IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 2, April 2025, pp. 1540~1547
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i2.pp1540-1547

Improved convolutional neural networks for aircraft type classification in remote sensing images

Yousef Alraba'nah[1], Mohammad Hiari[2]
[1]Department of Software Engineering, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[2]Department of Networks and Cybersecurity, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan

Article Info

Article history:

Received Sep 13, 2024
Revised Oct 20, 2024
Accepted Oct 23, 2024

Keywords:

Aircraft
Classification
Convolutional neural networks
Deep learning
Remote sensing images

ABSTRACT

With the exponential growth of available data and computational power, deep convolutional neural networks (CNNs) have become as powerful tools for a wide range of applications, ranging from image classification to natural language processing. However, during last decade, remote sensing imagery has emerged as one of the most prominent areas in image processing. Variations in image resolution, size, aircraft types and complex backgrounds in remote sensing images challenge the aircraft classification task. This study proposes an effective aircraft classification model based on CNN architecture. The CNN network architecture is improved to achieve more accuracy rate and to avoid overfitting and underfitting problems. To validate the proposed model, a new public aircraft dataset called multi-type aircraft remote sensing images 2 (MTARSI2) has been used. Through an analysis of existing methodologies and experimental validation, the model shows the superior performance of the proposed CNN model in comparison to state-of-the-art deep learning approaches.

Corresponding Author:

Yousef Alraba'nah
Department of Software Engineering, Faculty of Information Technology, Al-Ahliyya Amman University
Amman, Jordan
Email: y.alrabanah@ammanu.edu.jo

1. INTRODUCTION

The satellite image processing is now concentrated on the detection and identification of objects, which are also considered major topics. The development of new satellites equipped with cameras that can take high-resolution photographs of specific parts of Earth has been a motivation for further studies in this area [1]. Research on satellite images includes areas such as the definition of storm levels, help for weather forecasts, determining crops, and analysis of poverty levels in residential areas. The use of remote sensing images can be highly important, especially for military purposes or in the defense sector. In key defense system applications such as unmanned aerial vehicles, results obtained from the target detection process assist decision support systems that are based on high-level information extracted from the images [2]. The difficulty with finding objects in remote sensing images is that they depend on the object's surrounding environment. A major contrast is observed between the detected target and the background itself as the airport serves as the background, thereby leading to a discrepancy in magnitude between the two areas and the detection target. In addition, it becomes even more challenging if we want to locate an object of a smaller scale [3].

Traditional object detection methods exhibit drawbacks such as limited generalization capability and insufficient rotation invariance. These methods focus on extracting low-level features from remote sensing images and often fail to fully leverage high-level features. Additionally, the subjective nature of feature selection and extraction makes the process complex. The advent of big data and increased computing capacity has led to the rapid development and widespread adoption of deep learning methods across various domains [4]. Currently, deep learning demonstrates more achievements in machine learning and computer vision research fields, particularly in surveillance tasks, classification, biometrics, satellite imageries and medical imaging [5]. Object detection, a fundamental challenge in computer vision, has been a key focus of deep learning. The convolutional neural network (CNN), recognized for its effectiveness in feature extraction, has garnered significant attention in image recognition and detection. CNNs demonstrate notable performance in tasks such as object identification or detection, image generation, semantic segmentation, and high-resolution image reconstruction [6].

As hardware efficiency continues to expand and advance, deep neural networks have transformed the field of analysing remote sensing satellite images. Image recognition involves the process of identifying and acknowledging elements within digital images or videos [7]. The recognition of objects in digital images typically commences with preprocessing procedures such as image enhancement and noise removal. These procedures are followed by feature extraction to identify segments, lines, and potential areas with distinct characteristics [2]. Apart from the complex structure, variations in aircraft can occur in shapes, colors, dimensions, or patterns, even within specific parts of the aircraft. Intensity and texture often vary in different conditions. Additionally, recognition is frequently challenged by various instabilities, including altered contrasts, clutter, and inconsistencies related to anxiety [8].

With the rapid advances in deep learning, numerous scholars are now turning to deep learning algorithms for target identification in remote sensing image analysis. In contrast to traditional methods, deep learning offers superior detection accuracy and faster processing, particularly with end-to-end detection techniques [9]. Over the past decade, the CNN system has seen continuous enhancement and enrichment through the introduction of classical models and structures by scholars, including but not limited LeNet-5 [10], AlexNet [11], VGG-16 [12], ResNet-50 [13], region-based convolutional neural network (R-CNN) [14], Fast R-CNN [15], Faster R-CNN [16], Mask R-CNN [17], OverFeat [18], and YOLO [19]. A detailed review of different CNN architectures and models can be found in [20].

Zhao *et al.* [21] found that using YOLOv3 for detecting small objects such as aircraft in remotely sensed imagery has given remarkable detection results with low processing overhead. The study determined that YOLOv3 outperformed Faster R-CNN and single shot multibox detector (SSD) in terms of average precision and detection time. Utilizing Google Earth images and the DOTA dataset, the study used 224 images for training and 70 images for testing, with resolutions ranging from 600×600 to 1500×1500 pixels.

A VGGNet-16 was used in recognition of boats and planes in remote sensing images by combining of deep features gained from CNN, the combined features are then used to train a support vector machine. The authors proposed the using of fisher vector pooling strategy that enhanced the recognition performance on the training set. The evaluation was conducted on a dataset of size 1,000 images which composed of 10 aircraft types with 100 image images in each type [22].

Li *et al.* [23] introduced a novel aircraft detection framework that integrates a CNN model with reinforcement learning. This approach effectively identifies the precise positions of aircraft in remote sensing images. The experiments involved a dataset of 3000 images obtained from Google Earth via QuickBird. However, their method exhibited a drawback in terms of processing time compared to state-of-the-art techniques.

Wu *et al.* [24] conducted simulations using multi-type aircraft remote sensing images (MTARSI) dataset to evaluate recognition of aircraft with various state-of-the-art models. They achieved highest accuracy of 89.7% using EfficientNet deep learning model. By employing composite scaling to balance system depth, breadth, and resolution, the authors obtained the second-highest accuracy of 89.6% with ResNet. Additionally, they applied other models such as GoogleNet, DenseNet, VGG, AlexNet, living learning community (LLC), and ScSPM, yielding accuracies of 86.5%, 89.1%, 87.5%, 85.6%, 64.9%, and 60.6%, respectively. The study also combined (HOG + SVM) and (SIFT + BOVW) which achieved low accuracy with 61.3% and 59.0%.

Wang *et al.* [25] introduced enhanced significance pre-detection technique aimed to achieving multi-scale fast and coarse locating aircraft objects in synthetic aperture radar (SAR) images. Subsequently, the authors used CNN for precise detection of these object targets, resulting in good detection accuracy with extended testing time. Without employing data improvement techniques, the algorithm achieved detection rate of 86.33%.

Zhao *et al.* [26] proposed a fast detection technique for identifying aircraft targets in SAR images in intricate environments and big scenes. Their technique enhances the entire detection process, refining the extraction of airport regions using grayscale features and employing coarse detection of aircraft targets via CNN. The technique achieved a detection rate of 74.0% with a false alarm rate of 6.9%.

Han *et al.* [27] proposed a simple and efficient attention network (SEAN) which is intended to detect aircraft in SAR images. The network sidesteps the need for a complicated deep spine and parallel associated feature pyramid network (FPN) neck; instead, increasing both accuracy of detection and speed significantly while reducing network parameters and floating-point operations per second (FLOPs). The performance of SEAN has been tested with the Gaofen-3 SAR target dataset, and it shows an outstanding average precision of 97.7%. Its real-time speed is 83.3 frames per second. The experiments indicate that

SEAN has two advantages: detection accuracy and speed for finding SAR aircraft targets in complicated contexts more effectively than typical algorithms used for identifying targets.

The MTARSI2 dataset, a recently introduced extensive data set, was thoroughly examined [28]. The researchers conducted a comprehensive analysis to evaluate the dataset's attributes, strengths, weaknesses, and its performance when compared to well-known algorithms. The authors assessed various algorithms, including EfficientNetB4, Inception ResNetV2, InceptionV3, MobileNetV2, and ResNet50. Their findings indicated that ResNet50 achieved the highest accuracy rate at 90%, while InceptionResNetV2 exhibited the lowest accuracy at 78%. In this context, this research aims to explore the using of CNN architecture in classification of aircrafts. Particularly, the study seeks to examine how to improve the CNN to achieve a high accuracy rate in aircrafts classification using one of large, new dataset called MTARSI2.

This paper is structured as follows. In the second section, we describe the methodology used to design and build the proposed architecture. In the third section, the results of the study are presented, followed by a detailed discussion of the findings. Finally, the paper concludes by summarizing the key insights and offering recommendations for future research.

2. METHOD

CNNs are a type of advanced neural network which has gained widespread attention for their remarkable effectiveness in various computer vision applications, specifically for processing grid-like data. These applications include classification and segmentation of images, and object detection [29]. Drawing inspiration from the intricate organization of the visual cortex in animals, CNNs surpass in extracting hierarchical features. This design choice, along with their utilization of neural network structures with several hidden layers, has boosted CNNs to exceptional results in competitions. The ability to efficiently extract information from images and videos is highly regarded in these models. Every layer in a CNN plays a role in extracting meaningful patterns and obtaining higher-level image elements representations [30]. It is worth noting that each layer builds upon the knowledge it has acquired, continuously improving its ability to recognize invariant features. The initial layers are dedicated to identifying simple image characteristics like colors and edges, the later layers concern with detecting more complex patterns. A typical CNN model integrates three primary layers: convolutional, pooling, and fully connected layers. The former layer, set as the cornerstone, performs as the foundational unit and could be seamlessly combined with more pooling or convolutional layers. The latter layer, executes critical classification tasks depending on features collected from preceding layers and their filters. Simultaneously, the pooling layer executes downsampling processes to streamline input parameters and reduce the overall dimensionality of the data. As the input image moves through the network's layers, the CNN progressively recognizes larger parts. This hierarchical feature learning allows the network to build up from simple elements to complex, holistic understanding of image content [31].

Convolutional layers are named due to their core operation, convolution. This linear process involves multiplying an array of input data by a two-dimensional weight array known as a kernel or filter. The result is a feature map which is a condensed representation of the detected features in the input. This process involves a filter, typically smaller than the input, moves across the entire input horizontally and vertically. As the filter traverses the input, it captures and summarizes local patterns, creating a comprehensive feature map that highlights important characteristics of the original image [32].

The application of a filter on the whole input is an important concept. It allows the filter to detect significant features regardless of their location in the input. This characteristic, often called translation invariance, means the network focuses on a feature's presence rather than its exact position. However, the convolution process's output size is influenced by specific hyperparameters like number of filters which determines the output's depth [33]. For instance, utilizing three filters produces feature maps with a depth three. Stride defines how far the filter moves over the input data in each step. Smaller strides (two or lower) are typically better, while more values result in smaller outputs [34]. Padding is a technique maintains the input data's spatial dimensions after convolution. When filters don't fit the input perfectly, padding adds zeros around the input's edges to ensure consistent output dimensions. These parameters allow fine-tuning of the convolutional layer's behavior and output characteristics [35].

After each convolution process, CNNs perform rectified linear unit (ReLU) function, which is a nonlinear activation function to find the feature maps. The ReLU function operates by replacing all negative values to zero in the feature maps, while positive values are remained as are. A key advantage of ReLU, compared to other functions, is its capability to prevent all neurons from activating simultaneously. This characteristic allows for faster training and often leads to better performance than networks using alternative activation functions [36]. The ReLU function is mathematically expressed as (1):

$$\mathrm{ReLU}(x) = \max(0, x) \tag{1}$$

In a CNN, a pooling layer acts as a downsampling intermediary between consecutive convolutional layers. Its main purpose is to reduce the spatial dimensions of feature maps, which leads to fewer parameters and computations, improving overall efficiency and enabling the next layer to concentrate on wider areas of the input representation [37]. The pooling process involves moving a window of fixed size across the feature maps and performing subsampling functions to create medium-level features. While this subsampling may result in some information loss, it offers key benefits in preventing overfitting and simplifying the network's complexity. Two common pooling methods are used, max pooling that selects the highest value within the window, and average pooling which finds the average of all window values. Max pooling is generally preferred due to its excellent performance [38].

In CNNs, fully connected layers are similar to classical neural networks. In these layers, each neuron is connected to every neuron in the former layer. Typically, there are two or three such layers in a CNN. Their task is to take the results from the previous layer which forms learned features and classify this output into the desired target class. The last layer has an output size that matches the number of labels in the classification task. Before entering the fully connected layer, the output from the previous layer which could be a convolutional or pooling layer is flattened into a one-dimensional array [39]. These layers often use activation functions like softmax or sigmoid for accurate classification. In multi-class classification tasks, the softmax function is useful as it produces values between 0 and 1, which can be interpreted as probabilities [40]. The softmax function is defined as (2):

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{2}$$

The core structure of the proposed architecture is built upon the CNN design, as depicted in Figure 1. The architectural layers are introduced as following: It begins with an input layer, succeeded by two convolution layers each using 32-channel filters. The convolution is performed with same padding, allowing the filters to extend beyond the image boundaries and process whole input values. Every convolutional unit in the CNN acts as a detector, recognizing features within the image. For example, if a feature is in the top left corner, the feature map's top left corner will show a strong response, while a feature in the bottom right corner will result in a stronger response in that part of the feature map. The output from these layers is fed into a max pooling layer with a 2×2 window size, which reduces the size of the representation. This is followed by three convolutional layers, each with 64 filters, and one pooling layer. The output from this pooling layer is then processed by additional convolutional layers with 128 filters. All convolutional layer use 3×3 filters and a stride of 2. The output from the previous layers is flattened into a single-dimensional vector, and then fed into fully connected layers, which perform the final classification task. The last layer of this network has 40 nodes, each corresponding to a different type of aircraft.

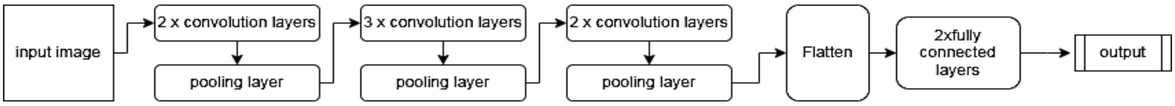

Figure 1. The proposed architecture

3. RESULTS AND DISCUSSION

3.1. MTARSI2 dataset

The proposed work is evaluated using the MTARSI2 dataset, which is an extended version of MTARSI dataset. MTARSI consists of 9,385 remote sensing images generated from Google Earth satellite imagery. This dataset underwent meticulous labelling by experts specializing in interpreting remote sensing imagery. However, the original MTARSI dataset [24] underwent modifications to include 42 classifications, along with additional data augmentation within these categories, resulting in the creation of MTARSI2 [41]. Within the dataset, each image exclusively features a single, complete aircraft, totalling 40 different aircraft types. It is worth to note that, MTARSI2 includes two image groups that did not belong to any aircraft type. The dataset exhibits variations in aircraft images, including differences in color, poses, viewpoints, backgrounds, and resolutions [28]. To augment the dataset, the researchers performed processes such as airplanes segmentation, performing rotations and flips, and changing backgrounds, Figure 2 shows a sample of each aircraft type. The overall number of images in MTARSI2 dataset is 10483, where the number of these images varies per category, ranging from 28 to 759, with specific class distributions outlined in Figure 3. MTARSI2 dataset serves as an illustration of an unbalanced dataset, presenting challenges associated with varying light conditions and viewing angles.

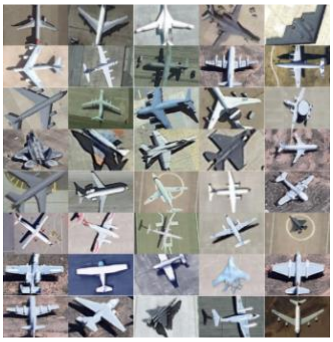

Figure 2. Samples of MTARSI2 dataset

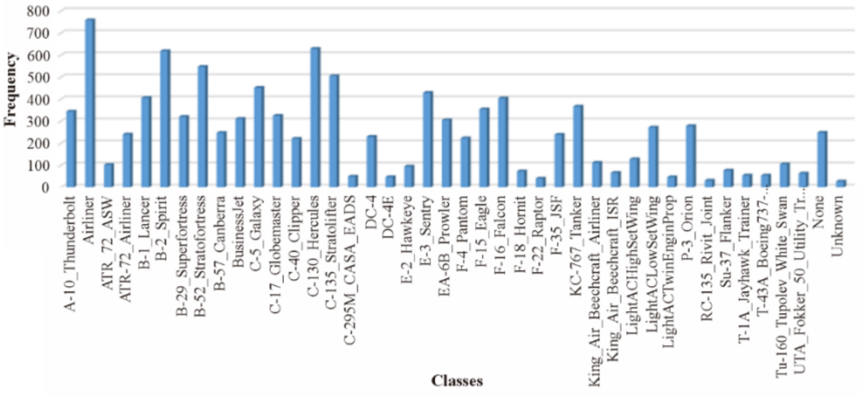

Figure 3. MTARSI2 class distribution

3.2. Experimental evaluation and discussion

The experiments are conducted on Kaggle of GPU runtime, utilizing a RAM with 13 GB and 73 GB hard disk. The implementation is carried out in Python, employing the Keras library with TensorFlow. Given the varied sizes of aircraft images, the dataset undergoes preprocessing, specifically resizing images to 224×224 pixels. Additionally, image normalization is applied, ensuring pixel values fall within the range of 0 to 1 through min-max scaling which divides every pixel value by 255. The MTARSI2 dataset is primarily partitioned into a 70% training set and a 30% testing set. Moreover, division allocates 90% to the training set and 10% to the validation set. The architecture undergoes training for 50 epochs. A categorical crossentropy loss function is employed, coupled with an Adamax optimizer featuring an initial learning rate of 0.001. We selected an initial learning rate of 0.001 for the following reasons. First, it is a widely accepted starting point in the literature for training CNNs, especially when using the Adam optimizer, which adjusts the learning rate dynamically during training. This value strikes a balance between fast convergence and stability, avoiding the risk of divergence with higher rates while ensuring sufficient progress compared to lower rates. Additionally, our empirical experiments, including testing rates of 0.01, 0.001, and 0.0001, showed that 0.001 provided the best trade-off between training speed and final model accuracy. Categorical crossentropy is a standard choice for multi-class classification, evaluating how well predicted probabilities align with true labels for each data point. The training objective is to minimize this loss, ensuring the model produces predicted probabilities closely matching the true class distributions. The Adam optimizer is chosen due to its efficiency in training deep neural networks. Its key advantage is the automatic adjustment of learning rates for individual parameters, leading to quicker convergence and better generalization. Adamax, a variant of Adam, facilitates the calculation of second-moment, which helps in reducing memory consumption and speeding up training. ReLU function acts as the activation function in convolutional layers, while Batch normalization follows each layer's activation function to normalize activations, enhancing training settlement and speeding up convergence. The CNN's final layer uses the softmax function to generate results. To prevent overfitting, the model employs early stopping technique, it tracks the model's performance on a separate validation dataset during the model training, and interrupts the training process if the model's validation performance seems to decline, even if the training loss is still decreasing. This approach helps ensure the model generalizes well to new data rather than memorizing the training set.

Performance evaluation is based on the accuracy metric, a key measure for assessing the effectiveness of deep learning models classification. This metric calculates the rate of correctly classified images out of overall number of images in the dataset. Figure 4 shows the evaluation results for the training set and validation set. The training accuracy achieves 96% within the initial 5 epochs and steadily increases thereafter, ultimately reaching a maximum accuracy of 100%. Regarding validation, the highest accuracy, which is 97.9%, is achieved at epoch 19, marking the optimal epoch. Notably, the training does not extend to 50 epochs due to the implementation of early stopping. This mechanism halts the process of training once the accuracy of validation stops to exhibit improvement. The study incorporates additional measurements such as precision and recall. Precision evaluates the accuracy of images predicted as positive. Conversely, recall measures how accurately the model identifies actual positive images in the dataset. For multiclass classification, recall is calculated separately for every class by considering the class as positive while considering all other classes as negative. Table 1 provides an overview of accuracy, recall, and precision results for training, validation, and testing sets.

To evaluate the performance of the proposed model on the MTARSI2 dataset, the study conducts thorough comparisons with state-of-the-art models that used MTARSI2 dataset in [28]. The results of these comparisons are illustrated in Figure 5, indicating that our proposed model surpasses in terms of overall accuracy state-of-the-art models. Specifically, our model achieves an accuracy of 93.21%, surpassing ResNet50 (90%), MobileNetV2 (88%), InceptionV3 (79%), Inception ResNetV2 (78%), and EfficientNetB4 (86%). Which we attribute to the enhanced feature extraction capabilities in the early convolutional layers. The proposed architecture leverages a combination of depth-wise separable convolutions and batch normalization to optimize for computational efficiency while preserving high classification accuracy, making it particularly suitable for real-time aircraft detection in resource-constrained environments. However, MTARSI2 dataset poses several challenges such as significant variations in size, orientation, and lighting, which may serve as a benchmark for future research in this domain. Our CNN architecture uses preprocessing techniques that are particularly effective in overcoming these challenges.

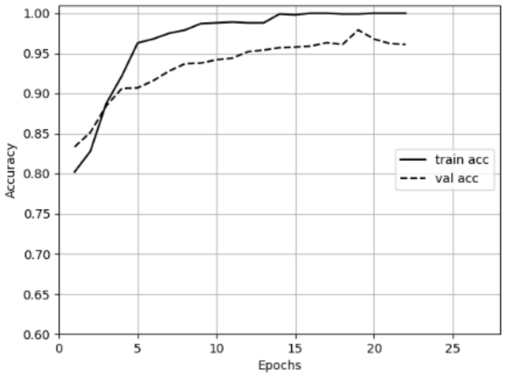

Figure 4. Training and validation accuracy

Table 1. Training, validation and testing results

Set	Accuracy (%)	Recall (%)	Precision (%)
Training set	96.68	97.47	97.67
Validation set	93.46	94.43	94.63
Testing set	93.21	93.70	94.10

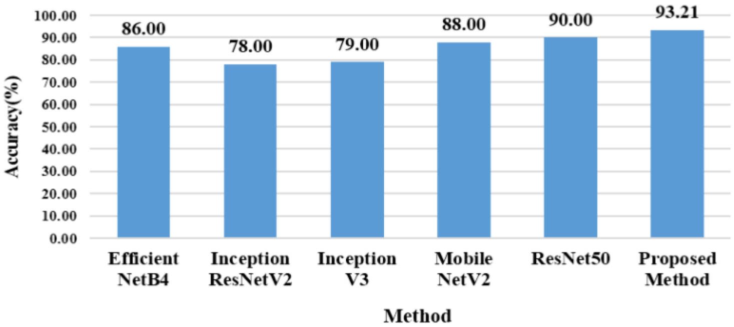

Figure 5. Comparison with state-of-the-art models

4. CONCLUSION

The application of CNNs for aircraft type detection demonstrates significant potential and accuracy. This study introduces a CNN-based approach for identifying aircraft types from remote sensing imagery. The proposed CNN architecture achieved an impressive accuracy of 93.21% on the testing set, surpassing other models and underscoring the effectiveness of tailored solutions in this domain. This high accuracy reflects the robustness of our approach in capturing and recognizing aircraft features. While the proposed architecture builds on existing CNN frameworks, its domain-specific optimizations, and high performance on the challenging task of aircraft classification demonstrate its practical value for real-world applications. Overall, CNNs have proven to be powerful tools for aircraft type detection, offering a reliable and efficient solution for this challenging task. Future work will explore incorporating multi-task learning to handle additional aircraft-related tasks, in addition to exploring the self supervised learning. The model will also be tested on other datasets to generalize it.

REFERENCES

[1] J. Rábago and M. Portuguez-Castro, “Use of drone photogrammetry as an innovative, competency-based architecture teaching process,” *Drones*, vol. 7, no. 3, p. 187, Mar. 2023, doi: 10.3390/drones7030187.

[2] W. Li, G. Wu, H. Sun, C. Bai, and W. Bao, “Dim and small target detection in unmanned aerial vehicle images,” in *Proceedings of 2022 International Conference on Autonomous Unmanned Systems (ICAUS 2022)*, 2023, pp. 3143–3152, doi: 10.1007/978-981-99-0479-2_289.

[3] W. Han *et al.*, “Methods for small, weak object detection in optical high-resolution remote sensing images: A survey of advances and challenges,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 4, pp. 8–34, 2021, doi: 10.1109/MGRS.2020.3041450.

[4] P. Shamsolmoali, M. Zareapoor, J. Chanussot, H. Zhou, and J. Yang, “Rotation equivariant feature image pyramid network for object detection in optical remote sensing imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022, doi: 10.1109/TGRS.2021.3112481.

[5] A. Goel, A. K. Goel, and A. Kumar, “The role of artificial neural network and machine learning in utilizing spatial information,” *Spatial Information Research*, vol. 31, no. 3, pp. 275–285, Jun. 2023, doi: 10.1007/s41324-022-00494-x.

[6] X. Yuan, J. Shi, and L. Gu, “A review of deep learning methods for semantic segmentation of remote sensing imagery,” *Expert Systems with Applications*, vol. 169, May 2021, doi: 10.1016/j.eswa.2020.114417.

[7] Z. Amiri, A. Heidari, N. J. Navimipour, M. Unal, and A. Mousavi, “Adventures in data analysis: a systematic review of deep learning techniques for pattern recognition in cyber-physical-social systems,” *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 22909–22973, Aug. 2023, doi: 10.1007/s11042-023-16382-x.

[8] Q. Liu, X. Xiang, Y. Wang, Z. Luo, and F. Fang, “Aircraft detection in remote sensing image based on corner clustering and deep learning,” *Engineering Applications of Artificial Intelligence*, vol. 87, Jan. 2020, doi: 10.1016/j.engappai.2019.103333.

[9] Y. Xiao *et al.*, “A review of object detection based on deep learning,” *Multimedia Tools and Applications*, vol. 79, no. 33–34, pp. 23729–23791, 2020, doi: 10.1007/s11042-020-08976-6.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012, vol. 2, pp. 1097–1105, doi: 10.1145/3065386.

[12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv-Computer Science*, vol. 1, pp. 1-14, Sep. 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[14] Y. Qian, H. Zheng, D. He, Z. Zhang, and Z. Zhang, “R-CNN object detection inference with deep learning accelerator,” in *2018 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, Aug. 2018, pp. 297–302, doi: 10.1109/ICCChinaW.2018.8674519.

[15] L. Sommer, N. Schmidt, A. Schumann, and J. Beyerer, “Search area reduction fast-RCNN for fast vehicle detection in large aerial imagery,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 3054–3058, doi: 10.1109/ICIP.2018.8451189.

[16] Y. Xiao, X. Wang, P. Zhang, F. Meng, and F. Shao, “Object detection based on faster R-CNN algorithm with skip pooling and fusion of contextual information,” *Sensors*, vol. 20, no. 19, Sep. 2020, doi: 10.3390/s20195490.

[17] S. Shivajirao, R. Hantach, S. B. Abbes, and P. Calvez, “Mask R-CNN end-to-end text detection and recognition,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Dec. 2019, pp. 1787–1793, doi: 10.1109/ICMLA.2019.00289.

[18] L. Liu *et al.*, “Deep learning for generic object detection: A survey,” *arXiv-Computer Science*, vol. 1, pp. 1–35, Aug. 2019.

[19] R. Menaka, N. Archana, R. Dhanagopal, and R. Ramesh, “Enhanced missing object detection system using YOLO,” in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Mar. 2020, pp. 1407–1411, doi: 10.1109/ICACCS48705.2020.9074278.

[20] K. Ahmed, M. A. Gad, and A. E. Aboutabl, “Performance evaluation of salient object detection techniques,” *Multimedia Tools and Applications*, vol. 81, no. 15, pp. 21741–21777, Jun. 2022, doi: 10.1007/s11042-022-12567-y.

[21] K. Zhao and X. Ren, “Small aircraft detection in remote sensing images based on YOLOv3,” *IOP Conference Series: Materials Science and Engineering*, vol. 533, no. 1, May 2019, doi: 10.1088/1757-899X/533/1/012056.

[22] B. Jiang, X. Li, L. Yin, W. Yue, and S. Wang, “Object recognition in remote sensing images using combined deep features,” in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Mar. 2019, pp. 606–610, doi: 10.1109/ITNEC.2019.8729392.

[23] Y. Li, K. Fu, H. Sun, and X. Sun, “An aircraft detection framework based on reinforcement learning and convolutional neural networks in remote sensing images,” *Remote Sensing*, vol. 10, no. 2, Feb. 2018, doi: 10.3390/rs10020243.

[24] Z.-Z. Wu *et al.*, “A benchmark data set for aircraft type recognition from remote sensing images,” *Applied Soft Computing*, vol. 89, Apr. 2020, doi: 10.1016/j.asoc.2020.106132.

[25] S. Wang, X. Gao, H. Sun, X. Zheng, and X. Sun, "An aircraft detection method based on convolutional neural networks in high-resolution SAR images," *Journal of Radars*, vol. 6, no. 2, pp. 195–203, 2017, doi: 10.12000/JR17009.

[26] Y. Zhao, L. Zhao, and G. Kuang, "Fast detection aircrafts in complex large scene SAR images," *Chinese Journal of Radio Science*, vol. 35, pp. 594–602, 2020.

[27] P. Han, D. Liao, B. Han, and Z. Cheng, "SEAN: A simple and efficient attention network for aircraft detection in SAR images," *Remote Sensing*, vol. 14, no. 18, 2022, doi: 10.3390/rs14184669.

[28] D. Hejji, O. Gouda, A. Bouridane, and M. Abu Talib, "Evaluation of MTARSI2 dataset for aircraft type recognition in remote sensing images," in *2022 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, Apr. 2022, pp. 1–9, doi: 10.1109/ICNS54818.2022.9771536.

[29] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00444-8.

[30] 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, 2022, doi: 10.1109/TNNLS.2021.3084827.

[31] M. Krichen, "Convolutional neural networks: A survey," *Computers*, vol. 12, no. 8, 2023, doi: 10.3390/computers12080151.

[32] 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.

[33] H. R. Naseri and V. Mehrdad, "Novel CNN with investigation on accuracy by modifying stride, padding, kernel size and filter numbers," *Multimedia Tools and Applications*, vol. 82, no. 15, pp. 23673–23691, 2023, doi: 10.1007/s11042-023-14603-x.

[34] R. Riad, O. Teboul, D. Grangier, and N. Zeghidour, "Learning strides in convolutional neural networks," *arXiv-Computer Science*, vol. 1, pp. 1-17, Feb. 2022.

[35] G. Liu *et al.*, "Partial convolution for padding, inpainting, and image synthesis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 6096–6110, 2023, doi: 10.1109/TPAMI.2022.3209702.

[36] J. Wu, Y. Chua, M. Zhang, G. Li, H. Li, and K. C. Tan, "A tandem learning rule for effective training and rapid inference of deep spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 446–460, 2023, doi: 10.1109/TNNLS.2021.3095724.

[37] A. Zafar *et al.*, "A comparison of pooling methods for convolutional neural networks," *Applied Sciences*, vol. 12, no. 17, 2022, doi: 10.3390/app12178643.

[38] S. Lee and D. Kim, "Deep learning based recommender system using cross convolutional filters," *Information Sciences*, vol. 592, pp. 112–122, May 2022, doi: 10.1016/j.ins.2022.01.033.

[39] H. Benradi, A. Chater, and A. Lasfar, "A hybrid approach for face recognition using a convolutional neural network combined with feature extraction techniques," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 2, pp. 627–640, 2023, doi: 10.11591/ijai.v12.i2.pp627-640.

[40] G. Xia and C.-S. Bouganis, "Augmenting softmax information for selective classification with out-of-distribution data," in *Computer Vision – ACCV 2022*, 2023, pp. 664–680, doi: 10.1007/978-3-031-26351-4_40.

[41] R. Rudd-Orthner and L. Mihaylova, "A benchmark for aircraft recognition using the MTARSI2 dataset," *Zenodo*, 2021, doi: 10.5281/zenodo.5044950.

BIOGRAPHIES OF AUTHORS

Yousef Alraba'nah received his B.Sc. degree in software engineering from Zarqa university (Jordan) in June 2012. In February 2013, he obtained fully-funded scholarship from Zarqa University to complete M.Sc. in computer science. He graduted 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.

Mohammad Hiari is a lecturer in Al-Ahliyya Amman University. He received his first degree in software engineering from Philadelphia University, Jordan, in August 2004 and master degree in computer science from Al Balqa Applied University, Jordan in February 2016. His research area of interest includes VoIP and cybersecurity data mining and optimization. He can be contacted at email: m.hyari@ammanu.edu.jo.