Intelligent Method for Counting Cars from Satellite Images

Myroslav Komar [1], Ruslan Savchyshyn [2], Khrystyna Lipianina-Honcharenko [1] and Oleksandr Osolinskyi [1]

[1] *West Ukrainian National University, 11 Lvivska St, Ternopil, 46009, Ukraine*

[2] *MagneticOne Group, 11211 Taylor Draper Ln #115, Austin, TX 78759, USA*

Abstract

The aim of Intelligent parking systems in cities is to enable quick identification of available parking spaces. In this project, an intelligent method for vehicle counting using satellite imagery and machine learning was developed. An existing approach was analyzed, and the Amazon SageMaker platform which is based on Geodatas of Ternopil, Ukraine was chosen for implementation. The system was tested from 2004 to 2019, and during this period, 28,579 vehicles were successfully identified. The forecast for the year 2030 indicates a significant (critical) increase in both the number of vehicles and the amount of CO2 emissions.

Keywords

parking spaces; neural network, AWS, Amazon SageMaker

1. Introduction

Analyzing the number of cars in big cities, can see that it is constantly growing. The transport industry is one of the large sectors which determines the development of the industry as a whole and agriculture in any country, including the European Union (EU). One car consumes an average of 1 ton of oxygen per year and emits about 600-800 kg of carbon dioxide, 40 kg of nitrogen oxides, and 200 kg of unburned hydrocarbons [28]. The intensive growth of vehicles in the European Union over the past decade has contributed to the economic development of countries and their integration but is accompanied by a negative impact on the environment and human health.

The relevance of this topic is in the fact that the development of the intelligent method for counting cars from satellite images based on machine learning will give the possibility to easily and quickly determine the number of cars in a locality, and calculate the amount of emissions.

This article is presented in the following structure: Section 2 discusses the analysis of related works, and Section 3 presents an Intelligent method for counting cars from satellite images based on machine learning. Section 4 presents the implementation of the method, and Section 5 is conclusions.

2. Related Work

Given to account the increase in urban population and traffic jams, smart parking is always a strategic issue [1] that should be addressed not only in the research field but also from economic interests. A work [3] discusses the problem of predicting the number of free parking spaces in a parking space. Paper [13] proposes a model that predicts the availability of parking spaces in real-time based on the movement of vehicles in a supermarket parking space. The work [6] proposes an alternative localization technique based on the presence of vehicles in the neighborhood and known fixed infrastructure, such as common radio access points. Article [2] presents an intelligent parking space detection system based on image processing techniques that captures and processes a brown

International Scientific Symposium «Intelligent Solutions» IntSol-2023, September 27–28, 2023, Kyiv-Uzhhorod, Ukraine
EMAIL: ruslan@magneticone.com (A.1); xrustya.com@gmail.com (A.2); oso@wunu.edu.ua (A.3); mko@wunu.edu.ua (A.4)
ORCID: 0000-0001-6541-0359 (A.1); 0000-0002-2441-6292 (A. 3); 0000-0002-0136-395X (A. 4)

rounded image drawn in a parking space and provides information about empty parking spaces. Authors of this work [4] present a new methodology based on deep learning using recurrent neural networks to predict parking space occupancy. In [12], the number of parking spaces was modeled using many types of regression. Paper [14] proposes a new convolutional hybrid model capable of capturing long-term time dependencies on two types of data - on-street and off-street parking. Authors of [7] propose a new framework based on a recurrent network that uses a long-term short-term memory (LSTM) model to predict parking spaces several steps ahead. The main idea is that both the level of occupancy of on-street parking in a certain region and the probability of a car leaving is used as a metric for forecasting efficiency.

The authors of [8] proposed to use data from various sources (parking data, pedestrian data, traffic data) to predict free parking spaces at fifteen-minute intervals. The authors investigated the relationship between the number of pedestrians and parking demand in specific neighborhoods. This data was then used to predict conditions on holidays and during special events, when the number of pedestrians increases dramatically.

In [9], the authors focused on freight transportation within the city, collecting and analyzing data on urban freight transportation and parking areas for an optimized urban freight transport system. Articles [5, 10] studied parking occupancy data published by the city councils of Birmingham, Glasgow, Norfolk, and Nottingham to test several forecasting strategies (polynomial fitting, Fourier series, K-means clustering, and analysis of their results). In this paper [11], the authors focused on curbside parking and developed a systematic framework called Curb Parking Demand Estimation (CPDE), which allows modeling the demand for public parking in cities, taking into account the duration of parking and regional characteristics. The authors of [16] presented an effective method implemented using the cloud environment to check the availability of parking spaces and reservations using the "smart parking" approach with the possibility of booking.

For intelligent vehicle detection and counting, works such as [23] are devoted to the field of vehicle detection and counting. It proposes a system for detecting and counting vehicles based on machine vision. The authors of [24] have developed a framework that integrates computer vision and traffic modeling to link real transportation systems and working virtual traffic models to optimize signal synchronization at multiple intersections.

Paper [25] aims to provide a simpler, more accurate, and less expensive solution for traffic management using deep neural network (DNN) methods, namely Faster R-CNN, Mask R-CNN, and ResNet-50, for vehicle detection, classification, and counting. This work [26] proposes a method for counting vehicles based on the mechanism of single object detection using attention (SSD) and state detection. An efficient approach for vehicle counting based on double virtual lines (DVLs) is presented in [27]. There are also several close analogues [23, 26, 27] that analyze the application domain, but they do not provide the ability to recognize objects from satellite images and, based on this data, predict the amount of CO_2 emissions.

Also, companies that own servers are trying to develop platforms that allow automating the machine learning process: AWS SageMaker [15]; Azure ML Studio [17]; IBM Watson Studio [18]; Google Cloud AutoML [19]; Oracle AutoML Pipeline [20]; Dataiku [21]; DataRobot [22].

3. Proposed method

An intelligent method for counting cars from satellite images based on machine learning can be described in several steps:

Step 1. Data preprocessing (Figure 1): This stage includes the collection of satellite images of the selected area.The images are then divided into squares with defined boundaries to simplify processing and analysis. Data preprocessing, especially when working with satellite imagery, is an important process that usually includes the following steps:

1.1. Image collection: First of all, a series of satellite images are collected from information providers such as NASA, ESA or commercial providers such as Google Earth.

1.2. Geometric correction: Images require geometric correction to correct any distortions caused by satellite movement, shooting angle, etc. This process usually involves image registration algorithms that align the image with a map or other images.

1.3. Normalisation: It is also important to normalise the intensity of the pixels, especially when using images captured at different times of the day or year when the lighting may vary. This typically involves calculating the mean and standard deviation of the pixel intensity and transforming each pixel so that its intensity is relative to the mean intensity.

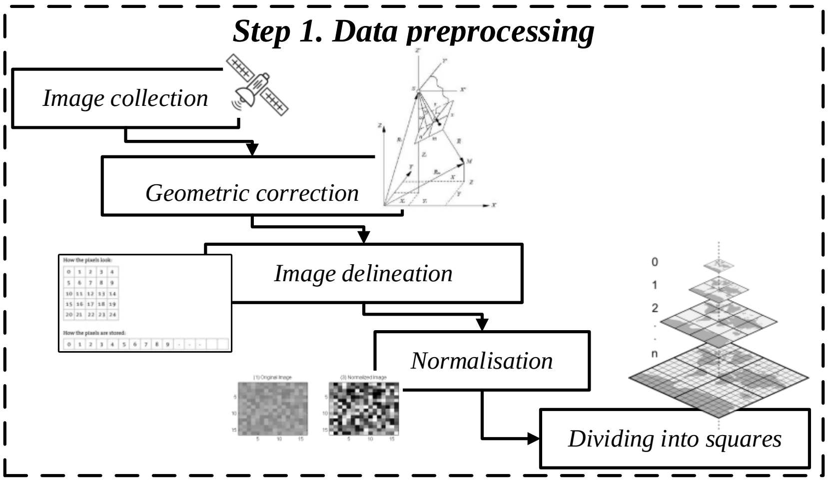

Figure 1. Data preprocessing

1.4. Dividing into squares: In order to simplify data processing and improve the efficiency of algorithms, large satellite images are divided into smaller squares, or "tiles". The size of these tiles can vary greatly depending on the task and computing resources, but it is usually chosen so that each tile contains enough information for analysis (e.g. 256x256 pixels).

The images processed using this method can then be used to train a machine learning model or perform other analytical procedures.

Step 2: Training the machine learning model (Figure 2): Using a limited number of marked images (i.e. images where cars have already been identified and marked by humans), a machine learning model (e.g. deep neural networks such as convolutional neural networks) is trained.

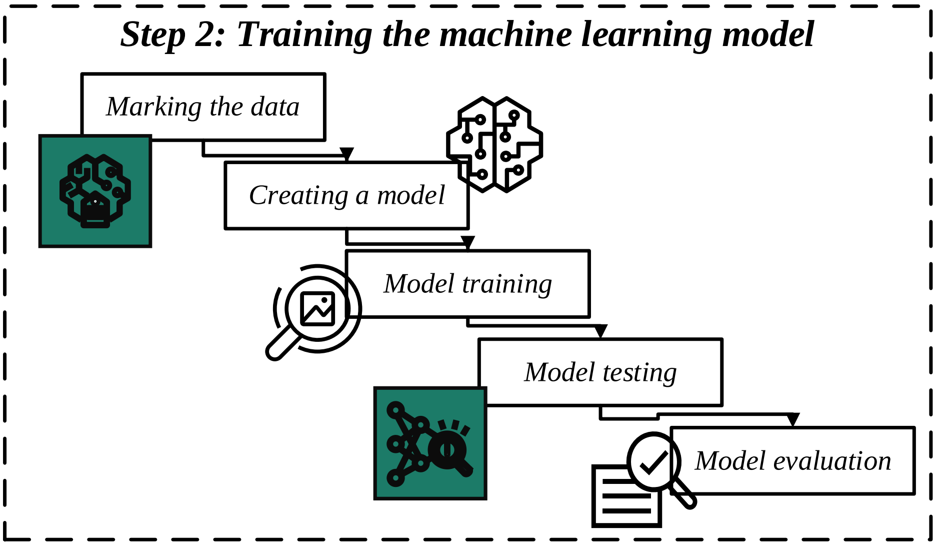

Figure 2. Training the machine learning model

This model is trained to detect the presence of a car in the image. The process of training a machine learning model, in particular a deep neural network (DNN) such as a convolutional neural network (CNN), can be considered as follows:

2.1. Marking the data: First of all, a marked data set is required, where each image has a corresponding label indicating whether a car is present in the image. Marks can be binary (e.g., 1 if the car is present and 0 if it is not), or they can be more complex, such as indicating the position of the car in the image.

2.2. Creating a model: A CNN model is included several layers. The first layers, commonly known as "convolutional layers", are used to identify key features in the image, such as edges, textures, colors, etc. Each convolutional layer consists of several filters, each of which defines a specific feature. Next layers, known as "fully connected layers", are used to combine these features and determine the final mark.

2.3 Model training: The model is trained by inputting the marked images into the model and adjusting the weighting factors to minimize the difference between the predicted marks of the model and the actual marks. This process is known as "backward error propagation" and usually involves the use of optimization algorithms such as stochastic gradient descent.

2.4. Model testing: After training the model on the training dataset, it needs to be validated on a separate dataset known as the "test set". This helps to ensure that the model can generalize its learning to new data.

2.5. Model evaluation: The model is evaluated using various metrics such as precision (proportion of correctly classified images), recall (proportion of true positive cases that were correctly identified), accuracy (proportion of cases that were correctly identified as positive), F1 metric (harmonic mean between precision and recall), and others.

Step 3. Vehicle detection (Figure 3): Once the model has been properly trained, it can be applied to a large set of unlabeled satellite images.

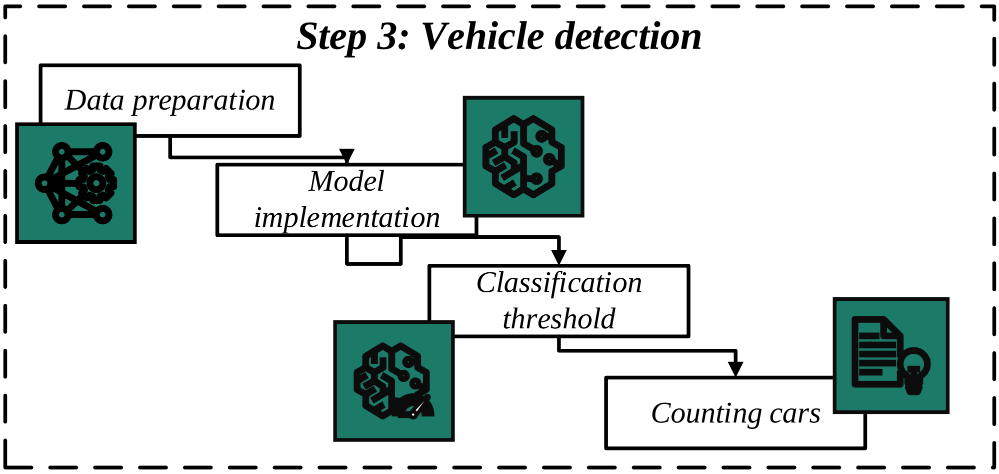

Figure 3. Vehicle detection

The model looks at each square of the image, determines whether a car is present, and counts the number of cars. Once trained, a machine learning model (e.g., a convolutional neural network) can be applied to a new set of satellite images to detect cars. The process of using the model for car detection can be described as follows:

3.1. Data preparation: New images to be applied to the model are prepared in the same way as the training images. This means that the images are divided into squares (or "tiles") of the same size as in the training set.

3.2. Model implementation: The machine learning model is then applied to each tile. It takes the tile as input and produces an output that indicates the probability of a car being present on the tile.

3.3 Classification threshold: The probability produced by the model is usually compared to a certain threshold to determine whether a tile contains a car. For example, if the model produces a probability of 0.7 and the threshold is 0.5, then the tile is classified as containing a car.

3.4. Counting cars: The number of cars in the image is calculated by counting the number of tiles that were classified as containing a car.

This process is repeated for all new images.

Step 4. Emissions analysis and forecasting (Figure 4): Once the number of cars on the maps is determined, this information can be used to calculate the current level of emissions.

This can be done using different emission models for different types of vehicles. The data (Table 1) can also be used to predict future emissions levels based on trends in vehicle numbers, whether they are increasing or decreasing.

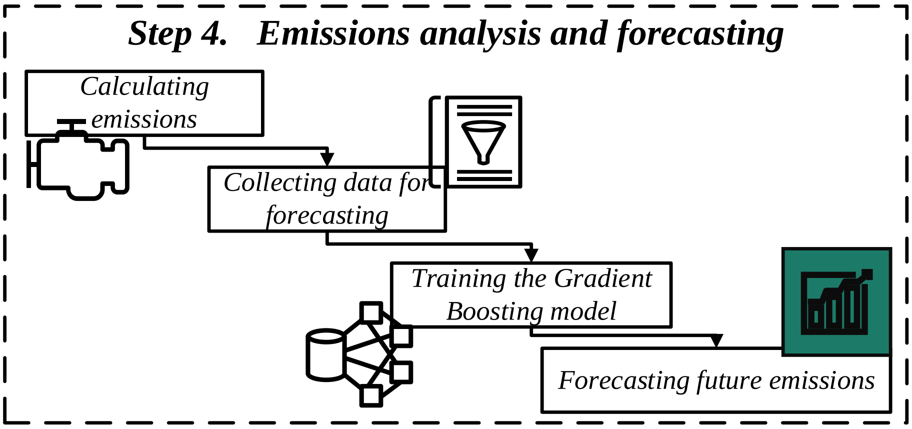

Figure 4. Emissions analysis and forecasting

Table 1. Average CO2 emission in Ternopil

Year	Numbers of cars	CO2 tones
2004	28579	71447,5
2005	28931	72327,5
2006	29821	74552,5
2007	33211	83027,5
2008	36601	73202
2009	39992	79984
2010	43382	86764
2011	45975	91950
2012	49379	98758
2013	57783	104009,4
2014	59186	106534,8
2015	62590	112662
2016	65893	98839,5
2017	67530	101295
2018	69405	104107,5
2019	70258	105387

Emissions forecasting and analysis can be done using statistical methods and machine learning algorithms such as Gradient Boosting. The steps of this stage are presented below:

4.1 Calculating emissions: To calculate current emissions, a model is applied that relates the number of cars to CO2 emissions. This model can be based on various parameters, such as the average age of cars, average mileage, engine type, etc. For example, can use data from official sources or scientific research to determine the average CO2 emissions per car per year.

4.2. Collecting data for forecasting: To forecast future emissions, a dataset is used that includes a time series of vehicle counts (or calculated CO2 emissions) for previous time periods. Other parameters can also be included, such as economic indicators, demographic information, policy and climate information, and so on.

4.3. Training the Gradient Boosting model: Gradient Boosting is a machine learning technique used to make predictions based on previous data. It creates an ensemble of simple models (often decision trees), each of which attempts to correct the errors of the previous model. The model learns by minimising the loss through gradient descent. The sequence of steps is described below:

4.3.1 Initialisation: Starts with a simple model. This can be any model, but often a very simple model is used that simply predicts the average value of the target variable. It serves as a baseline for subsequent models.

4.3.2 Boosting: In the next step, create a new model, that attempts to correct the mistakes made by the previous model. That provides training a new model on the "leftovers", or errors, of the previous model. For example, if the previous model predicted that CO2 emissions would be 10 tonnes, but the actual value was 8 tonnes, the residuals would be -2 tonnes.

4.3.3 Combining models: The new model is combined with the previous model to create a more accurate model. This is realized by adding the predictions of the new model to the predictions of the previous models. In addition, each new model is trained in small steps (often known as the "learning rate") to avoid overtraining.

4.3.4 Iteration: This process is repeated many times (often hundreds or thousands of iterations), each time creating a new model that attempts to correct the errors of previous models. The final model is simply the sum of the predictions of all the individual models.

Gradient Boosting takes into account model losses or errors and uses a gradient descent method to minimise these losses. During the gradient descent process, the model parameters are continuously adjusted to reduce the overall model error until the best possible parameters are reached.

4.4. Forecasting future emissions: Once the model has been trained, it can be used to predict future emissions based on the latest available data.

This approach allows for a variety of factors and patterns to be incorporated into the predictions, making it a very powerful tool for emissions analysis and prediction and vehicle detection.

This method can be useful for a variety of applications, including determining the effectiveness of emission reduction policies, monitoring urbanization trends, and developing traffic management systems.

4. Experimental Results and Discussion

To implement the described method, satellite images of Ternopil City were collected from 2004 (Fig. 5) to 2019 (Fig. 6). Each image was divided into sectors (or squares). This was chosen to simplify the analysis, as each sector can be processed separately.

To implement the described method, the Amazon SageMaker platform was chosen. Satellite images of Ternopil City from 2004 to 2019 were uploaded to Amazon S3 for further processing. Amazon SageMaker tools and services, such as Jupyter Notebook and Amazon Sagemaker Ground Truth, were used to divide the images into sectors and mark cars on the images.

After that, a machine learning model was built using Amazon SageMaker Autopilot, which allowed the model to automatically train to recognise cars on satellite images.

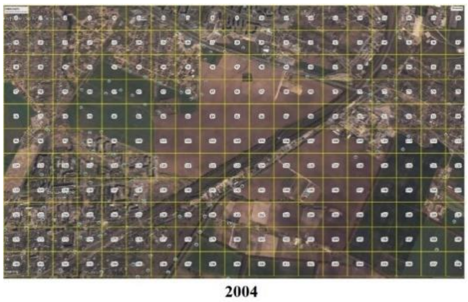

Figure 5. A snapshot from 2004

Figure 6. A snapshot of 2019

Using a trained machine learning model on the number of marked cars in each sector recognized cars in each sector (Figure 7). The number of cars was tracked over the period 2004-2019.

Based on the car count data, a change index was built for each sector. This index shows how much the number of cars in each sector has changed over the period. The change index was visualized on a map (Figure 8), where darker squares correspond to larger changes. Once the number of cars has been determined for each sector, these data can be summed to give the total number of cars in Ternopil in 2004 (Figure 9). As a result, 28,579 cars were recognised in 2004. In this way, it is possible to track how the total number of cars has changed over time and to make predictions about the future number of cars in the city.

Figure 7. Car recognition in a square

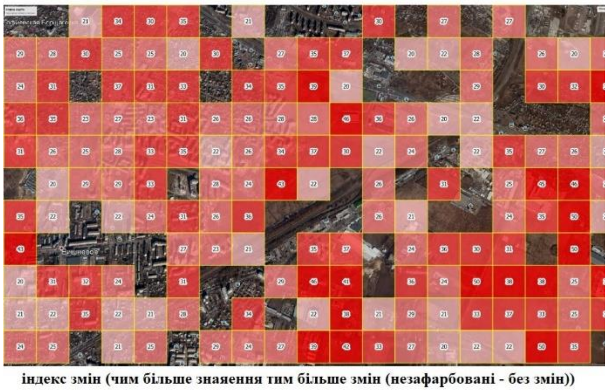

Figure 8. Index of changes

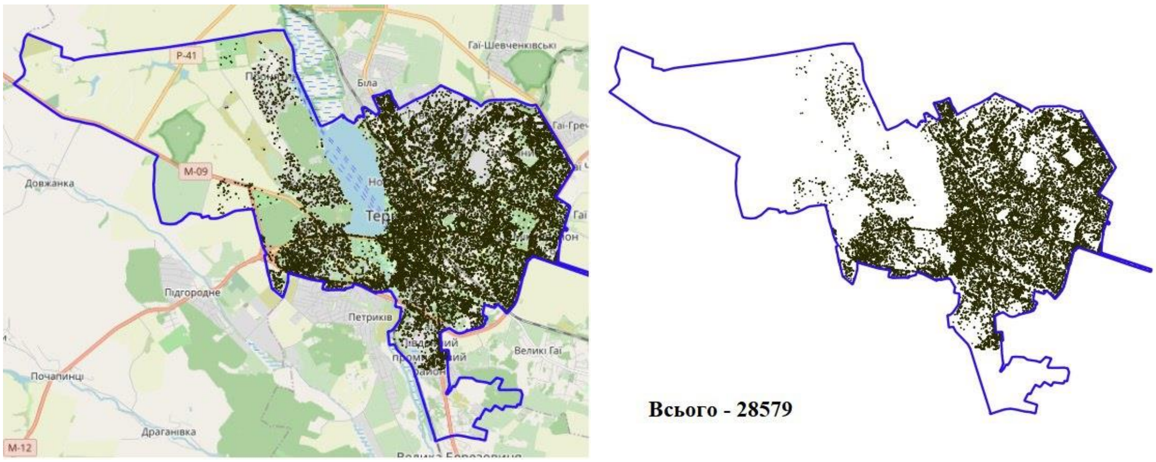

Figure 9. The total number of cars in Ternopil city

The Gradient Boosting Regressor model was used to forecast the amount of CO2 emissions and the number of cars in Ternopil for the period from 2020 to 2030.

The following conclusions can be drawn from the analysis of the predicted number of cars (Figure 10) and CO2 emissions (Figure 11) for the years 2004-2025:

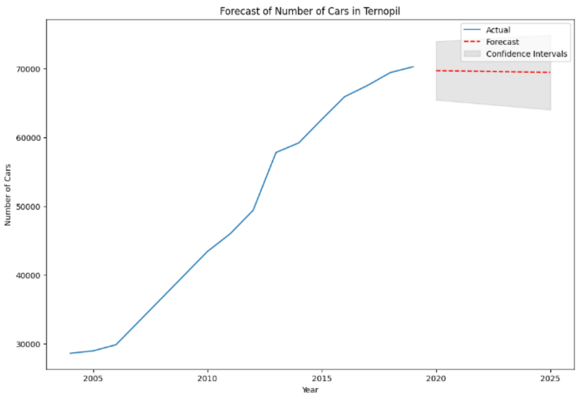

Figure 10. Forecast of Number of Cars in Ternopil

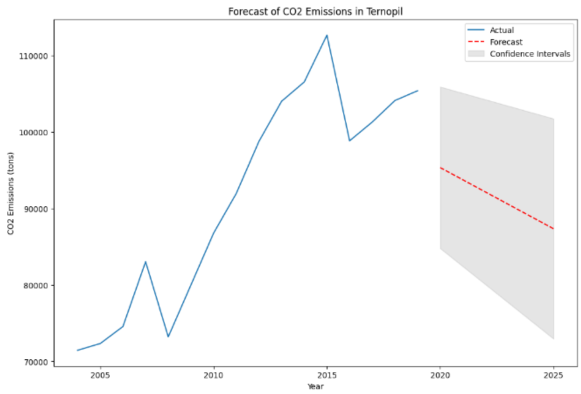

Figure 11 Forecast of CO2 Emissions in Ternopil

- The number of cars and CO2 emissions were based on actual data from 2004 to 2019.
- The forecasted vehicle numbers for the years 2020-2025 are increasing over time.
- The forecasted CO2 emissions also increase over time, although some years may have smaller increases compared to other years.

The forecasted CO2 emissions for 2020-2025 show a general upward trend but with deviations in different years. Emission values may vary from year to year due to the introduction of CO2 emission standards, such as EU regulations, which force car manufacturers to reduce emissions.

Given to account the forecasts, can expect an increase in the number of cars and, consequently, an increase in CO2 emissions in Ternopil in the future. The values of these forecasts can be used for planning and decision-making on sustainable development of the transport system and reducing environmental impact. The developed method combines two key functions - object recognition on satellite images and forecasting the number of cars in the context of emissions accounting and analysis. Thus, the developed method opens up new possibilities for accurate vehicle counting and emissions forecasting, which has significant potential in many areas requiring traffic flow monitoring and environmental sustainability.

5. Conclusions

In this work, a method was developed that combines car detection in satellite images with car count and CO2 emissions forecasting. For this purpose, satellite images of the city of Ternopil from 2004 to 2019 were used, which were divided into quadrants for further analysis. In addition, the Gradient Boosting Regressor model based on time series analysis was used to forecast CO2 emissions. The actual and projected data show an increase in the number of cars and CO2 emissions in Ternopil. These predicted values show a stable trend to increase but with variations in different years. Thus, the developed method can be useful for car counting and forecasting emissions based on satellite images. This can be important for monitoring traffic flow, infrastructure planning, and developing environmental strategies to reduce the environmental impact of vehicles. Possible further research could include analysing the impact of traffic on air quality and developing emission reduction strategies, studying the impact of infrastructure on car traffic, and investigating the effectiveness of environmental policies in the city.

6. References

[1] T. Lin, H. Rivano and F. Le Mouël, (2017) , A Survey of Smart Parking Solutions, in IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 12, pp. 3229-3253, doi: 10.1109/TITS.2017.2685143.

[2] Yusnita, R., Norbaya, F., & Basharuddin, N. (2012). IEEE Intelligent parking space detection system based on image processing. International Journal of Innovation, Management and Technology, 3(3), 232-235.

[3] Klappenecker, A., Lee, H., & Welch, J. L. (2014). Finding available parking spaces made easy. Ad Hoc Networks, Elsevier, 12, 243-249.

[4] Camero A., Toutouh J., Stolfi D.H., Alba E. (2019) Evolutionary Deep Learning for Car Park Occupancy Prediction in Smart Cities. In: Battiti R., Brunato M., Kotsireas I., Pardalos P. (eds) Learning and Intelligent Optimization. LION 12 2018. Lecture Notes in Computer Science, vol 11353. Springer, Cham, 386-401 https://doi.org/10.1007/978-3-030-05348-2_32

[5] Stolfi D.H., Alba E., Yao X. (2017) Predicting Car Park Occupancy Rates in Smart Cities. In: Alba E., Chicano F., Luque G. (eds) Smart Cities. Smart-CT 2017. Lecture Notes in Computer Science, vol 10268. Springer, Cham, 107-117. https://doi.org/10.1007/978-3-319-59513-9_11

[6] W. Balzano and F. Vitale, (2017), DiG-Park: A Smart Parking Availability Searching Method Using V2V/V2I and DGP-Class Problem, IEEE 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, Taiwan, pp. 698-703, doi: 10.1109/WAINA.2017.104.

[7] Shao W., Zhang Y., Guo B., Qin K., Chan J., Salim F.D. (2019) Parking Availability Prediction with Long Short Term Memory Model. In: Li S. (eds) Green, Pervasive, and Cloud Computing. GPC 2018. Lecture Notes in Computer Science, vol 11204. Springer, Cham. 124-137. https://doi.org/10.1007/978-3-030-15093-8_9

[8] Alajali W., Wen S., Zhou W. (2017) On-Street Car Parking Prediction in Smart City: A Multi-source Data Analysis in Sensor-Cloud Environment. In: Wang G., Atiquzzaman M., Yan Z., Choo KK. (eds) Security, Privacy, and Anonymity in Computation, Communication, and Storage. SpaCCS 2017. Lecture Notes in Computer Science, vol 10658. Springer, Cham. 641-652. https://doi.org/10.1007/978-3-319-72395-2_58

[9] Mrazovic, Petar; Eravci, Bahaeddin; Larriba-Pey, Josep Lluis; Ferhatosmanoglu, Hakan; Matskin, Mihhail (2017). [IEEE 2017 18th IEEE International Conference on Mobile Data Management (MDM) - Daejeon, South Korea (2017.5.29-2017.6.1)] 2017 18th IEEE

International Conference on Mobile Data Management (MDM) - Understanding and Predicting Trends in Urban Freight Transport. 124–133. doi:10.1109/MDM.2017.26
[10] Stolfi, Daniel H.; Alba, Enrique; Yao, Xin (2019). Can I Park in the City Center? Predicting Car Park Occupancy Rates in Smart Cities. Journal of Urban Technology, 27-41. doi:10.1080/10630732.2019.1586223
[11] Y. Nie, W. Yang, Z. Chen, N. Lu, L. Huang and H. Huang, (2021), Public Curb Parking Demand Estimation With POI Distribution, IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2020.3046841.
[12] Errousso, Hanae; Malhene, Nicolas; Benhadou, Siham; Medromi, Hicham (2020). Predicting car park availability for a better delivery bay management. Procedia Computer Science, Elsevier, 170, 203–210. doi:10.1016/j.procs.2020.03.026
[13] Brožová, H., & Růžička, M. (2020). The prediction of parking space availability. Transport, 35(5), 462-473. https://doi.org/10.3846/transport.2020.14016
[14] Jomaa, Hadi Samer; Grabocka, Josif; Schmidt-Thieme, Lars; Borek, Alexander (2019). [IEEE 2019 International Joint Conference on Neural Networks (IJCNN) - Budapest, Hungary (2019.7.14-2019.7.19)] 2019 International Joint Conference on Neural Networks (IJCNN) - A Hybrid Convolutional Approach for Parking Availability Prediction. 1–8. doi:10.1109/ijcnn.2019.8852400
[15] Joshi A.V. (2020) Amazon's Machine Learning Toolkit: Sagemaker. In: Machine Learning and Artificial Intelligence. Springer, Cham. 233-243. https://doi.org/10.1007/978-3-030-26622-6_24
[16] Karthi, M.; Harris, Preethi (2016). [IEEE 2016 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM) - Bangalore, India (2016.10.19-2016.10.21)] 2016 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM) - Smart Parking with Reservation in Cloud Based Environment. , (), 164–167. doi:10.1109/CCEM.2016.038
[17] Etaati L. (2019) Azure Machine Learning Studio. In: Machine Learning with Microsoft Technologies. Apress, Berkeley, CA. Springer, Cham. 201-223 https://doi.org/10.1007/978-1-4842-3658-1_12
[18] Cecil, R. R., & Soares, J. (2019). IBM Watson studio: a platform to transform data to intelligence. In Pharmaceutical Supply Chains-Medicines Shortages (pp. 183-192). Springer, Cham.
[19] Bisong, E. (2019). Google AutoML: cloud vision. In Building Machine Learning and Deep Learning Models on Google Cloud Platform (pp. 581-598). Springer, Cham.
[20] Yakovlev, A., Moghadam, H. F., Moharrer, A., Cai, J., Chavoshi, N., Varadarajan, V., ... & Agarwal, N. (2020). Oracle automl: a fast and predictive automl pipeline. Proceedings of the VLDB Endowment, 13(12), 3166-3180.
[21] Dataiku. (2021). https://www.dataiku.com/.
[22] Data Robot. [n. d.]. Data Robot: Automated Machine Learning for Predictive Modeling. Retrieved 06-Oct-2019 from https://datarobot.com
[23] Song, H., Liang, H., Li, H. et al. Vision-based vehicle detection and counting system using deep learning in highway scenes. Eur. Transp. Res. Rev. 11, 51 (2019). https://doi.org/10.1186/s12544-019-0390-4
[24] Zhou, S., Ng, S. T., Yang, Y., & Xu, J. F. (2021). Integrating computer vision and traffic modeling for near-real-time signal timing optimization of multiple intersections. *Sustainable Cities and Society*, Elsevier, *68*, 102775.
[25] H. Tahir, M. Shahbaz Khan and M. Owais Tariq, "Performance Analysis and Comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the Detection and Counting of Vehicles," 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2021, pp. 587-594, doi: 10.1109/ICCCIS51004.2021.9397079.
[26] Yang, T., Liang, R. & Huang, L. Vehicle counting method based on attention mechanism SSD and state detection. Vis Comput (2021). https://doi.org/10.1007/s00371-021-02161-y
[27] Xu, H., Zhou, W., Zhu, J., Huang, X., Wang, W.: Vehicle counting based on double virtual lines. Signal Image Video Process. 11, 905–912 (2017). https://doi.org/10.1007/s11760-016-1038-7
[28] Deng, C., Jin, Y., Zhang, M., Liu, X. and Yu, Z. (2018). Emission Characteristics of VOCs from On-Road Vehicles in an Urban Tunnel in Eastern China and Predictions for 2017–2026. Aerosol Air Qual. Res. 18: 3025-3034. https://doi.org/10.4209/aaqr.2018.07.0248.