Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática
Especialidad en Sistemas Embebidos

Street Mapping of Traffic Signs through an Automated Intelligent Device

TRABAJO RECEPCIONAL que para obtener el **GRADO** de **ESPECIALISTA EN SISTEMAS EMBEBIDOS**

Presenta: **MARTHA ANGÉLICA MERCADO RAMOS, MARCELA ALEJANDRA ROSALES JIMÉNEZ, CRISTIAN SHAID DE JESÚS GARCÍA**

Asesor **MTRO. ÁLVARO IZAGUIRRE SERRANO**

Tlaquepaque, Jalisco. Julio de 2019.

Acknowledgements

The authors would like to thank M.Sc. Álvaro Izaguirre Serrano, co-author and technical advisor during the development of this project, whose continuous guidance lead us to the completion of this effort.

The authors would also like to thank CONACyT, who provided the scholarship to partially finance the development of this project. Scholarship numbers: 930774, 931793 and 935282.

Also, the authors thank ITESO University, who provided the infrastructure and guidance for the development of this project.

Finally, the authors thank the moral support of their families and friends during this year of research.

Abstract

Most commercial mapping services tend to use either data generated by them or data provided by external entities, usually at a price. Efforts that try to keep this information freely available are likely to depend on the help of volunteers, for instance the OpenStreetMap project. To support them, this document presents a system that reduces the work of mapping streets manually. It is a proof of concept of a street mapping device that captures video, keeps useful images, and identifies traffic signs. Due to the limited storage capacity, the device analyzes the images and discards the ones without relevant differences, for this it uses the Structural Similarity index to decide the images that will be stored. Then, it uses an AlexNext-based neural network to identify traffic signs and provide extra useful information about the streets, which later is added to the metadata of the images. This proof of concept illustrates a modular system flow to automate the street mapping task, which can later be extended and improved for more robust and accurate results. This system can be applied to generate more mapping information in cities, such as the Guadalajara Metropolitan area.

1. Introduction

Street mapping is an activity that requires economical resources and infrastructure to generate useful information for routing and research. This information can be managed through Geographic Information Systems (GIS) to produce useful maps and query information. Corporations like Google Maps, Waze or Apple Maps have resources and infrastructure to gather geographical data. Since these companies own the information they produce, they can decide how they expose and use this information.

Another approach for street mapping is through open source projects, as OpenStreetMap (OSM), whose aim is to obtain and distribute free geospatial data for anyone to use and share [1]. OSM shares information, such as points of interest, parks, and traffic signs. As in many open source projects, it is driven by volunteers who collect street images, tag points of interest, and associate a GPS location.

In the Guadalajara Metropolitan area, there is a lack of traffic sign information on OSM. One strategy to achieve a broader description of the city is through manual contributions to OSM servers, without the requirement of specialist on geospatial collection [2]. This strategy relies on individual contributions, though the work implied could be lessened by using an embedded system.

This paper proposes a software-based proof of concept of a street mapping device that can capture images and identify traffic signs within each image. The device is an embedded system that is intended to be mounted on a car, and once it is turned on, it will automatically capture street images. These images are associated with a GPS location, then, a computer vision algorithm identifies traffic signs. Due to the finite amount of hard drive resources, the software should determine what images should be kept and which ones could be deleted.

This document describes in Section II the elements of the street mapping device, starting with the system flow, the discard algorithm, the traffic sign identification, and lastly, the traffic sign dataset. Section III reports the results of the proof of concept, and Section IV describes the conclusions, and further challenges for a street mapping device.

2. Street Mapping Device

2.1. System architecture

The system architecture of the street mapping device is modular and sequential, as it is shown in “Fig. 2-1”. The flow begins with the capture of an image, which in this proof of concept, is a video recorded in the streets of the Guadalajara Metropolitan area. Then, a pre-processing stage converts the image to black and white, and reshapes it to be a 227 by 227 pixel matrix. The pre-processing is made to diminish the workload during the following steps: discard algorithm and object recognition. Later, a discard algorithm compares how similar the captured image is with the previous one, in order to decide whether to keep it or not. Finally, an object recognition neural network identifies a traffic sign and that information is added to the metadata of the image.

Fig. 2-1. System flow diagram

2.2. Discard algorithm

The street mapping device is intended to capture images for many hours during a car trip. It is assumed that during a journey, if images are captured on a fixed time frame, several pictures are going to be similar, and thus, they are not useful for mapping. In addition, a limitation of the proof of concept is the restricted hard disk space. Hence, to increase the work duration of the street mapping device, a discard algorithm based on the similarity of contiguous images was implemented.

Four approaches were analyzed to obtain a similarity measurement suitable for this proof of concept. Three were based on the feature detectors: Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), and KAZE. The last approach was based on the Structural

Similarity (SSIM) index. The algorithms were tested using an OpenCV framework implementation through Python programming language bindings.

SIFT, SURF and KAZE are feature detectors that are scale, rotation, and affine invariant [3]. For the experimentation of the proof of concept, these detectors were used to extract the feature key points of two contiguous images. Then, a feature-matching stage using k-nearest neighbors (KNN) algorithm determined the amount of common key points. Finally, this measurement was normalized to generate a percentage metric of similarity. In the fourth approach, the SSIM index, which is a metric of the image quality versus a reference image, was used [4].

2.3. Traffic sign identification

To correctly identify traffic signs on the road, the street mapping device uses as a basis the AlexNet convolutional neural network, which is 8 layers deep, and can classify up to 1000 object categories [5]. This neural network is comprised of layers, which are classified by the task executed in the process: convolution, non-linear, pooling, and classification. The convolutional layer calculates the features of the object on an image, learning specific patterns that will be later recognized. The output of this layer is a feature map. The non-linear Rectifier Linear Unit (ReLU) layer allocates the output of convolutional layers to an activation function. In this phase, all the pixels with a negative value are replaced by zero. Then, max-pooling layers reduce the dimensions of the input image to reduce the computational complexity. Since the output image is smaller, the network has a lower weight to compute. This reduction is usually executed by creating sub-matrices out of the input matrix (feature map), and the largest values inside the submatrix are for the output matrix.

All the layers are connected to each other by their neurons and the network generates a vector with the number of categories that it is trained to recognize. The probability of the image that belongs to a certain class is stored in each position of this vector, and the sum of the values is 100%. The neural network output refers to the object category with the highest probability. For this, the neural network is configured, and retrained using MatLab software. “Fig 2-2” shows the image of the retraining neuronal network procedure.

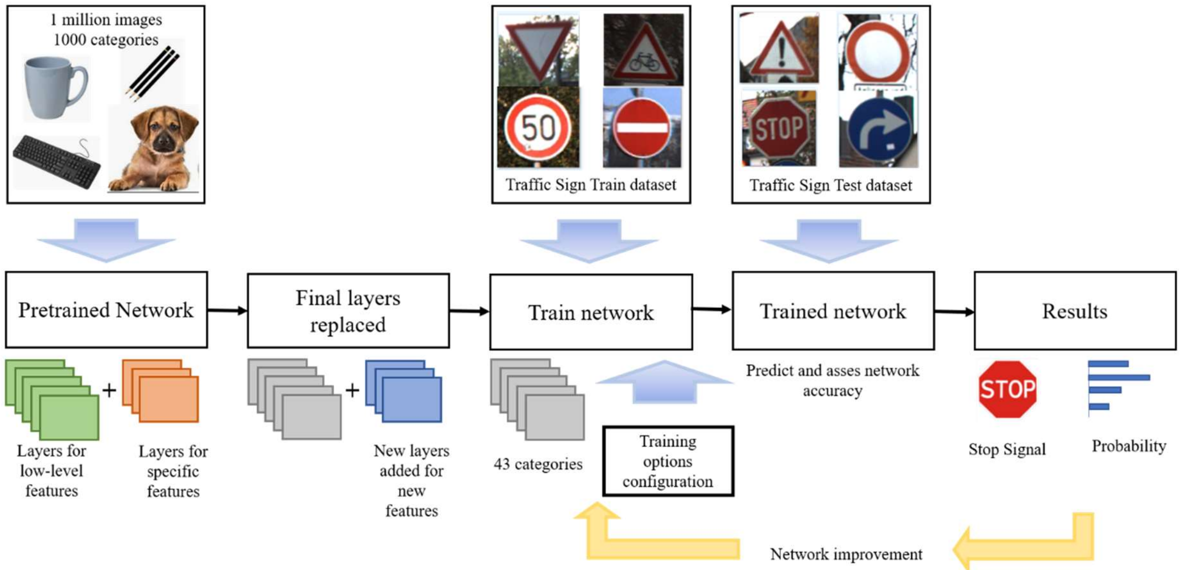

Fig. 2-2. Reuse of pretrained AlexNet convolutional neural network with new categories

2.4. German Traffic Sign Recognition Benchmark (GTSRB) dataset

The neural network needs to be trained with a local traffic sign dataset, however of the datasets analyzed, none had enough images of the signs located in Mexican streets. To bridge this gap, the GTSRB dataset [6] was selected because it has similar traffic signs to those used in Mexico. This dataset is public and was designed for a competition at the International Joint Conference on Neural Networks (IJCNN) in 2011, with the following properties: single-image, multiclass classification problem, more than 40 classes, and more than 50,000 images.

3. Results

3.1. Discard algorithm

It is assumed that contiguous frames have a high similarity between them, so the discard algorithm first selects one frame every ten frames so the frequency of the further steps will not overload the processing capacity. “Fig. 3-1” shows the flow of images of a 191-frame video after this first discarding phase, the algorithm reduced the video to 20 frames.

Fig. 3-1. Flow of pictures from a video recorded in the Guadalajara Metropolitan area

Then, the discard algorithm calculates a similarity percentage in order to decide whether to keep the new image or not. In order to calculate the similarity between two images, four methodologies were analyzed: SURF, SIFT, KAZE, and SSIM. After a visual analysis of each of them, it was decided that the SSIM index was the most suited for the proof of concept. The criteria was that a new image should have less than 70% of the SSIM score with respect to the previous image in order to continue with the system flow, otherwise it is disposed. “Fig. 3-2” shows the resulting images after the completion of the discard algorithm.

Fig. 3-2. Flow of pictures after executing the discard algorithm

3.2. Neural network

At this phase, before the image recognition assessment, the image is split into 36 sub-images, this in order for the algorithm to have the chance of identifying the traffic sign in each image. The AlexNet network takes each subimage and returns a label that corresponds to an identified object and its probability, the architecture of the network is shown in “Fig. 3-3”.

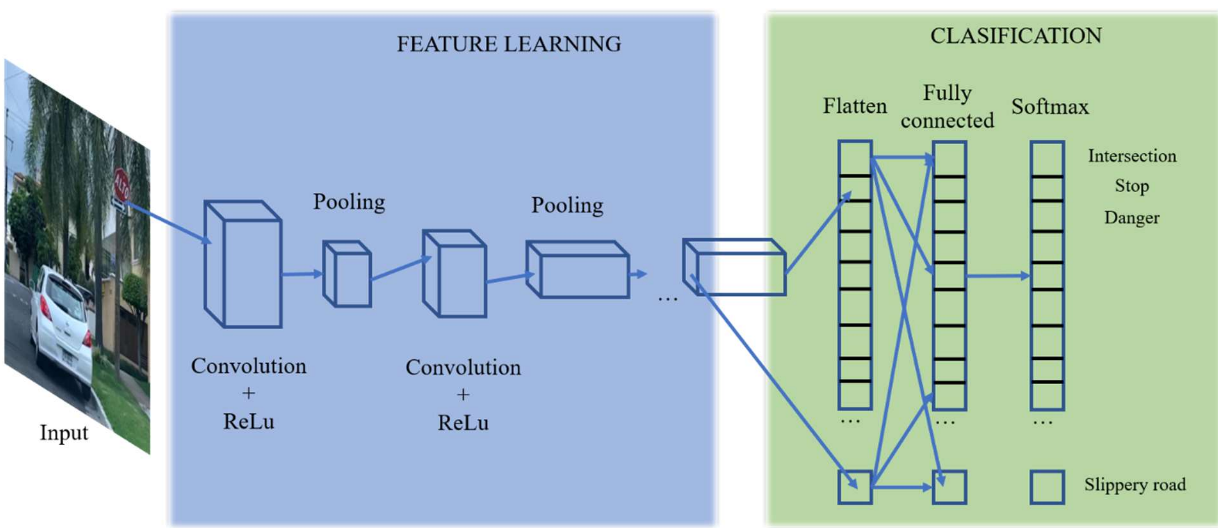

Fig. 3-3. Neural network

A neural network takes hyperparameters to determine its structure, and the way in which this is trained. Some of the parameters used are: classes or the objects that can be identified by the network, the Weight Learn Factor Rate or the parameter that controls how much the model is changed in response to the estimated error after each training cycle, and the Bias Learn Rate or a constant that helps the model to the best fit with any given data. In this proof of concept, the neural network that is pretrained was configured to be retrained, and thus, be able to classify new images. The final three layers were adapted changing the following parameters: 43 classes, a Weight Learn Factor Rate value of 10, and a Bias Learn Rate Factor equal to 20.

As it is shown in “Fig. 3-4”, the result of the neural network retrain is the ReLU graph, that shows the validation accuracy, the training time, epochs, iterations, and frequency. The validation accuracy result was 99.35% after 48 minutes, with 5 epochs, and 13720 iterations.

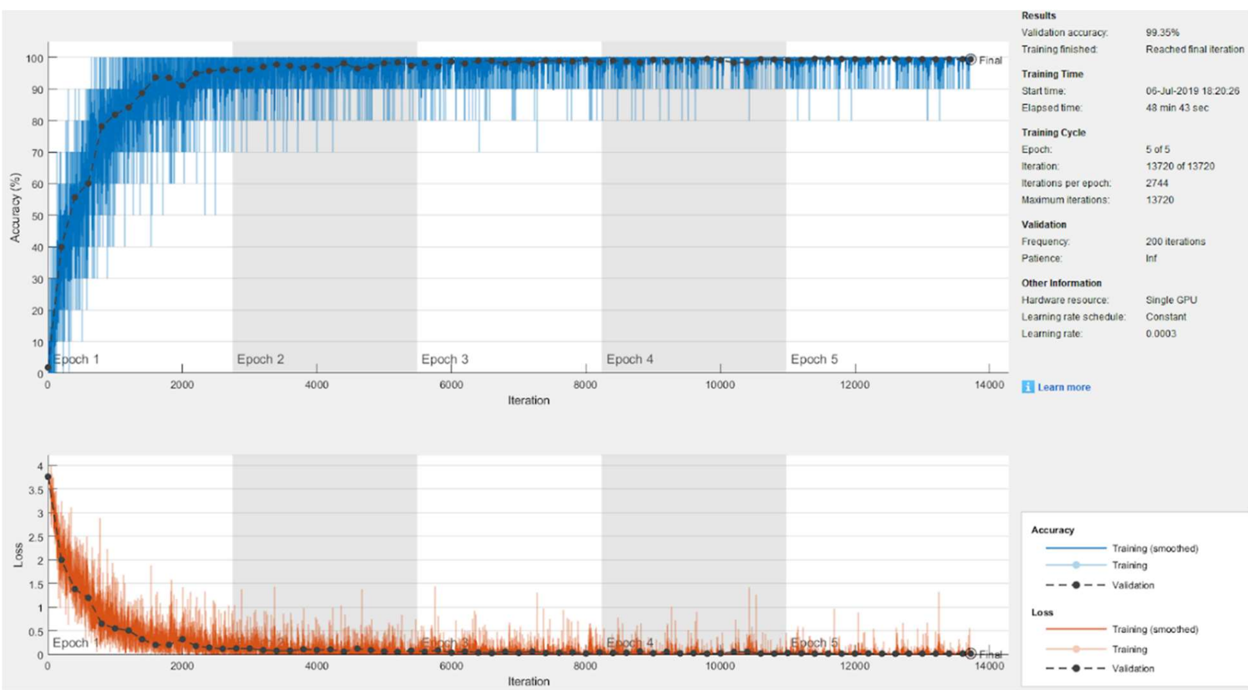

Fig. 3-4. Training cycles and its accuracy after each epoch of the training

Conclusions

A modular system is presented for the proof of concept of a street mapping device that could support OpenSourceMap volunteers in the task of mapping cities. The flow of the system begins with the capture, and preprocessing of an image. Then, a discard algorithm based on SSIM index discards images to optimize storage capacity. Finally, a street object recognition algorithm based on AlexNet neural network identifies if there is a traffic sign in the image.

This modular attribute of the proof of concept allows future improvements. For example, a segmentation algorithm that recognizes common traffic signs shapes could crop the appropriate shapes, so only one image is processed by the AlexNet network, instead 36 subimages as in the current implementation. Moreover, this proof of concept software could be optimized for an embedded device, so it could perform in real time. Also, a software module could upload the images to OSM servers from the device, without human intervention.

References

[1] M. Haklay and P. Weber, “OpenStreetMap: User-Generated Street Maps,” IEEE Pervasive Computing, vol. 7, no. 4, pp. 12–18, Oct. 2008.

[2] A. Basiri et al., “Quality assessment of OpenStreetMap data using trajectory mining,” Geo-spatial Information Science, vol. 19, no. 1, pp. 56–68, Jan. 2016.

[3] S. A. K. Tareen and Z. Saleem, “A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK,” in 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2018, pp. 1–10.

[4] K. N. Raju and K. S. P. Reddy, “Comparative study of Structural Similarity Index (SSIM) by using different edge detection approaches on live video frames for different color models,” in 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2017, pp. 932–937.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017.

[6] “German Traffic Sign Benchmarks.” [Online]. Available: http://benchmark.ini.rub.de/index.php?section=gtsrb&subsection=dataset#Downloads. [Accessed: 02-Jul-2019].