# Position Estimation of Unmanned Aerial Vehicles in Contested Environments using Dense Matching Networks

**Bruno Klaus de Aquino Afonso**[1], **Willian Dihanster Gomes de Oliveira** [12]
**Jéssica Domingues Lamosa**[1], **Lilian Berton**[1]

[1]Institute of Science and Technology
Federal University of São Paulo - UNIFESP
São José dos Campos – SP – Brazil

[2]Latam Datalab Serasa Experian
São Paulo – SP – Brazil

{dihanster.willian, jd.lamosa, lberton}@unifesp.br

***Abstract.*** *This paper presents an approach to predicting the trajectory and the position of unmanned aerial vehicles (UAVs) in contested global navigation satellite system (GNSS) environments. Our approach utilizes dense matching networks to analyze and predict patterns in UAV movement and then effectively recovers temporal trajectory information and accurately predicts UAV position changes using images and altitude data. This problem was part of the KDD-BR 2022 Kaggle Competition, where we obtained a Root Mean Squared Error (RMSE) of 0.29603, which made it the runner-up solution. This could have significant implications for the future of UAV navigation, potentially leading to safer and more efficient operations in various applications.*

## 1. Introduction

In recent years, there has been a significant increase in the use of Unmanned Aerial Vehicles (UAVs) for a variety of applications, from environmental monitoring to package delivery [Erdelj et al. 2017, Barmpounakis et al. 2016]. However, a fundamental challenge in operating UAVs is trajectory recovery and position change prediction accuracy. The ability to retrieve past trajectories and accurately predict the UAV's position change is essential to ensure a safe and efficient flight. This is crucial to allow flight controllers and navigation systems to plan ahead and make informed decisions about the UAV's next move. Precise trajectory recovery and position prediction facilitate smoother communication, as operators can better anticipate UAV behavior and respond effectively to changing circumstances.

UAVs are often utilized for tasks where precise navigation and positioning are crucial, such as surveillance, mapping, delivery, and disaster response. Accurate trajectory recovery and position change prediction allow UAVs to optimize their flight paths, reducing travel time and energy consumption. This efficiency translates to quicker task completion and extended operational range. Moreover, maintaining a predictable trajectory and accurately predicting position changes are essential for avoiding collisions, especially in airspace shared with other aircraft or in environments with obstacles. Reliable trajectory recovery helps UAVs to react promptly to unexpected events, preventing accidents and ensuring the safety of both the UAV itself and people on the ground.

UAVs can collect a large amount of data quickly, but analyzing and interpreting this data can be challenging [Rovira-Sugranes et al. 2022]. The effectiveness of UAVs depends on their ability to process, transmit and analyze data in real time, which requires advanced data processing systems and machine learning algorithms (ML). These algorithms can be trained to predict UAV positions based on historical data, sensor measurements, and environmental factors. However, algorithms must deal with noisy and inconsistent data, which requires preprocessing techniques and robust algorithms to ensure accurate predictions.

This paper presents an innovative approach for temporal trajectory recovery and position change prediction of UAVs using dense matching networks [Truong et al. 2021]. It is a deep neural network architecture that excels at learning complex features and relationships. Dense networks are designed with direct connections between all layers, allowing information to flow freely from one layer to another. This feature is particularly relevant for temporal path recovery as it allows for capturing complex dependencies and UAV movement patterns. We apply the Probabilistic Dense Correspondence Network (PDC-Net) proposed by [Truong et al. 2021] pre-trained with the MegaDepth dataset. Taking into account the altitude value and the time delta, we calculate the new altitude of the drone. The proposed approach of temporal trajectory recovery and position change prediction of UAVs was proposed for KDD-BR 2022 Kaggle Competition and led to good results. Our team got second place in the private leaderboard.

The remainder of the paper is organized as follows: Section 2 presents some related work in the area. Section 3 presents the materials and methods, which include the datasets, the algorithm of dense matching networks employed, the extracted features, and the step of training the regression model to predict the UAV's position change. Section 4 presents the results and discussion. Finally, Section 5 presents the conclusion and future work.

## 2. Related Work

Integrating computer vision techniques into artificial intelligence (AI) in Unmanned Aerial Vehicles (UAVs) has been a focal point of recent research. These techniques have addressed various challenges, from routing protocols to position estimation. [Rovira-Sugranes et al. 2022] reviews AI-enabled routing protocols designed for aerial networks. Network nodes use radar-based and visual target tracking to perceive the network topology. Mobility in UAVs can be predicted using different techniques:

- Data-driven: There are data mining and fuzzy methods that analyze large datasets to identify frequent movement patterns. These methods indirectly capture the influence of natural and artificial textures, users' behavioral habits, and spatial and temporal variations in node mobility. In modeling movement patterns, similar methods are proposed to model movement patterns of pedestrians, vehicles, animals, and other mobile users. In addition, traffic distribution trends are also extracted.
- Model-based (based on models): Uses the smoothness of motion paths to predict the future locations of moving objects. Methods include slice segmentation, Hidden Markov Models (HMM), Levy flight process, Bayesian methods, manifold learning, Kalman filtering, fuzzy zone-based methods, and Gaussian mixture models.

Some works have focused on solving the problem using more traditional AI techniques. For example, in [Mostafa et al. 2018], the Performance Visualized Assessment (PVA) model was introduced to evaluate UAV performance in indoor settings. This model combined traditional assessment tools with AI components, notably the Chi-square Inference (CSI) module. However, the model's reliance on specific modules like the CSI might make it less adaptable to environments with varying characteristics. Additionally, the real-world applicability of the model in diverse indoor scenarios needs further validation. In [Anderson et al. 2019], a visual odometry algorithm was developed for UAVs to navigate indoors without GPS. This algorithm harnessed sequential pairs of RGBD camera images, complemented by AI techniques, for real-time dynamic uncertainty estimation. The algorithm's efficacy might be jeopardized in low-lit environments or situations with rapid object movements. Furthermore, its dependence on RGBD imagery introduces potential data processing and storage bottlenecks. [Liu et al. 2015] introduced a novel UAV position and attitude estimation method, leveraging visual sensors combined with georeferenced images. Designed to be robust in noisy environments, this method offers a promising alternative, especially in scenarios where traditional GPS signals are unavailable or unreliable. Preliminary results indicate accurate position estimations, even in sparsely textured regions. However, challenges such as dependency on visual data clarity and the need for a hybrid approach integrating GPS signals underscore areas for future research. This work sets the stage for further exploration into AI-enhanced UAV navigation, aiming to achieve consistent and reliable results across diverse operational conditions.

Different works have centered on using more robust techniques to deal with the problem using deep neural networks. For example, in [Kruber et al. 2020], a deep neural network tailored for object detection was trained on two distinct datasets to estimate vehicle positions. The study underscored the limited availability of datasets in this domain, potentially affecting the model's generalizability. Moreover, the accuracy of position estimation can be influenced by environmental factors and the quality of aerial imagery. In [Dilshad et al. 2023], the "LocateUAV" framework was introduced. It determines the UAV's location using real-time processing of visual sensor data via a streamlined convolutional neural network (CNN). Assuming the UAV operates within an Internet of Things (IoT) setting, the system initially applies object detection techniques to pinpoint the object of interest (OOI), notably signboards. Subsequently, optical character recognition (OCR) is employed to extract relevant contextual information. This data is then communicated to a map application programming interface (API) to locate the UAV accurately.

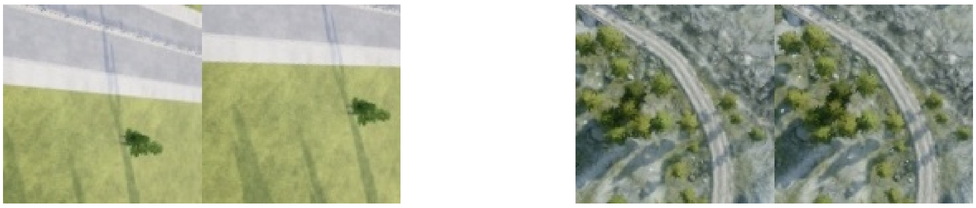
As research advances in Unmanned Aerial Vehicles (UAVs) integrated with artificial intelligence and computer vision techniques, it becomes evident that, despite significant progress, substantial challenges remain to be overcome. Limitations identified in current studies underscore the importance of exploring and developing new techniques and approaches. Furthermore, the use of neural networks seems promising for the proposed challenge. However, no work found used Dense Matching Networks to solve the problem, which will be explored in this work.

## 3. Materials and Methods

This section will detail the materials and methods used to develop the work. Section 3.1 details the dataset used. Section 3.2 presents the dense matching networks, which is the model used to extract features from the images used. Section 3.3 presents the proposed method to estimate the trajectory performed by the UAV. In Section 3.4, the evaluation metric is defined. Finally, Section 3.5 details the proposed methodology.

### 3.1. Dataset

The dataset contains 146,262 images of aircraft flights with dimensions $240 \times 120$ (considering image pairs). The data is divided into 91,221 images for training and 55,021 for testing. The dataset was made available by the DroneComp Research Group[1], which research and develops computer systems and computational methods for next-generation air transport systems. Figure 1 represents some of the images available in this dataset.



(a) (b)

**Figure 1. Examples of images contained in the datasets, captured in sequence, at time $t$ and $t + 1$.**

Each example, in addition to its image at time $t$ and $t+1$, is accompanied by some metadata, which refers to the change in position of the UAV for time $t + 1$. This metadata is detailed in Table 1.

**Table 1. Metadata for dataset samples.**

| Attribute | Description |
|---|---|
| Filename | Filename with the image pair (both north-directed and facing down) |
| Altitude | UAV altitude (sea level) at time $t$ (left image) |
| Delta | UAV altitude change between image pair |
| North | Position change between images considering the North-South axis |
| East | Position change between images considering the East-West axis |

The North and East attributes are the attributes to be predicted, i.e., given a pair of images, the regression task is to define how much the position of the UAV has changed concerning the North and East. Therefore, this information is not available for the test set.

For example, for Figure 1(a), we have the following metadata: Altitude: 179.521103; Delta: 0.713089; North: -0.857068; and East: -1.399377. For Figure 1(b), we have the following metadata: Altitude: 200.46283; Delta: -0.248322; North: -0.18989; and East: -1.648725.

### 3.2. Feature Extraction with Dense Matching Networks

In this section, we will detail the use of dense matching networks, an important model for extracting and processing images.

[1] https://www.drone-comp.ita.br/

Finding correspondence between pairs of images, including pairs of images representing consecutive frames of a video, is common in computer vision problems. In this context, the dense methods emerged, which predict a match for every single pixel in the image and do not require the detection of salient and repeatable key points [Truong et al. 2021].

An example of a dense method is the Probabilistic Dense Correspondence Network (PDC-Net) proposed by [Truong et al. 2021] for joint learning of dense flow estimation along with its uncertainties. This model learns to predict the conditional density probability of the dense flow between two images, parametrized as a constrained mixture model. Furthermore, the model can create a pixel-wise confidence map, indicating the reliability and accuracy of the prediction.

When considering a pair of consecutive images, we can apply the PDC-Net model to perform tasks such as keypoint matching, pose estimation, 3D reconstruction, and other tasks related to the geometry of the images. For example, we can input the pair of consecutive images into the PDC-Net model and obtain key point correspondences between the two images. These correspondences can be used for object tracking, motion detection, or even establishing correspondences between different scene viewpoints. So, it became the backbone of our solution, extracting features using this PDC-Net model.

Their proposed model was tested under different datasets for geometric matching and optical flow, obtaining state-of-the-art results. One of these datasets is MegaDepth [Li and Snavely 2018], which is a dataset of images collected from the internet on various topics. The PDC-Net model trained with this dataset is available on the author's Github[2].

Then, we apply the PDC-Net model that was pre-trained with the MegaDepth dataset to the entire image set, and the features are extracted. Each example contains relevant information extracted by the dense matching model, like pixel correspondences, depth maps, and other geometric matching data. So, in addition to the altitude features and altitude delta for time $t + 1$, the datasets also incorporate features derived from the PDC-Net algorithm, which will be detailed in the 3.2.1 section.

### 3.2.1. PDC-Net Features

The PDC-Net algorithm performs homography estimation, resulting in the features $ef$ (estimated flow) and $mat$ (matches from flow). The homography matrix, a geometric transformation encompassing translation, rotation, scaling, and projective deformation, aligns two correlated images. Furthermore, it is important to note the inclusion of the "conf" (confidence map) feature within the PDC-Net framework. Here is the meaning of each feature:

- *Estimated Flow* ($ef$): This feature represents an image sequence's estimated optical flow for a particular point or region. Optical flow refers to the object's apparent motion pattern between consecutive video frames. The estimated flow provides information about the direction and magnitude of the movement of objects in the scene.

[2] https://github.com/PruneTruong/DenseMatching/tree/main

- *Matches from Flow* ($mat$): This feature indicates the matches obtained from the optical flow estimation. It represents the correspondence between points or regions in consecutive frames based on the estimated flow. These matches help establish the relationship between different points or regions in the image sequence.
- *Confidence Map* ($conf$): The confidence map feature represents the level of confidence or reliability associated with the estimated optical flow. It measures how accurately the optical flow has been estimated for different points or regions in the image sequence. A higher confidence value indicates a more reliable estimation, while a lower value suggests potential errors or uncertainty in the estimated flow.

These PDC-Net features are extensively employed in various computer vision applications such as motion analysis, object tracking, and scene understanding. They offer valuable insights into the motion and dynamics of objects within a sequence of images. A $3 \times 3$ homography matrix is constructed as a fundamental component in our study. These features are calculated for each image, and following the image ordering method for estimating the UAV trajectory in Section 3.3, the features of each example will also incorporate the features of the two preceding images.

The term shift_$n$_PDCNet_$x$_$m$ refers to a specific feature generated by the PDC-Net algorithm. In this context:

- The parameter $n$ represents the shift value, indicating the temporal displacement relative to the original image, where 0 corresponds to the original figure.
- The parameter $x$ in shift_$n$_PDCNet_$x$_$m$ can take the values $ef$, $mat$ or $conf$.
- The parameter $m$ at the end of shift_$n$_PDCNet_$x$_$m$ signifies that the feature is related to the $3 \times 3$ matrix (homography matrix) generated by the PDC-Net algorithm.

Overall, shift_$n$_PDCNet_$x$_$m$ represents a feature derived from PDC-Net that captures the temporal shift and the association with the $3 \times 3$ homography matrix. This feature enables the analysis of temporal dynamics, correspondences, and confidence levels within the context of a sequence of images.

### 3.3. Recovering Temporal Information for Better Spliting

Upon careful analysis of the dataset, it became evident that each image is a segment of a recording capturing the flight trajectory of a drone. Given the consecutive nature of these frames, a correlation exists among them. To address this requirement, we employ the following approach to determine the subsequent frame or time trajectory of the UAV. The methodology revolves around the concept of utilizing the altitude information and its delta for time $t + 1$.

For each image, taking into account its altitude value and the time delta, we calculate the new altitude of the drone. This combination of altitude and delta serves as a key factor in determining the next frame or time trajectory:

- In the case of exact matches, that is, if a single match is found for the calculated altitude, we can confidently assume that it corresponds to the next frame in the video sequence;

- However, when encountering approximate matches, where there are either zero or more than one close match, we adopt an alternative approach. In such instances, we obtain the identification numbers (IDs) associated with the 100 closest altitudes and subsequently select the ID that exhibits the lowest squared error when compared to the image under consideration.

We can use altitude and delta information to establish the temporal correlation among consecutive frames of the drone's flight trajectory dataset. This approach enables us to accurately identify the next frame in the video sequence based on either exact or approximate matches, further enhancing the quality of dataset splitting and validation processes.

So with our UAV trajectory ordering estimate, we separate the first 90% images and features for model training and the remaining 10% for validation.

### 3.4. Evaluating

The metric of interest used to validate the proposed model is the Root Mean Squared Error (RMSE). This metric considers the difference squared in the value of the predictions, penalizing larger errors more. It is worth noting that the smaller the RMSE value, the better the model is. A perfect model would have an RMSE of 0.

The formula is defined by the Equation 1, where $\hat{y}_i$ is the model prediction for UAV prediction, $y_i$ is the actual value for UAV prediction, and $n$ is the total number of examples.

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{1}$$

### 3.5. Proposed Methodology

The first modeling step is to apply the dense matching network model, the pre-trained PDC-Net, to the dataset images. Next, we must define the UAV flight path to divide the dataset into training and validation. In this way, with the features already created and the dataset already divided, a Machine Learning model can now be applied to predict the position of the UAV.

A Gradient Boosting Decision Tree algorithm known for its speed and good accuracy is LightGBM [Ke et al. 2017]. Thus, LightGBM was the chosen algorithm for modeling, and then two LightGBMs were trained, one to predict the change in position of the UAV towards the North and the other to predict the change in position towards the East. Finally, evaluation is performed using Root Mean Squared Error (RMSE), comparing predicted and actual UAV positions on each axis.

Figure 2 details the workflow for the proposed methodology, where PCD-Net processes the training and test datasets for feature extraction. Then, the trajectory recovery method is applied to estimate the trajectory and order of the images available. This training dataset is divided into training and validation, where the first 90% of the data is defined as training, and the remaining 10% of the images are used to validate the trained model. That way, two LightGBM are trained, one to predict the position of the UAV relative to the North and the other to the East. Finally, the trained models are applied to the test dataset, and the predictions are saved for submission to Kaggle, the platform chosen for the KDD BR 2022 Competition.

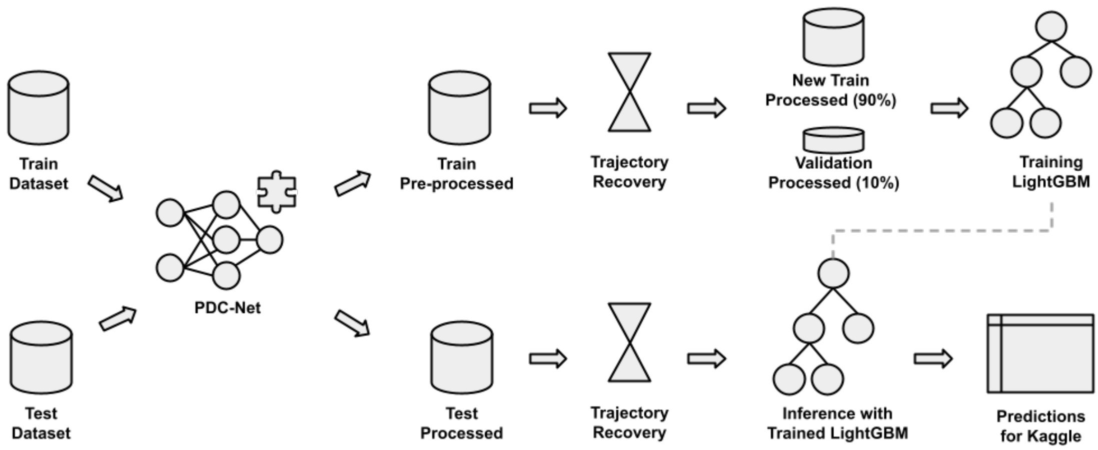



**Figure 2. Schematic of the proposed methodology workflow.**

## 4. Results and Discussion

This section encompasses the results obtained. Section 4.1 details the RMSE results obtained from local validations and Kaggle. Section 4.2 presents model interpretation results.

### 4.1. Results

To first validate our model locally, we previously split our dataset between training and validation and thus have an expectation of how much our model should reach on Kaggle's leaderboard.

Thus, as we trained two LightGBMs, one for the predictions concerning the North and the other for the East, we have different results in our validation dataset, according to Table 2, where we can see that the results in RMSE for the model for the East are smaller than for the North model.

**Table 2. RMSE results from local validation testing.**

| RMSE (North) | RMSE (East) |
|---|---|
| 0.2932 | 0.2560 |

This result is good, considering the baseline solutions released by the DroneComp group, which is an RMSE of 0.51124 for the Beta model and 0.61386 for the Alpha model, which was calculated with the predictions for North and East and together.

Then the next test is to apply the model to the blind testing dataset, available only in Kaggle, based on the competition's final results. Table 3 presents the results for the private leaderboard in the competition on Kaggle, where our solution ranked second with an RMSE of 0.29603.

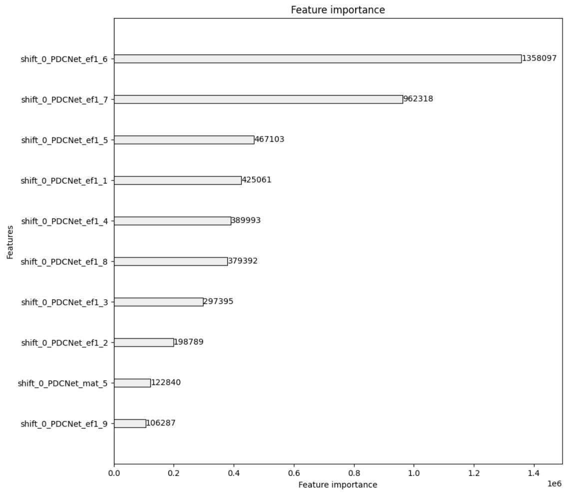
Table 3. Top 10 teams according to RMSE in Kaggle's private leaderboard.

| Team | RMSE |
|---|---|
| Team 1 | 0.24452 |
| **Team 2 (Ours)** | **0.29603** |
| Team 3 | 0.30213 |
| Team 4 | 0.42446 |
| Team 5 | 0.44856 |
| Team 6 | 0.46151 |
| Team 7 | 0.48451 |
| Team 8 | 0.48608 |
| DroneComp Beta | 0.51124 |
| DroneComp Alpha | 0.61385 |
| Team 9 | 1.112016 |
| Team 10 | 1.26917 |

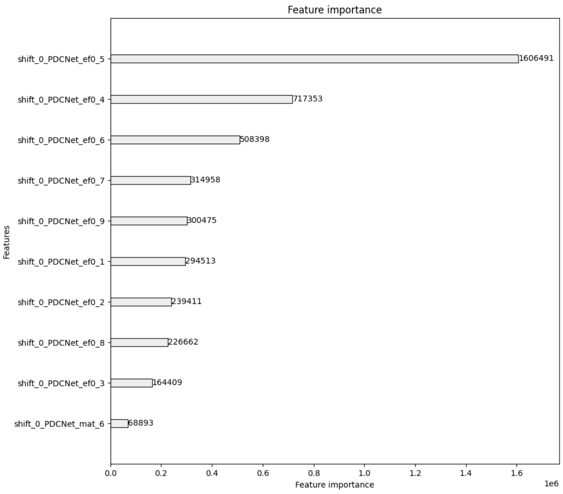
DroneComp's baseline solution is between the 8th and 9th place. It is also noted that only the top 3 teams achieved an RMSE less than 0.30213, while the rest of the competitors achieved RMSEs greater than 0.40. Thus, this indicates that our obtained results were good. Unfortunately, the analyzed works in Section 2 used datasets and metrics different from those used in this article, so it is not possible to compare them directly.

### 4.2. Interpretability

We noticed that the role of the PDC-Net and the temporal split feature were significant in predicting the drone's position and determining the order of images, respectively. In this way, firstly, we use the feature importance of the trained LightGBM models to analyze them, using information gain as a measure of feature importance. Figure 3 presents the most important features of the North and East LightGBM models.




(a) Top 10 Features for the North model




(b) Top 10 Features for the East model

**Figure 3. Feature Importance for the two trained LightGBM models.**

The feature importance in Figure 3 indicates the relevance of different features in a predictive model. In the case of our analysis of the features from PDC-Net, the feature importance reveals that the $ef$ (estimated flow) feature consistently appears in the top positions among the most important features. This suggests that the $ef$ feature carries significant predictive power and contributes significantly to the model's performance.

On the other hand, the $mat$ (matches from flow) feature appears less frequently among the top features in the importance plot. This indicates that the $mat$ feature may have a comparatively lower impact on the model's predictions. The correspondences derived from optical flow estimation might not provide as strong of a signal for the prediction task at hand.

Interestingly, the $conf$ (confidence map) feature does not appear in the top features of the important plot. This implies that the confidence level associated with the estimated optical flow, as captured by the $conf$ feature, may have limited influence on the model's predictions. It suggests that the model may rely more on other features to make accurate predictions rather than relying on the confidence level of the estimated optical flow.

Overall, based on the feature importance plot, the $ef$ feature stands out as the most important feature in our analysis. In contrast, the $mat$ feature has relatively less importance, and the $conf$ feature appears to have minimal impact on the model's predictions. For now, all features have been used. However, a feature selection step can be used in future experiments and can reduce the number of features and potentially improve the results obtained.

## 5. Conclusion and Future Work

In conclusion, this paper has explored the use of dense matching networks to predict the path and future position of UAVs in GNSS environments. Our findings suggest that dense networks can predict patterns in UAV movement, even in contested GNSS environments. We also noticed that estimated flow features were essential for the trained model. Features such as flow matches had less importance, and confidence maps had little importance.

Our approach yielded an RMSE of 0.29603, making it the runner-up team out of more than ten teams that participated in the KDD BR 2022 Competition. This could lead to better UAV navigation and operations in various applications and make UAV flights safer and more efficient in the future. Looking ahead, more research is needed to further understand and improve this method since image-based methods have extremely challenging contexts, such as trajectories over grass regions, where the image and a consecutive frame can be extremely similar.

## 6. Acknowledgments

The authors thank CAPES Grant Numbers 23038.014333/2020-46, 88887.508531/2020-00, 88887.50693/2020-00, 88887.513581/2020-00 and CNPQ Grant Numbers 441016/2022-0 and Latam DataLab Serasa Experian for support.

*In memoriam* of Bruno Klaus de Aquino Afonso, a brilliant student and exemplary colleague, whose dedication, intelligence, and friendship left an indelible mark on our hearts.

## References

Anderson, M., Brink, K., and Willis, A. (2019). Real-time visual odometry covariance estimation for unmanned air vehicle navigation. *Journal of Guidance, Control, and Dynamics*, 42:1–17.

Barmpounakis, E. N., Vlahogianni, E. I., and Golias, J. C. (2016). Unmanned aerial aircraft systems for transportation engineering: Current practice and future challenges. *International Journal of Transportation Science and Technology*, 5(3):111–122.

Dilshad, N., Ullah, A., Kim, J., and Seo, J. (2023). Locateuav: Unmanned aerial vehicle location estimation via contextual analysis in an iot environment. *IEEE Internet of Things Journal*, 10(5):4021–4033.

Erdelj, M., Natalizio, E., Chowdhury, K. R., and Akyildiz, I. F. (2017). Help from the sky: Leveraging uavs for disaster management. *IEEE Pervasive Computing*, 16(1):24–32.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3149–3157, Red Hook, NY, USA. Curran Associates Inc.

Kruber, F., Morales, E. S., Chakraborty, S., and Botsch, M. (2020). Vehicle position estimation with aerial imagery from unmanned aerial vehicles. *arXiv.org*.

Li, Z. and Snavely, N. (2018). Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050.

Liu, X., Wang, H., Fu, D., Yu, Q., Guo, P., Lei, Z., and Shang, Y. (2015). An area-based position and attitude estimation for unmanned aerial vehicle navigation. *Science China. Technological sciences*, 58(5):916–926.

Mostafa, S. A., Mustapha, A., Shamsudin, A. U., Ahmad, A., Ahmad, M. S., and Shamini Gunasekaran, S. (2018). A real-time autonomous flight navigation trajectory assessment for unmanned aerial vehicles. In *2018 International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR)*, pages 1–6. IEEE.

Rovira-Sugranes, A., Razi, A., Afghah, F., and Chakareski, J. (2022). A review of ai-enabled routing protocols for uav networks: Trends, challenges, and future outlook. *Ad Hoc Networks*, 130:102790.

Truong, P., Danelljan, M., Van Gool, L., and Timofte, R. (2021). Learning accurate dense correspondences and when to trust them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5714–5724.