

Navigation Application with Safety Features

Andrew Usama¹, Moustafa Waly², Habiba Elwany³,
Monica Medhat⁵, Youssef Mobarak⁶
Misr International University
Computer Sciences
Cairo, Egypt

Mohamed H. ElGazzar⁴
IoT Senior Solutions Architect
Vodafone International Services
Cairo, Egypt

Abstract—In 2017, the number of car accidents that occurred was astronomically high, even though, infrastructural road systems are being continuously built and renewed to make it more efficient. But a significant problem which still remains is that a staggering number of accidents is exactly what should be avoided. In order to address this issue, this paper will serve to survey and discuss some of the solutions proposed, both software and hardware, for this problem. This will include some of the implemented safety features while also exploring how to make the system more interactive and smooth to meet user needs.

Keywords—Traffic safety; navigation systems; neural networks

I. INTRODUCTION

In this survey paper, we gathered several papers with the specific purpose and aim of building a navigation application with safety features and how to implement them while also keeping the application interactive and user-friendly. Safety features include how to detect road junctions and how to identify straight road network sections using software such as famous machine learning algorithms or hardware such as the Velodyne rangefinder. This paper further aims to discuss several algorithms with their advantages and disadvantages, in order to reach a consensus on which algorithms are best suited for the tasks presented. This paper is organized in the following order Abstract, Introduction, Related Work, Conclusion, and lastly the references.

II. RELATED WORKS

A. Efficient Filtering and Clustering Mechanism for Google Maps

The work presented in [1] aims to solve the problem of displaying a large number of markers on the map without greatly affecting the interactivity and the reaction time for the application.

The algorithms used to solve this problem are Grid-based Filtering algorithm and DBSCAN, density-based spatial clustering of application with noise, algorithms [2], [3], and [4] which are used to be able to display large-scale geospatial data on the map. The first algorithm used is Grid-Based filtering which is used to prevent markers from overlapping when displaying hundreds or even thousands of them. The Algorithm divides the viewport into several grid cells for effective filtering. After that, it will get the top K places from each grid cell and performs the algorithm. Then, we give each cell a unique reference key given using Cartesian coordinate systems which save computation time when defining the reference from

each point's longitude and latitude. And then select the top k ranked places in each of these grid squares and display them to avoid overlapping markers. But even then, sometimes we can still get several markers that overlap which makes it difficult for the user to pick a marker if it's being overlapped and in that's where DBSCAN is used. DBSCAN is known for good performance. It is suited for tasks relating to large datasets, with noise, and can identify clusters with different sizes and shapes. It works by clustering regions with a high density of points. This algorithm needs three input parameters: For an experiment, a dataset of 74000 restaurant locations was gathered to test the algorithms proposed and then divided into hotspots for clearer results. One of the hotspots tested, was around the Busan area and the result are shown in Table I.

TABLE I. RESULTS OF THE BUSAN AREA EXPERIMENT [1]

	All	Naive Top-k	Proposed Algorithm
#markers	370	125	77
#clusters	-	-	6
Distance	5 km	5 km	5 km
Zoom level	13	13	13

Using these algorithms significantly improved the response time by 20ms as shown in Fig. 1.

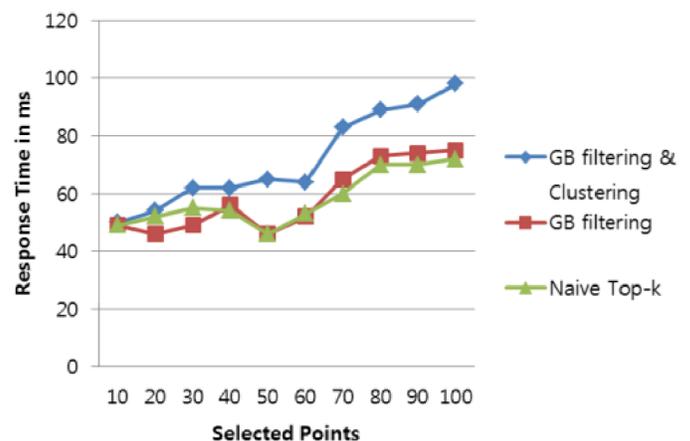


Fig. 1. Response time for selected points [1]

The main drawback of this paper is that it does not try any algorithms other than the ones proposed. Instead, it compares a hybrid of Grid-based filtering and DBSCAN with naive plotting and while the results are significantly better; it is not a fair comparison since naive plotting is too basic to be compared to 2 algorithms.

B. Cloud Aided Safety-based Route Planning

How to use the work described in [5] in order to create a neural network and use clustering techniques to produce a reliable risk assessment for safe route planning. While achieving the goal of guiding the driver toward the route with a low accident risk.

Developing a safety-based multi-objective safety route planning is done through an assessment a road risk module, which is proposed through features like road risk index, crash prediction, neural network, hybrid neural network model, and Crash rate history in order to model the risk level. Furthermore, another approach for road risk index emerged in DE Leur's and Sayed's work [6] which studied the driver assessment of existing road risks. A function for exposure of crash rates is often modeled as road risks, in which the amount of crashes occurrence are the exposure representation. Finally, the crash rate is the number of crashes per unit exposure and the most common measurement of exposure is the miles a vehicle has traveled.

Using these factors which were gathered from the road segment and crash data, a crash rate prediction was modeled. This model has a multi-state database that includes accident data, roadway inventory files, state supplemental inventory containing curves, and grade data. Furthermore, 70 percent of the samples in the dataset was used for training while 15 percent were reserved for the testing and validation of the neural network. The mean square error is the measurement of the quality of the module and for the [7] R-value regression is the relation between the expected output and the actual output. The mean square error used was 9.21 and the R-value regression was 0.80. Consequently, a hybrid neural network model was developed that aims to improve neural network models. This model was developed by partitioning the raw input-output data into three clusters using the fuzzy C-means clustering algorithm. For each cluster, we get a neural network model as shown in Fig. 2. Finally, the accident rates can be

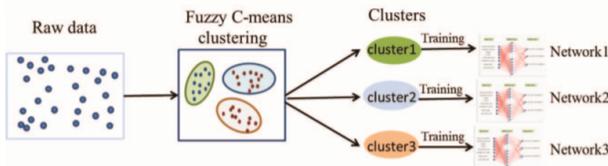


Fig. 2. A hybrid neural network model [5]

predicted, and the risk index is generated using the following formula:

$$RRI(i) = F\left(\sum_{i=1,2,3} n_{ij} S_j / AADT_i\right) \quad (1)$$

,where:

- i is the road segment number
- $RRI(i)$ is the risk index of road segment i
- n_{ij} is the predicted number of accidents
- S_j is the cost of an accident of type
- The $AADT_i$ represents the annual average daily traffic on the road segment

A real-world route experiment, shown in Fig. 3, was performed which is a route planning from Scioto Downs Inc to Delaware Ohio by first abstracting into graph the road network. The main road intersection was represented as nodes. The aim is to test the models and see the result from the first node to the final node with the minimum cost in the time expected to travel and the risk index. The output was computed by Cplex for $\alpha = 0$ and $\alpha = 0.2$, where α is the weight on cumulative RRI reflecting the driver's safety preferences. for $\alpha = 0$, for the travel time expected was 42 minutes and the total risk index was 161.86. While when $\alpha = 0.2$, the travel time expected was 44 minutes and total risk index was 103.57. The second

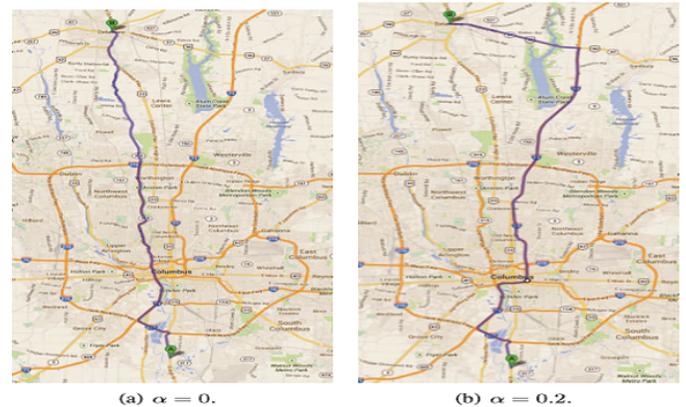


Fig. 3. Optimal routes with different α s [1]

route had a 36 percent higher risk when compared to the first route but required 2 more minutes which shows that the safe route often differs from the fast route. Finally, the deficiency of modeling a road risk index using crash rate cannot be used for roads with no historical crash data. And it cannot predict the dependence on changing factors like traffic density or weather condition.

C. Identification of Curves and Straight Sections on Road Network from Digital Vector Data

According to [8], a method was proposed for the automatic identification of curves and their characteristics, particularly the horizontal size of the curves and their length on roads. According to the World Road Association (WRA), accidents tend to occur 1.5 to 4 times on curves than on straight roads. The method proposed was built on the principle of determination of the radius of an osculating circle [9].

The problem is that there is no ideal data available, due to the fact that breakpoints are somewhat equally apart from one another. The data is neither "dense" nor "sparse". And there are only very small measurement errors, but that type of data is not available. Therefore, the circumscribed circle calculation

TABLE II. THE VALUES IN PERCENTAGE INDICATE HOW MANY BREAK POINTS WERE CORRECTLY CLASSIFIED [8]

Threshold value of radius (m)	II/446		II/444	
	Circumscribed circle	Osculating circle	Circumscribed circle	Osculating circle
100%	65%	65%	54%	56%
300%	73%	72%	74%	76%
500%	74%	73%	80%	80%
1000%	70%	78%	80%	81%
1500%	70%	74%	78%	80%
2000%	72%	75%	77%	78%

of the radius may lead to the incorrect determination of curves due to the relatively small radii assignment. That's why a method called the osculating circle was used, it has none of the above drawbacks because it is merged with the Douglas-Peucker algorithm.

They compared their method with 3 other methods which are based on the identification of the circle circumscribed radius of the road segment, or on the calculation of the break-points angle formed by a road line. In this paper, data from the Road and Motorway Directorate of the Czech Republic (RMD CR) was used because they are directly acquired by the roads administrator. The authors collect the maps on which the curves exist, and then adds a geometric line generalization which reduces and simplifies the data without losing important information.

Then, the Douglas-Peucker algorithm was used which connects the curve's start and end points. Then, finds a point formed by all the connected points with the biggest distance from the line segments. While also considering the Euclidean distance formula and then comparing the distance retrieved with the selected and acceptable tolerance. And, if the accuracy tolerance is more than the distance, terminate the calculation if not, then it finds a simplified curve with fewer points. And then pass the result to an osculating circle calculation that detects the curves to approximate the line curvature at a specific point. The change of the tangent to the curve slope directly changes the radius.

Therefore, by drawing a circle on each point and running the equation using the three consecutive points of a circumscribing circle radius and the cumulative angle of three points; they are able to identify the curve.

Using the osculating circle, the authors had a positive predictive value of 95.9 percent which allows them to identify the curves correctly. The method is based on calculating the radius of the osculating circle and comparing it with three methods alternatively. But, the methods of calculating the radius of the curve significantly appeared better than those based on the lines of the break angle calculation. Table II shows the success rate of the given classified values for the boundary of the radii.

On the other hand, the determination of margins for identifying curves was left for further research hence it was not properly addressed in this work.

D. Detecting Road Junctions by Artificial Neural Networks

Nowadays, Road junction detection has become an important and essential factor of navigation systems while being

very difficult to detect. And, most of the early approaches were not successful. This paper presents a method for road junction detection using medium to a high-level resolution aerial images. This method should be able to deal equally well with three and four arm junctions based on raster and vector information. In this paper, image processing algorithms such as edge and line detection were used.

In [10], artificial neural networks (ANN) were used for object detection. Because neural networks have proven their universality in several technical fields such as image processing. They are used to recognize signalized control points in photogrammetric images and to detect airplanes in images as mentioned in many papers that used ANN. But the results were unsatisfactory so the Lanser algorithm [11] was added which is a rotation invariant version of the Deriche edge finder for edge detection. Then, the Ramer algorithm was used for edge smoothing and a circle centered on the junctions was used. Moreover, the Levenberg-Marquard algorithm was used because it's known for its high efficiency. Then, ambiguous image samples were removed due to wrong lighting condition issues, trees, and buildings causing the shadowing of the image which produced clean data. Then, the neural network was trained using that data from the images. Afterward, the ANN was left to detect the junctions. The detection of the junctions was accomplished by passing a predefined size window over the image. The result of the detection was either it's a junction or a non-junction. The window has a center pixel that the image stores the calculated radius of the circle drawn in it.



Fig. 4. Detected junctions in the other part of the study area [10]

An ANN classifier was used and a feed-forward neural network was chosen because it is frequently implemented in many programs. The testing was done using 0.4 M. resolution, black-and-white orthoimages which covered a region near Germany. The result was that the program was able to detect several potential junctions. All recognized junctions were marked with circles as shown in Fig. 4 and 5 and had their radius computed and any additional potential crossings

were marked with squares. It also detected bridges because they resemble crossing road segments. Finally, the advantages of this paper are that they made it easy to detect road junctions using ANN along with multiple other algorithms.

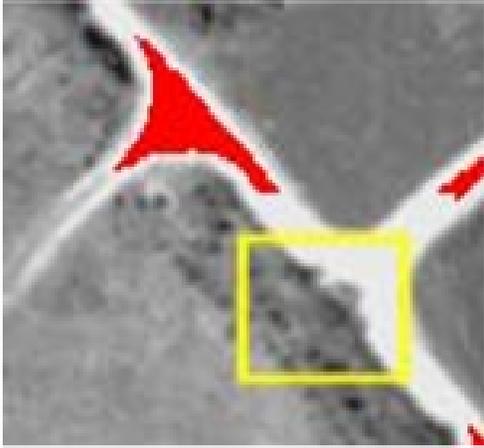


Fig. 5. Enlargement of figure 4 with tree shadows [10]

The disadvantages of this paper are that the ANN was trained on clean data to detect the junctions, which is inapplicable in real life. Additionally, the number of training samples was small which means that the result might not have been accurate. Although a larger training sample with more variety could have been used since one of the main advantages of ANN is its ability to handle large datasets. Consequently, it detected bridges as road junctions.

E. Road Junction Detection from 3D Point Clouds

In this paper [12], the main focus falls on how to detect the changing traffic conditions because they're of vital importance when it comes to the safety of autonomous cars that navigate urban environments. One of the most critical traffic problems that require a lot of attention is road junctions. This work presents several machine learning techniques in order to detect junctions including Support vector machines, Adaptive boosting, and Artificial neural networks. During operation, 3D range finders create frames, which are point clouds created at regular intervals and each of those frames contains a set of environmental readings from a specific and singular location. The Objective algorithm's main objective is to classify any single frame into either a road junction or a road. Three phases are required to accomplish this task which are: feature extraction, initial classification, and finally Structured Classification. A Laser rangefinder collects three-dimensional point cloud environmental data. Features are, then, extracted from the point cloud using the method described in [13]. This method was chosen because it is rotation invariant since cars might change directions at junctions. Then, the features are passed on to the second phase: classification which as mentioned before includes the ANN, SVM, and AdaBoost

TABLE III. TEST RESULTS [12]

Dataset	Classifier	Accuracy	Precision	Recall
SC2	ANN	0.7088	0.870	0.8100
	AdaBoost	0.8700	0.6478	0.8466
	SVM	0.7220	0.7854	0.4987
K2	ANN	0.8049	0.6606	0.6867
	AdaBoost	0.9027	0.8736	0.8139
	SVM	0.3626	0.8267	0.3009

to see which one will yield the best results. And then the third phase comes in which is Structured Classification

Artificial neural networks [14] are a nonlinear statistical machine learning method that is mostly used in pattern recognition. An ANN's objective is to adjust all the edges' weights and all the weights within each neuron, the weights of the previous layers are added and then they are put through an activation function like Sigmoid functions which restricts the output to either 0 or 1.

Adaptive Boosting [15] belongs to a family of methods called boosting that also trains boosted classifiers. Adaptive boosting uses dataset D for training, each iteration of t adds an extra layer to a t-1 layer classified f_{t-1} . These new layers that are constructed by Adaboost are each assigned a weight A_t to a possible weak classifier's hypothesis h, so that the error E_t is minimized:

$$E_t = \sum_{x \in D} E[F_{t-1}(\mathbf{X}) + \alpha_t h(\mathbf{X})] \quad [17] \quad (2)$$

The aforementioned process neglects any classifiers that don't improve the predictive percentage of the model which also helps avoid overfitting.

Support Vector Machine [16] is a supervised learning model that can be used for classification. Any data given to an SVM is represented in the form of points in a space. SVM then attempts to fit a hyperplane through the input space. And the hyperplane that leaves the largest distance between different classes is the best fit.

The result of using the aforementioned three classifiers which are ANN, AdaBoost, and SVM yielded the results shown in Table III.

The next step is the structured classification in which the sensor produces 10 frames per second so that a number of consecutive frames belonging to the same class can be taken. This is done so that the classifier is given noisy predictions. In the experiments and results phase, 2 separate tests were run, the first using Velodyne 32 laser collected by members of the Carina 2 project and the second using a Velodyne 64, which is a more advanced version of Velodyne 32, which generates denser point clouds with twice as many per frame.

The tests were done to compare the performance of the three base classifiers' used, each algorithm was trained and tested separately and the accuracy, precision, and recall were calculated.

TABLE IV. TEST RESULTS WHEN USING CROSSROAD DETECTION THROUGH CURB DATA AND CROSSROAD DETECTION THROUGH ROAD SURFACE DATA [17]

(A) 404-2			
		Predicted	
		CR	NCR
Actual	CR	340	22
	NCR	27	461
Accuracy		94.24 %	
MSE		0.11517	

(B) 404-5-2			
		Predicted	
		CR	NCR
Actual	CR	327	35
	NCR	25	463
Accuracy		92.94 %	
MSE		0.13592	

(C) 404-10-2			
		Predicted	
		CR	NCR
Actual	CR	345	17
	NCR	33	455
Accuracy		94.12 %	
MSE		0.11853	

(D) 210-2			
		Predicted	
		CR	NCR
Actual	CR	334	4
	NCR	1	112
Accuracy		98.67 %	
MSE		0.01941	

(E) 210-5-2			
		Predicted	
		CR	NCR
Actual	CR	328	10
	NCR	2	112
Accuracy		97.35 %	
MSE		0.05046	

(F) 210-10-2			
		Predicted	
		CR	NCR
Actual	CR	332	6
	NCR	1	113
Accuracy		98.45 %	
MSE		0.03331	

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad [12] \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

Where TP are the true positives, TN are the true negatives, FP are the false positives, and FN are the false negatives.

The Drawbacks of this paper is that only 3 specific classifiers were tested and not more which gave no room for the possibility of other algorithms that might have performed better.

F. Crossroad Detection using Artificial Neural Networks

According to [17], Autonomous ground mobiles may be the key to increasing traffic flow and highway capacity because they're capable of optimizing cars' navigation in the streets. Another one of their benefits is their ability to avoid collisions and therefore reducing road accidents. Autonomous vehicles must have a steadfast environment perception system to rely on, in order to safely navigate the streets and avoid collisions. This paper proposed an approach that takes advantage of our autonomous car's perception components to identify crossroads using a neural network. In order to detect curbs, obstacles are detected by analyzing the scan ring compression. Different obstacles have and create different compression patterns. For example, curbs make the sensor return distances between rings that are smaller than the ones flat terrain generates.

$$\begin{aligned} \Delta r_i^{plane} - r_{i-1}^{plane} \\ = \frac{h}{\tan \theta_i} - \frac{h}{\tan \theta_{i-1}} \\ = h((\tan \theta_i)^{-1} - (\tan \theta_{i-1})^{-1}) \end{aligned} \quad (6)$$

$$I_i = [\gamma \Delta r_i^{plane}, \delta \Delta r_i^{plane}] \quad [18] \quad (7)$$

The output of both the curb and the road surface detector were adapted to feed the neural network. And, in order to fix the number of points that will be inputted in the network, the points obtained from both detectors were placed in a grid comprising of cells that had the same dimensions. The first step of building the aforementioned grid is to store the number of points that fall in, in each cell. Then, this number is divided by the maximum number of points observed in the grid's cells Which results in a proportionate grid and limits any cell's value into a [0,1] range. So that cells with fewer points have lower values while cells with a larger amount of points have higher values. Training the aforementioned neural network for the classification of crossroads was performed in two ways: The first was by using road curb examples and the second was by using road surface examples. Each data set was separately trained in three different network topologies which are x-2, x-5-2, and x-10-2, where the input layer size x was 210 for road surface data and 404 for curb data. Then the dataset was split into 2 parts: training and testing. Training was assigned 66.6% of the dataset while testing was assigned the remaining 33.3% . And after testing the result were as follows:

Through the "Receiver operating characteristic", the cross-road classification performance comparison was conducted between curb and road surface data. 0.940589 was the AUC obtained for curb while 0.981209 was the AUC obtained for road surface data. In this method, it was deduced that using

the data obtained from the road surface was more appropriate in the verification of a crossroad's presence in the street as shown by the results in Table IV. And since the AUC was near 1.0, It was demonstrated that this classifier provided adequate true and false positives proportions. It was confirmed, by Analyzing distinct neural network topologies, that the single-layer perceptron brings higher accuracy and lower MSE to classify unseen patterns.

The drawbacks of this paper are that it only tested its data in a single-layer perceptron classifier. It, also, didn't test other classifiers; it only tested different data.

G. Parallel Hyper-Heuristic Algorithm for Multi-Objective Route Planning in a Smart City

According to [18], the commercial navigation application for route planning only focuses on parameters such as distance or time while completely neglecting safety which is one of the most important parameters. The safety parameter consists of many things like crime rate which are one of the risk issues found around the world. Most people, tourists in particular, often don't have information about safe or dangerous areas which is why people need an application that gives them the safest and fastest route. So, first, we must know the crime risk index which is extracted through several features, some temporally related and others spatially related. Spatially related features include the number of police stations and their proximity to a given region which has a high correlation and impact on a region's crime risk. while the temporally related features include traffic flow which influences urban safety. After extracting the spatial and temporal features, the next step begins which is to formulate the road network as multi-objective planning with the objective of getting the route that has the lowest crime risk index. The paper proposed a Multi-Objective Hyper-Heuristic algorithm for planning the route and, then, the CUDA framework was added to increase the speed of the planning. Finally, we get a safety-aware routing that takes the crime risk into account as well as the distance and time. Many experiments were conducted to compare the proposed algorithm with other similar algorithms. The results showed that the proposed algorithm was 173 times faster than the EMLS algorithm, 5.3 times faster than RL-MOHH, and 3.1 faster than the parallel NSGAIL.

III. CONCLUSION

Surveys with different work and efforts yielded different results with some algorithms being noticeably better than others at accomplishing specific tasks than others. In several papers [5], [10], [17], ANN seems to be the best solution for detecting junctions and straight road network sections. while DBScan combined with Grid-based Filtering [1] seem to give the best result with regard to interactivity and an application's reaction time. As a Future work, it would be extremely beneficial to add an accident Heat-map based on previous accident rates going back a certain number of years. as well as blocking off certain areas that have been known to have a high crime rate based on previous incident reports that can be retrieved from public records. And finally, using the official weather reports to avoid certain roads that do not function well under specific weather conditions.

REFERENCES

- [1] G. K. Habibullayevich, X. Chen, and H. Shin, "Efficient filtering and clustering mechanism for google maps," *Journal of Advanced Management Science*, vol. 1, no. 1, pp. 107–111, 2013.
- [2] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- [3] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering personally meaningful places: An interactive clustering approach," *ACM Transactions on Information Systems (TOIS)*, vol. 25, no. 3, p. 12, 2007.
- [4] K. Mumtaz and K. Duraiswamy, "An analysis on density based clustering of multi dimensional spatial data," *Indian Journal of Computer Science and Engineering*, vol. 1, no. 1, pp. 8–12, 2010.
- [5] Z. Li, I. V. Kolmanovsky, E. M. Atkins, J. Lu, D. Filev, and J. Micheline, "Cloud aided safety-based route planning," *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2495–2500, 2014.
- [6] P. de Leur and T. Sayed, "Development of a road safety risk index," *Transportation Research Record*, vol. 1784, pp. 33–42, 01 2002.
- [7] H. Wu and Z. Zhang, "A framework for developing road risk indices using quantile regression based crash prediction model," *Tech. Rep.*, 2012.
- [8] R. Andrášik, M. Bíl, Z. Janoška, and V. Valentová, "Identification of curves and straight sections on road networks from digital vector data."
- [9] J. J. Watkins, "Mathematical omnibus: Thirty lectures on classic mathematics by dmitry fuchs and serge tabachnikov," *The Mathematical Intelligencer*, vol. 32, no. 2, pp. 71–72, Jun 2010. [Online]. Available: <https://doi.org/10.1007/s00283-009-9101-7>
- [10] A. Barsi and C. Heipke, "Detecting road junctions by artificial neural networks," in *2003 2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas*. IEEE, 2003, pp. 129–132.
- [11] S. Lanser and W. Eckstein, "Eine modifikation des deriche-verfahrens zur kantendetektion," vol. 290, 01 1991, pp. 151–158.
- [12] D. Habermann, C. E. Vido, F. S. Osório, and F. Ramos, "Road junction detection from 3d point clouds," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 4934–4940.
- [13] K. Granström, T. B. Schön, J. I. Nieto, and F. T. Ramos, "Learning to close loops from range data," *The International Journal of Robotics Research*, vol. 30, no. 14, pp. 1728–1754, 2011. [Online]. Available: <https://doi.org/10.1177/0278364911405086>
- [14] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.
- [15] Z. Q. John Lu, "The elements of statistical learning: Data mining, inference, and prediction," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 173, no. 3, pp. 693–694, 2010.
- [16] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.
- [17] A. Hata, D. Habermann, D. Wolf, and F. Osório, "Crossroad detection using artificial neural networks," in *Engineering Applications of Neural Networks*, L. Iliadis, H. Papadopoulos, and C. Jayne, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 112–121.
- [18] Y. Yao, Z. Peng, and B. Xiao, "Parallel hyper-heuristic algorithm for multi-objective route planning in a smart city," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10307–10318, 2018.