Identification and Tracking of Unmanned Aerial Vehicles Based on Radar Data

Lesia Mochurad, Roman Bliakhar, Nataliia Reverenda

Lviv Polytechnic National University, 12 Bandera street, Lviv, 79013, Ukraine

Abstract
With the development of technology, Unmanned Aerial Vehicles (UAVs) have become indispensable assistants in commercial activities and have gained popularity for personal use. They have found wide applications in many areas, such as photography and video shooting, logistics, military activities, geomorphology, etc. Also of particular relevance is the problem of identifying detected objects based on the information provided. The paper proposes an algorithm and designs an automated system for solving the problem of tracking and identifying drones using data from a radar station and a photo or video camera. Thus, a Kalman filter was implemented to localize "noisy" radar measurements and CNN for binary classification of input images. As a result, more than ten times closer to the true measurements than the "noisy" ones were achieved. Also, a corresponding model was trained for classification, with an accuracy of 92% according to the F1 measure.

Keywords 1
Drone identification, drone tracking, convolutional neural network, Kalman filter.

1. Introduction

The problem of effectively countering the illegal use of UAVs has become particularly relevant [1-4]. After all, the drone manufacturing industry is actively developing, but there is no universal method of combat. The relative availability and ease of use of drones make them a convenient tool and potential illegal abuse. In many countries, particularly Ukraine [5], there are no clear requirements for purchasing and registering UAVs. In this regard, criminals can and do use the latter to transport contraband, illegal information gathering, surveillance, and terrorist acts. Hence the need for effective countermeasures.

With the appearance of drones, the means of combating them appeared, as well as the UAVs themselves - first in the military and defense industry and later in the civilian sector. Several dozens of commercial solutions designed to counter the illegal use of drones have been proposed to date. Two main steps are distinguished in the fight against them. The first step is tracking. It consists of an identification step to determine the type of object detected (for example, a drone or a bird). If, as a result of the identification, the identified object carries a potential threat, then the UAV flight path is tracked, and the second step is taken - neutralization. There are three main approaches for neutralization: radio suppression, interception using UAVs, and various small arms [6]. Sometimes there is also a third step – tracking the operator.

However, it should be understood that there is no perfect solution. After all, the UAV manufacturing industry is constantly developing, both hardware-wise and structurally. Also, the effectiveness of using specific means directly depends on the infrastructure and terrain relief of the protected area.

Today, there are a large number of UAV varieties. They differ both in terms of purpose and construction. However, there are still some similarities between them. Since this is an aircraft, it must have an engine (or several), the operation of which is accompanied by the release of noise and heat. The drone is controlled remotely using transfers of radio frequency signals.

Modern location mechanisms are aimed at detecting the above features. There are four main types of drone monitoring equipment [6,7]: radar, optical, acoustic, and radio frequency.

More than one of the above methods is often needed to track drones correctly. Birds can serve as an example of obstacles. Since it is a living organism, it emits heat. Also, when flying, the bird flaps its wings, which is accompanied by a sound. Which, in the case of optical-thermal and acoustic equipment, can be determined by the system as a UAV. No less critical are the topography and infrastructure of the area, which can make it difficult to use some methods. So, the city's noise will introduce many obstacles, practically making it impossible for the acoustic equipment to work properly. Therefore, complex systems using several sensors are used to avoid such ambiguities.

Also, statistical analysis and artificial intelligence methods are widely used in modern solutions to improve the quality and automation of detection processes [8-10]. The main task is identifying detected objects based on the provided information.

Therefore, our work aims to develop an algorithm to solve the problem of correctly identifying and tracking UAVs based on data received from radars and sensors.

The object of the study is a software application for the correct tracking and identification of UAVs based on data from radars and sensors.

The subject of research is methods and means of identification and tracking of drones.

2. Analysis of literary sources

In recent years, more and more research on this topic has appeared. Thus, in the article [11], an overview of the main methods of detecting and classifying drones using machine learning (ML) methods were carried out. Also, the authors considered the main means of probing and the advantages and disadvantages of using each of them. In particular, such as:

- The problem of erroneous perception by radars of ground objects that are not drones (for example, birds).
- The sensitivity of acoustic equipment to ambient noise, as well as the negative impact of wind and meteorological changes.
- Optical systems are no less sensitive to meteorological factors, for example: dust, fog, rain. Changes in natural lighting during the day can also introduce inaccuracies into the work.
- • The impossibility of detecting drones flying autonomously without any communication channels using the radio frequency method.

The authors devoted the main part of the work to the review of identification methods. They distinguished two types of object identification: identification as a binary classification applied to determine which object is being observed (e.g., "drone" or "not a drone," "drone" or "bird"), and classification with constructional identification - hardware features of the object (determination of the type, parameters or model of the UAV). Although a large number of methods and approaches were considered in the work, their comparison is considered inappropriate. A quote from the text: "... most of the performed research is experimental, and the outcomes from different papers can hardly be compared. A general requirement-driven specification for the problem of drone detection and classification problem is still missing as well as reference datasets which would help evaluate different solutions". In summary, the authors highlight the absence of distance-based regression models and propose such an approach as a research area.

In [12], a new approach to detecting drones based on data from radio frequency sensors is proposed. During the research, the authors concluded that the lower range of the signal (between the drone and the controller) is sufficient to be used as a feature vector for ML. However, the upper range also carries some information. This approach, combined with the XGBoost algorithm (an ensemble method based on decision trees) [13], allows you to significantly reduce the time spent on the calculation and obtain satisfactory accuracy. According to the results of the experiments, the use of

XGBoost gave about an 8% increase in classification accuracy compared to DNN. In conclusion, it can be noted that this approach will be a good choice when creating rapid response systems.

The next considered work [14] presents a CNN that uses a decision tree and an ensemble structure to obtain a complete characterization of a UAV. The result of the model is information about the orientation in space and design features of the drone. This approach made it possible to determine the parameters of the flight (such as the angle of inclination and course) and classify the body's different parts (engines, body, cameras). As a result of the conducted experiments, more than 90% accuracy was obtained, making it possible to use the model in working with real data. The authors propose to use the model in combination with existing object detection technologies, which will allow obtaining the full characteristics of the drone.

Radar equipment is widely used to detect drones, but using radar measurements for classification is problematic. The study [15] proposed using spectrograms constructed from measurements obtained from radars to classify objects using a convolutional network further. The authors emphasize the need for a large amount of data to obtain satisfactory accuracy due to training. Therefore, a set of spectrogram images of drone and bird flights was created. The resulting images were used to train two models: GoogLeNet and CNN, of its architecture. According to the results of the experiments, accuracy was achieved – 99% and 94%, respectively. However, performance indicators for the latter were better.

It can be said that two main stages are distinguished in combating the illegal use of drones - tracking and neutralization. There is a wide selection of equipment, but each has significant disadvantages. So, for example, radar measuring devices are considered the best solution for tracking. However, based on their information, it is impossible to unambiguously determine whether the detected object is a potential threat (for example, a bird may be detected).

Over the past decade, significant progress has been made in image processing and computer vision. Today's technologies make it possible to detect objects in a video stream and image and classify them [16, 17]. These technologies can potentially be used in the given task.

Therefore, the research task in this work is to develop an algorithm and design an automated system to solve the problem of tracking and identifying drones based on data receivers from radar and photo or video cameras. The developed software should process radar data and analyze images using artificial intelligence.

3. Methods and means of research

3.1. Input data

The system developed in this paper uses data received from radar and camera. The information received from the radar may differ depending on the configuration features of the equipment. In the most trivial case, the data provided by the locator contains information about the speed, distance, and angle relative to the radar itself. For the possibility of further use in automated systems, this format requires calculations to determine the spatial coordinates of the observed objects. However, most modern radars implement this calculation mechanism with their hardware capabilities, so they return spatial geographic coordinates, speed, and accompanying information as output data, which is not considered within the scope of this study.

The camera provides a digital image of the observed space. The data comes in the form of a file that contains a two-dimensional array of vectors. Each array element describes the image's corresponding pixel (color point). A pixel is represented as a vector whose elements correspond to a certain color space. The presented work uses the RGB (Red, Blue, Green) space, where each element of the three significant vectors represents the intensity of the corresponding color.

Therefore, the input data are the spatial coordinates and speed of the observed object, which are received from the radar with a certain frequency (for example, every second), as well as the image received from the camera, which is presented in the form of a two-dimensional array of RGB vectors.

3.2. Output data

The data coming from the radar has some deviation from the "true." Therefore, for their further use, it is necessary to localize these trajectories to bring them closer to real indicators.

Images are analyzed by AI methods to obtain useful information within the defined task. This study is a classification for object identification, i.e., the useful information is the determined type of the observed object.

Therefore, the output data of the proposed system are localized spatial coordinates and information about the type of the observed object.

3.3. Proposed algorithm

As mentioned above, the input data sources are radar measurements and the image provided by the camera. Since this information arrives simultaneously and needs immediate processing, we consider these stages parallel in the proposed algorithm.

The tracking procedure can be described as an iterative process. Probing by the locator occurs with a certain frequency, so we will consider each radar cycle an iteration. Data is sent from the radar to the system at each iteration (provided that an object is detected in the surveillance sector). They are transferred to the system component responsible for their processing, namely localization. In parallel with this, an image of the observed sector, which contains the detected object, is received from the camera. It is proposed to use the Kalman filter to solve the localization problem [18]. Object identification within this work's framework is reduced to a binary classification of the input image using convolutional neural networks (CNN) [19].

The recursive Kalman filter algorithm works in the "prediction-correction" cycle. The work begins with some assessment of the initial state $\hat{x}_0$, and error covariance matrices P_0. The "prediction-correction" format is applied cyclically at each stage of localization. The state vector is predicted from the dynamic state equation (1).

$$\hat{x}_{t|t-1} = F_{t-1}\hat{x}_{t-1}, \tag{1}$$

where, $\hat{x}_{t|t-1}$ – predicted state vector, $\hat{x}_{t-1}$ – previous state estimation vector, F – transition matrix between two states. It is worth noting that «t/t-1» is an abbreviated designation of the state at a discrete moment in time t, taking into account its previous state at a discrete moment in time $t-1$, that is, the system model adjusted in the previous step is used for prediction.

Next, the error covariance matrix is calculated (2).

$$P_{t|t-1} = F_{t-1}P_{t-1}F_{t-1}^T + Q_{t-1}, \tag{2}$$

where, $P_{t|t-1}$ – matrix of covariances of the predicted state error, P_{t-1} – pre-estimated state error covariance matrix, and Q – process noise covariance matrix. Again, «t/t-1» indicates that this is the expected covariance matrix at t based on the system model at $t-1$.

After obtaining the predicted values, the gain matrix K is calculated in (3).

$$K_t = P_{t|t-1}H_t^T(H_t P_{t|t-1}H_t^T + R_t)^{-1}, \tag{3}$$

where H – transition matrix between state and measurement, and R – measurement noise covariance. The Kalman filter calculates a gain matrix for each measurement that determines the effect of the input vector on the estimation of the system state. In other words, when a noisy measurement comes in, the transmission coefficient will trust its estimate more than the new inaccurate information.

After calculation, the gain matrix K is used to weight the measured and predicted values. Based on this, a new assessment of the state of the system is built in (4).

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(z_t - H_t\hat{x}_{t|t-1}), \tag{4}$$

where z – the input measurement at this point in time.

Similarly, the state error covariance matrix is updated (see equation (5)).

$$P_{t|t} = (I - K_t H_t)P_{t|t-1}, \tag{5}$$

where I – identity matrix.

In the presented work, CNN is used for binary classification of the detected object based on the input image. For practical use, CNN requires a preliminary training stage. The development of the classification module starts with the training of the model. Since this work solves the problem of binary classification of images, therefore, the dataset for training consists of two classes of images -

drones and birds [20]. The selected set contains 826 images, of which 428 are drone images and 398 are images. The images were premixed and divided into training and validation samples in the ratio of 80% to 20%, respectively. Also, an annotation file was created for each of the samples, containing the names of the files and the classes to which the images belong.

All input images are reduced to a given size (150x150), to save computing resources during further training. The principle of the scaling operation is similar to the blending layer, with an aggregation function applied to a certain neighborhood of pixels. The disadvantage is that this approach loses some information.

Then, using the Python programming language and the Keras library, we created a CNN model, the architecture of which is shown in Figure 1.

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 148, 148, 32)      896

conv2d_5 (Conv2D)            (None, 146, 146, 32)      9248

max_pooling2d_3 (MaxPooling  (None, 73, 73, 32)        0
2D)

dropout_4 (Dropout)          (None, 73, 73, 32)        0

conv2d_6 (Conv2D)            (None, 71, 71, 32)        9248

max_pooling2d_4 (MaxPooling  (None, 35, 35, 32)        0
2D)

dropout_5 (Dropout)          (None, 35, 35, 32)        0

conv2d_7 (Conv2D)            (None, 33, 33, 32)        9248

max_pooling2d_5 (MaxPooling  (None, 16, 16, 32)        0
2D)

dropout_6 (Dropout)          (None, 16, 16, 32)        0

flatten_1 (Flatten)          (None, 8192)              0

dense_2 (Dense)              (None, 64)                524352

dropout_7 (Dropout)          (None, 64)                0

dense_3 (Dense)              (None, 2)                 130

=================================================================
Total params: 553,122
Trainable params: 553,122
Non-trainable params: 0
_____
```

Figure 1: Structure of the model architecture CNN

As you can see from Figure 1, the model consists of four folded layers (Conv2D), three unifying (MaxPooling2D), and two layers of perceptron (Dense). Also, at the intermediate stages, dropout regularization was added (Dropout). Dropout works by randomly disconnecting neurons and their respective connections. This regularization aims to prevent overtraining of the model, which is relevant in this case since the training set is quite small.

Then the model is trained. The training process can be described as follows: a certain number of images are fed as input, and the network returns a prediction for each. Then the predictions are compared with the "true" values using a loss function. The loss function evaluates the correspondence

of the obtained predictions to the true ones. Based on the results of the loss function, the model is adjusted. The process is repeated for the entire training set.

The validation process is similar to training, but the model is not adjusted based on the loss function results. Validation is performed after each training epoch.

4. Numerous experiments

To generate a dataset that would simulate the measurements of the locator, we took the files of drone flight logs [21]. The flight logs contain time-sampled geographic coordinates of the drone, throughout the entire flight trajectory (see Figure 2).
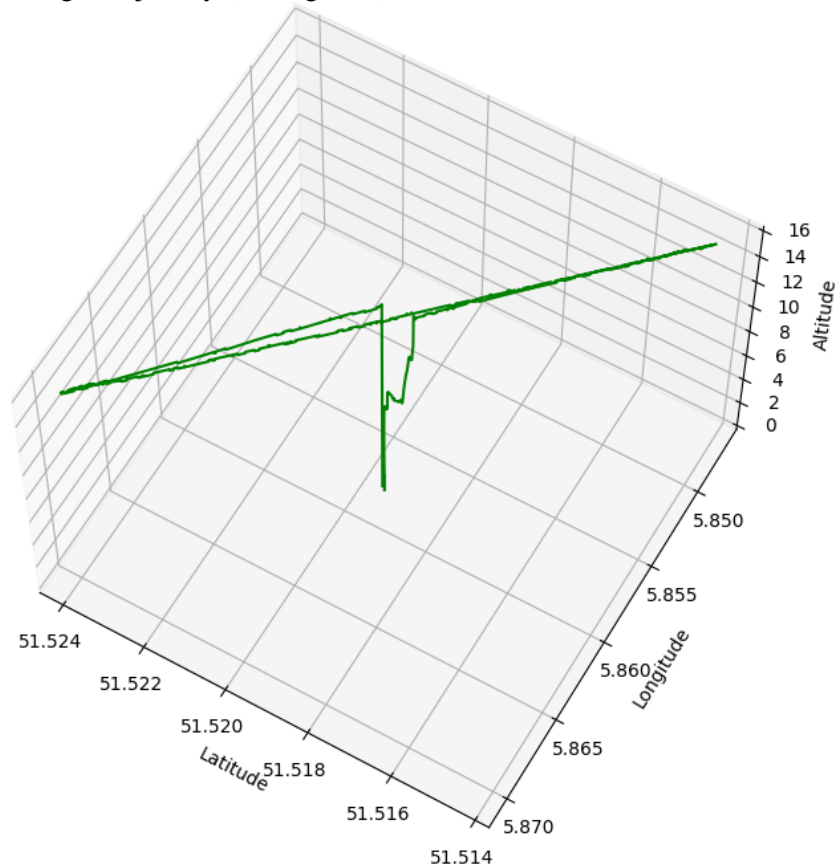

Figure 2: Three-dimensional visualization of the drone's flight path

The geolocation data of the logs are received at intervals of 0.2 seconds. To simulate radar measurements, records were selected at 1 second intervals. Then, "noise" was added to these data to simulate the radar measurement error. Noise, in this case, is a random number generated by a normal (Gaussian) distribution. For a random variable, the normal distribution is defined by the formula (6).

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \qquad (6)$$

where μ – arithmetic mean, σ – standard deviation. To generate noise: $\mu = 0$, σ – permissible measurement error (since the radar equipment has a permissible measurement error specified in the documentation). On Figure 3 shows the generated data (blue dots) relative to the "true" data (red dashed line) in a certain sector. For convenience, only the longitude and latitude trajectory in a two-dimensional coordinate system is shown.
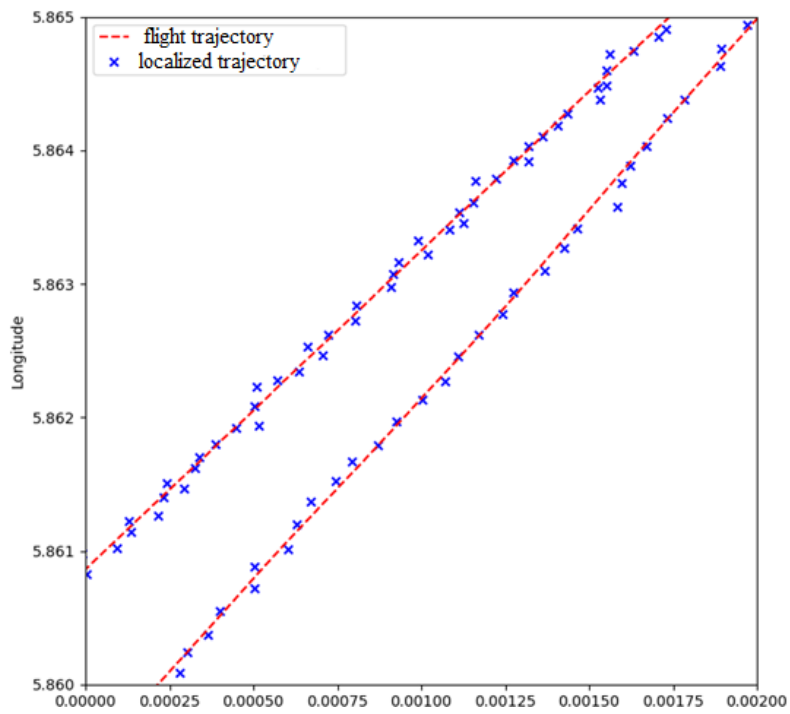
Figure 3: Flight path and radar measurements in a certain sector

The above operation was performed for each flight log (3 in total). As a result, three files simulating radar measurements were obtained and will be used in further work for simulations and testing.

On Figure 4 shows the localization result (green line) relative to the radar measurements(blue dots) and the "true" trajectory (red dotter line).
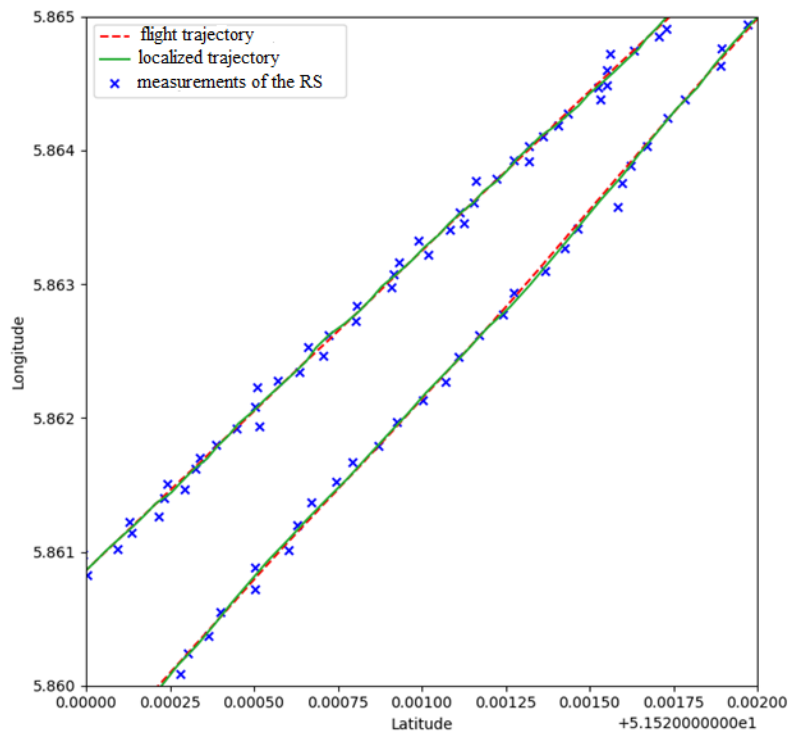

Figure 4: Localization results relative to radar measurements and "true" trajectory

Figure 4 is convenient for visualizing the results, but it is difficult to judge the filter's performance from the image. Therefore, for a more objective assessment, for each file with "noisy" radar measurements, we will find the root mean square error relative to the "true" trajectory. We will perform the same operation for the localized data, which will allow for further comparison in the "before-after" format. The results of the calculations are shown in Table 1.

Table 1

Results of calculating the root mean square error

Flight log number	Root mean square error	
	«Noisy» measurements	Localized measurements
1	2.7×10^{-8}	6×10^{-9}
2	3.2×10^{-8}	5.1×10^{-9}
3	2.9×10^{-8}	5.3×10^{-9}

Table 1 shows that for localized data, the root mean square error is ten times lower than the error for "noisy" data. It should be noted that the measurements are presented in the form of geographic coordinates, and therefore so are the deviations. In this case, the root mean square error, when converted to the metric system, can be several meters. Therefore, the obtained localized data reflects a significant approximation to the true data, which can also be seen in Figure 4, where the green line describes the localized trajectory, and the red dashed line is the true trajectory.

In this study, the model was trained for 90 epochs, resulting in a prediction accuracy of about 94%. Below, Figures 5 and 6 show plots of the loss function values against the accuracy estimates, respectively. Visualization of the dynamics of these indicators in relation to epochs allows us to assess the learning process.
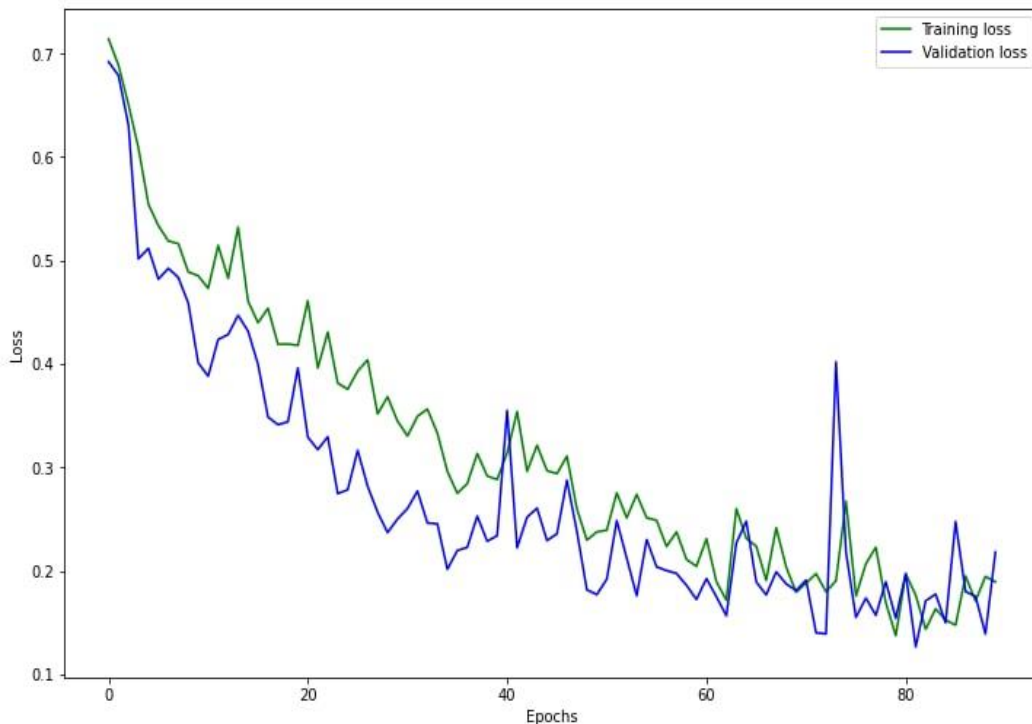

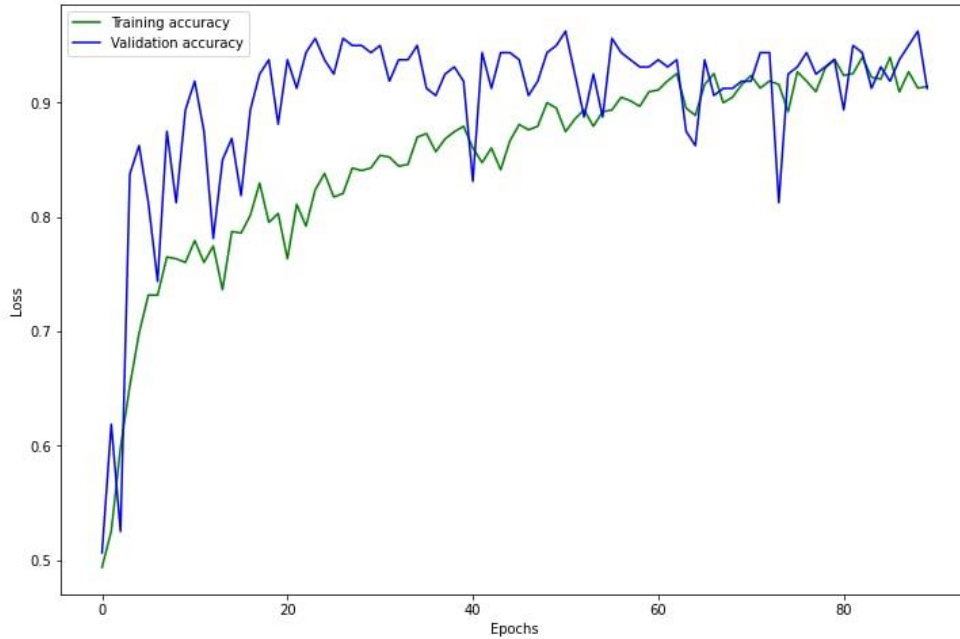**Figure 5**: The value of the loss function at each of the epochs

Figure 6: The value of the forecast estimate in each of the epochs

Several metrics were used to evaluate accuracy, and the results for each of them are presented in Table 2.

Table 2

Estimates of class predictions by several metrics

class	Metric		
	Precision	Recall	F1
Dron	0.97	0.86	0.91
Bird	0.88	0.98	0.93

Figures 5 and 6 show plots of the loss and prediction function values at each epoch for the training and validation samples. Visualization of these curves is used to diagnose the learning process [22]. For example, from the graph of the loss function values, one can conclude how well the model is trained. In the case of undertraining or overtraining, the curves for the training and validation samples at the end of the training process are divergent (not converging - undertraining, diverging - overtraining), as can be seen from the above figures, in this case, the curves coincide, which indicates a normal course of the training process.

Also, both figures show that the curves are "broken" throughout the training process. The reason for this is dropout regularization on the intermediate layers. However, as mentioned earlier, in this case, it is a necessity since the training data set is small.

Next, let's analyze the metric scores for each class, which are presented in Table 2. The (Precision) score for the "drone" class is 97%, close to perfect. However, the (Recall) score is 86%. This difference may be due to inappropriate annotations or uninformative images in the training data set (e.g., images that do not contain the objects in question). In the case of the bird class, the situation is the opposite, with 88% accuracy and 98% completeness, which may indicate a lack of training data for this class. The absolute score for the F1 metric is 92%, which is satisfactory.

5. Conclusions

This paper proposes an algorithm to solve the problem of automated identification and tracking of drones using real-time data from radar and a camera.

The article analyzes the existing problem and existing solutions in the field of drone tracking. Based on the review, a new conceptual model of drone tracking and identification is proposed, where the data sources are radar and a camera. Using the implemented Kalman filter, the trajectory tracked by the locator is localized by a limited number of parameters, namely, spatial coordinates. According to the results of the experiments, the output localized trajectory is more than ten times closer to the true one compared to the input "noisy" data. The detected object was also identified using the image provided by the camera. For this purpose, a CNN model for binary classification was developed. The model was trained on a small set of images of birds and drones. According to the experimental results, the model's accuracy is 92%.

The developed system can serve as a starting point for creating a full-fledged tool to counteract the misuse of drones. The main directions for further research are: solving the problem of tracking and identifying several objects in the monitored sector simultaneously; modernization of the classifier to identify the type or model of the detected drone.

6. References

[1] J. Śledziowski, P. Terefenko, A. Giza, P. Forczmański, A. Łysko, W. Maćków, G. Stępień, A. Tomczak, A. Kurylczyk. "Application of Unmanned Aerial Vehicles and Image Processing Techniques in Monitoring Underwater Coastal Protection Measures." Remote Sensing (2022) 14(3):458. doi:10.3390/rs14030458.

[2] Y. Li, M. Liu, D. Jiang. "Application of Unmanned Aerial Vehicles in Logistics: A Literature Review." Sustainability (2022) 14, 14473. doi:10.3390/ su142114473.

[3] A. Michalska, K. Karpinska. "Capabilities of the Unmanned Aerial Vehicles in Logistic Support." Safety & Defense, 4 (1) (2018) 22–26.

[4] Szymon Śledź, Marek W. Ewertowski, Jan Piekarczyk. "Applications of unmanned aerial vehicle (UAV) surveys and Structure from Motion photogrammetry in glacial and periglacial geomorphology." Geomorphology, Volume 378, (2021), 107620. doi:10.1016/j.geomorph.2021.107620.

[5] State aviation service of Ukraine. [Electronic resource] URL: https://avia.gov.ua/bezpilotni-povitryani-sudna-2/.

[6] V.U. Castrillo, A. Manco, D. Pascarella, G. Gigante. "A Review of Counter-UAS Technologies for Cooperative Defensive Teams of Drones." Drones. 2022; 6(3):65. doi:10.3390/drones6030065.

[7] S. Bjorklund, N. Wadstromer. "Target Detection and Classification of Small Drones by Deep Learning on Radar Micro-Doppler." 2019 International Radar Conference (RADAR). (2019). doi:10.1109/radar41533.2019.171294.

[8] M.Z. Anwar, Z. Kaleem, A. Jamalipour. "Machine Learning Inspired Sound-Based Amateur Drone Detection for Public Safety Applications." IEEE Transactions on Vehicular Technology. (2019), 68(3):2526–2534. doi:10.1109/tvt.2019.2893615.

[9] Y. Zhao, X. Xu, K. Wang. "Research on small target pedestrian detection based on improved YOLO." International Journal of Wireless and Mobile Computing. (2021), 20(3):281. doi:10.1504/ijwmc.2021.10038592.

[10] Y.J. Teoh, C.K Seow. "RF and Network Signature-based Machine Learning on Detection of Wireless Controlled Drone." 2019 PhotonIcs & Electromagnetics Research Symposium – Spring (PIERS-Spring), (2019). doi:10.1109/piers-spring46901.2019.9017231.

[11] B. Taha, A. Shoufan. "Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research." IEEE Access. (2019), 7:138669–138682. doi:10.1109/access.2019.2942944.

[12] O.O. Medaiyese, A. Syed, A.P. Lauf. "Machine Learning Framework for RF-Based Drone Detection and Identification System." 2021 2nd International Conference On Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), (2021). doi:10.1109/icon-sonics53103.2021.9617168.

[13] T. Chen and C. Guestrin. "XGBoost: A Scalable Tree Boosting System." in KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2016):785–794. doi:10.1145/2939672.2939785.

[14] S. Scholes et al. "DroneSense: The identification, segmentation, and orientation detection of drones via neural networks." IEEE Access. (2022). doi:10.1109/access.2022.3162866.

[15] S. Rahman, D.A. Robertson. "Classification of drones and birds using convolutional neural networks applied to radar micro-Doppler spectrogram images." IET Radar, Sonar & Navigation. (2020). 14(5):653–661. doi:10.1049/iet-rsn.2019.0493.

[16] L. Mochurad, A. Dereviannyi, U. Antoniv. "Classification of X-Ray Images of the Chest Using Convolutional Neural Networks." IDDM 2021 Informatics & Data-Driven Medicine. Proceedings of the 4th International Conference on Informatics & Data-Driven Medicine. Valencia, Spain, November 19 - 21, (2021). 269-282.

[17] L.I. Mochurad. "Canny Edge Detection Analysis Based on Parallel Algorithm, Constructed Complexity Scale and CUDA". Computing and Informatics, (2022). 41(4): 957–980. doi:10.31577/cai_2022_4_957.

[18] Pei Yan, Biswas Swarnendu, Fussell Donald S., and Keshav Pingali. "An Elementary Introduction to Kalman Filtering." *ArXiv*, (2017). Accessed February 3, 2023. https://doi.org/10.48550/arXiv.1710.04055.

[19] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, Tsuhan Chen. "Recent advances in convolutional neural networks." Pattern Recognition, (2018), 77:354-377. doi:10.1016/j.patcog.2017.10.013.

[20] Birds vs Drone Dataset. [Electronic resource] URL: https://www.kaggle.com/datasets/harshwalia/birds-vs-drone-dataset.

[21] Drone identification and tracking. [Electronic resource] URL: https://www.kaggle.com/competitions/icmcis-drone-tracking/data.

[22] Akira Tanimoto, So Yamada, Takashi Takenouchi, Masashi Sugiyama, Hisashi Kashima. "Improving imbalanced classification using near-miss instances". Expert Systems with Applications, (2022), 201(117130). doi:10.1016/j.eswa.2022.117130.