Convolutional Neural Networks for Autonomous UAV Navigation in GPS-denied Environments

Ricardo Serras Santos[1], João P. Matos-Carvalho[1,2], Slavisa Tomic[1,2], Marko Beko[1,3], Carlos T. Calafate[4]

[1] COPELABS, Universidade Lusófona, Campo Grande 376, 1749-024 Lisbon, Portugal
[2] Center of Technology and Systems (UNINOVA-CTS) and Associated Lab of Intelligent Systems (LASI), 2829-516 Caparica, Portugal
[3] Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisbon, Portugal
[4] Department of Computer Engineering (DISCA) Universitat Politècnica de València, Camino de Vera, S/N, Valencia, 46022, Spain

{ricardo.santos, joao.matos.carvalho, slavisa.tomic, beko.marko}@ulusofona.pt
calafate@disca.upv.es

Abstract. This work addresses the challenge of autonomous Unmanned Aerial Vehicle (UAV) navigation in Global Positioning System (GPS)-denied environments by proposing a new approach that is an amalgamation of data-driven and model-based philosophies. The proposed method exploits datasets acquired from existing frameworks like the Generalized Trust Region Sub-problem (GTRS) and the Weighted Least Squares (WLS). These datasets are then used to feed the proposed Convolutional Neural Network (CNN) specially tailored to create models for UAV navigation. Afterwards, these models are used to make predictions of an optimal trajectory. The obtained numerical results reveal that the proposed CNN reveals improvements in accuracy and robustness to noise when compared to other Machine Learning approaches, while reducing the required training time.

Keywords: Navigation, Convolutional Neural Network (CNN), Weighted Least Squares (WLS), Unmanned Aerial Vehicle (UAV), Generalized Trust Region Sub-Problem (GTRS).

1 Introduction

The usage of Unmanned Aerial Vehicles (UAVs) is predicted to be indispensable in future Smart Cities to provide services such as aerial delivery services (for instance medications or organs for transplantation) [1], infrastructure inspection (allowing to inspect places where a human cannot see, like damage monitoring on bridges, buildings, *etc.*) [2] and first responder services (after a natural disaster, for instance) [3]. Additionally, due to their maneuverability and rapid deployment, these types of vehicles are operating in a variety of applications, for instance search and rescue [4], telecommunications [5], wild-fire detection [6] and agriculture [7]. Nonetheless, currently available UAVs encounter technical hurdles, including cybersecurity risks, concerns about privacy, and issues related to public safety. [8]. Additionally, they lack the necessary autonomy to execute complex tasks independently and heavily depend on the Global Positioning System (GPS) for navigation. While GPS offers reliable accuracy in outdoor environments, it becomes ineffective in indoor and densely urban areas due to possible signal obstructions. Consequently, alternative methods must be employed to control UAVs in such environments.

The main motivation for the development of this work is the fact that, in contrast to GPS in outdoor environments, there is no uniquely accepted solution to navigate a UAV in GPS-denied environments. Motivated by this fact, and by the desire of creating / improving UAV navigation algorithms, this work was created. The research question for this work can be formulated as "*How to improve currently existing algorithms with the help of Artificial Intelligence tools*?"

The research question posed above is addressed with the introduction of a custom designed CNN composed of a 1D convolution layer to improve currently available methods, such as the two existing methods in [10] and [11]. This is achieved by training the CNN using one of the algorithms to acquire enough information to perform its own position predictions.

The remainder of this work is organized as follows: in the following section we briefly contextualize our contribution in the scope of Human-Centric Systems addressed in DoCEIS 2024. The related work and formal problem formulation are presented in section 3 and 4, respectively. Then, the proposed CNN method is presented in section 5 and the numerical results in section 6. Finally, the conclusions drawn from this work are outlined in Section 7, along with references to future research.

2 Contribution to *Technological Innovation for Human-Centric Systems*

Human-Centric Systems are planned and developed with the user's needs in mind. They aim to improve or develop human skills in technological environments.

In this work, an improved UAV navigation algorithm is proposed focusing on applications that help humans in a variety of tasks. With the proposed work, the UAV can navigate autonomously, enabling tasks such as pest control within greenhouses, organ (and other medical supplies) transport for urgent transplants, critical infrastructure inspection (e.g. tunnels), warehouse stock monitoring and counting, *etc*. Thus, it represents a good fit to the DoCEIS 2024 theme. More specifically, it falls within the Embedded Artificial intelligence area since the proposed algorithm is fit for embedded implementation in a real UAV.

3 Related Work

One potential approach to address the challenge of UAV navigation in GPS-denied environments involves utilizing Received Signal Strength (RSS) data and refining it with a Kalman Filter (KF). Subsequently, employing a Particle Filter (PF) allows for the estimation of the drone's position [9]. The authors in [10] resort to Bayesian methodology and non-linear optimization to convert the navigation problem into a Generalized Trust Region Sub-problem (GTRS) whose solutions are found using the bisection procedure. To improve localization precision, the authors suggested integrating position estimations with observations from odometry using a Kalman Filter. In [11], the authors estimate the position and navigate the UAV by exploiting radio signals combined with a Weighted Least Squares (WLS) criterion solved in the closed-form. More recently and derived from the new discoveries in artificial intelligence, new algorithms that utilize this method have also been created [12, 13]. While, in [12], the authors proposed employing an LSTM that gathers information from estimates predicted by another (model-based) solution, such as GTRS in [10], to independently predict its own solution thereafter. In [13], the authors introduced a deep neural network comprising a Long Short-Term Memory (LSTM) and a Convolutional Neural Network (CNN). The CNN aims to extract local spatial features, while the LSTM focuses on extracting temporal features from the measurement model.

4 Modelling the Problem

Considering the navigation problem in two (or three) dimensions with N fixed reference points and a drone, where $\boldsymbol{a_i}, \boldsymbol{x}^{(t)} \in R^q$ represent, respectively, their true positions with a time instant being denoted by t. To accurately navigate the drone and resorting to intermediate points located inside the area of interest, the problem is formulated as

$$\boldsymbol{x}^{(t+1)} = S\left[(\boldsymbol{x}^{(t)})^T, (\boldsymbol{v}^{(t)})^T\right]^T, \tag{1}$$

being $\boldsymbol{x}^{(t)}$ the drone's location at instant t and $\boldsymbol{S}$ the matrix that represents state transitions as

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 & 0 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

with the sampling interval between two subsequent time steps represented by Δ, while

$$\boldsymbol{v}^{(t)} = \boldsymbol{v}^{(t)} \begin{bmatrix} \cos(\varphi^{(t)})\sin(\varrho^{(t)}) \\ \sin(\varphi^{(t)})\sin(\varrho^{(t)}) \\ \cos(\varrho^{(t)}) \end{bmatrix} \in \mathbb{R}^3 \quad \textbf{(2)}$$

represents the drone's velocity vector, being the azimuth between the current drone position and the desired destination represented by $\varphi^{(t)}$, and the elevation angle, represented by $\varrho^{(t)}$. Note that, the drone's velocity is dependent on the distance to the intermediate point, represented by τ. Hence, if $\tau > 4$, $\boldsymbol{v}^{(t)}$ is computed by

$$\boldsymbol{v}^{(t)} = \boldsymbol{\omega}^{(t)} \cdot P_{vmax} \quad \textbf{(3)}$$

where P_{vmax} is the maximum allowed speed permitted to the drone. Otherwise, $\boldsymbol{v}^{(t)}$ is computed according to

$$\boldsymbol{v}^{(t)} = \boldsymbol{\eta}^{(t)} \cdot \left(\frac{\lambda^{(t)}}{\tau}\right)^{\gamma} \quad \textbf{(4)}$$

being $\boldsymbol{\eta}^{(t)} = \boldsymbol{x}_{\text{dest}} - \hat{\boldsymbol{x}}^{(t)}$, $\boldsymbol{\lambda}^{(t)} = \|\boldsymbol{\eta}^{(t)}\|$ the distance between the desired destination and the current location estimation of the drone, while $\boldsymbol{\gamma}$ is a smoothing factor.

Before controlling the movement of the drone, it is necessary to estimate its location. This is achieved by exploiting terrestrial radio measurements. Starting with distance (range) observations obtained resorting to the received signal using RSS [14,15], Time-Of-Flight (TOF) [16,17] or some other property. In general, any range measurement k ($1 \leq k \leq K$) among the i-th stationary point and the UAV at time instant t can be modelled as

$$d_{i,k}^{(t)} = \|\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\| + n_{i,k}^{(t)},\ i = 1, \ldots, N,\ k = 1, \ldots, K, \quad \textbf{(5)}$$

with $n_{i,k}^{(t)}$ representing the measurement noise, defined as a Normal random variable with mean zero and variance $\sigma_{i,k}^{(t)}$, *i.e.*, $n_{i,k}^{(t)} \sim \mathcal{N}(0, \sigma_{i,k}^{(t)})$ and the symbol $\|\cdot\|$ denoting the L_2 norm. For simplicity and to introduce greater robustness to outliers, the median of the K distance observations, $d_i^{(t)}$, is employed in the subsequent derivation.

Resorting to **(5)** and according to the maximum likelihood (ML) criteria [18], the drone's position at instant t could be obtained resorting to

$$\hat{\boldsymbol{x}}^{(t)} = \arg\min_{\boldsymbol{x}^{(t)}} \sum_{i=1}^{N} \frac{1}{2(\sigma_i^{(t)})^2} (d_i^{(t)} - \|\boldsymbol{x}^{(t)} - \boldsymbol{a}_i\|)^2 \,. \quad \textbf{(6)}$$

Note that, due to being highly non-convex, **(6)** might possess multiple local optima resulting in the impossibility of tackling the ML estimator directly.

5 Proposed CNN Method

In this section, a detailed description on the proposed CNN is presented and is divided as follows. Firstly, an explanation of the network architecture is provided and secondly, an explanation on how the network was trained is described.

5.1 Network Architecture

To allow the navigation of the UAV, the network architecture was created so that it can extract features from the X- and Y-axes of a particular trajectory. Moreover, the choice of CNNs was made since they can capture the relationship between local features and can provide more detailed information than an LSTM for instance. Thus, the proposed network architecture is visible in Fig. 1 and the CNN architecture parameters are summarized in Table 1.
As shown in Fig.1, first a 1D Convolution layer is applied to extract features from the provided dataset. Then, a MaxPooling is performed to extract the most relevant features in each instant. Afterwards, a Flatten layer is applied so that it is possible to apply the 2 Dense layers (Dense and Output) to finally reach the location prediction from the network.

Table 1. The proposed CNN architecture.

Layer	Output	Hyperparameters
CONV1D	(2, 64)	0
MaxPooling1D	(1, 64)	0
Flatten	64	0
Dense	50	3250
Output	3	153

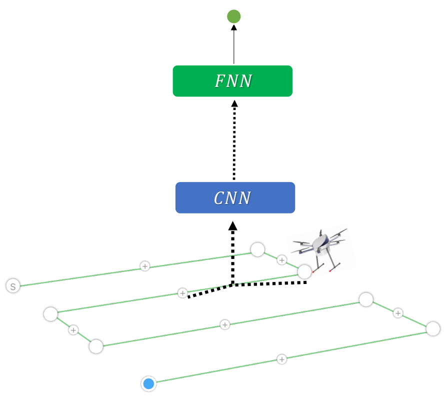

Fig. 1. The proposed CNN.

5.2 Network Training

The CNN proposed in this work can be trained using existing algorithms for UAV positioning, since it only requires in its input the predictions made by one of those algorithms. This training is performed using a dataset containing predictions for the desired trajectory using one of those algorithms. For demonstration purposes, in this work the GTRS in [10], and WLS in [11], were chosen. In the subsequent sub-sections, a brief explanation on both is presented.

GTRS The solution found to overcome the ML problem was to convert it into a GTRS resorting to the use of weights as

$$w_i^{(t)} = \frac{(d_i^{(t)})^{-1}}{\sum_{i=1}^{N} (d_i^{(t)})^{-1}} \tag{7}$$

with the goal of assigning more trust to *closer* links. Next, by resorting to a squared-range methodology and developing the squared-term, the positioning problem can be posed in a vector form as

$$\begin{aligned} &\underset{\boldsymbol{z}^{(t)}}{\text{minimize}} \|\boldsymbol{A}^{(t)}\boldsymbol{z}^{(t)} - \boldsymbol{b}^{(t)}\|^2 \\ &\text{subject to} \\ &(\boldsymbol{z}^{(t)})^T \boldsymbol{D} \boldsymbol{z}^{(t)} + 2\boldsymbol{f}^T \boldsymbol{z}^{(t)} = 0, \end{aligned} \tag{8}$$

with $\boldsymbol{z}^{(t)} = \left[\left[\left(\boldsymbol{x}^{(t)}\right)^T \left(\boldsymbol{v}^{(t)}\right)^T\right]^T, \|\boldsymbol{x}^{(t)}\|^2\right]^T \in R^7$,

$$\boldsymbol{A}^{(t)} = \begin{bmatrix} \left[\sqrt{w_1^{(t)}} 2\boldsymbol{a}_1^T, \boldsymbol{0}_{1\times 2}\right] & -\sqrt{w_1^{(t)}} \\ \vdots & \vdots \\ \left[\sqrt{w_N^{(t)}} 2\boldsymbol{a}_N^T, \boldsymbol{0}_{1\times 2}\right] & -\sqrt{w_N^{(t)}} \\ (\widehat{\boldsymbol{P}}^{(t|t-1)})^{-1/2} & \boldsymbol{0}_{4\times 1} \end{bmatrix} \in \mathbb{R}^{(N+6)\times 7}, \quad \boldsymbol{b}^{(t)} = \begin{bmatrix} \sqrt{w_1^{(t)}}\left(\|\boldsymbol{a}_1\|^2 - \left(d_1^{(t)}\right)^2\right) \\ \vdots \\ \sqrt{w_N^{(t)}}\left(\|\boldsymbol{a}_N\|^2 - \left(d_N^{(t)}\right)^2\right) \\ \left(\widehat{\boldsymbol{P}}^{(t|t-1)}\right)^{-\frac{1}{2}} \widehat{\boldsymbol{\theta}}^{(t|t-1)} \end{bmatrix} \in \mathbb{R}^{N+6},$$

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 4} \\ \boldsymbol{0}_{4\times 3} & \boldsymbol{0}_{4\times 4} \end{bmatrix} \in \mathbb{R}^{7\times 7}, \boldsymbol{f} = \begin{bmatrix} \boldsymbol{0}_{6\times 1} \\ -1/2 \end{bmatrix} \in \mathbb{R}^7,$$

with $\widehat{\boldsymbol{\theta}}^{(t|t-1)}$ and $\widehat{\boldsymbol{P}}^{(t|t-1)}$ denoting the mean and the covariance matrix of the predicted state in a single step (more details can be found in [10]), whereas $\boldsymbol{0}_{\boldsymbol{p}\times\boldsymbol{q}}$ and $\boldsymbol{I}_{\boldsymbol{G}}$ being respectively the zero matrix with p rows and q columns and the identity matrix of order G.

Due to the existence of an interval on which the GTRS is a monotonically decreasing function the solution can be found by applying the bisection procedure [19].

WLS The WLS methodology in [11] is another solution found to overcome difficulties encountered within the ML problem. Similar to the GTRS case, weights are introduced using **(7)**. Then, the WLS estimator [11] is formed as

$$\underset{\boldsymbol{x}^{(t)}}{\text{minimize}} \| \boldsymbol{W}^{(t)} \left(\boldsymbol{H}^{(t)} \boldsymbol{x}^{(t)} - \boldsymbol{h}^{(t)} \right) \|^2, \quad \textbf{(9)}$$

where $\boldsymbol{W}^{(t)} = \text{diag}\left(\boldsymbol{w}^{(t)}\right)$, $\boldsymbol{w}^{(t)} = \left[w_i^{(t)}\right]^T$, $\boldsymbol{H}^{(t)} = [(\boldsymbol{u}_i^{(t)})^T] \in \mathbb{R}^{N\times 3}$, $\boldsymbol{h}^{(t)} = [d_i^{(t)} + (\boldsymbol{u}_i^{(t)})^T \boldsymbol{a}_i] \in \mathbb{R}^{N\times 1}$, $\boldsymbol{u}_i^{(t)} = \left[\cos\left(\hat{\phi}^{(t)}\right) \sin\left(\hat{\alpha}^{(t)}\right),\ \sin\left(\hat{\phi}^{(t)}\right) \sin\left(\hat{\alpha}^{(t)}\right), \cos\left(\hat{\alpha}^{(t)}\right)\right]^T$, and

$$\hat{\phi}_i^{(t)} = \arctan\left(\frac{\hat{x}_{i,2}^{(t)} - a_{i,2}}{\hat{x}_{i,1}^{(t)} - a_{i,1}}\right), \hat{\alpha}_i^{(t)} = \arccos\left(\frac{\hat{x}_{i,3}^{(t)} - a_{i,3}}{\left\|\hat{x}_i^{(t)} - \boldsymbol{a}_i\right\|}\right),$$

where the estimated azimuth from the coarse estimate of the drone's position, $\hat{\boldsymbol{x}}_i^{(t)}$, from each anchor node i is represented by $\hat{\phi}_i^{(t)}$ and the elevation angle is represented by $\hat{\alpha}_i^{(t)}$. The final solution is found in closed form as

$$\hat{x}^{(t)} = ((\boldsymbol{H}^{(t)})^T (\boldsymbol{W}^{(t)})^T \boldsymbol{W}^{(t)} \boldsymbol{H}^{(t)})^{-1} \times ((\boldsymbol{H}^{(t)})^T (\boldsymbol{W}^{(t)})^T \boldsymbol{W}^{(t)} \boldsymbol{h}^{(t)}). \quad \textbf{(10)}$$

UAV Navigation Since one of the goals is also the navigation of the UAV, and having the estimated drone’s position (resorting to existing methodologies such as GTRS), it is possible (resorting to intermediate destination points) to compute the direction the UAV needs to follow by

$$\hat{\varphi}^{(t)} = \arctan\left(\frac{x_{\text{dest},2} - \hat{x}_2^{(t)}}{x_{\text{dest},1} - \hat{x}_1^{(t)}}\right), \quad \textbf{(11a)}$$

$$\hat{\varrho}^{(t)} = \arccos\left(\frac{x_{\text{dest},3} - \hat{x}_3^{(t)}}{\left\|\boldsymbol{x}_{\text{dest}} - \hat{\boldsymbol{x}}^{(t)}\right\|}\right), \quad \textbf{(11b)}$$

where the azimuth between the estimated position, $\hat{\boldsymbol{x}}^{(t)}$, to a destination at t is represented by $\hat{\varphi}^{(t)}$, and the elevation angle is represented by $\hat{\varrho}^{(t)}$.

Thus, the drone navigation can be performed by

$$\hat{\boldsymbol{x}}^{(t+1)} = \boldsymbol{S}[(\hat{\boldsymbol{x}}^{(t)})^T, (\hat{\boldsymbol{v}}^{(t)})^T]^T, \quad \textbf{(12)}$$

being the estimated velocity vector at t defined as

$$\hat{\boldsymbol{v}}^{(t)} = \boldsymbol{v}^{(t)} \begin{bmatrix} \cos(\hat{\varphi}^{(t)}) \sin(\hat{\varrho}^{(t)}) \\ \sin(\hat{\varphi}^{(t)}) \sin(\hat{\varrho}^{(t)}) \\ \cos(\hat{\varrho}^{(t)}) \end{bmatrix} \in \mathbb{R}^3 \quad \textbf{(13)}$$

Note that, unlike in **(1)** where the true position is considered, in **(12)** an estimate of the drone's location, $\hat{\boldsymbol{x}}^{(t)}$ is used.

To summarize the section, a pseudo-code of the proposed CNN algorithm is presented in Algorithm 1.

Algorithm 1 The proposed CNN algorithm

1: **Initialization**: $Ts \leftarrow 3$, $EPS \leftarrow 3000$
2: **Model definition**: *Model*
// Model training
3: *train* ← Read **GTRS** or **WLS** dataset file with $\sigma = 1$
// Format dataset
4: *X, y* = split_sequences(*train*, *Ts*)
5: *Model*.fit(*X, y*)
// Save Model
6: *Model*.save()
// Model Prediction
7: *test* ← Model Predict for $\sigma = 2$ or $\sigma = 5$

6 Numerical Results

In this section the performance of the proposed method is going to be analyzed. The following results were obtained using the Python programming language on a machine containing an INTEL CORE® i9-9880H 2.3 GHZ, 16 GB of RAM, and running MacOS 14.3 and Python 3.9. For enhanced understanding of the subsequent analysis, it's imperative to define two parameters. Starting with the number of *TimeSteps* (*Ts*), that are used in the input of the CNN, followed by the number of *Epochs* (*Eps*), that defines the number of times the learning model is trained using the full dataset. In all simulations outlined in this section, unless explicit said otherwise, the value of $Ts = 3$ and $Eps = 3000$ was considered. Moreover, the datasets used in the training phase were created by taking advantage of the GTRS [10] and WLS [11] methodologies in MatLab® R2023a using $\sigma = 1$.

6.1 Training using the GTRS algorithm

To analyze how the proposed method would predict the trajectory for different σ values, a trained model using fixed $\sigma = 1$ (m) is used to predict the trajectory for $\sigma = 2$ and 5 (m). The obtained results can be seen in Fig. 2.

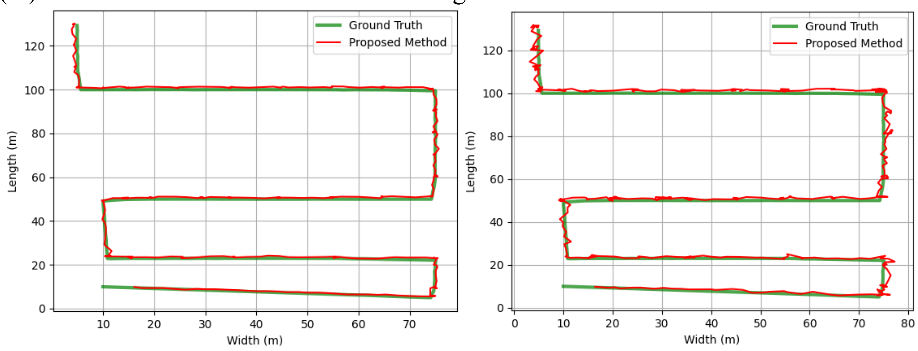

Fig. 2. Comparison between the ground truth trajectory and the proposed CNN trained with GTRS when $\boldsymbol{\sigma} = 2$ (left) and 5 (m) (right).

The results shown in Fig. 2 highlight that, even though it was trained with σ = 1 (m), the proposed method achieves an almost ideal trajectory when σ = 2 (m) and when σ = 5 (m). Also note that, since the model is being trained using the GTRS algorithm, that introduces some error in its predictions, some of the deviations from the trajectory are produced due to this factor. This is most noticeable when σ = 5 (m), as expected. Nonetheless, the proposed method proves to be robust against high noise inferences and matched the already good results of the GTRS algorithm.

6.2 Training using the WLS algorithm

Similarly to the previous subsection, the trained model using σ = 1 (m) was used to predict the trajectory for σ = 2 (m) and 5 (m). Note that this time the training was done using the WLS algorithm. The obtained results can be seen in Fig. 3.

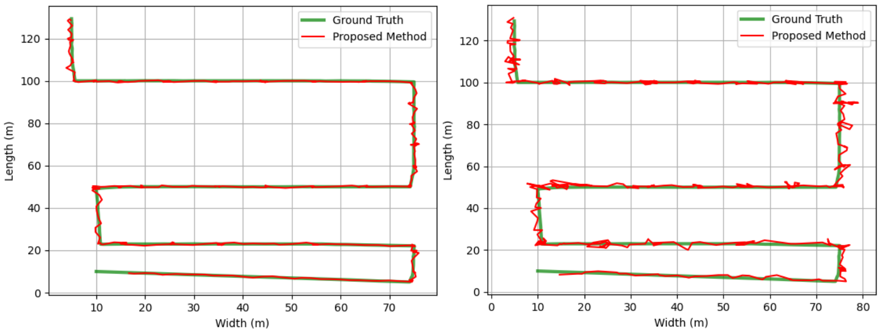

Fig. 3. Comparison between the ground truth trajectory and the proposed CNN trained with WLS when $\boldsymbol{\sigma}$ = 2 (left) and 5 (m) (right).

From Fig. 3 it is possible to see that the proposed method achieves an almost ideal trajectory when the value of σ = 2 (m), but it shows some variations when σ = 5 (m), as anticipated. This can be explained due to the error present in the WLS predictions. Despite this fact, the trajectory is still very similar to the ground truth one. Similarly, with the GTRS algorithm in the previous subsection, the proposed method also matches the WLS algorithm results. To conclude this section, the Root Mean Squared Error (RMSE) for the training with the GTRS and WLS was computed. The results can be found in Table 3.

Table 3. RMSE for the proposed method when trained with the GTRS and WLS.

	RMSE Training GTRS (m)	RMSE Training WLS (m)
σ = 2 (m)	3.82	3.84
σ = 5 (m)	3.85	3.96

6.3 Comparison with other Machine Learning approaches

To further validate the proposed method, a comparison with other machine learning methods is quite relevant. To do so, the proposed method trained with GTRS predictions was tested against the method proposed in [12], where the authors proposed the use of an LSTM to improve the GTRS in [10].

The LSTM was trained using a *LookBack* (*i.e*, the number of previous time stamps being used to compute a prediction for the subsequent time instant) equal to 10 in the following simulations. Both the proposed method and the LSTM in [12] where trained using *Eps* = 100 and $\sigma = 1$ (m). Then, the trained model was used to make a prediction for $\sigma = 5$ (m). The obtained results are shown in Fig. 4.

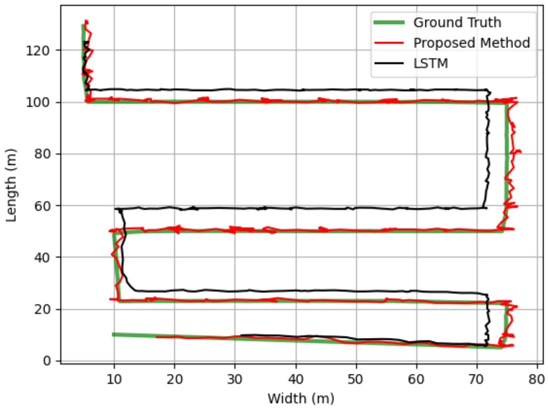

Fig. 4. Comparison between the ground truth trajectory, the proposed method, and the LSTM in [12] with $\boldsymbol{\sigma} = 5$ (m).

Table 4. Training Time for each method.

	Training Time (s)
Proposed Method	18
LSTM	1781

Fig. 4 reveals that the trajectory of the proposed CNN is better in terms of error compared to the LSTM proposed in [12] but notice that in some parts it introduces high trajectory fluctuations, while the LSTM does not. It is important to note that even the slightest deviation from the trajectory could potentially lead to the UAV colliding with an obstacle. The figure also reveals that not only is the proposed CNN more robust to noise, but also that it reveals a lower error when compared to the LSTM.

To finalize this section, a comparison between the training time of the LSTM and the proposed CNN is shown in Table 4.

Table 4 reveals that the proposed CNN is roughly 100× faster in the training stage when compared to the LSTM, since the LSTM is trained in each of the three axes (X, Y and Z) of the Cartesian coordinate system separately, while the CNN trains on the 3 axes at the same time. Note that this drastic reduction in training time is very important in dynamic scenarios because it is necessary to train the network again for a different trajectory. Therefore, the proposed algorithm can adapt to this type of scenarios faster than the LSTM.

7 Conclusions and Future Work

This work proposes a new method that combines Machine Learning with existing solutions to solve the UAV navigation in GPS-denied environments problem. The proposed algorithm can have several applications in the real world, for instance, it can be utilized in warehouse stock monitoring, allowing for real-time stock counting. Finally, it is worth mentioning that testing in real hardware is one of the objectives for the near future. By feeding the proposed CNN network with predictions created by the GTRS or WLS algorithms, a model is created. With the model, a prediction for a known trajectory is made. The obtained results revealed that not only does the proposed method obtain a better robustness to noise and a better overall trajectory estimation when compared to other machine learning approaches, but also reduced the training time by roughly 100×. However, it is important to note that despite having advantages, such as higher robustness to noise comparing, for instance, to the algorithms used in the training phase, it naturally faces some limitations. For instance, when it comes to dynamic scenarios the CNN must be trained every time the scenario suffers (significant) alterations, like changes in environmental conditions or in the trajectory.

Upcoming research will focus on integrating novel machine learning methodologies and incorporating obstacle avoidance strategies to enhance and expand upon the current work.

Acknowledgment

This research was partially funded by Fundação para a Ciência e a Tecnologia under Projects UIDB/04111/2020, 2021.04180.CEECIND, UIDB/50008/2020, CEECINST/00147/2018/CP1498/CT0015 and ROBUST EXPL/EEIEEE/0776/2021 and by the European Union's Horizon Europe Research and Innovation Programme under the Marie Skłodowska-Curie grant agreement No. 101086387.

References

1. Innocenti, E., Agostini, G., Giuliano, R.: UAVs for medicine delivery in a smart city using fiducial markers. Information 13(10) (2022).
2. Máthé, K., Buşoniu, L.: Vision and control for UAVs: A survey of general methods and of inexpensive platforms for infrastructure inspection. Sensors 15(7), 14887–14916 (2015).
3. Matos-Carvalho, J.P., Moutinho, F., Salvado, A.B., Carrasqueira, T., Campos-Rebelo, R., Pedro, D., Campos, L.M., Fonseca, J.M., Mora, A.: Static and dynamic algorithms for terrain classification in UAV aerial imagery. Remote Sensing 11(21), 2501 (2019).
4. Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I.L., Ruess, F., Suppa, M., Burschka, D.: Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. IEEE Robotics & Automation Magazine 19(3), 46–56 (2012).

5. Khan, S.K., Naseem, U., Siraj, H., Razzak, I., Imran, M.: The role of unmanned aerial vehicles and mmWave in 5G: Recent advances and challenges. Transactions on Emerging Telecommunications Technologies 32(7), e4241 (2021).
6. Correia, S.D., Fé, J., Tomic, S., Beko, M.: Drones as sound sensors for energy-based acoustic tracking on wildfire environments. In: Camarinha-Matos, L.M., Heijenk, G., Katkoori, S., Strous, L. (eds.) Internet of Things. Technology and Applications. pp. 109–125. Springer International Publishing (2022).
7. Salvado, A.B., Mendonça, R., Lourenço, A., Marques, F., Matos-Carvalho, J.P., Miguel Campos, L., Barata, J.: Semantic navigation mapping from aerial multispectral imagery. In: 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE). pp. 1192–1197 (2019).
8. Pedro, D., Mora, A., Carvalho, J., Azevedo, F., Fonseca, J.: Colanet: A UAV collision avoidance dataset. In: Camarinha-Matos, L.M., Farhadi, N., Lopes, F., Pereira, H. (eds.) Technological Innovation for Life Improvement. pp. 53–62 (2020)
9. Shen, Y., Hwang, B., Jeong, J.P.: Particle filtering-based indoor positioning system for beacon tag tracking. IEEE Access 8, 226445–226460 (2020).
10. Matos-Carvalho, J.P., Santos, R., Tomic, S., Beko, M.: GTRS-based algorithm for UAV navigation in indoor environments employing range measurements and odometry. IEEE Access 9, 89120–89132 (2021).
11. Santos, R., Matos-Carvalho, J.P., Tomic, S., Beko, M.: Wls algorithm for UAV navigation in satellite-less environments. IET Wireless Sensor Systems 12(3), 93–102 (2022).
12. Santos, R., Matos-Carvalho, J.P., Tomic, S., Beko, M., Correia, S.D.: Applying deep neural networks to improve UAV navigation in satellite-less environments. In: 2022 International Young Engineers Forum (YEF-ECE). pp. 63–68 (2022).
13. Yang, B., Li, J., Shao, Z., Zhang, H.: Robust UWB indoor localization for NLoS scenes via learning spatial-temporal features. IEEE Sensors Journal 22(8), 7990–8000 (2022).
14. Rappaport, T.S.: Wireless communications - principles and practice. Prentice Hall (1996)
15. Tomic, S., Beko, M., Dinis, R.: RSS-based localization in wireless sensor networks using convex relaxation: Noncooperative and cooperative schemes. IEEE Transactions on Vehicular Technology 64(5), 2037–2050 (2015).
16. Comuniello, A., Angelis, A., Moschitta, A., Carbone, P.: Using Bluetooth Low Energy technology to perform ToF-based positioning. Electronics 11, 111 (12 2021).
17. Hashem, O., Harras, K., Youssef, M.: Accurate indoor positioning using IEEE 802.11mc round trip time. Pervasive and Mobile Computing 75, 101416 (05 2021).
18. Kay, S.M.: Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice Hall Upper Saddle River, NJ, USA, 1st ed. edn. (1993)
19. Tomic, S., Beko, M., Dinis, R.: 3-d target localization in wireless sensor networks using RSS and AoA measurements. IEEE Transactions on Vehicular Technology 66(4), 3197–3210 (2017).