2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE) | 978-1-6654-6280-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/LARS/SBR/WRE56824.2022.9995889

Navigation Aids Based on Optical Flow and Convolutional Neural Network

1st Leonardo Silveira
Computer Science Division
Aeronautics Institute of Technology - ITA
São José dos Campos-SP, Brazil
leonardo.silveira@ga.ita.br

2st Mateus Rodrigues
Computer Science Division
Aeronautics Institute of Technology - ITA
São José dos Campos-SP, Brazil
mateusrodrigues4991@gmail.com

3st Bruno S. Faiçal
Computer Science Division
Aeronautics Institute of Technology - ITA
São José dos Campos-SP, Brazil
bruno.faical@ga.ita.br

4st Alexandre S. Quirino da Silva
Embraer
São José dos Campos-SP, Brazil
asqsilva@embraer.com.br

5st Cesar Marcondes
Computer Science Division
Aeronautics Institute of Technology - ITA
São José dos Campos-SP, Brazil
cmarcondes@ita.br

6st Marcos R. O. A. Maximo
Autonomous Computational
Systems Lab - LAB-SCA
Computer Science Division
Aeronautics Institute of Technology - ITA
São José dos Campos-SP, Brazil
mmaximo@ita.br

7st Filipe A. N. Verri
Computer Science Division
Aeronautics Institute of Technology - ITA
São José dos Campos-SP, Brazil
verri@ita.br

***Abstract*—Unmanned Aerial Vehicles (UAV) have been employed in various activities, such as search and rescue missions. Global Navigation Satellite System (GNSS) is the main tool used by UAVs to identify their global location and to be able to complete flights safely. However, UAVs can suffer attacks that invalidate the global positioning information or simply loose for a period the GNSS signal. The lack of the aircraft's global positioning information can result in incomplete missions and accidents. In this paper we propose an Image-Based Localization System (IBLS), which allows the global position of the aircraft to be inferred based on images captured by a camera pointed at the ground. Our proposal is based on the concepts of optical flow to infer the displacement between two images captured sequentially with a CNN. Based on the displacement, we use the haversine formula to estimate the new global position (latitude and longitude) of the UAV. IBLS uses a calibration step learned during flight with available GNSS signal, which allows to reduce the error in inferring the new position. The results achieved on simulated and real data sets demonstrate that our proposal is able to infer the position of the UAV with lower error than in GNSS. The main contribution of this work is the investigation of a system based on the concept of Optical Flow and Convolutional Neural Network to estimate the geographical coordinate during UAV flight.**

***Index Terms*—Computer vision, Unmanned aerial vehicle, Autonomous vehicles, Convolutional Neural Network**

This work was supported by Embraer and FAPESP Grant 2021/06872−6.

978-1-6654-6280-8/22/$31.00©2022 IEEE

I. INTRODUCTION

Unmanned Aerial Vehicles (UAV) use the Global Navigation Satellite System (GNSS) signals as their primary location tool. Knowing the global position (latitude and longitude coordinates) allows the flight control system to be able to perform missions in outdoor environments. However, this system is susceptible to various types of attacks and also interruptions in signal reception [1], [2].

During the aircraft's flights, especially when the UAV is performing some search and rescue (SAR) missions, it is important to estimate the position of the aircraft even if the GNSS information is not available [3]. Several approaches have been investigated, such as sensor fusion, Inertial Measurement Units (IMUs) and image-based inference. Image-based navigation use several strategies to estimate location: learning landmarks, odometry, among other [4], [5], [6], [7] Usually, theses methods have high computational cost due to the use of image processing pipeline required.

We find in the literature works that propose image-based methods to estimate the displacement from a reference point (e.g., initial position) and avoid collision [8], [9], [10]. However, we did not find related work on using CNN and dataflow to estimate the global position of UAVs. The use of lightweight Artificial Neural Networks (ANN) models in image-based strategies can allow the decrease of the computational cost in the inference step. An approach that can be used in conjunction with ANNs is to consider that the GNSS signal is available

Authorized licensed use limited to: ITA - Instituto Tecnológico de Aeronáutica. Downloaded on March 30,2023 at 16:04:26 UTC from IEEE Xplore. Restrictions apply.

for part of the flight, and use this information to calibrate the inferred result of the model. The use of a calibration procedure in addition to the ANN model can enable the use of smaller prediction models that can be employed in real time.

Additionally, Convolutional Neural Networks (CNN) have achieved great success in computer vision pipelines, becoming the standard building block for image processing using machine learning techniques [11], [12]. In comparison with traditional fully connected ANN layers, CNNs make better use of local spatial patterns observed in images, and require a much smaller number of parameters than the former [13].

In this regard, we present an Image-Based Localization System (IBLS) which draws from the concepts of image data flow: we propose a lightweight CNN to infer the meters traveled between two sequential images. The inferred value is used in the haversine formula to calculate the new geographic coordinate (latitude and longitude) of the vehicle. In this first step, we assume that the direction of flight and the height of the vehicle are obtained from sensors (e.g. compass and radio altimeter). Importantly, IBLS uses a calibration mechanism in the inference performed by the CNN to reduce the impact of the accumulated error in long flights.

The remaining of this paper is organized as follows. Section II, we describe the works found in the scientific literature close to the proposal. Section III details the proposed system. Section V details the datasets used, the evaluation methodology and the results. Finally, Section VI we present our conclusions and future work.

II. Scientific Literature

Numerous works in the literature highlight the importance of investigating methods and approaches for different vehicles (ground or air) to have location information without the dependency on external sources [8], [9], [10]. In general, these works investigate approaches that allow vehicles to move safely and reach their goal.

Visual approaches, such as visual odometry, have attracted attention for several reasons, among them price, accessibility, accuracy, and the independence on external signals, as in the case of GNSS-based methods [14], [15]. Visual odometry is the process of estimating the motion of an agent (e.g., vehicle, human, and robot) using only the input from a single or multiple cameras [16].

There are works describing good results on merging visual odometry information with other information to achieve better accuracy in location inference [17], [18]. The results show that merging information from visual odometry with other sensors can increase the accuracy of the positioning and movement information. Visual odometry commonly uses an important concept called optical flow.

Optical flow has been used to detect the motion of objects and scenery to help to autonomously drive vehicles and avoid collisions [19]. An example of this scenario can be seen in [20], where the proposed method (named LiteFlowNet2) is evaluated on datasets from different contexts. The MPI Sintel dataset is a dataset derived from the open source 3D animation short film, Sintel. In this setting the method receives a pair of sequential images and its output is a segmented image of the regions occupied by the characters' movements in the time interval between the images received as input. Another dataset used is KITTI [19], [21], which is a set of images captured by a car on urban routes. In this dataset, the method is evaluated for the goal of detecting the surrounding scenery in motion.

One can observe in the literature works that use the concepts above to estimate the movement of unmanned aerial vehicles [22], [23]. The proposed methods can be used in outdoor and indoor environments.

However, it is common for navigation systems to use geographic coordinate information to manage UAV flight. Considering this context, we were not able to find works with the objective of inferring the geographic coordinate of UAVs during flight.

III. Image-Based Localization System

In this section, we describe the proposed ANN model and its training procedure, as well as the calibration process used during inference.

A. Network Architecture

The proposed neural network model needs to be able to receive two images as its input, and compute the displacement between these two frames. Therefore, this can be defined as a regression problem, and we design a Convolutional Neural Network (CNN) model to solve for it.

Our network can be decomposed in two parts:

1) The first part is a Siamese network, where the model receives a pair of images taken at consecutive time steps, and pass these images thorough identical neural network layers and weights. We chose the design of a Siamese network with the aim that the images can be processed independently, and the neural network can find useful coarse patterns in them, like edges, before joining both images.
2) The output of the Siamese network is concatenated, effectively superposing both image maps. The concatenated image maps then goes through a regular CNN pipeline, ending in a MLP prediction head containing three fully connected layers, of which the last one has only a single output followed by a linear activation function.

During the search of the best hyperparameters for the neural network, we find that larger filter sizes work best for our problem, and we ultimately employ 7x7 filters throughout the network. Apart from the filter size, our design choices took inspiration from the VGGNet architecture [24]: at every new set of convolutional layers we double the number of filters, as well as reducing by half the spatial dimensions of the image maps by means of applying MaxPooling layers with kernel size of 2x2. In summary, our model is composed by 6 convolutional layers and 3 fully connected layers. In total, the neural network is 9 layers deep. The detailed architecture can be seen at Fig. 1.

Authorized licensed use limited to: ITA - Instituto Tecnológico de Aeronáutica. Downloaded on March 30,2023 at 16:04:26 UTC from IEEE Xplore. Restrictions apply.

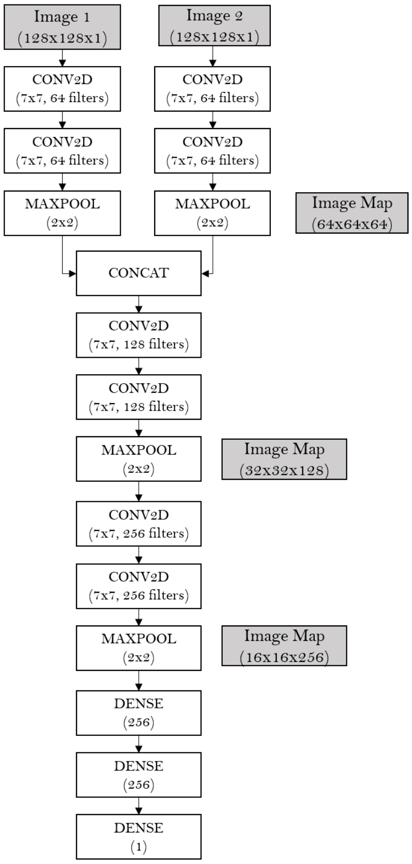

Fig. 1. Architecture of the proposed neural network model, and image map sizes at different locations. The model counts with 23,473,601 trainable parameters.

Once the model is to be employed in a small unmanned aircraft, computational power and storage are important restrictions. With this in mind, we design our neural network to receive gray-scale images of dimensions 128x128. The use of gray-scale images should suffice for the task of displacement prediction, once the color information transmitted by the red, green and blue channels should not give any insights about the movement of the vehicle.

B. Training

For training our model, we employed ReLU activation functions throughout the network, except in the last layer, where we used a linear activation function. Additionally, we used batch-normalization layers after every weight layer, and dropout layers with probability 0.3 in part 1 of the network and probability of 0.2 in part 2, with the exception of the last fully connected layer.

We trained the network on Tensorflow framework, and applied Adam optimizer with a learning rate of 0.001. For our loss function, we used Mean Squared Error, and we trained the network for 30 epochs.

Additionally, during training the image pairs were shuffled before been presented to the neural network: This assures that we do not feed correlated data to the model, once the image pairs seen in a single batch will not be all from the same flight or will not have been taken in a sequence by the camera mounted on the drone.

C. Calibration

We assume that the proposed model for displacement prediction may be employed alongside a GNSS receiver, as a redundancy, thus being activated in case the GNSS presents malfunctioning behavior or becomes unavailable. This allows us to use the signal from the GNSS to calibrate our model in the early stages of flight.

In this setting, while the GNSS is available, we use its signal to correct the prediction of our model by fitting a linear regression model, where the input of the model is our CNN model prediction, and the output is the *ground-truth vehicle displacement* from the GNSS. This calibration procedure continues as long as the GNSS system is working.

When the GNSS becomes unavailable, we feed the output of our CNN model in our linear regression model, and use its calibrated output as our prediction.

IV. Generating the Datasets

To evaluate the proposed model we use two datasets generated by us and publicly available at zenodo. These are: (i) dataset with real images [25] and (ii) simulated images dataset [26], [27], [28]. The diversification in the datasets used contributes to a more comprehensive evaluation of the proposed method. Fig. 2 exemplifies the existing images in each dataset. The columns refer to the real and simulated datasets, while the rows are the order in which the images were obtained.

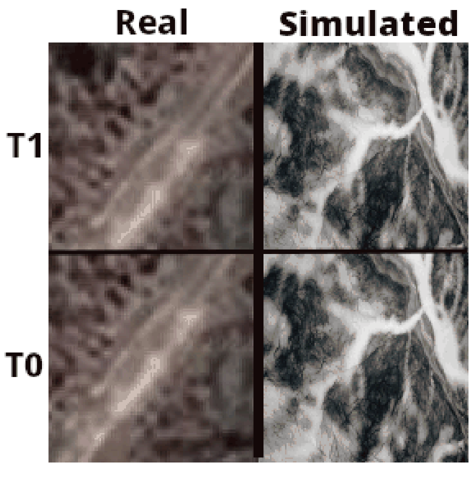

Fig. 2. Example of the images contained in the datasets.

A. Dataset with real images

We implemented an algorithm capable of generating datasets with real images considering two flight routes with the aircraft camera directed to the ground. The routes can

https://zenodo.org/

Authorized licensed use limited to: ITA - Instituto Tecnológico de Aeronáutica. Downloaded on March 30,2023 at 16:04:26 UTC from IEEE Xplore. Restrictions apply.

be performed in two patterns, such as straight line and area coverage (see Fig. 3). In both routes the images have variable displacement between images, assuming that acceleration and deceleration actions or weather influence can change the distances between pairs of images.

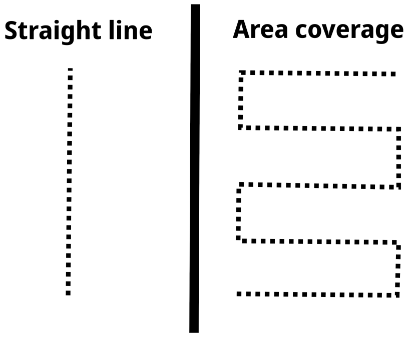

Fig. 3. Routing patterns supported by the algorithm.

Parameters such as distance, variation between images, height, dimensions of the images, initial position (latitude and longitude), direction of the final position, and path distance are configurable in our algorithm. The real images are obtained through the Maps Static API , which requires registration to receive an access key.

Given the characteristic of the image having the top always facing north, we use a heuristic of requesting images from the API with dimension 640×640 and crop the image according to the defined size and direction of flight. In this way, the top of the image is always the direction where the front of the UAV is pointing. In initial experiments to set up the algorithm, we were able to obtain rotated images with dimension up to 440×440.

B. Simulated images dataset

In order to develop a large-scale simulated dataset to the task of UAV displacement estimation, we leverage the capabilities of AirSim [29], which is an open-source simulator for autonomous vehicles, including self-driving cars and drones.

We simulated drone flying in three different scenarios: a mountainous arctic region, a tropical forest and a small city with green area. These scenarios include artifacts like lakes, rivers, different sizes of streets, buildings and vegetation. Besides the inherent heterogeneity of these choices of maps, we also varied the weather conditions, adding dust, mist and rain, creating a diversified range of settings. This variability in the scenarios are important if we want the machine learning model trained on this data to be able to generalize well to scenes never seen before, which is paramount once we employ these models in real-world applications.

The images taken from the drone have resolution of 720x480 pixels. Even though most applications may need smaller resolution, the operation of resizing in the data pipeline is very cheap, and on the plus side, the choice of a large resolution gives the user a choice to apply operations of data-augmentation, like random-cropping, that need an image larger than the input of the network. The use of data-augmentation operations in the preprocessing step can greatly increase the generalization ability of the models trained, as well as virtually increasing the dataset size.

The images are taken from a camera mounted under the vehicle, pointing vertically to the ground. In this configuration the captured images should obtain the maximum information about the drone movement in the horizontal plane. Additionally, the images are taken with approximately 3 frames per second, a slow rate but which is sufficient to obtain enough superposition between each pair of images, and which is easy to be replicated.

From this, it can be seen that the horizontal position of the vehicle will vary at each time-step, while the height of the flight will remain constant. Additionally, the velocity sampled for each segment of the flight comes from a uniform distribution, varying from 2 m/s to 12 m/s.

Also, each image has an associated file with the ground truth information of the complete status of the drone in the instant the picture was taken, which includes linear and angular speeds, linear and angular accelerations, position, latitude and longitude, and attitude of the vehicle. We explicitly annotated the images with this rich set of information to make it possible for researchers to use this dataset for other ends, besides the prediction of the vehicle displacement.

V. Experiments and Results

We emphasize that we performed the experiments with two objectives, (i) to evaluate the capacity of the proposed method to map pairs of images (input) into meters of displacement (output) between the images; and (ii) to evaluate the behavior of the model in a test case, in which we simulate a UAV flight where the GNSS signal becomes unavailable. Thus, the method is used to infer the displacement of the aircraft and calculate its new geographical coordinate (lat and long). It is important to mention that in this case study we consider that the heading and height are obtained by other sensors, such as compass and radio altimeter. The protocol for the training process is the same for all datasets used.

A. Learning evaluation

The images in each dataset were organized into sequential image pairs and the label is the displacement between the images, i.e. the difference in meters from the center of one image to the center of the other image. Thus, all flights existing within the same dataset are organized into pairs of sequential images. The only division that exists is between the datasets, which are not mixed.

The simulated dataset has a total of $163{,}358$ images, resulting in $163{,}357$ image pairs. The training set consists of $143{,}357$ images, while the validation and test sets consist of $10{,}000$ images. On the other hand, the real images dataset consists of two routes, of which only one (the larger route) was used for training ($35{,}792$ images, resulting in $35{,}791$ image

https://developers.google.com/maps/documentation/maps-static

Authorized licensed use limited to: ITA - Instituto Tecnológico de Aeronáutica. Downloaded on March 30,2023 at 16:04:26 UTC from IEEE Xplore. Restrictions apply.

pairs). The training set for this route consists of 28,633 image pairs, while the each test and validation sets have 3,579 image pairs.

Table I shows the performance achieved by the method on the simulated and real images dataset. We observed that the generated model had a stable performance during the validation and testing phases, obtaining similar MSE and RMSE values.

TABLE I
PERFORMANCE OF THE MODELS FOUND BY THE PROPOSED METHOD.

Dataset	Phase	MSE	RMSE
Simulated	Validation	0.0587	0.2423
	Test	0.0578	0.2404
Real	Validation	0.0139	0.1178
	Test	0.0185	0.1360

We attribute the difference between the MSE and RMSE values of the simulated and the real dataset to the characteristics of the images in each dataset. However, we emphasize that the proximity of the errors in the validation and testing phases in each dataset evidence the stability of the model.

Given the results presented, we consider that the proposed model is able to map the input images to the meters displaced by the aircraft.

B. Test case

Considering the results obtained in the previous experiments, we evaluate our model in a two test cases that simulate flights without GNSS or with loss of GNSS signals in part of the flight. In the second case, we performed a calibration based on linear regression of the prediction with the ground-truth available from the GNSS in the part of the flight performed where it is available.

For these test cases we use the smaller route from the real images dataset, which was not used during training. The route is 9.996 km long in a straight line with the displacement varying between images.

Flying without GNSS signal

In this scenario the UAV takes off vertically and we assume that it does not change its position during takeoff. Therefore its initial position is known and an image is recorded at that moment. The flight is initiated and we assume that the flight direction is unique, being a straight line flight. The images are taken at constant frame intervals, but the displacement is variable because weather conditions (such as wind) can influence the displacement of the aircraft.

From the beginning of the UAV's displacement, new images are captured and the model infers the displacement based on the pair of images (previous and current). With the displacement information in meters, the flight system uses the haversine function [30] to calculate the new coordinate. This function receives as input the previous coordinate, displaced meters, direction of displacement and returns as output the new coordinate pair.

Table II shows the performance achieved in this scenario. Although we consider the average error to be low, the recurrence of the error interferes with the following predictions and results in a high cumulative error.

TABLE II
MODEL PERFORMANCE ON A FLIGHT WITHOUT GNSS SIGNAL.

MSE	RMSE	Average error (m)	Accumulated error (m)
0.0185	0.1359	0.0051	50.7130

Flying with loss of GNSS signals

Similar to the previous case, this flight starts with the same premises, except that the GNSS signals is present for part of the flight. The GNSS signals is present at the beginning of the flight and remains for varying lengths of time (percentages of the flight). Thus, we evaluate the performance in inferring the new UAV coordinate in scenarios varying with GNSS signals availability from 10% to 90% of the flight.

We emphasize that the way to calculate the new coordinate in the UAV is the same as the one used in the previous scenario.

The table III shows the performances achieved by the model using the prediction calibration in a flight with a trajectory of 9.996 km. We highlight line 4, where the initial 40% of the route with GNSS signals is used to calibrate the model prediction, resulting in a significant improvement in accuracy.

TABLE III
MODEL PERFORMANCE ON A FLIGHT WITH PARTIAL GNSS SIGNALS.

RMSE	Average error (m)	Accumulated error (m)	Accumulated error without calib. (m)	Flying with GNSS signals (%)
0.0841	0.0016	14.54	47.53	10
0.0836	0.0011	8.45	42.26	20
0.0837	0.0009	6.55	37.75	30
0.0835	**0.0005**	**3.24**	**32.09**	**40**
0.0837	0.0008	4.17	23.53	50
0.0826	0.0005	1.96	19.26	60
0.0829	0.0008	2.53	17.50	70
0.0819	0.0019	3.75	13.32	80
0.0843	0.0030	2.95	8.33	90

It is important to note that the prediction of the new coordinate improves over the flight without GNSS signals from the first calibration level: using 10% of the flight length for calibration, the error already reduces to 28.59% of the error without calibration.

The use of calibration prominently improves the prediction of the UAV coordinate up to level 4, where it uses 40% of the route's length with GNSS signals. At this level it reduces the error to 6.05% of the error when calibration is not used, having a cumulative error of 3.24 meters.

The smallest error obtained is 1.96, which occurs when the flight has a GNSS signals for 60% of the flight. We note that from level 4 the error does not maintain a constant reduction, indicating that it is not possible to guarantee that the smallest error will be reached with more time of available GNSS signals.

Authorized licensed use limited to: ITA - Instituto Tecnológico de Aeronáutica. Downloaded on March 30,2023 at 16:04:26 UTC from IEEE Xplore. Restrictions apply.

VI. CONCLUSIONS

Inferring the position of a UAV with high accuracy without the use of GNSS is an obstacle with several studies in the scientific literature. The increasing evolution of UAVs and the high range of possible contexts in which they can be applied further highlights the need for independence from the GNSS signals for safe navigation.

Our proposed method demonstrates that it is possible to infer the geographic position of a UAV in flight from images collected with a camera attached to the vehicle. We consider the results obtained with the calibrated model satisfactory: they present a lower error than the GNSS error, considering absolute position information. One important feature observed so far is that the image pairs need to overlap so that the displacement vector can be estimated.

Lots of things must be done in the future:

- Test the architecture in data from different altitudes;
- Generalize our calibration method to tackle varying-altitude flights;
- Propose improvements in our CNN architecture to reduce even further the number of parameters while maintaining satisfactory prediction results;
- Use simulated data to pre-train the network;
- Propose a more robust calibration method that automatically uses only relevant data (too many data from the past might hinder performance);
- Test our proposed model during a UAV flight.

ACKNOWLEDGMENT

Hidden for review.

REFERENCES

[1] A. Oruc, "Potential cyber threats, vulnerabilities, and protections of unmanned vehicles," *Drone Systems and Applications*, vol. 10, no. 1, pp. 51–58, 2022.

[2] A. B. Moore and M. Johnson, "Geospatial support for agroecological transition through geodesign," in *Drones and Geographical Information Technologies in Agroecology and Organic Farming Contributions to Technological Sovereignty*. CRC Press, 2022, pp. 174–203.

[3] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif, "Towards the unmanned aerial vehicles (UAVs): A comprehensive review," *Drones*, vol. 6, no. 6, 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/6/147

[4] J. R. G. Braga, H. F. C. Velho, G. Conte, P. Doherty, and É. H. Shiguemori, "An image matching system for autonomous UAV navigation based on neural network," in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016, pp. 1–6.

[5] G. da Penha Neto, H. F. de Campos Velho, and E. H. Shiguemori, "UAV autonomous navigation by image processing with uncertainty trajectory estimation," in *Proceedings of the 5th International Symposium on Uncertainty Quantification and Stochastic Modelling*, J. E. S. De Cursi, Ed. Cham: Springer International Publishing, 2021, pp. 211–221.

[6] A. Gupta and X. Fernando, "Simultaneous localization and mapping (slam) and data fusion in unmanned aerial vehicles: Recent advances and challenges," *Drones*, vol. 6, no. 4, p. 85, 2022.

[7] C. Noviello, G. Gennarelli, G. Esposito, G. Ludeno, G. Fasano, L. Capozzoli, F. Soldovieri, and I. Catapano, "An overview on down-looking UAV-based GPR systems," *Remote Sensing*, vol. 14, no. 14, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/14/3245

[8] B. S. S. Rathnayake and L. Ranathunga, "Lane detection and prediction under hazy situations for autonomous vehicle navigation," in *2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer)*, Sep. 2018, pp. 99–106.

[9] N. Bergman, L. Ljung, and F. Gustafsson, "Terrain navigation using Bayesian statistics," *IEEE Control Systems*, vol. 19, pp. 33–40, 1999.

[10] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology, Second Edition*, 2004.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[14] M. O. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, "Review of visual odometry: types, approaches, challenges, and applications," *SpringerPlus*, vol. 5, no. 1, pp. 1–26, 2016.

[15] Y. Yu, C. Hua, R. Li, and H. Li, "Pose estimation method based on bidirectional recurrent neural network in visual odometry," *Available at SSRN 4155315*.

[16] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[17] J. Rehder, K. Gupta, S. Nuske, and S. Singh, "Global pose estimation with limited gps and long range visual odometry," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 627–633.

[18] G.-S. Cai, H.-Y. Lin, and S.-F. Kao, "Mobile robot localization using gps, imu and visual odometry," in *2019 International Automatic Control Conference (CACS)*. IEEE, 2019, pp. 1–6.

[19] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[20] H. W., T. X., and C. C. Loy, "A lightweight optical flow cnn—revisiting data fidelity and regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9018073/

[21] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3061–3070.

[22] J. Goppert, S. Yantek, and I. Hwang, "Invariant kalman filter application to optical flow based visual odometry for UAVs," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2017, pp. 99–104.

[23] H. Romero, S. Salazar, O. Santos, and R. Lozano, "Visual odometry for autonomous outdoor flight of a quadrotor UAV," in *2013 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2013, pp. 678–684.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[25] B. S. Faiçal, L. Silveira, A. S. Q. da Silva, M. Rodrigues, J. D. D. an Cesar Marcondes, and F. A. N. Verri, "Global Positioning System Based on Optical Flow and Convolutional Neural Network," Aug. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6954593

[26] L. Silveira and M. Rodrigues, "Dataset for UAV velocity prediction_pt1," June 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6670189

[27] ——, "Dataset for UAV velocity prediction_pt2," June 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6671685

[28] ——, "Dataset for UAV velocity prediction_pt3," June 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6672408

[29] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[30] C. C. Robusto, "The cosine-haversine formula," *The American Mathematical Monthly*, vol. 64, no. 1, pp. 38–40, 1957.

Authorized licensed use limited to: ITA - Instituto Tecnológico de Aeronáutica. Downloaded on March 30,2023 at 16:04:26 UTC from IEEE Xplore. Restrictions apply.