information remains. Different to classic approaches such as the principal component analysis (PCA) [26, Ch. 12.1], Gaussian process latent variable models using non-parametric and non-linear regression for latent space estimation. In [31], the latent space is estimated using variational inference [32]. This approach needs less data to create useful models, which is a strong benefit compared to artificial neuronal nets, where a small number of examples is critical [24].

Note, that this approach must not lead to a latent representation useful for processing but unfolds the structure in the data. This process is visualised in figure 1, where a 3D dataset $\mathbf{Y}$ is compressed to the latent representation $\mathbf{X}$. We aim to find a function, which allows us to project data into a latent representation and back using GPLVMs. Note, that currently, we do not use a deep architecture.

III. Methods

In CNN applications [7], [33], image matrices are used for processing. Using latent variable models such as GPLVMs, grey-scale images[1] with R rows and C columns $\mathcal{I}_j \in \mathbb{R}^{R \times C}$ must be "flattened" to $\vec{i}_j \in \mathbb{R}^{R \cdot C \times 1}$, where $\vec{i}_j = \left({}_j p_{0,0}, \ldots, {}_j p_{C,0}, {}_j p_{1,0}, \ldots, {}_j p_{C,R}\right)^T$. ${}_j p_{a,b}$ describes the pixel in the a-th row and b-th column in image j. Latent variable models estimates a compressed representation of each sample based on the dataset finding representative information in the dataset. Note, that this representation is strongly connected to model restrictions, which limits the capability of modelling real coherences.

Historically, the PCA [26, Ch. 12.1] [34, Ch. 23.1] were used as a latent variable models for feature extraction. The PCA was successfully used for autonomous driving [35], [36], face detection [37] and computer vision based quality management [38].

The PCA estimates a projections matrix $\mathbf{W} \in \mathbb{R}^{D \times d}$, which projects the dataset in the latent space. $\mathbf{W}$ is estimated using

$$\underset{\mathbf{W}}{\operatorname{argmin}} \sum_{i=1}^{d} ||\vec{y}_i - \mathbf{W}^{-1}\mathbf{W}\vec{y}_i||_2^2 \tag{1}$$

The samples are projected into the feature space using $\mathbf{W}^T \vec{x}_j$.

Note, that the PCA assumes a Gaussian distributed dataset, which is a strong restriction. This restriction will not hold in real applications. Extensions of the PCA, namely independent component analysis (ICA) [36], [39] or reconstruction ICA [40], soften this restrictions.

In comparison, Gaussian process latent variable models use non-linear and non-parametric regression rather than linear projections for latent space estimation. Different to PCA, ICA or reconstruction ICA, GPLVM [19], [27], [30] use Gaussian process regression (GP) [26, Ch. 6.4] [41], a non-linear and non-parametric regression approach, for latent space estimation. Gaussian processes were successfully used in transition modelling for autonomous driving [42]–[45], stochastic extensions of differential equations [46] or global optimization [47], [48] for hyperparameter estimation. Based on [29], who proved, that a multi-layer perceptron with infinite neurons can converge to a Gaussian process several extensions [49], [50] are available today.

The GPLVM formulates the problem using the mean field family by introducing an independent latent space [19]

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{d=1}^{D} p(\vec{y}_d|\mathbf{X}) = \prod_{d=1}^{D} \mathcal{N}\left(\vec{y}_d|\vec{0}, \mathbf{K}_{NN} + \beta^{-1}\mathbf{I}_N\right) \tag{2}$$

Where $\mathbf{K}_{NN}$ is a kernel matrix, β is the precision and $\mathbf{I}_N$ is the identity matrix. Note, that for each dimension, a Gaussian process is used to estimated the original data $\mathbf{Y}$, based on the latent representation $\mathbf{X}$. Further, equation is a

[1] Note, that this approach is not limited to grey-scale images. Using RGB images, the flattening procedure would lead to $\vec{i}_j \in \mathbb{R}^{R \cdot C \cdot 3 \times 1}$.

general approach, where the number of latent dimensions is not defined.

The estimation of this function can be implemented using several approaches [30], where the location of the latent representation is estimated. The extension to a fully Bayesian approach, the Bayesian GPLVM (bGPLVM), is discussed in [31].

The number of latent dimensions is still unknown. In this work, we used bGPLVM to estimate a latent representation of images of street signs and the reconstruction capability (see equation 1) for the estimation of the latent dimension d^*. Thus, we estimate d^* using

$$d^* = \underset{d}{\operatorname{argmin}} \sum_{n=1}^{N} \sqrt{\sum_{k=1}^{P} \left({}_n p_k - {}_{n,d}\, p_k^*\right)^2} \tag{3}$$

Where ${}_n p_k$ is the original k-th pixel of the n-th image and ${}_{n,d} p_k^*$ is the corresponding reprojected pixel value of the bGPLVM using d latent dimensions. P is the number of pixel in image $\mathcal{I}_n$. d^* is the latent dimension, where the reproduction error is minimal.

Note, that this is a novel approach in comparison to [18], where the latent dimension is estimated using the lower bound [32]. We used the reconstruction error to separate the model estimation and model selection.

We used d^* for the estimation of the latent dimension ${}_{d^*}\mathbf{X}$. ${}_{d^*}\mathbf{X}$ and the correspondence class labels (e.g.: street sign class) are then used for classification, where we used state-of-the-art machine learning algorithms.

Note, that we used a fixed number of inducing points [31] and the optimal number of inducing points can be estimated equivalently to the latent space optimization.

For visualization of the extracted features, we calculated d main components using ${}_d\vec{f} = \begin{pmatrix} {}_d f_1 & \dots & {}_d f_d \end{pmatrix}^T$ for each latent dimension and

$${}_d f_j = \begin{cases} \max\limits_{\mathbf{n}}(\vec{x}_{:,d}) & \text{if } j = d \\ \operatorname{mean}\limits_{\mathbf{n}}(\vec{x}_{:,d}) & \text{else} \end{cases} \tag{4}$$

Where $\vec{x}_{:,d}$ describes the $d-th$ column vector of ${}_d\mathbf{X}$. Thus for each estimatet latent variable model, a set of d^* images visualizing the features can be generated.

For the estimation of d^*, a street sign database based on [51] was used. Due to limited computational power, the database was reduced to 20 street sign classes including total 3054 images. Later, the full dataset (43 classes) and d^* are analysed.

The street signs were extracted from background and resized to $[25 \times 25]$ pixels due to processing limitations.

IV. Experimental Results

The images were centered and scaled, were unit variance and zero mean were set. Based on [31] and the implementation in [52], ${}_d\mathbf{X}$ were extracted, for $d = 1 : 100$, the inducing points are fixed to 50. For each ${}_d\mathbf{X}$, the reproduction error and visualization according to equation 3 and 4 are calculated.

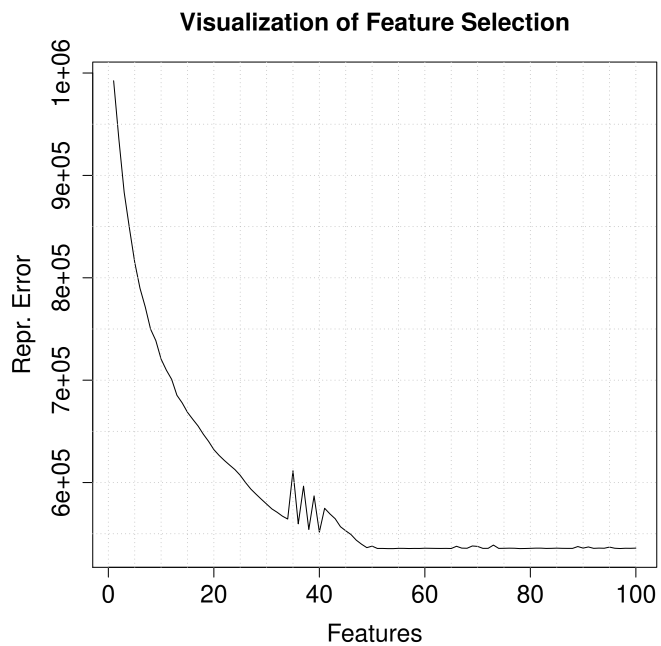

Fig. 2: The reproduction error for $d = 1 : 100$. Note, that the minimum of the error is reached at $d^* = 53$. For the purpose of this work, this dimension is used.

The reproduction error according to the latent dimension is shown in figure 2. Note, that we are using d^* based on the reproduction error, no further information measurement is used. Further, this measurement estimates the latent dimension based on the capability of information reprodcution rather then information gain or capability of feature processing according to prior knowledge such as image label. This is a major different to CNN [6], [7], [20], where the feature extraction is based on the capability of the model and prior information. According to [53], [54], neuronal net-like structures tend to confounding. Thus, the benefit of our approach is the explicability of the model.The reconstruction of the all features of d^* is visualized in figure 3. Note, that the features are sorted according to the automatic relevance determination capability of the bGPLVM implementation [52] visualized in figure 4. We used the radial base function kernel, which is defined as [26, Ch. 6][2]

$$k(\vec{x}, \vec{x}') = exp\left(-\tfrac{1}{2}\sum_{j=1}^{d^*} \tfrac{1}{l_d^2}(x_j, x_j')^2\right) \tag{5}$$

$\vec{x}$ and $\vec{x}'$ are two samples and l_d is the lengthscale of dimension[3] d. Thus, the lowest values corresponds to the highest relevance. Note that the main shapes of the street signs in the dataset, namely circle-like or triangular-like shapes, are clearly visible in figure 3.

[2]The implementation in [52] is described in https://gpy.readthedocs.io/en/deploy/GPy.kern.src.html#module-GPy.kern.src.rbf

[3]The ARD values (e.g.: l_d values) are stored inverted in [52].

Fig. 3: Visualization of all features using d^* latent dimensions.

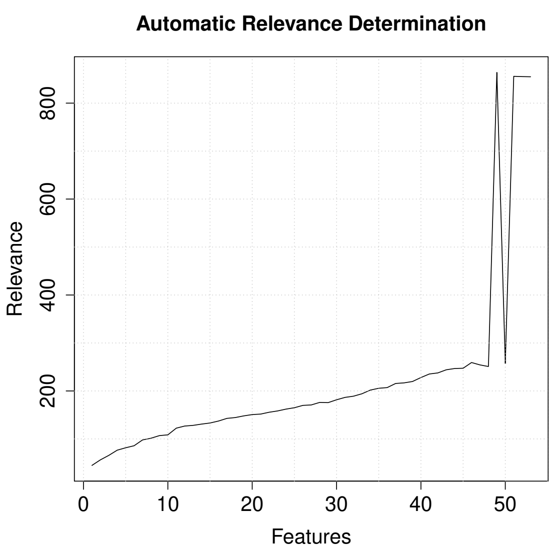

Fig. 4: Automatic relevance determination at d^*. Note, that the lengthscale is inverted and thus, the lowest value represents the features with highest relevance.

A. *Classification Using Initial Dataset*

Since the optimization of the latent dimensions using the reconstruction error focuses on the original dataset reproduction rather than feature processing, classification using $_{d^*}\mathbf{X}$ must not lead to satisfying classification results. Thus, we do classification to evaluate the usefulnes of the estimated data structure.

In this work, classic classifications algorithms were used. We implemented the classification approach using logistic regression (LR) [26, Ch. 4.3.2] and support vector machines (SVM) [26, Ch. 7.1], where the ν-SVM approach [55], [56] is extended using global Bayes optimization [48] for the estimation of the kernel parameter γ and SVM hyperparameters ν. For the training and testing of the classifier, the reduced dataset (20 classes) was divided in 75% training and 25% evaluation data. The classification approaches were done using 5 fold cross validation, where the training datasets were split randomly. The result of the classification is shown in table I, where $\nu = 0.075$ and $\gamma = 0.01253$ was estimated by the optimization approach.

B. *Classification of all Classes*

Based on the described method and d^*, we estimated a latent representation of all 43 classes. We used 1960 examples of the dataset (39209 examples) for latent space estimation. 37244 examples, which are not used for latent space estimation, including all 43 classes were projected in the latent space. Afterwards, the data were split 50% training and 50% evaluation examples.

Again, we used a ν-SVM including hyperparameter optimization for classification and 5 fold cross validation with a random split. The result is shown in table II, where $\nu = 0.0495$ and $\gamma = 0.0251$ was estimated by the optimization approach. The minimal classification accuracy is 91.84%. The mean accuracy is 98.02%.

V. Discussion

We presented a combination of supervised and unsupervised image processing, which were combined in a fully explicable model, where no image pre-processing or image feature extraction was performed. The ν-SVM classification result lies between 93.62% and 100%. In comparison to the original benchmark [51], we used $\left[25 \times 25\right]$ pixels instead of $\left[40 \times 40\right]$. Further, no further image feature (e.g.: HOG features) were used which leads to less classification accuracy.

Regardless of the application, the backbone of the approach is a sparsed Gaussian process [49], where a fixed number of inducing points is used for prediction. This is critical for real-time applications and current part of research [57]. Thus, the benefits of (b)GPLVM are currently restricted by computational limitations. Simultaneously, the rise of computational power will lead to significant gain in computational power and thus, the shortcomings of GPLVM will decrease. The GPU optimized implementation [57] based on [52] is currently in development.

Further, due to the usage of unsupervised learning approach the latent representation can lead to inferior classification results in contrast to supervised approaches such as CNNs. Thus, the bGPLVM approach is used for unsupervised feature extraction rather than feature selection. For better classification results, feature selection must be performed additional to the latent space estimation. This is visualized in table III, where 40 dimensions performs slightly better than 50 dimensions.

TABLE I: Class accuracy of bGPLVM based classification of half dataset.

Classifier	Classes 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ν-SVM	0.993	0.975	0.965	0.936	0.907	0.964	0.994	0.961	0.843	1.000	1.000	0.927	0.934	0.961	1.000	1.000	0.968	1.000	0.989	0.821
Log. Regr	0.949	0.916	0.921	0.883	0.753	0.973	0.852	0.919	0.893	0.978	0.963	0.847	0.865	0.942	0.986	0.987	0.917	0.959	0.938	0.765

TABLE II: Class accuracy of bGPLVM based classification on full dataset

ν-SVM	Classes 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	1.000	0.995	0.978	0.954	0.958	0.938	0.993	0.984	0.995	0.936	0.997	0.997	0.924	0.972	0.974	0.970	0.991	0.974	0.986	0.987	0.918	0.994

ν-SVM	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
	0.975	0.995	0.997	0.949	0.976	0.993	0.959	0.987	0.999	0.990	0.999	0.980	0.989	0.998	0.995	0.989	0.967	0.985	0.997	0.999	0.995

TABLE III: Class Accuracy of bGPLVM Based Classification using different latent dimensions

Class	Latent dimension 10	20	30	40	50
1	0.99	0.98	0.99	0.99	0.98
2	0.81	0.88	0.95	0.96	0.95
3	0.84	0.93	0.95	0.91	0.98
4	0.68	0.94	0.87	0.96	0.93
5	0.64	0.83	0.90	0.81	0.81
6	0.97	1.00	1.00	0.97	0.96
7	0.88	0.90	0.97	0.96	0.91
8	0.96	0.90	0.96	0.96	1.00
9	0.82	0.83	0.76	0.83	0.95
10	0.97	1.00	1.00	1.00	1.00
11	0.98	0.98	0.99	0.99	0.98
12	0.72	0.89	0.83	0.93	0.89
13	0.86	0.89	0.94	0.96	0.94
14	0.86	0.91	0.96	0.92	0.96
15	0.81	0.85	0.83	0.90	0.95
16	1.00	1.00	1.00	1.00	0.95
17	0.91	0.96	0.96	0.98	0.98
18	0.92	1.00	1.00	0.95	1.00
19	0.90	0.97	0.97	0.98	0.95
20	0.85	0.93	0.90	0.93	0.80
Mean	0.87	0.93	0.93	0.94	0.94

Finally, we use the methodology presented in this work for latent dimension estimation, which is the base to investigate the full effect of inducing points. This is a current part of research.

VI. Summary and Outlook

In this work, we used Gaussian latent variable models for non-parametric and non-linear feature extraction without any image pre-processing. The extracted latent representation was used for classification using ν-SVM and logistic regression, where the SVM clearly outperforms the logistic regression approach. In contrast to state-of-the-art algorithms, we did not use any image feature extraction such as HOG features. Further, due to computational limitations, we resized the images significantly. Both, the omission of image features and significant resizing leads to a mean accuracy of 98.17%.

Our next steps are the optimization of the latent space estimation to overcome computational limitations, the extension to street sign detection and the usage of a fully Bayesian approach including classification. Based on that, we will be able to used image sizes similar to [51], which is necessary for state-of-the-art comparison.

Further, CNN-like structures, deep architectures and object detection are current part of research, where we will use fully Bayesian approaches instead of SVM based classification.

References

[1] M. Swathi and K. V. Suresh, "Automatic traffic sign detection and recognition: A review," in *2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*, Feb 2017, pp. 1–6.

[2] S. Wali, "Comparative survey on traffic sign detection and recognition: a review," *PRZEGLD ELEKTROTECHNICZNY*, vol. 1, pp. 40–44, 12 2015.

[3] A. Allamehzadeh and C. Olaverri-Monreal, "Automatic and manual driving paradigms: Cost-efficient mobile application for the assessment of driver inattentiveness and detection of road conditions," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 26–31.

[4] Y. Huang, Y. Le, and F. Cheng, "A method of detecting and recognizing speed-limit signs," in *2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, July 2012, pp. 371–374.

[5] P. Gil-Jimnez, S. Lafuente-Arroyo, S. Maldonado-Bascn, and H. Gmez-Moreno, "Shape classification algorithm using support vector machines for traffic sign recognition," vol. 3512, 06 2005, pp. 873–880.

[6] Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, pp. 541–551, 1989.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[8] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013. [Online]. Available: http://arxiv.org/abs/1311.2524

[9] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

[10] A. Hechri and A. Mtibaa, "Automatic detection and recognition of road sign for driver assistance system," in *2012 16th IEEE Mediterranean Electrotechnical Conference*, March 2012, pp. 888–891.

[11] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[12] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. A. Mujica, A. Coates, and A. Y. Ng, "An empirical evaluation of deep learning on highway driving," *CoRR*, vol. abs/1504.01716, 2015. [Online]. Available: http://arxiv.org/abs/1504.01716

[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." *CoRR*, vol. abs/1511.00561, 2015.

[14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6199

[15] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *CoRR*, vol. abs/1710.08864, 2017. [Online]. Available: http://arxiv.org/abs/1710.08864

[16] Z. Zhou and J. Feng, "Deep forest: Towards an alternative to deep neural networks," in *IJCAI*. ijcai.org, 2017, pp. 3553–3559.

[17] R. Ranganath, L. Tang, L. Charlin, and D. Blei, "Deep exponential families," *Journal of Machine Learning Research*, vol. 38, pp. 762–771, 1 2015.

[18] A. Damianou and N. Lawrence, "Deep gaussian processes," in *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, C. M. Carvalho and P. Ravikumar, Eds., vol. 31. Scottsdale, Arizona, USA: PMLR, 29 Apr–01 May 2013, pp. 207–215.

[19] N. D. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, pp. 1783–1816, 2005.

[20] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.

[22] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks," *CoRR*, vol. abs/1709.08429, 2017.

[23] A. Milioto, P. Lottes, and C. Stachniss, "Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns," *CoRR*, vol. abs/1709.06764, 2017.

[24] G. Marcus, "Deep learning: A critical appraisal," *CoRR*, vol. abs/1801.00631, 2018.

[25] M. Wiering, M. Schutten, A. Millea, A. Meijster, and L. Schomaker, "Deep support vector machines for regression problems," in *International Workshop on Advances in Regularization, Optimization, Kernel Methods, and Support Vector Machines: theory and applications*, 2013, pp. 53–54.

[26] Bishop, C.M., *Pattern Recognition and Machine Learning*. Springer Science+Business Media, 2006.

[27] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," vol. 16, 03 2004.

[28] J. Lee, Y. Bahri, R. Novak, S. Schoenholz, J. Pennington, and J. Sohldickstein, "Deep neural networks as gaussian processes," *International Conference on Learning Representations*, 2018.

[29] R. Neal, "Bayesian learning for neuronal networks," Ph.D. dissertation, University of Toronto, 1995.

[30] P. Li and S. Chen, "A review on gaussian process latent variable models," 11 2016.

[31] M. K. Titsias and N. D. Lawrence, "Bayesian gaussian process latent variable model," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, 2010, pp. 844–851. [Online]. Available: http://www.jmlr.org/proceedings/papers/v9/titsias10a.html

[32] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 1–9.

[34] S. Shalev-Shwart and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[35] W. Wöber, D. Szuegyi, W. Kubinger, and L. Mehnen, "A principal component analysis based object detection for thermal infra-red images," *Proceedings ELMAR-2013*, pp. 357–360, 2013.

[36] Wöber, W., "Development of a Multivariate Analysis Based Object Detection System for Thermal Infra-Red Images," Master's thesis, University of Applied Science Technikum Wien, 2013.

[37] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, pp. 71–86, 1991.

[38] J. Martinez-Alajarin, J. D. Luis-Delgado, and L.-M. Tomás-Balibrea, "Automatic system for quality-based classification of marble textures," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 35, pp. 488–497, 2005.

[39] Pierre, C., "Independent component analysis, a new concept," *Signal Processing*, vol. 36, pp. 287–314, 04 1994.

[40] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, "ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 1017–1025.

[41] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[42] W. Wöber, G. Novotny, M. Aburaia, R. Otrebski, and W. Kubinger, "Estimating a Sparse Representation of Gaussian Processes Using Global Optimization and the Bayesian Information Criterion." Austrian Robotics Workshop, 2018.

[43] Reece, S.; Roberts, S., "An introduction to Gaussian processes for the Kalman filter expert," in *2010 13^{th} Conference on Information Fusion (FUSION)*. IEEE.

[44] Hartikainen, J.; Srkk, S., "Kalman filtering and smoothing solutions to temporal Gaussian process regression models," in *2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2010.

[45] Ko, J.; Fox, D., "GP-BayesFilters: Bayesian filtering using Gaussian Process prediction and observation models," 2009, (online) https://rse-lab.cs.washington.edu/papers/gp-bayesfilter-arj-09.pdf (Last access: 18.2.2018).

[46] Garcia, C.A.; Otero, A.; Felix, P.; Presedo, J.; Marquez, D.G., "Nonparametric Estimation of Stochastic Differential Equations with Sparse Gaussian Processes." Physical Review E, 2017.

[47] Bergstra, J.; Bardenet, R.; Benigo, Y.; Kegl, B., "Algorithms for hyper-parameter optimization." NIPS'11 Proceedings of the 24^{th} International Conference on Neural Information Processing Systems, 2011.

[48] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 2951–2959.

[49] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 5, 2009, pp. 567–574.

[50] Snelson, E.; Ghahramani, Z., "Sparse Gaussian Processes using Pseudo-inputs." Advances in Neural Information Processing Systems 18 (NIPS 2005), 2005.

[51] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: A multi-class classification competition," in *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 1453–1460.

[52] GPy, "GPy: A gaussian process framework in python," http://github.com/SheffieldML/GPy, since 2012.

[53] S. Greenland, J. Pearl, and J. M. Robins, "Causal diagrams for epidemiologic research," in *Epidemiology*, vol. 10, 1999, pp. 37–48.

[54] J. Pearl and D. Mackenzie, *The Book of Why: The New Science of Cause and Effect*, 1st ed. New York, NY, USA: Basic Books, Inc., 2018.

[55] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[56] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch, *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2017, r package version 1.6-8.

[57] A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. Leon-Villagra, Z. Ghahramani, and J. Hensman, "GPflow: A Gaussian process library using TensorFlow," *Journal of Machine Learning Research*, vol. 18, no. 40, pp. 1–6, apr 2017. [Online]. Available: http://jmlr.org/papers/v18/16-537.html