A two-stage approach for aircraft detection with convolutional neural network

Wael Toghuj[1], Yousef Alraba'nah[2]
[1]Department of Computer Science, Al-Ahliyya Amman University, Amman, Jordan
[2]Department of Software Engineering, Al-Ahliyya Amman University, Amman, Jordan

Article Info

Article history:

Received Jan 2, 2024
Revised Mar 6, 2024
Accepted Apr 16, 2024

Keywords:

Aircraft detection
Convolutional neural network
Deep learning
Remote sensing system
Two-stage detection algorithm

ABSTRACT

Over the past few years, object detection has experienced remarkable advancements, primarily attributable to significant progress in deep learning architectures. Nonetheless, the task of identifying aircraft targets within remote sensing images remains a challenging and actively explored area. Presently, there are two main approaches employed for this task: one utilizing convolutional neural network (CNN) techniques and the other relying on conventional methods. In this work, a CNN based architecture is proposed to recognize aircraft types using remote sensing images. The experiments performed on multi-type aircraft remote sensing images (MTARSI) dataset show that the proposed architecture achieves 97.07%, 94.81%, and 94.44% accuracy rates for training, validation and testing sets. The results approve that, the architecture outperforms state of the art models.

Corresponding Author:

Wael Toghuj
Department of Computer Science, Al-Ahliyya Amman University
Al-Saro, 19328, Amman, Jordan
Email: w.toghuj@ammanu.edu.jo

1. INTRODUCTION

A remote sensing system measures the physical characteristics of an area by measuring its reflected and emitted radiation from afar (usually from a satellite or aircraft). Aircraft detection in remote sensing images, which falls under the realm of computer vision, is a crucial aspect of image processing utilizing deep learning techniques. In recent times, numerous advanced algorithms have emerged and been utilized effectively for aircraft detection across various scenarios. As an example, manned and unmanned aircraft operations may be subject to mid-air collisions (MACs) or near mid-air collisions (NMACs), especially at low altitudes [1]. On the one hand, due to the emergence of economic globalization, aircraft have assumed a prominent role within the aviation industry. Consequently, the detection of aircraft objects holds great significance, offering valuable guidance in this regard. Also, the challenges associated with object detection in remote sensing images are closely tied to the background environment in which the objects are situated. When the background consists of an airport, clouds or sea, there exist substantial disparities between the detected targets and the surrounding environment. This imbalance between the background and the detection target presents an additional obstacle. Furthermore, the task of locating small-scale objects poses an even greater level of difficulty [2].

Applications of geospatial intelligence rely heavily on object detection in satellite imagery. For example, emergency responders need overhead imagery to assess disaster damage quickly. Crowd-sourced action may be taken following a major crisis, assisted by pre- and post-event images for damage assessment. With the Tomnod website, anyone with an internet connection can help out with search tasks through DigitalGlobe's crowdsourcing platform. The search for Malaysia Airlines flight MH370 in 2014, for instance, involved more than 10 million people. Using this crowdsourcing platform, satellite imagery was provided over one million square kilometers [3].

A rapid advancement in satellite technology has greatly improved the spatial and temporal resolution of remote sensing products for both civil and military purposes. A key concern when it comes to remote sensing images for detecting objects is aircraft. Identifying aircraft types is critical both in military and civil applications, as it is required to recognize targets in remote sensing images. Due to fine-grained features, however, the task is extremely challenging with small changes between classes being caused by highly comparable subcategories, and large changes within classes being caused by differences in size, position, and angle. The J-20, Su-57, F-22 and F-35 are just a few examples of the different types of fighter aircraft illustrated in Figure 1. Even though their roles and purposes are distinct, these aircraft look similar.

This detection task needs to be reliable to enable automation of site analysis, especially to generate alerts on unusual events. Achieving robustness in the face of various challenges such as noise interference, shadows, fluctuations in lighting conditions, or alterations in ground texture poses significant difficulties. However, it is imperative for practical applications to address these challenges effectively. Only through robust and reliable detection mechanisms can the full potential of automated site analysis be realized, facilitating proactive responses to emerging situations and enhancing overall operational efficiency as shown in Figures 2(a) and 2(b) [4].

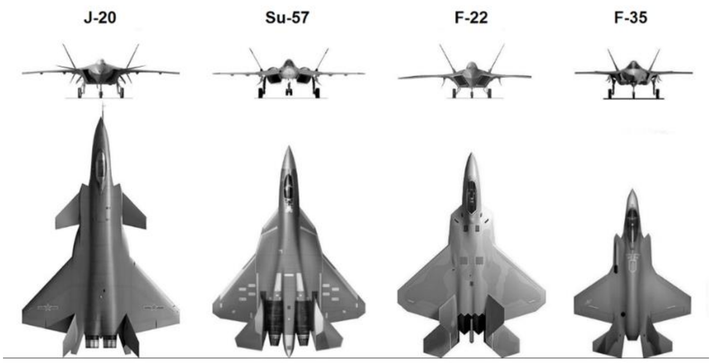
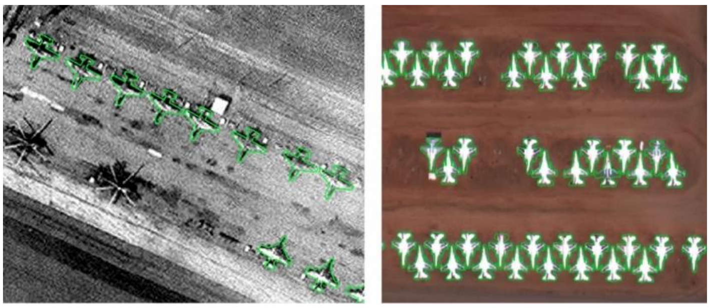


Figure. 1. Four types of fighter aircraft: J-20, Su-57, F-22 and F-35



(a) (b)

Figure 2. Illustration of the data diversity (with ground truth), (a) SU-25 and (b) F-16

2. RELATED WORK

In the past, the analysis of remote sensed images required specialized equipment and trained personnel. However, thanks to the advancements in machine learning techniques, these laborious tasks can now be accomplished with reduced human intervention. Machine learning encompasses automated computational procedures that have the ability to learn and solve problems based on existing examples. This learning process involves gathering knowledge, comprehending the acquired knowledge, and acquiring skills through experience. As the method continues to learn and gain more experience, its performance improves. This approach is particularly effective in scenarios where a significant amount of data is available for the learning process [5]. With the rapid progress of deep learning, deep learning algorithms for object detection have increasingly gained prominence. These algorithms can be broadly classified into two main branches. The first branch is the one-stage object detection method, which treats object detection as a combined problem of regression and classification. The second branch is the two-stage object detection method, which relies on regional candidate proposals to detect objects. These two approaches represent the prevailing trends in deep learning algorithms for object detection. This set of algorithms primarily consists of [6]–[10]:

- Region-based convolutional neural network (R-CNN), a two-stage detection algorithm.
- Spatial pyramid pooling (SPP)-net, a convolutional neural architecture that employs spatial pyramid pooling to remove the fixed-size constraint of the network.
- Faster R-CNN, a single-stage model that is trained end-to-end.
- Mask R-CNN, convolutional neural network (CNN) and state-of-the-art in terms of image segmentation.

As opposed to the two-stage method, the one-stage method treats object detection as a classification and regression problem. The method can therefore be directly implemented into convolution neural networks to target object classification and localization and does not require the generation of many regional candidate proposals to lead the method. This series of algorithms mainly include [11]–[13]:

- You only look once (YOLO) which aims to predict a class of an object and the bounding box that defines the object location on the input image.
- Single Shot Detector (SSD) it has no delegated region proposal network and predicts the boundary boxes and the classes directly from feature maps in one single pass.
- Deconvolutional single shot detector (DSSD) with deconvolutional path, which improves the previous SSD algorithm.
- Feature-fusion SSD (FSSD) is a fast and efficient algorithm for subgroup set discovery.

Two-stage methods have higher accuracy in location, but their training time is too long, which slows down detection speed. One-stage methods, on the other hand, have fast detection speeds, but have lower accuracy in location, and have a series of missed detection problems, especially for small targets. Ultimately, the choice between the two-stage and one-stage methods depends on the specific requirements of the application.

For aircraft detection, a variety of deep learning methods have been developed. One of the prominent International Conference on Civil Aviation [14], which has been regarded as the leading detector, achieving top accuracy in the challenging common objects in context (COCO) benchmark. Nevertheless, over the past few years, one-stage detectors like the feature pyramid network (FPN) [15] have achieved comparable accuracy to the most intricate two-stage detectors on the COCO benchmark. In their research, Lin *et al.* [16] pinpointed a crucial challenge that has been impeding one-stage detectors from attaining state-of-the-art accuracy, namely, the issue of class imbalance during the training process. To tackle this challenge head-on, they introduced an innovative loss function designed to eliminate this obstacle. Simultaneously, they integrated enhancements such as feature pyramid network (FPN) [15] into their model, referred to as RetinaNet [16]. This combination of strategies effectively overcame the class imbalance issue, resulting in improved detection performance. Yan [17] focused on addressing the challenge of accurately detecting aircraft in remote sensing images, particularly under complex conditions. To tackle this issue, a novel method called center-based proposal regions and invariant features (CPIF) was developed. This method was specifically designed to handle challenging image deformations, with a particular emphasis on rotations, enabling precise aircraft detection. The objective of Wang *et al.* [18] in his research was to address the challenge of the high cost and difficulty in obtaining spaceborne optical remote sensing images. In order to overcome this issue, they proposed an aircraft detection algorithm that could effectively detect aircraft objects even with limited samples. The algorithm aimed to enhance early warning capabilities by accurately identifying aircraft objects in the images. In their work, Wang *et al.* [19] introduced a weakly supervised learning algorithm based on AlexNet for aircraft detection. They emphasized that their method eliminates the need for object location data during the training phase. By utilizing only image-level labeled data, the proposed model achieved promising results comparable to established methods such as YOLOv3 and Faster-RCNN in terms of aircraft detection performance. In the study described in [20], a deep learning-based aircraft detection model was developed. The authors introduced a modified version of the RCNN network that incorporated a SoftMax layer for classification instead of the conventional support vector machine (SVM) approach. During the training and testing phases, a comprehensive dataset comprising two classes, namely "Plane" and "NPlane", was utilized. To tackle the challenge of detecting small-scale aircraft, Zhou *et al.* [2] adopted the multiscale detection network (MSDN) structure. This approach involved dividing the input images into smaller grids, enabling the detection of small-sized aircraft. Additionally, the authors introduced the deeper and wider module (DAWM) to counteract the influence of background noise caused by complex backgrounds. The DAWM module enhanced the network's perceptual field, allowing it to better handle the complexities of the background and improve the accuracy of aircraft detection.

The remainder of the paper is organized as follows: section 3 presents the outline of the method of the proposed aircraft detection system with a CNN explanation in brief. Section 4 provides the details of the methodology of the proposed model by presenting a description of the datasets used in the study and the experimental studies performed. Findings from the experiments are also presented in this section. In section 5, the conclusion paragraph is included.

3. PROPOSED MODEL

This section introduces a comprehensive framework designed to address the challenges of aircraft detection from remote sensing images. First, we discuss the MTARSI dataset, which has been used in the proposed model. Then we discuss the proposed model and its CNN structure. Figure 3 shows the steps that we follow to detect aircraft type.

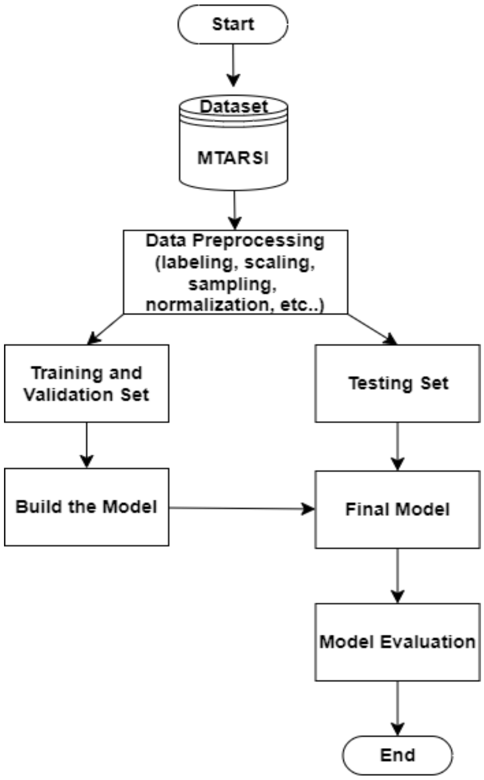


Figure 3. Aircraft type detection framework

3.1. MTARSI dataset

Multi-type aircraft remote sensing images (MTARSI) is the dataset that is used to evaluate the proposed work. MTARSI has 9,385 remote sensing images, which have been acquired from Google Earth satellite imagery. These sample images have undergone thorough labeling by seven experts specializing in the interpretation of remote sensing imagery. The dataset contains a wide range of variations within its aircraft images, including instances where aircraft of the same model appear in different colors, poses, viewpoints, with varying backgrounds, and at different resolutions [21].

Each image in the dataset exclusively contains a single, complete aircraft. The dataset is made up of 20 different types of aircraft. The aircraft types included in the dataset are B-1, B-2, B-29, B-52, Boeing, C-130, C-135, C-17, C-5, E-3, F-16, F-22, KC-10, C-21, U-2, A-10, A-26, P-63, T-6, T-43. The number of model aircraft images in each category varies, ranging from 230 to 846. Furthermore, MTARSI dataset does not represent real-world distribution, with a disproportionate representation of military aircraft. It is worth mentioning that the MTARSI dataset is limited to only 36 airports and includes many images of uncommon military aircraft. Consequently, certain aircraft are shown in multiple images, although they were captured under varying imaging conditions. To enlarge the dataset, the researchers conducted augmentation processes, which involved segmenting airplanes, applying rotations and flips, and ultimately altering backgrounds.

3.2. The proposed model

CNNs are a deep learning neural networks class that is primarily designed for manipulating structured grid-like data, such as videos and videos. CNNs have proven to be effective in several computer vision operations, including image classification, segmentation, and object detection [22]. CNNs are inspired by the visual cortex organization in animals, which is known for its hierarchical and feature-extraction abilities. CNNs are a widely adopted approach that leverages neural network architectures comprising multiple hidden layers. CNNs demonstrated exceptional performance in the image net large scale visual recognition competition (ILSVRC) and are acknowledged for their effectiveness in capturing spatial information from 2D images and videos. Each layer within a CNN serves to semantically extract patterns and acquire higher-level representations of image elements [23]. Moreover, each layer passes its learned weights to the subsequent layer, progressively developing invariant features. For instance, initial layers specialize in detecting basic image attributes like edges and colors, while higher layers focus on patterns that are more intricate. Any CNN model is composed of three primary types of layers: convolutional layers, pooling layers, and fully connected layers. The convolutional layer, positioned at the forefront, serves as the fundamental building block of CNNs and can be stacked with additional convolutional or pooling layers. The final layer, the fully connected layer, carries out the classification operation based on the features that are extracted from previous layers and their filters. Meanwhile, the pooling layer conducts down sampling operations to reduce input parameters and dimensionality. As the input image traverses through each layer, the CNN progressively identifies larger segments until it ultimately recognizes the entire image [24].

Within convolutional layers, the central operation that lends its name to the network is known as convolution. Convolution is a linear process that includes the multiplication of an input data array by a 2-D array of weights referred to as a filter or kernel. This operation results in the creation of a feature map that summarizes the detected features within the input. The filter is typically smaller than the input data and systematically traverses the input, moving in both horizontal and vertical directions. As it moves, the filter computes dot products between its entries and the corresponding entries in the input [25].

The concept of applying the same filter across the entire input data is a powerful one, as it enables the filter to identify the desired features anywhere within the input. This property is often referred to as translation invariance, which means that the network is concerned with whether a feature exists rather than precisely where it exists in the input. However, certain hyperparameters, like the number of filters, strides, and padding, influence the output size from the convolution operation. The number of filters determines the depth of the output [26]. For example, using three filters would result in three depth feature maps. Stride dictates the distance or the number of elements the filter moves over the input data. Typically, stride values of two or less are preferred, while larger stride values are rarely used and result in smaller output [27]. Padding is employed to maintain the spatial dimensions of the input data following convolution. In cases where the filters do not perfectly fit the input, padding sets all elements outside the input to zero, ensuring that the output maintains the same dimensions [28]. Following each convolution operation, CNNs apply a nonlinear activation operation known as the rectified linear unit (ReLU) to compute the feature maps. The ReLU function replaces all negative values in the feature maps with zeros while leaving positive values unchanged. One primary benefit of employing the ReLU function in contrast to alternative activation functions lies in its ability to avoid simultaneous activation of all neurons. As a result, ReLU facilitates faster training, often outpacing its counterparts with different activation units [29].

A pooling layer functions as a down sampling layer positioned between two consecutive convolutional layers in a CNN. Its primary role is to conduct spatial dimensionality reduction, thereby decreasing the size of feature maps. This reduction in size serves to lower the number of parameters and computations, enhancing overall efficiency. By providing an input for the subsequent layer, the pooling layer enables it to focus on broader areas within the input representation [30]. Similar to convolutional layers, pooling layers entail the movement of a window of specified dimensions across the feature maps. During this process, subsampling functions are applied, resulting in the generation of medium-level features. Although some information may be lost in the pooling layer due to this subsampling, it helps mitigate the risk of overfitting and simplifies the network's complexity. Commonly used pooling functions include max pooling and average pooling, with max pooling being the favored choice for its superior performance [31].

Fully connected layers in CNNs resemble traditional neural networks, where neurons establish complete connections with all neurons in the preceding layer. These layers typically consist of a series of perceptron layers, usually two or three layers deep. A fully connected layer takes the output from the preceding layer, which represents the learned features, and classifies it into the appropriate target class. The final fully connected layer possesses an output size equal to the number of class labels. The layer immediately preceding the fully connected layer may be a convolutional or pooling layer, with its output being flattened before being passed into the fully connected layer [32]. In most cases, fully connected layers employ activation functions such as SoftMax or sigmoid to properly classify their inputs. The SoftMax function yields values within the [0, 1] range and is preferred in multi classification tasks [33].

The CNN architecture represents the backbone structure for the proposed architecture. Figure 4 illustrates the proposed CNN architecture. The layers that build up the architecture can be outlined as follows: the architecture starts with an input layer, followed by two convolution layers. The convolution layers use 32 channels convolution filters. Padding is set to "same" to enable convolution across input images. With "same" padding, the filter window can extend beyond the input boundaries, ensuring that the filter operates on all input values. Each convolutional unit within the CNN functions as a detector capable of identifying the target within the image. For instance, when the target is positioned in the image upper left corner, the upper left corner of the feature map generated by the convolutional layer will exhibit a stronger response. Conversely, if the target is located in the lower right corner, the corresponding region in the feature map's lower right corner will display a more pronounced response. The output of the layers is forwarded to a pooling layer. The pooling layer performs pooling operation using max pooling function, resulting in a less-size input representation. The pooling window size is set to 2×2. The next layers are three convolution layers with 64 filters followed by another pooling layer. The result of the pooling layer is fed to next convolution layers that have 128 filters. All convolution layers utilize filter dimensions of 3×3, and a consistent stride value of 2. The output is flattened and passed to the fully connected layers, the flatten transforms the output features map into a one-dimensional characteristic vector, therefore spatial information of the image gets lost. The fully connected layers responsible for classification tasks. The output of the fully connected layer could be one of 20 nodes, where each node forms an aircraft type.

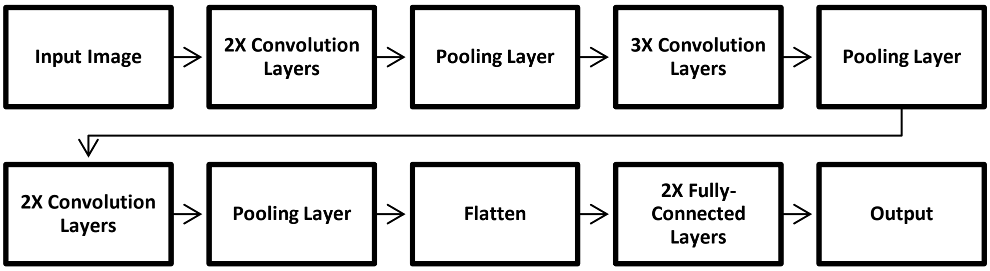


Figure 4. The proposed CNN architecture

4. RESULT AND DISCUSSION

Our experiments are implemented on Kaggle with GPU runtime using 13 GB RAM and 73 GB hard disk, using Python programming language and Keras deep learning library with TensorFlow. As the aircraft images have different sizes, the dataset is first preprocessed by resizing the image to 224×224 pixels. The images are also normalized to be in the range 0 to 1 using a min-max scaler by dividing each pixel value by 255. MTARSI dataset mainly split into two sets: 80% training and 20% testing sets. The training set is further split into 90% training and 10% validation set. The architecture is trained using 40 epochs. Further, a categorical cross-entropy loss function is used with Adamax optimizer with initial learning rate 0.001. Categorical cross-entropy loss function is commonly used in multi-class classification tasks, and it calculates a value for each data point, quantifying how well the predicted probabilities match the true labels [34]. In the training process, the goal is to minimize the categorical cross-entropy loss, which effectively trains the model to produce predicted probabilities that are as close as possible to the true class distributions. This leads to accurate multiclass classification. Adam is effective in training deep neural networks because it dynamically adapts learning rates for each parameter, providing faster convergence and better generalization, however, Adamax simplifies the second-moment calculation, which can lead to memory savings and faster training [35]. For activation function in convolutional layers, ReLU function is used. Moreover, to normalize the activations, Batch normalization is applied after the activation function in each layer, which improves training stability and speeds up convergence. To get the result, the output layer leverages SoftMax function. We utilize early stopping as a regularization technique, which is used to reduce overfitting during model training. It monitors the performance of a model on a validation dataset during the training process and stops the training when the performance of the model on the validation dataset starts to reduce, even if the training loss continues to decrease.

The performance is evaluated by the accuracy metric, which is a fundamental metric to evaluate the performance of classification models in deep learning. Accuracy measures the percentage of correctly predicted images out of the total number of images in the dataset. Figure 5 shows the results of evaluation on training and validation sets.

The proposed model in this study tends to have high detection accuracy as the training accuracy reaches 95% in first 5 epochs, and then it increases continuously. The maximum attained accuracy is 100%. For validation, the maximum attained accuracy is 93.3% at epoch 22, which is the best epoch. We can notice that the training does not continue for 40 epochs, as we use early stopping, which stops the training when the validation accuracy starts to show no progress. This study also uses other metrics such as recall and precision. Recall measures how many of the actual positive images in the dataset were correctly predicted. In multiclass classification, recall is calculated for each class independently, treating each class as the positive class in turn, while the rest are treated as negative classes. However, precision measures how many of the positively predicted images are actually correct. Table 1 shows the overall accuracy, recall and precision for training, validation and testing sets.

To validate our architecture performance on the MTARSI dataset, we conducted comparisons with state of the arts models. The outcomes of these comparisons can be found in Figure 6. The figure shows that the proposed custom CNN model is outperforming state of the arts models in terms of overall accuracy.

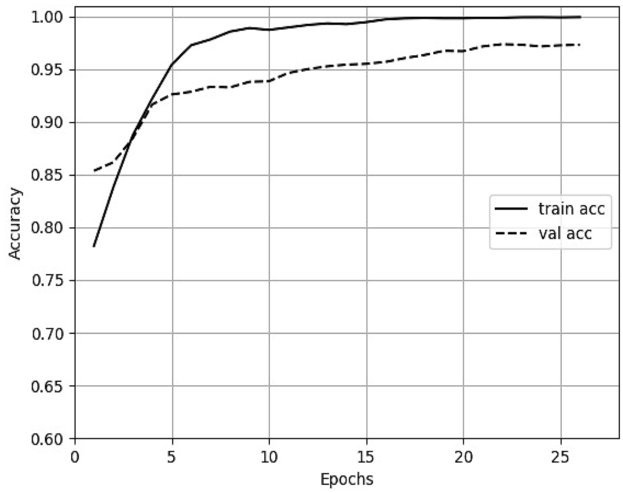


Figure 5. Training and validation accuracy

Table 1. The experiments result

Set	Accuracy	Recall	Precision
Training	97.07%	97.59%	98.12%
Validation	94.81%	94.92%	96.50%
Testing	94.44%	95.20%	96.89%

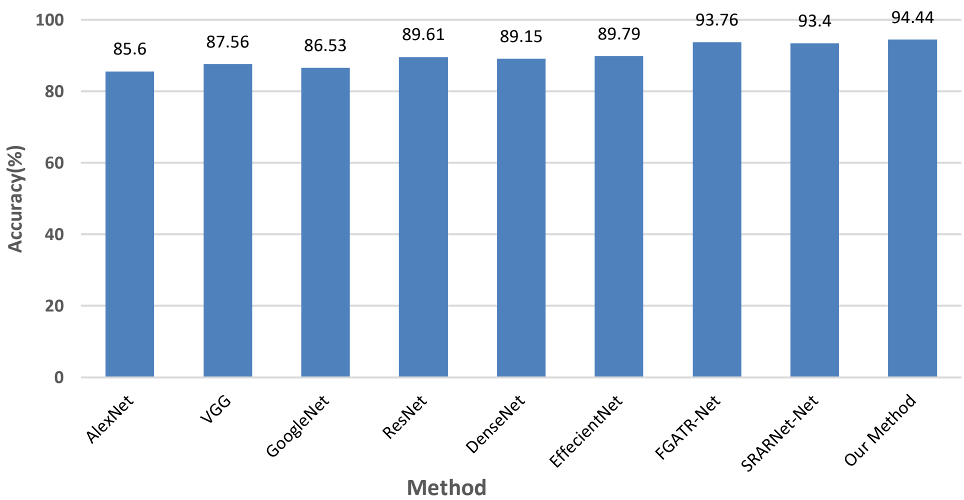


Figure 6. Comparison with other methods

This study explored the detection of aircraft types from remote sensing images using deep learning architecture, by designing a custom CNN. However, further and in-depth studies may be needed to confirm its performance on other remote sensing images dataset. This verification process is particularly crucial concerning the parameters' size and detection time, warranting comprehensive scrutiny for robustness and generalization.

5. CONCLUSION

The identification of aircraft types from remote sensing images has attracted significant research interests. CNNs are commonly used for this task due to their ability to learn hierarchical representations from visual data. This paper presents a CNN based method for recognizing aircraft type using remote sensing images. Experimental results on MTARSI dataset show that the proposed method achieves a high accuracy rate with 94.44% on the testing set, which demonstrates that our method is better than the state-of-the-art methods. Future works will focus on improving the method performance by leveraging information from diverse sources. We will also concentrate on designing a CNN network using transfer learning to reduce the computational complexity.

REFERENCES

[1] S. Ghosh, J. Patrikar, B. Moon, M. M. Hamidi, and S. Scherer, "AirTrack: onboard deep learning framework for long-range aircraft detection and tracking," in *Proceedings - IEEE International Conference on Robotics and Automation*, May 2023, vol. 2023-May, pp. 1277–1283, doi: 10.1109/ICRA48891.2023.10160627.

[2] L. Zhou *et al.*, "Aircraft detection for remote sensing images based on deep convolutional neural networks," *Journal of Electrical and Computer Engineering*, vol. 2021, pp. 1–16, Aug. 2021, doi: 10.1155/2021/4685644.

[3] Y.-C. Lin and W.-D. Chen, "Automatic aircraft detection in very-high-resolution satellite imagery using a YOLOv3-based process," *Journal of Applied Remote Sensing*, vol. 15, no. 01, Jan. 2021, doi: 10.1117/1.jrs.15.018502.

[4] D. Grosgeorge, M. Arbelot, A. Goupilleau, T. Ceillier, and R. Allioux, "Concurrent segmentation and object detection CNNs for aircraft detection and identification in satellite images," *International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 276–279, May 2020, doi: 10.1109/IGARSS39084.2020.9323338.

[5] T. Santha and M. M. B. V, "The significance of real-time, biomedical and satellite image processing in understanding the objects and application to computer vision," in *Proceedings of 2nd IEEE International Conference on Engineering and Technology, ICETECH 2016*, Mar. 2016, pp. 661–670, doi: 10.1109/ICETECH.2016.7569333.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8691 LNCS, no. PART 3, pp. 346–361, Jun. 2014, doi: 10.1007/978-3-319-10578-9_23.

[8] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Dec. 2015, vol. 2015 Inter, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[10] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Oct. 2017, vol. 2017-Octob, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.

[11] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of YOLO algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2021, doi: 10.1016/j.procs.2022.01.135.

[12] L. Cheng, Y. Ji, C. Li, X. Liu, and G. Fang, "Improved SSD network for fast concealed object detection and recognition in passive terahertz security images," *Scientific Reports*, vol. 12, no. 1, Jul. 2022, doi: 10.1038/s41598-022-16208-0.

[13] Q. Wang, H. Zhang, X. Hong, and Q. Zhou, "Small object detection based on modified FSSD and model compression," *2021 6th International Conference on Signal and Image Processing, ICSIP 2021*, pp. 88–92, Aug. 2021, doi: 10.1109/ICSIP52628.2021.9688896.

[14] D. Wang, X. Li, and M. Hao, "Aircraft target detection in remote sensing images based on improved faster R-CNN," in *Proceedings of 2023 IEEE 5th International Conference on Civil Aviation Safety and Information Technology, ICCASIT 2023*, Oct. 2023, pp. 302–306, doi: 10.1109/ICCASIT58768.2023.10351602.

[15] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[16] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Oct. 2017, vol. 2017-Octob, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[17] H. Yan, "Aircraft detection in remote sensing images using centre-based proposal regions and invariant features," *Remote Sensing Letters*, vol. 11, no. 8, pp. 787–796, Jun. 2020, doi: 10.1080/2150704X.2020.1770364.

[18] T. Wang *et al.*, "An aircraft object detection algorithm based on small samples in optical remote sensing image," *Applied Sciences (Switzerland)*, vol. 10, no. 17, p. 5778, Aug. 2020, doi: 10.3390/APP10175778.

[19] Z. Z. Wu, T. Weise, Y. Wang, and Y. Wang, "Convolutional neural network based weakly supervised learning for aircraft detection from remote sensing image," *IEEE Access*, vol. 8, pp. 158097–158106, 2020, doi: 10.1109/ACCESS.2020.3019956.

[20] F. Ucar, B. Dandil, and F. Ata, "Aircraft detection system based on regions with convolutional neural networks," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 8, no. 3, pp. 147–153, Sep. 2020, doi: 10.18201/ijisae.2020363534.

[21] Z. Z. Wu *et al.*, "A benchmark data set for aircraft type recognition from remote sensing images," *Applied Soft Computing Journal*, vol. 89, p. 106132, Apr. 2020, doi: 10.1016/j.asoc.2020.106132.
[22] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.
[23] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.
[24] M. Krichen, "Convolutional neural networks: a survey," *Computers*, vol. 12, no. 8, 2023, doi: 10.3390/computers12080151.
[25] Z. Han, M. Jian, and G.-G. Wang, "ConvUNeXt: an efficient convolution neural network for medical image segmentation," *Knowledge-Based Systems*, vol. 253, Oct. 2022, doi: 10.1016/j.knosys.2022.109512.
[26] D. Setyawan, R. Wardoyo, M. E. Wibowo, E. E. Herdiana Murhandarwati, and J. Jamilah, "Malaria classification using convolutional neural network: a review," *2021 6th International Conference on Informatics and Computing, ICIC 2021*, Nov. 2021, doi: 10.1109/ICIC54025.2021.9632998.
[27] R. Riad, O. Teboul, D. Grangier, and N. Zeghidour, "Learning strides in convolutional neural networks," *arXiv preprint arXiv:2202.01653.*, Feb. 2022.
[28] K. Rimal, K. B. Shah, and A. K. Jha, "Advanced multi-class deep learning convolution neural network approach for insect pest classification using TensorFlow," *International Journal of Environmental Science and Technology*, vol. 20, no. 4, pp. 4003–4016, Jun. 2023, doi: 10.1007/s13762-022-04277-7.
[29] L. Parisi, D. Neagu, R. Ma, and F. Campean, "Quantum ReLU activation for convolutional neural networks to improve diagnosis of Parkinson's disease and COVID-19," *Expert Systems with Applications*, vol. 187, Jan. 2022, doi: 10.1016/j.eswa.2021.115892.
[30] A. Zafar *et al.*, "A comparison of pooling methods for convolutional neural networks," *Applied Sciences (Switzerland)*, vol. 12, no. 17, Aug. 2022, doi: 10.3390/app12178643.
[31] R. Nirthika, S. Manivannan, A. Ramanan, and R. Wang, "Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study," *Neural Computing and Applications*, vol. 34, no. 7, pp. 5321–5347, 2022, doi: 10.1007/s00521-022-06953-8.
[32] V. Pomazan, I. Tvoroshenko, and V. Gorokhovatskyi, "Development of an application for recognizing emotions using convolutional neural networks," *International Journal of Academic Information Systems Research*, vol. 7, pp. 25–36, 2023.
[33] G. Xia and C. S. Bouganis, "Augmenting SoftMax information for selective classification with out-of-distribution data," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13846 LNCS, Springer Nature Switzerland, 2023, pp. 664–680.
[34] U. Ruby and V. Yendapalli, "Binary cross entropy with deep learning technique for image classification," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 5393–5397, Aug. 2020, doi: 10.30534/ijatcse/2020/175942020.
[35] K. S. Kim and Y. S. Choi, "Hyadamc: a new Adam-based hybrid optimization algorithm for convolution neural networks," *Sensors*, vol. 21, no. 12, Jun. 2021, doi: 10.3390/s21124054.

BIOGRAPHIES OF AUTHORS

Wael Toghuj received the B.Sc. and M.Sc. degrees in computer engineering from Kabardino-Balkarian State University, Nalchik, Russian Federation, in 1997. He received Ph.D. in computer science from Stavropol State University (North-Caucasus Federal University), Stavropol, Russian Federation, in 2009. Currently, he is an associate professor at the Department of Computer Science, Al-Ahliyya Amman University, Jordan. His research interests include data reliability and security, fault-tolerant systems, multi-errors correcting codes, human-computer interaction, simulations and multimedia systems. He can be contacted by this email: w.toghuj@ammanu.edu.jo.

Yousef Alraba'nah received his B.Sc. degree in software engineering from Zarqa university (Jordan) in June 2012. In February 2013, Yousef obtained fully-funded scholarship from Zarqa university to complete M.Sc. in computer science. He graduated in June 2015 with excellent degree. He currently works as a lecturer in the faculty of information technology at Al-Ahliyya Amman University, Jordan. His main research interests include distributed systems, networks security, and machine learning. He can be contacted at email: yrabanah@gmail.com.