

# Classification of Aircraft in Remote Sensing Images Based on Deep Convolutional Neural Networks

Youssef Ben Youssef<sup>1</sup>, Mohamed Merrouchi<sup>2</sup>, Elhassane Abdelmounime<sup>2</sup>, Toufiq Gadi<sup>2</sup>

<sup>1</sup>*Hassan first university Settat, Ecole Nationale des Sciences Appliquées, Berrechid, Morocco*

<sup>2</sup>*Hassan first University Settat, FST of Settat, Morocco*

**Abstract** *Convolutional Neural Network (CNN)* is a component of *Deep Learning (DL)* recently exploited in different fields. In this work, we improve the performance of multi-label classification based on CNN for remote sensing images of aircraft types. Intensive preprocessing limits the classification rate in previous studies. In order to avoid under-fitting and over-fitting problems, we optimized the architecture and Network parameters. To validate our method the recent public image dataset called *Multi-Type Aircraft Remote Sensing Images (MTARSI)* is used. Extensive experiments prove the effectiveness of the proposed method in terms of classification rate.

**Keywords** Computer vision, Machine learning, Deep learning, Convolutional Neural Network, Classification

**AMS 2010 subject classifications** 62P30, 68U10

**DOI:** 10.19139/soic-2310-5070-1143

## 1. Introduction

Image classification is one of the most important field of computer vision and machine learning. Assigning automatically predefined labels to images is that the aim of image classification. One of the important issues in remote sensing image processing is aircraft type classification, and it has been widely used in civil and military applications. To solve this issue, researchers have designed and implemented several methods for the image's classification. Machine learning (ML) has emerged united of the foremost successful artificial intelligence techniques and has achieved impressive performance within the field of computer vision and image processing, with applications like image medical classification [1][2], remote sensing image scene classification[3], aircraft detection [17], and aircraft classification [5]. All algorithms in ML are based on many handcrafted features available from images for doing the classification; those methods are named also handcraft descriptors in classification. Recently in remote sensing image classification based on DL is growing. It has been widely applied in diverse areas study, including vegetated areas, urban areas, wetlands, and forest areas[6]. As a result, the details of ground objects, such as contour, structure, and texture information, can be obtained conveniently. Among DL algorithms used in classification, CNN have gained popularity. Since 2012, CNN has attracted more attention because of the increasing computing power, availability of lower-cost hardware, open-source algorithms, and the rise of big data [7]. Getting deeper is an important typical trend of CNN [8]. By increasing depth, CNN can approximate the target function with increased non-linearity and get better descriptor representations. However, the complexity of the network is increasing, which makes the network more difficult to optimize and easier to get under-fitting or over-fitting. The main contributions of this work can be summarized as follows:

---

\*Correspondence to: Youssef Ben Youssef (Email: youssef.benyoussef@uhp.ac.ma). Ecole Nationale des Sciences Appliquées, Avenue de l'université, B.P :218 Berrechid. Morocco(26100).

- CNN structure is designed to increase classification rate;
- The underfitting and overfitting are reduced by optimizing the parameters in training and testing of the model.

The rest of this paper is organized as follows. In Section 2, we briefly review the related work. The Dataset MTARSI and CNN model of deep learning are presented in Section 3, whereas Section 4 results and discussion are largely explained. The paper is ended with a conclusion in section 5.

## 2. Related Work

Initially, the modern CNN is presented in [9]. The authors developed a multi-layer artificial neural network called LeNet-5 that could classify handwritten digits[10]. Subsequently, many CNN models have been proposed to classify images and have been considered in different topics for images. Previous research projects are studied for three typical CNN application cases in remote sensing image classification: scene classification, object detection and object segmentation are presented [11]. Due to the state-of-the-art in image classification using CNN such as VGG [12], GoogLeNet [8], ResNet[13], DenseNet [14], and EfficientNet[15] were successfully applied to the ImageNet dataset available on line[16]. Recognition of the aircraft from remote sensing images focuses on deciding whether an object is an aircraft or not based on reinforcement learning and convolutional neural networks[17]. Wu and Prasad suggested a recurring neural network in which some convolutional layers are followed by recurring layers. Middle level and locally invariant descriptors are extracted from raw HSI, and spectral context features are extracted from the descriptors generated by convolutional layers [18]. Fu et al. propose a fine-grained aircraft recognition method for remote sensing images. Their multi-class activation mapping uses two sub networks, the target net and the object net, to fully use the descriptors of discriminative object parts[19]. Zhao et al. propose the aircraft type recognition issue by detecting the landmark points of an aircraft using a vanilla network designed a keypoint detection model based on CNN and a keypoint matching method to recognize aircraft, transforming the aircraft recognition issue into the landmark detection problem[?] . All of the above works have been trained and tested using different dataset. The only common point is the scene classification with intensive preprocessing, which consequently affects the performance of the systems proposed. Zhi-Ze Wu et al. examined the performance of five state-of-the-art CNN structures namely VGG, GoogLeNet, ResNet, DenseNet, and EfficientNet and found that the result that EfficientNet is better than others in term of classification rate [21]. Marmanis et al. used a CNN pretrained from the ImageNet challenge and used it to extract an initial set of representations for Earth observation classification[22]. These methods for identifying the object type from remote sensing images have achieved significant results, but there are still many challenges. In the literature, there are numerous variants of CNN architectures with huge parameters that need to be adapted for each case studied.

## 3. Proposed Approach

This section provides information on the MTARSI data set used in this work, a detailed description of the evaluated network architecture is provided below

### 3.1. Dataset

Database available in the literature are divided into three groups based on three of the basis of classification tasks: scene classification, object detection and object segmentation. Among this Dataset, MTARASI dataset has the advantage that labeled images include a single type of aircraft in different orientations. The MTARSI dataset was used for training and testing our proposed method. It is an open-source dataset for Aircraft Type Recognition from Remote Sensing. The MTARSI dataset contains a total of 9,385 remote sensing images that were taken from Google Earth satellite images and manually expanded. It contains 20 different types of aircraft and different sizes covering 36 airports. Each image contains exactly one complete labeled aircraft. The spatial resolution of the images varies in a range of 0.3 to 1, and contains various orientations, aspect ratios, and pixel sizes of the objects. In addition, the images vary depending on to the altitude, nadir-angles of the satellites, and the illumination. Some image patches

have some cropped objects, and some examples are black and white images. These variations in the MTARSI allow the trained aircraft classification architectures to achieve similar performance in different image conditions. The aircraft may differ on type and model as illustrated in Figure 1, where a sample of image aircraft types extracted from MTARSI dataset is depicted.



Figure 1. MTARSI dataset samples.

### 3.2. Convolutional Neural Network Architecture

A mathematical model function for a neural network can be viewed as input  $x$  to output values  $y$ . In supervised



Figure 2. Network function model.

classification, the function  $\psi$  assigns an input data to a given set of a predefined classes in the output. Classification problem can be described more formally, given a training set of data, the objective is to learn a function  $\psi$  called a hypothesis, so that  $\psi(x)$  is an optimal predictor for the corresponding value of  $y$ . In dataset MATRASI used in this work, we have extracted  $N = 8779$  total images. Each image is  $200 \times 200$  pixels, then each image is represented as  $D = 200 \times 200 \times 3 = 120.000$  distinct values, and a total of  $C=20$  class labels for aircraft.

$$y = \psi(x) \quad (1)$$

where  $y \in \{c_1, c_2, \dots, c_{20}\}$  and  $x \in \{1, \dots, 120000\}$  The loss function is the cross entropy between the predicted probability and the true label  $y$ ; it measures the error by comparing the target label vector  $y$  and the predicted label

vectors  $\hat{y}$ . For N training samples, Loss function is defined as

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N y_i \log \hat{y}_i \quad (2)$$

The cross-entropy loss is numerically stable in training and is faster in term of convergence rate when coupled with softmax normalization [23]. Loss function is used in the training process to find the parameter values for model proposed. The loss is returned on training and testing process and its interpretation is how well the model is doing for these two process. DL architectures have been developed and have been applied in different fields and have performed several algorithms in visual recognition. The structure of CNNs allows the model to learn highly abstract feature detectors and to map the input descriptors into representations that can clearly outperform the performance of the subsequent example. The AlexNet structure contains filters of  $11 \times 11$ , the recent trends to wards using smaller filters. The ResNet architecture does not contain filters larger than  $3 \times 3$ . The advantage of the CNN is its flexibility to add or reduce the number of layers in its structure for a given task. Furthermore, there are many optional techniques that can be used to train it. The CNN design is depicted in Figure 3. Generally, a CNN

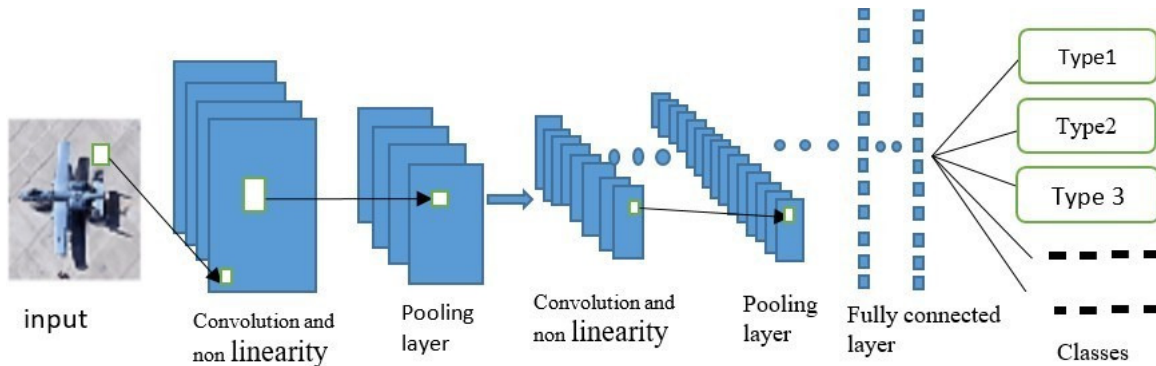


Figure 3. CNN design.

mainly consists of three parts: convolutional layers, pooling layers, and fully connected layers.

3.2.1. *Convolutional layers* It has a nonlinear sufficiently intense and will not be able to model the response variable (as a class label). The convolutional layer computes the output feature map:

$$y^s = \sum_{i=1}^q W_i^s * x_i + b \quad (3)$$

where  $*$  is a two-dimensional discrete convolution operator,  $W^s$  is weight and  $b$  is a bias parameter. The parameters weight  $W^s$  and bias vector  $b$  are adjusting in training process.

3.2.2. *Nonlinear layer* in this layer, a nonlinear function applied to each component of a mapped component. The nonlinear layer is added after each convolution operation. It has an activation function that effects the nonlinear property. The output of ReLu is the maximum value between zero and the input value. The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time and ReLU train several times faster than their equivalents with other units. *Rectified Linear Unit (ReLU)* is commonly chosen and defined as :

$$ReLU(x) = \max(0, x) \quad (4)$$

**3.2.3. Pooling Layer** it follows a convolutional layer and it is used to reduce the dimensionality of feature maps and improve the robustness of the extracted features. This allows us to reduce the number of parameters, which both decrease the training time and avoid over-fitting. It is usually placed between two convolutional layers. There are two types of basic pooling operation which are the most commonly used: average pooling and max pooling. Detailed theoretical analysis can be found in [7]. Max pooling returns the maximum value from the portion of the image covered by the Kernel.

**3.2.4. Fully connected layer** In this layer, the output maps of the last convolution layer or pooling layer are flattened into vectors, serving as the inputs to the first fully connected layer. The output of the final fully connected layer is the learned feature, forming the result of which is extracted from the input image by the convolutional network. In the training of the model, the flattened output is fed to a feed-forward neural network whose back propagation algorithm is applied to each iteration [24]. Over a fixed number of epochs, the model is able to distinguish between dominating and certain low-level descriptors in images and classify them using the softmax function used in classification technique. Softmax is usually used in the multi-classification tasks defined below :

$$f(x) = \frac{e^{(x_i)}}{\sum_i e^{(x_i)}} \quad (5)$$

Softmax function returns value in the range [0,1], it can be viewed as form of a probability distribution. It defines a flexible learning task with adjustable margin[25].

## 4. Experiment Results and Discussions

For implementation, we used a PC with an HP i5 8th generation CPU processor 1.80GHz, 4 Go RAM, ×64, with Python 3.7 for Windows 10 in Keras environment.

### 4.1. Data Preprocessing

In this study, 8779 images were extracted from the MATRASI dataset libeled in 20 classes of aircraft. Before training our CNN, because all image are in different sizes all image were resized to the size of  $200 \times 200$ ; and normalized using ImageDataGenerator function. This method consists on dividing all the pixels of the image by 255 to range min-max values between 0 and 1. Once we have images loaded, we put them into a CNN that does the classification.

### 4.2. Model adopted

After data preprocessing process, the model built contains 3 convolution layers paired with a batch normalization, three max-pooling layers and a fully connected layer with 2 hidden layers. Each three convolution layers is accompanied by a pooling layer. There are 20 final output nodes at the end of the CNN. Each node represents an aircraft type. The CNN model proposed in this work is shown in Figure 4. Our CNN architecture has more than 8 million parameters. After several experiences as aim to increase the classification rate and avoid overfit and underfit, the CNN structure built has 3 convolution layers paired with a batch normalization and 3 Max pooling layers followed by 2 fully connected layers. The output is a score matrix for the weights for each class. Each filter has a kernel size of  $3 \times 3$ .

### 4.3. Training and Testing Process

After building the CNN, it was trained on 8,779 images for 20 epochs with batch size of 128, compiled with categorical crossentropy loss function and RMS optimizer with the learning rate 0.0001. RMSprop is a gradient based optimization technique; it was developed as a stochastic technique for mini-batch learning [26]. The classification rate or accuracy is one metric used to measure how often the algorithm classifies an image correctly

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 198, 198, 32)	896
max_pooling2d (MaxPooling2D)	(None, 99, 99, 32)	0
conv2d_1 (Conv2D)	(None, 97, 97, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 48, 48, 64)	0
conv2d_2 (Conv2D)	(None, 46, 46, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 23, 23, 128)	0
flatten (Flatten)	(None, 67712)	0
dense (Dense)	(None, 128)	8667264
dense_1 (Dense)	(None, 20)	2580

Total params: 8,763,092  
 Trainable params: 8,763,092  
 Non-trainable params: 0

Figure 4. Structure of CNN adopted.

and defined as

$$Accuracy = \frac{\text{Correct predictions}}{\text{All predictions}} \tag{6}$$

After training process, our system was tested on 2,122 images selected randomly in file image training dataset. The following graphs in figure 5 shows the accuracy and loss vs the number of epochs in training and testing model. As we can see in Figure 5, the accuracy plots at each epoch shows that our model suffer little from over-fitting.

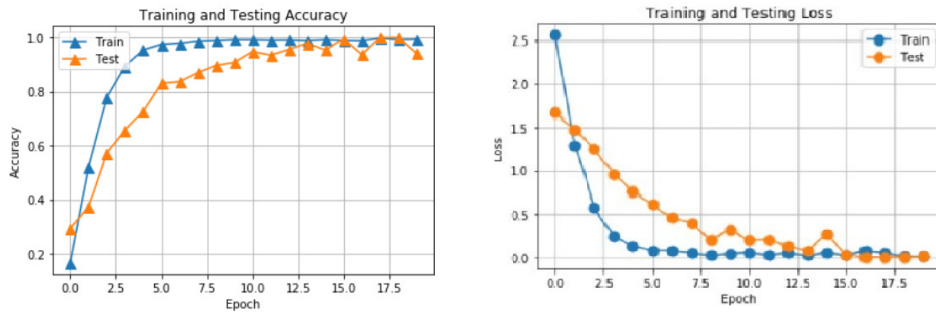


Figure 5. Training and testing accuracy vs epochs in left and in right Loss training and testing vs epochs.

Beyond an epoch 10, the test accuracy is slightly lower than the training accuracy. This implies that our model proposed obtains better performance with less complexity when we choose. The highest test accuracy is reached from epoch 10 and does not increase. And we can see, beyond epoch 10, the loss test and train decrease. The highest test accuracy at all the epochs is reported as the best score 99.90%. Compared to other algorithms developed by researchers using the same MTARSI dataset is presented in Table 1 in term of average accuracy related to all epoch.

### 5. Conclusion

An approach based on convolutional neural networks for the multi-classification of aircraft type is presented. The empirical results indicated that our approach provides superior results on MTARSI data sets. Classification rate (90,66%) is a good result obtained with only normalization data augmentation and without model regularization. CNN have been shown to be very powerful for image analysis and classification tasks in other field. In the future,



Table 1. Comparison Result of proposed method.

Method	Accuracy(%)
VGG	87,66
GoogLeNet	86,53
ResNet	89,6
DensNet	89,15
EfficientNet	89,79
Our approach	90,66

other techniques may be considered such as data augmentation to add more training images and develop better aircraft classification methods.

## REFERENCES

1. B. Ben Youssef, E. ABDELMOUNIM, and A. Belaaguid, *Mamograms Classification using Support Vector Machine*, in Handbook of Research on Advanced Trends in Microwave and Communication Engineering, edited by A. El oualkadi and J. Zbitou, IGI Global, pp. 546–569, 2017.
2. Y.B. Youssef, E. Abdelmounim, J. Zbitou, M. Elharoussi, and M. Boujida, *Comparison machine learning algorithms in abnormal mammograms classification*, International Journal of Computer Science and Network Security, vol. 17, no. 5, pp. 19–25, 2017.
3. G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, *When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs*, in IEEE Transactions on Geoscience and Remote Sensing, vol. 56, no. 5, pp. 2811–2821, 2018.
4. 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 Journal, vol. 10, pp. 243, 2018.
5. Y. B. Youssef, M. Merrouchi, E. Abdelmounim, and T. Gadi, *Aircraft Type Classification in Remote Sensing Images using Deep Learning*, in IEEE International Conference on Electronics, Control, Optimization and Computer Science, pp. 1–6, 2020.
6. L. Maa, Y. Liuc, X. Zhang, Y. Ye, G. Yind, and B. A. Johnson, *Deep learning in remote sensing applications: A meta-analysis and review*, Journal of Photography and Remote Sensing, vol. 1521, pp. 166–177, 2019.
7. A. Krizhevsky, I. Sutskever, G.E. Hinton, *Imagenet classification with deep convolutional neural networks*, in Proceedings of Conference on Neural Information Processing Systems, pp. 1106–1114, 2012.
8. C. Szegedy, *Going deeper with convolutions*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9, 2015.
9. Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L.D. Jackel *Handwritten digit recognition with a backpropagation network*, Proceedings of the Advances in Neural Information Processing Systems (NIPS), pp. 396–404, 1989.
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.
11. L. Zhang, L. Zhang, and B. Du, *Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art*, IEEE Geoscience and Remote Sensing Magazine, n.2, pp.22–40, vol.4, 2016.
12. K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, arXiv 1409.1556, 2014.
13. G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, *Densely connected convolutional networks*, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700–4708, 2017.
14. K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.
15. M. Tan and Q. V. Le, *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*, in Proceedings of the 36th International Conference on Machine Learning, 2019.
16. L. Fei-Fei, *ImageNet: crowdsourcing, benchmarking and other cool things*, In CMU VASC Seminar, vol.16, pp. 18–25, 2010.
17. Y. Li, K. Fu, H. Sun, and X. Sun, *An aircraft detection framework based on reinforcement learning and convolutional neural networks in remote sensing image*, Remote sensing, vol. 10, no. 2, pp. 243, 2018.
18. H. Wu and S. Prasad, *Convolutional Recurrent Neural Networks for Hyperspectral Data Classification*, Remote Sensing vol.9, pp.298, 2017.
19. Y.Z.M.K. Fu, W. Dai and X. Sun, *MultiCAM : Multiple Class Activation Mapping for Aircraft Recognition in Remote Sensing Images*, Remote sensing, vol. 11, 2019.
20. A. Zhao, K. Fu, S. Wang, J. Zuo, Y. Zhang, Y. Hu, and H. Wang, *Aircraft recognition based on landmark detection in remote sensing images*, IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 8, pp. 1413–1417, 2017.
21. Z.Z. Wu, S.H. Wan, and X. Fang, *A benchmark data set for aircraft type recognition from remote sensing images*, Applied Soft Computing Journal, vol.89, pp. 106–132, 2020.
22. D. Marmanis, M. Datcu, T. Esch, U. Stilla, *Deep learning earth observation classification using ImageNet pretrained networks* IEEE Geoscience and Remote Sensing Letters vol. 13, no. 1, pp. 105–109, 2015.
23. C. M. Bishop, *Neural networks for pattern recognition*, in neural networks for pattern recognition, 2005.

24. D. Rumelhart, G. Hinton, and R. Williams, *Learning representations by back-propagating errors*, Nature, vol. 323, no. 6088, pp. 533–536, 1986.
25. W. Liu, Y. Wen, Z. Yu, and M. Yang, *Large-margin softmax loss for convolutional neural networks*, in Proceedings of the 33rd International Conference on International Conference on Machine Learning, Vol. 48, pp.507–516, 2016.
26. J. Xu, Z. Zhang, T. Friedman, Y. Liang and G.V. den Broeck *A Semantic Loss Function for Deep Learning with Symbolic Knowledge*, Proceedings of the 35th International Conference on Machine Learning, arXiv, 2018.