Federated Deep Feature Extraction-based SLAM for Autonomous Vehicles

Christos Anagnostopoulos[1,2], Alexandros Gkillas[2] , Nikos Piperigkos[2]
, Aris S. Lalos[2]
[1]Department of Informatics and Telecommunication, University of Ioannina, Greece
[2]Industrial Systems Institute, Athena Research Center, Patras Science Park, Greece
{anagnostopoulos, gkillas, piperigkos, lalos}@isi.gr

***Abstract*—In this paper we investigate the impact of federated learning on deep learning-based feature extraction used for self-localization in autonomous vehicles. The accurate and reliable determination of the vehicle's position is crucial for both safety and efficiency purposes. Deep-learning based feature extractors have demonstrated benefits over model-based components and have started to replace them during the past few years. In addition, Federated Learning (FL) frameworks can provide advantages in terms of data privacy, resource allocations, and scalability. We apply FL principles on training a deep learning-based feature detector network and we evaluate the performance of the models by integrating them into a Simultaneous Localization and Mapping (SLAM) system and measuring the drift of generated estimated trajectories. The results showed us that the FL-based methods can perform similarly and sometimes even better in comparison to traditional centralized solutions.**

***Keywords*—SLAM, Deep Learning, Federated Learning, Autonomous Driving**

I. Introduction

Autonomous vehicles are one of the most significant technological advancements of our time and can play a pivotal role in transforming the transportation industry. One of their main anticipated advantages, is that self-driving cars have the potential to reduce the frequency of road accidents and significantly enhance road safety [1]. By incorporating a range of diverse sensors, autonomous vehicles can function as platforms that can respond with greater speed and precision than human drivers do. For autonomous vehicles, accurately and reliably executing self-localization is essential since it allows them to locate and orient themselves in an unknown area. SLAM is the process of estimating an agent's position and representing its surroundings using just raw sensory data, such as RGB images and LiDAR point clouds. The significance of this process is vast, since it guarantees safe vehicle operation and collision avoidance in new and unexplored environments. In the case of camera input these systems are called visual SLAMs (vSLAMs) and their efficiency is based on the extraction of robust and reliable interest point from the images. Deep learning-based feature extractors have started to replace model-based components during the past few years. On the other hand Federated Learning frameworks provide advantages in terms of data privacy, resource allocations, and scalability.

This work has received funding from the European Union's research and innovation programme TRUSTEE under grant agreement No 101070214.

979-8-3503-3959-8/23/$31.00 ©2023 IEEE

The present study aims to investigate whether the benefits of federated learning come at the cost of efficiency of vSLAM. More specifically, we created a synthetic multi-agent dataset and we have trained several models using both centralized and FL approaches. Then we evaluated these models on both synthetic and real data using the root mean square error of the absolute trajectory error, which is one of the most common evaluation metrics regarding measuring absolute drift of an estimated trajectory. The result showed us that the FL-based methods can perform similarly and sometimes even better in comparison to traditional centralized solutions.

The major contribution of this paper can be summarized in the following points:

- We propose a novel federated learning framework for deep feature extraction-based SLAM for autonomous vehicles, enabling connected vehicles to utilize diverse datasets from multiple sources. This approach enhances the derived model by incorporating data gathered from various locations and environmental conditions.
- We conduct extensive evaluations of the proposed federated learning schemes for vSLAM algorithms, using both synthetic and real-world datasets to thoroughly assess their performance and effectiveness. This comprehensive evaluation enables us to validate the benefits of our approach and compare it to existing centralized deep learning and traditional methods.

The remaining sections of the paper are structured as follows: Section II provides a short description of Deep Learning-based Visual Odometry and presents a series of works on Federated Learning in the Automotive, Section III provides a detailed description of the methodology followed, including the theoretical formulation of the SLAM problem, the presentation of the Deep Learning-based algorithm used and the Federated Learning SLAM, Section IV describes the experimental setup and the results of the experiments conducted, and finally, in Section V, we draw conclusions and summarize our findings.

II. Background

A. Deep Learning-based Feature Extraction

Visual odometry (VO) can be described as a process of estimating the motion of a moving object using only a sequence of images captured by a fixed camera as input. The

basic principle is to determine the displacement and orientation of the camera with respect to its initial position by examining the discrepancies between the photographs. It is a subset of the Structure From Motion (SfM) technique, which differs mainly in two ways: VO must meet real-time requirements and the images are arranged sequentially. The visual odometer alone provides poor results in estimating vehicle position because it cannot bound the drift of the trajectory. However, when combined with an optimization backend, the result is a system that can produce impressive results. These are the two building blocks of a vSLAM system. Depending on the method used by the visual odometer front end to correlate successive images, vSLAMs can be divided into direct and indirect systems.

Indirect or feature-based vSLAM systems share a common pipeline that involves the steps of finding, extracting, and matching geometric features. These features can be points, lines, or planes from images. This process is used to simultaneously determine the camera's position and create a map of the environment. Early feature detection and extraction techniques focused on finding corners, but these approaches often failed to provide accurate results. This led researchers to look for more reliable local image features that are invariant to scale. Today, modern VSLAM geometrical systems use feature extraction algorithms like SIFT [2], SURF[3], and ORB [4], which select robust features accompanied by a descriptor that's used for matching. However, geometry-based features rely heavily on low-level hand-designed features that don't always represent the complex real-world environment accurately. These features also struggle in highly dynamic or feature-poor environments. Deep learning-based techniques can offer more reliable and advanced features, and appear to be the logical evolution of vSLAM systems. Additionally, they have the capacity to understand abstract and comnplex features that it is very difficult to be extracted manually. Convolutional neural networks (CNNs), Recurrent Neural networks (RNNs), and autoencoders are examples of common deep neural networks that can be utilized for feature extraction. Example of such works are [5], [6], [7], [8], which share a common architectural approach, where a common deep learning-based backbone is followed by parallel branches that produce simultaneously interest points and descriptors for a given image.

B. Federated Learning in Automotive

Despite the extensive exploration of the federated learning framework in various fields, such as image processing and computer vision, its application in the autonomous driving domain remains significantly under-investigated. Existing literature presents a limited number of studies that explore the advantages of implementing federated learning in this context. For instance, study [9], utilized the federated learning framework to tackle the object detection problem in automotive scenes, achieving comparable results to centralized deep learning while accelerating local model training. Similar finding derived from studies in [10], [11], which employed the FedAvg approach aiming to design accurate local models to predict the wheel steering angle in self-driving vehicles. Moving on, studies [12], [9] provided a more theoretical analysis of the federated learning on vehicular networks, providing a discussion regarding the training procedure of the local models such as the data distribution and the non i.i.d. nature of the autonomous driving datasets. Finally, the study in [13] focused on semantic segmentation using federated learning. developing a benchmark platform with two datasets and multiple leading federated learning algorithms.

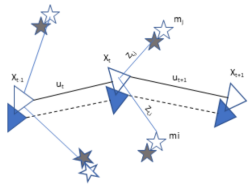

Fig. 1. The SLAM process involves making observations between the agent and real landmarks, since their true positions are not known. The estimated positions of the agent and observations are represented by filled triangles and stars, while the real values are depicted by empty shapes. The real vehicle's path is shown with continuous lines, while the estimated path is shown with dotted lines.

In contrast to previous works, the focus of our work is to investigate the advantages of Federated Learning for deep feature extraction tasks. Specifically, we aim to leverage a variety of data from connected autonomous vehicles that operate in diverse environmental and weather conditions. Our research seeks to explore how Federated Learning can enhance the effectiveness and efficiency of deep feature extraction in automotive scenes.

III. Methodology

A. SLAM problem formulation

SLAM problem, which was firstly theoretically described in [14], can be described as a process in which a mobile agent creates a map of the environment and localizes himself inside this environment without having any knowledge beforehand. It is also important to be noted that the whole procedure has to be executed on-line.

Let's consider a scenario where a mobile agent is moving through an environment and using a sensor mounted on it to take measurements [5]. At each time step, the system can be defined by the following parameters: the state vector, denoted by $\boldsymbol{x_t} \in \mathbb{R}^6$, which comprises the location and rotation of the agent, the control vector, denoted by $\boldsymbol{u_t} \in \mathbb{R}^6$, which includes the angular and linear velocity we need to apply at time $t-1$ to move to state $\boldsymbol{u_t}$, the location of each landmark, denoted by $\boldsymbol{m_i} \in \mathbb{R}^3$, and the observation of the i^{th} landmark, denoted by $\boldsymbol{z_{it}} \in \mathbb{R}^3$, taken at time instance t. For every instance t the following probability distribution has to be computed:

$$P(\boldsymbol{x_t}, \boldsymbol{m} | \boldsymbol{Z}_{0:t}, \boldsymbol{U}_{0:t}, \boldsymbol{x_0})$$

where $\boldsymbol{m}$ includes all the landmarks, $\boldsymbol{Z}_{0:t}$ is the set of all observations until t, $\boldsymbol{U}_{0:t}$ is the set of all the control vectors until t and $\boldsymbol{x}_0$ is the initial state of the agent.

B. Federated Learning SLAM

To formulate the proposed Deep SLAM federated learning problem, consider a network with N autonomous vehicles (agents) where contains a local private dataset. The agents aim to train local deep learning models for unsupervised SLAM by minimizing a local objective. Under the federated learning framework, the devices collaboratively train a global model

orchestrated by a central server, minimizing the aggregation of local objectives. This collaborative approach allows the model to learn and adapt more effectively to variations in terrain, lighting, and weather patterns, ultimately improving its overall performance and generalization capabilities.

1) Agent-side: On the edge-device side, each autonomous vehicle n leverages its private dataset to optimize a local deep learning model based on the SuperPoint architecture [5] for addressing the deep SLAM problem.

In more detail, the SuperPoint network employs a common VGG [15] based encoder to create a common representation of the original image that is scaled down by a factor of eight. This factor can be changed. The product of this operation is then fed into two parallel branches that simultaneously compute two outputs. The interest point decoder generates $\mathcal{X} \in \mathbb{R}^3$ which is then transformed into a tensor $\mathbb{R}^{HxW}$. The 65 channels are comprised of 64 non overlapping 8x8 regions of pixels plus one dustbin that represents no interest points. The final tensor is produced by applying a bit-wise softmax operation on $\mathcal{X}$ and discarding the dustbin. The second branch outputs $\mathcal{D} \in \mathbb{R}^{H_c \times W_c \times D}$ and then preforms a bi-cubic interpolation and a L2-normalization.

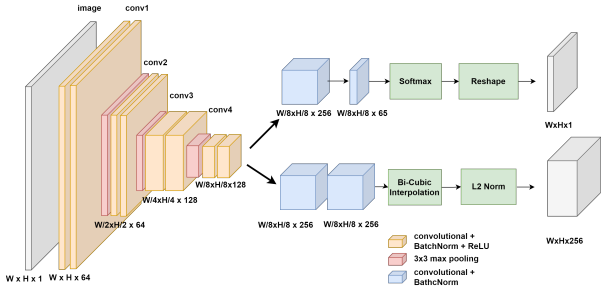

Fig. 2. Overview of the architecture of SuperPoint. The common VGG-based encoder is followed by two parallel branches. The upper branch produces the keypoints and the lower branch the descriptors.

The proposed system is fully self-supervised and operates on full-size images without extracting patches. It also introduces a homographic matching stage, in which pseudo-ground truth labels are generated by a network trained on online generated synthetic images of basic geometric shapes. The annotated dataset is then used to train the final model, which is based on the Siamese training scheme. In this method, a pair of synthetically distorted images connected by a randomly selected homography is used as input. For both images, we know the position of the interest points and the correspondences between them. The training loss is comprised of two parts, a fully convolutional cross-entropy loss of the feature detector and a weighted hinge loss of the feature descriptor. More specifically, the interest point detector loss, $\mathcal{L}_p$ over the cells $\boldsymbol{x}_{hw} \in \mathcal{X}$ is:

$$\mathcal{L}_p(\mathcal{X}, \mathcal{Y}) = \frac{1}{H_c W_c} \sum_{i=1,w=1}^{H_c,W_c} l_p(x_{hw}; y_{hw}),$$

where

$$l_p(x_{hw}; y_{hw}) = -\log\left(\frac{\exp(x_{hwy})}{\sum_{k=1}^{65} \exp(x_{hwk})}\right)$$

and $\mathcal{Y}$ are the ground truth labels generated in the Homography adaption step and H_c, W_c are the dimensions of the image divided by eight. The descriptor loss is calculated for every pair of descriptor cells $d_{hw} \in \mathcal{D}$ and $d'_{h'w'} \in \mathcal{D}'$ for the original and the warped image respectively. The formula that is used for the estimation of the loss is:

$$\mathcal{L}_d(\mathcal{D}, \mathcal{D}', \mathcal{S}) = \frac{1}{(H_c W_c)^2} \sum_{h=1,w=1}^{H_c,W_c} \sum_{h'=1,w'=1}^{H_c,W_c} l_d(d_{hw}, d'_{h'w'}, s)$$

where,

$$l_d(d_{hw}, d'_{h'w'}; s) = \lambda_d * s * max(0, m_p - \boldsymbol{d}^T \boldsymbol{d}') + (1 - s) * max(0, \boldsymbol{d}^T \boldsymbol{d}' - m_n)$$

and s equals to one when there is a match between the two patches and zero when there is not, m_p and m_n are positive and negative margins for the hinge loss and λ_d is a weighting term for balancing negative correspondences. Finally, the complete equation of the loss is given below:

$$\mathcal{L}(\mathcal{X}, \mathcal{X}', \mathcal{D}, \mathcal{D}', \mathcal{Y}, \mathcal{Y}', \mathcal{S}) = \mathcal{L}_p(\mathcal{X}, \mathcal{Y}) + \mathcal{L}_p(\mathcal{X}', \mathcal{Y}') + \lambda \mathcal{L}_d(mathcalD, \mathcal{D}', \mathcal{S})$$

2) Server-side: On the server-side, the main goal is to compute a global model by combining local models received from participating edge autonomous agents. Specifically, the server merges the local models ${\vartheta_n}_{n=1}^N$ to create a new global model, denoted as ϑ_g, using a weighted average fusion approach:

$$\vartheta_g = \frac{1}{N} \sum_{n=1}^{N} w_n \vartheta_i^m, \tag{1}$$

In equation (1), w_n denotes the size of the local dataset of the n-th device. After combining the local models, the centralized server distributes the new global model back to all devices. This process continues for T communication rounds until the global model reaches convergence.

IV. Experiments

The purpose of the conducted experiments was the validation of the proposed FL-based SLAM . Initially, we applied the FL principles on training SuperPoint networks using synthetic data on ten different and independent agents that correspond to ten vehicles. For each agent we used a training set of 3000 images and we also conducted a second training in which we limited the number of the training dataset to 1000 images. The batch size of each iteration was set to eight and each communication round consisted of 1000 iterations. Even though, the initial total number of round was set to 170, both models converged earlier, around round fifty. Furthermore, we trained two additional models. One model utilized a traditional centralized training approach where all data from all agents was accessible for training. The other model followed also an individual centralized training scheme but only had access to data from only one agent. The second part of our experiments involved the evaluation of the performance of the models when they are employed in the fronted of a SLAM system. For this purpose, we used a simple SLAM system and we evaluated the estimated trajectory for each model based on both synthetic

and real world sequences. We not only compared the models with each other, but also with ORB and SIFT, which are two well-known geometric algorithms widely used in relevant. literature.

A. Datasets

For the training of our models, we generated a dataset from Carla [16]. We spawned ten agents in Town03 of the simulator and we collected for each one of them a sequence of 3000 images, along with the corresponding ground truth. The extracted dataset follows the KITTI [17] odometry's dataset format. Regarding the evaluation of the models we have used both synthetic and real data. More, specifically we have used synthetic data from Carla [18] in the same environment as the training dataset. Additionally, we have used KITTI sequences in order to assess the ability of the model to generalize.

B. Metrics

In order to compare the SuperPoint models we utilized the validation loss using a separate dataset generated by Carla. Regarding the evaluation of the estimated trajectories we utilized the Absolute Trajectory Error (ATE), which measures the absolute deviation of the predicted pose from the ground truth pose. In the case of monocular vSLAM the estimated trajectory is given in different coordinate frame and scale than the ground truth. Therefore we have to execute and alignment[19], something which results to a 3D similarity transform $\mathcal{T}$. Let $\mathcal{P}$ be the estimated trajectory produced by the SLAM algorithm for n poses and $\mathcal{G}$ the corresponding ground truth, the ATE $\mathcal{E}$ for the frame $i \in [0, n]$ is $\mathcal{E}_i = \mathcal{G}_i^{-1}\mathcal{T}\mathcal{P}_i$, where $\mathcal{G}, \mathcal{P}, \mathcal{E} \in SE(3)$ and $\mathcal{T} \in Sim(3)$ which represent the Special Euclidean group and the Similarity group in three dimensions respectively. For all the time instances n, the Root Mean Square Error (RMSE) of the ATE is: $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\mathcal{E}_i^2}$.

C. Results

In figure 3 is demonstrated the convergence of the FL global model in the case of the reduced dataset, and its comparison to the highest accuracy achieved by both the centralized and individual training methods. The FL model can perform comparably to the centralized solution without off course the necessity of possessing a huge amount and at the same time exhibits a superior performance with comparison to the individual training model.

We evaluated the performance of three SuperPoint models integrated into a vSLAM system on five KITTI sequences and one Carla sequence using the RMSE of the ATE as a metric. For models $FL-3000$ and $FL-1000$ we followed the FL training paradigm, where each of the 10 agents had a training set of 3000 and 1000 images, respectively. The $client-indiv-learning$ model adopted an individual centralized training scheme, in which the agent had access only to his own dataset. We have also Incorporated the results of the experiments conducted using ORB and SIFT feature extractors in the frontend of the vSLAM. The results, which are depicted in figure 4, show a clear advantage of the deep learning based feature extractors compared to ORB and comparable, or even superior performance compared to SIFT. Furthermore, the FL-based models demonstrate a clear superiority over the individual model something that even though was expected is validated experimentally.

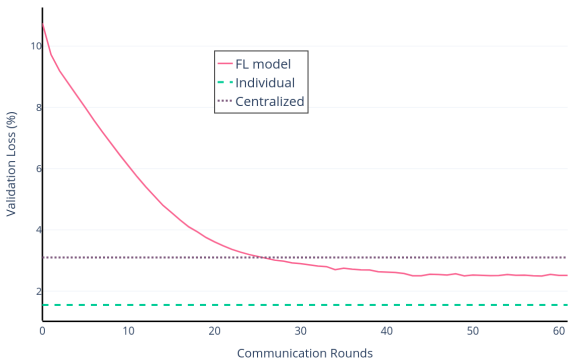

Fig. 3. The global model's validation loss resulting from the proposed FL approach in the case of 1000 images per agent is compared to the highest accuracy achieved by both the centralized and individual training methods.

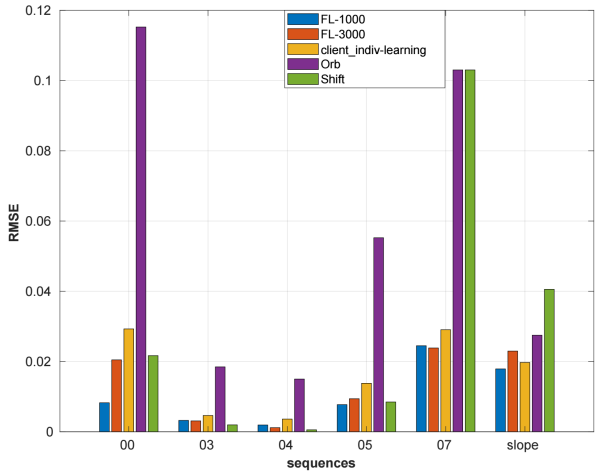

Fig. 4. The RMSE of the ATE for the five different front end configurations of a vSLAM on five KITTI sequences and on one Carla sequence

V. Conclusion

Accurate and reliable localization is extremely crucial for autonomous vehicles. In recent year deep learing-based techniques have become increasingly popular regrading the extraction of robust and reliable interest point from images. In addition, Federated Learning techniques can provide advantages in terms of data privacy, resource allocations, and scalability. We have a novel federated learning framework for deep feature extraction-based SLAM for autonomous vehicles, enabling connected vehicles to utilize diverse datasets from multiple sources. Futhermore, we conducted experiments evaluateing the proposed federated learning schemes for vSLAM algorithms, using both synthetic and real-world datasets. The result showed us that the FL-based methods can perform similarly and sometimes even better in comparison to traditional centralized solutions.

REFERENCES

[1] "Research on the impacts of connected and autonomous vehicles on traffic flow: summary report," https://assets.publishing.service.gov.uk, (Accessed on 03/29/2023).

[2] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.

[3] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Computer Vision – ECCV 2006*. Springer Berlin Heidelberg, 2006, pp. 404–417.

[4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[5] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, jun 2018.

[6] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2d2: Repeatable and reliable detector and descriptor," 2019.

[7] Z. Luo, T. Shen, L. Zhou, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, "Contextdesc: Local descriptor augmentation with cross-modality context," 2019.

[8] A. Barroso-Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, "Key.net: Keypoint detection by handcrafted and learned cnn filters," 2019.

[9] D. Jallepalli, N. C. Ravikumar, P. V. Badarinath, S. Uchil, and M. A. Suresh, "Federated learning for object detection in autonomous vehicles," in *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, 2021, pp. 107–114.

[10] H. Zhang, J. Bosch, and H. H. Olsson, "End-to-end federated learning for autonomous driving vehicles," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[11] A. Nguyen, T. Do, M. Tran, B. X. Nguyen, C. Duong, T. Phan, E. Tjiputra, and Q. D. Tran, "Deep federated learning for autonomous driving," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 1824–1830.

[12] S. Wang, C. Li, D. W. K. Ng, Y. C. Eldar, H. V. Poor, Q. Hao, and C. Xu, "Federated deep learning meets autonomous vehicle perception: Design and verification," *IEEE Network*, pp. 1–10, 2022.

[13] L. Fantauzzo, E. Fanì, D. Caldarola, A. Tavera, F. Cermelli, M. Ciccone, and B. Caputo, "Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 11 504–11 511.

[14] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 138–145.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," 2017.

[17] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[18] A. Kloukiniotis, A. Papandreou, C. Anagnostopoulos, A. Lalos, P. Kapsalas, D.-V. Nguyen, and K. Moustakas, "Carlascenes: A synthetic dataset for odometry in autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2022, pp. 4520–4528.

[19] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.