

Deep convolutional neural networks architecture for an efficient emergency vehicle classification in real-time traffic monitoring

Amine Kherraki, Rajae El Ouazzani

Image Laboratory, School of Technology, Moulay Ismail University of Meknes, Meknes, Morocco

Article Info

Article history:

Received Jun 12, 2021

Revised Dec 14, 2021

Accepted Dec 28, 2021

Keywords:

Convolution neural network

Deep learning

Emergency vehicle

Image classification

Image processing

ABSTRACT

Nowadays, intelligent transportation system (ITS) has become one of the most popular subjects of scientific research. ITS provides innovative services to traffic monitoring. The classification of emergency vehicles in traffic surveillance cameras provides an early warning to ensure a rapid reaction in emergency events. Computer vision technology, including deep learning, has many advantages for traffic monitoring. For instance, convolutional neural network (CNN) has given very good results and optimal performance in computer vision tasks, such as the classification of vehicles according to their types, and brands. In this paper, we will classify emergency vehicles from the output of a closed-circuit television (CCTV) camera. Among the advantages of this research paper is providing detailed information on the emergency vehicle classification topic. Emergency vehicles have the highest priority on the road and finding the best emergency vehicle classification model in real-time will undoubtedly save lives. Thus, we have used eight CNN architectures and compared their performances on the Analytics Vidhya Emergency Vehicle dataset. The experiments show that the utilization of DenseNet121 gives excellent classification results which makes it the most suitable architecture for this research topic, besides, DenseNet121 does not require a high memory size which makes it appropriate for real-time applications.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Amine Kherraki

Image Laboratory, School of Technology, Moulay Ismail University of Meknes

Marjane, Meknes 50050, Morocco

Email: amine.kherraki.9@gmail.com

1. INTRODUCTION

In recent years, intelligent transportation system (ITS) has gained much importance, due to the vast increase in the number of cars, and other types of vehicle on the road [1]. Smart systems contain a large amount of high-quality information for a smart and secure use [2]. For example, the ITS provides multiple services such as transport and traffic monitoring. The aim of this monitoring is to acquire and analyze vehicle movements, provide accurate data, and important statistics about the type and shape of the vehicle, as well as the evaluation of road traffic and safety [1], [3]. Thus, transport and traffic monitoring comes to facilitate the human labor, by vision-based tasks that a computer or automated system can perform [4]. Such a system is essential for effective real-time traffic monitoring that are able to detect changes in traffic characteristics in a timely manner, allowing regulatory agencies and authorities to respond quickly to traffic situations.

In literature, many applications for traffic video surveillance make vehicle re-identification in a multi-camera environment [5]. These applications can report important information, such as traffic flow or traffic information, and travel time in a distributed traffic control system. According to [6], and with regard to object localization, the authors proposed an approach to find an instance of a vehicle by estimating its position with ratio and size, which is widely used for vehicle tracking. In [7], the authors used a fixed camera-based

application for traffic video analysis and a central processing unit (CPU) based system to count the number of passing vehicles along with their speed on a highway. The analysis of the results shows that the application will not only count the number of vehicles but also estimate the speeds of the vehicle by providing more traffic flow information. Deep learning approaches and algorithms such as convolutional neural network (CNN) are widely used in many areas, applications, and issues in computer vision like image recognition, segmentation, detection, and classification [8], [9]. In fact, vehicle detection methods must be fast enough to operate in real-time, and to be immune to changes in lighting and different weather conditions, and have the ability to separate vehicles from image sequences accurately and efficiently [10]. Image-based vehicle classification is one of the most promising techniques for large-scale traffic data collection and analysis [11], and deep learning algorithms are widely used in this topic. For example, the authors in [12] have used CNN architectures to classify road vehicle images into six categories, including large buses, cars, motorcycles, minibuses, trucks, and vans. They show that using CNN for vehicle type classification provides the most recent results on previously cropped images containing only vehicles [12].

In this paper, we will discuss an important topic to which researchers had not given much importance before, knowing that emergency vehicles have the top priority on the road. Our contribution consists of making a comparison between the results of classification of emergency vehicles by using many CNN architectures. Therefore, our paper would help researchers and developers in the implementation of some efficient real-time emergency vehicle classification applications. The rest of the paper is organized as; section 2 presents related work on the classification of vehicles in general, as well as the classification of emergency vehicles. Section 3 contains the definition of CNN, with a detail of each architecture. Section 4 provides details about the research method, especially the dataset images preprocessing, and the implementation of CNNs. Section 5 presents, the experimental results, and performance analysis discussion. Finally, section 6 contains concluding observations and future work, with guidance in this area.

2. RELATED WORKS

2.1. Vehicle classification

Vehicle classification is a promising research task in ITS, and deep learning algorithms are the most used techniques for this task. Indeed, CNN has performed well in this area, in terms of real-time speed and accuracy compared to other machine learning algorithms, such as support vector machine (SVM), decision tree (DT). According to [13], in 2015, the authors proposed a semi-supervised CNN for vehicle detection from frontal view images. The authors used Laplacian filter learning to obtain the filters of the network on a large amount of unlabeled data. Besides, they used a Softmax classifier in the output layer, and they trained their model on a small amount of labeled data. A year earlier, the authors in [14] have introduced an unattended CNN for vehicle classification. They used CNN to learn characteristics of vehicles and then classify them by using Softmax regression. The proposed network filters are learned with a sparse filtering method. In [15], the authors used CNN with low-resolution video images to detect and classify vehicles. The preprocessing operation includes resizing each image and adjusting the contrast with histogram equalization. The proposed CNN architecture detects higher-level features such as edges and corners. In addition, the authors vary the number of filters and their sizes as well as the number of hidden layers [15]. After that, the authors in [16], show that, a simple CNN surpasses scale-invariant feature transform (SIFT) and support vector machine (SVM) models applied on vehicle classification. Finally in [17], [18], the authors used you only look once (YOLO) for vehicle detection and AlexNet model for vehicle classification. Further, the authors used AlexNet as an entity extractor and performed the classification of extracted entities by using a linear SVM.

2.2. Emergency vehicle classification

In this subsection, we will present some relevant work in emergency vehicle classification. According to [19], the authors used color segmentation, which adopted the hue saturation value (HSV) and red green blue (RGB) color models to characterize the emergency vehicle siren light. Subsequently, they took the light as a detection function for the identification and classification of emergency vehicles in the input video. Finally, they used SVM classifier to perform the vehicle classification. Basically, the authors cut the image horizontally so that they can detect the siren light in the highest part. However, sometimes the ambulances are not provided with a siren. In our opinion, to train correctly a model, we need a dataset containing images of multiple angles and positions with and without light and siren. In the real world, when we put sensors in the car, we will not be able to see the complete image of the car, because the angle of view or the visual field of the cameras is very limited to the places where it is necessary to put the sensors. Thus, we can say that it is impossible to make the decision through these criteria, because it is rare to see lights and sirens in emergency vehicle images.

Later in [20], neural network (NN) and structural traits are used to train and recognize ambulance characters in emergency vehicles. The authors explore the photography processing machine through “Thinning” and “Hilditch” algorithms, and the structural features of a character in the picture. In addition, they

diagnosed some features of the widespread language while, the implemented approach is based on the detection of the characters “AMBULANCE” or “108” on the vehicle. From our point of view, we find their idea not very practical, and sometimes there are other characters or just symbols like the moon, the cross, or another language outright on the ambulance.

3. CONVOLUTION NEURAL NETWORK

In recent years, neural networks (NNs) have become one of the most widely used machine learning algorithms. The latter have been proven decisively over time that they outperform other traditional algorithms in terms of accuracy and speed. According to [21], CNNs are a version of artificial neural networks (ANNs), and they correspond to the patterns of connectivity found in the visual cortex of animals [22]. Mathematically, these models of connectivity are described by a process of convolution. CNN is an improvised variant of the multilayer perceptron, typically it is composed of an input layer, an output layer, and many hidden layers [23]. CNNs are primarily used for computer vision tasks, such as facial recognition, image classification, identification and detection, and image processing in the field of autonomous vehicles.

As shown in Figure 1, a CNN is made up of two main parts. The first one is feature extraction (feature learning), where the network will perform the convolution and grouping operations so that it can detect the features [24]. The second one is classification, where the fully connected layers will serve as a classifier on the top of these extracted features, thus, they will assign a predicted probability about the object in the image [24]. CNNs are composed of four types of layers, including convolutional, pooling, rectified linear unit (ReLU), and fully connected (FC) [25]. In regard to convolutional layers, an input image is analyzed by a set of filters that produces a feature map. This output is then sent to a grouping layer to reduce the size of feature map, thus, it reduces the processing time by mapping the feature map on the most important information [25]. The convolution and grouping processes are repeated multiple times, and these repetitions depend on the CNN architecture, later, the outputs of the condensed feature map are sent to a series of FC layers. The latter flatten the maps together and check the probabilities of each feature occurring with the others to make the best classification [26], [27]. The ReLU layer is all about adding nonlinearity to a system because the convolution performs linear operations. It is simply a multiplication and a summation by element [28]. In the following, we will highlight some famous CNN architectures.

- VGG16: VGG16 is a popular choice when extracting CNN features from images. It has 16 layers, with 13 convolutional layers separated by 5 Max Pooling layers. In addition, it has 3 FC layers with 4096 neurons for each. The final dense layer is equal to the number of classes used for the final classification [26].
- VGG19: It has almost the same principle as VGG16, except that VGG19 contains 19 convolutional layers in total. It consists of 16 convolutional layers separated by 5 Max Pooling layers, 3 FC layers at the end, 4096 neurons for the first 2 FCs, and 1000 neurons for the last FC. Finally, a dense layer which is equal to the number of classes is used for the final classification [29].
- ResNet-50: Created in 2015, ResNet-50 introduced the idea of residual learning in order to make deeper CNNs. The input of the convolutional layer is replicated and added to the output of this layer, after this process the network recognizes practically learn residuals. ResNet-50 consists of 49 convolutional layers, with a Pooling layer, an Average Pooling layer, and an FC layer with 1000 neurons. However, the ResNet-50 benefit in terms of accuracy is related to the execution time and memory requirements [30].
- Inception-V3 and Xception: The main contribution of these architectures is that they combine many different convolution filters, for example, Conv (1×1), Conv (3×3), and Conv (5×5) in a multi-extractor [31]. The Inception-V3 architecture typically consists of 22 convolutional layers, with 5 pooling layers. Its variety Inception-V3 is too demanding in terms of memory, for this reason, a more optimized variant of the creation family has been proposed, it is called Xception where separable convolutions have been proposed in an attempt to reduce computational complexity [32].
- MobileNetV2: Using the idea of separable packaging, the MobileNetV2 model family achieves optimal performance at a low computational cost. The MobileNetV2 model consists of 27 layers, 13 deep Conv (3×3), 13 Conv (1×1), 1 Conv (3×3), with an Average Pooling layer and an FC layer. The MobileNetV2 architecture is distinguished by its low requirements in terms of parameters and memory [33].
- Inception-ResNet-V2: Inception-ResNet-V2 is a CNN with 164 deep layers, it combines the Inception architecture with residual connections. Inception-ResNet-V2 is a variant of Inception-V3, and it consists of 155 Convolution layers, 3 Average Pooling Layers, 4 Max Pooling layers, and 2 FC layers [34].
- DenseNet121: Densely networks involve the use of connection hopping in a different way. DenseNet121 is a model generated with 121 layers, 120 convolution layers including Conv (1, 1), Conv (3, 3), Conv (7, 7) in different blocks of Dense, separated by transition layers and a Pooling layer. The strong

point of DenseNet121 is that it outperforms the majority of CNN architectures in terms of accuracy, and does not require large memory [35], [36].

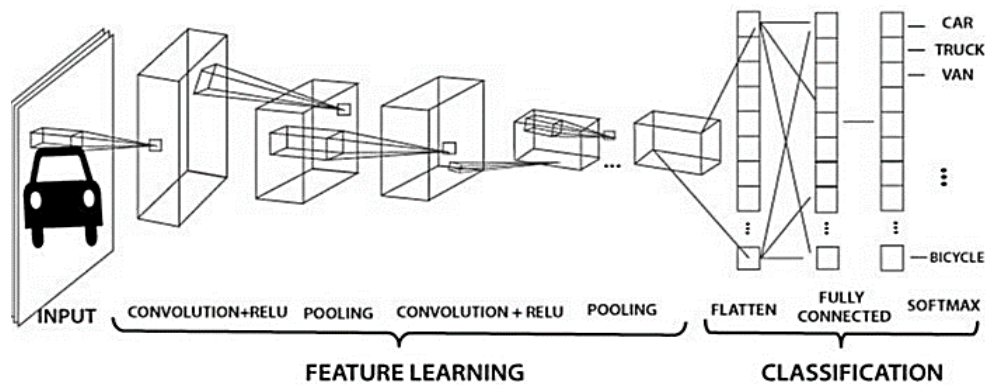


Figure 1. The Convolutional neural network architecture [23]

4. RESEARCH METHOD

In this section, we provide information about the general setup, and the dataset that we used in our experiments. Next, we present detailed information about the building and training of the CNN network. Figure 2 shows our workflow for classifying emergency vehicles.

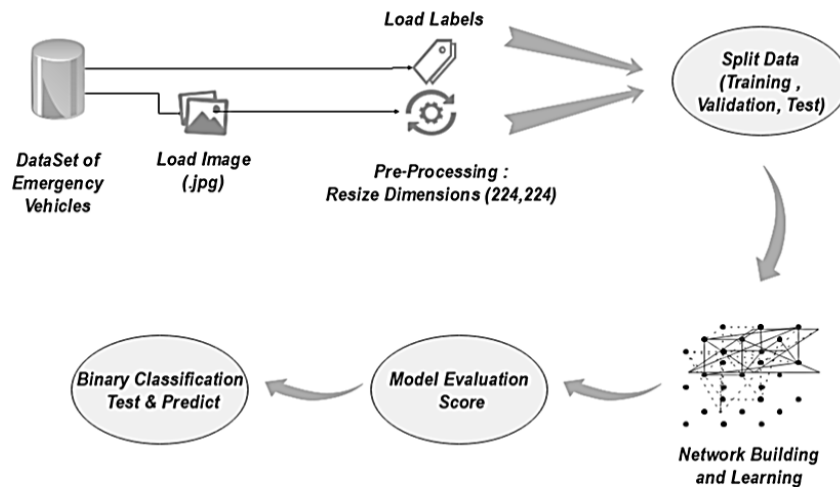


Figure 2. The workflow of the proposed method to classify emergency vehicles

4.1. Dataset and image pre-processing

For the experiments, we use a large-scale dataset named analytics vidhya emergency vehicle [37]. It contains 2352 vehicle images of different dimensions, 1361 images for normal vehicles and 991 for emergency vehicles such as police cars, ambulances, and fire brigades. Figure 3 shows some vehicle images taken in different angles and positions. All the experiments were implemented in Python language using the Keras and TensorFlow libraries on Kaggle simulator. Besides, we have used a laptop with an Intel i7-6500 Hq processor, a 2.5 GHz processor and 8 GB of memory. We have implemented several CNN architectures, including DenseNet121, MobileNetV2, Inception-ResNet-V2, Inception-V3, VGG19, VGG16, ResNet-50 and Xception to classify emergency vehicle images, and it takes about 20 hours for all models. We have performed a preprocessing on the images by resizing them to (224×224) as shown in Figure 4. After that, we randomly divided the data into two parts using the “sklearn” library. We have used 70% for training with validation, and 30% for test.



Figure 3. Examples of emergency and normal vehicle images from analytics vidhya emergency vehicle dataset [35]

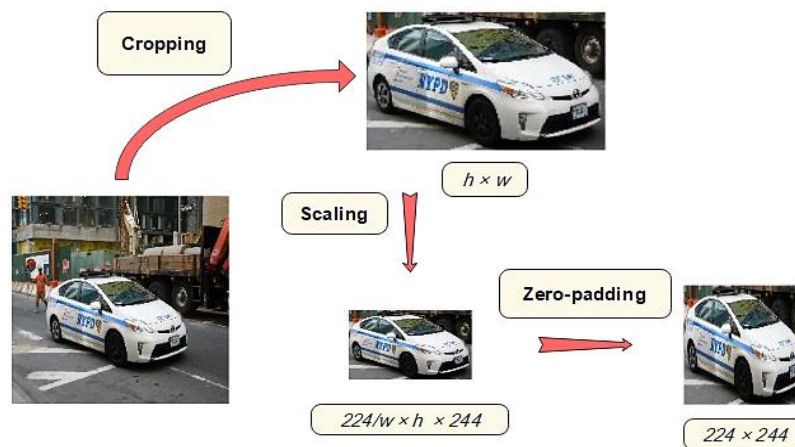


Figure 4. The workflow of image pre-processing

4.2. Network building and learning

After unifying the size of images, we have implemented eight CNN architectures, including DenseNet121, MobileNetV2, Inception-ResNet-V2, Inception-V3, VGG19, VGG16, ResNet-50, and Xception which are already predefined in the Keras. Next, we provide some parameters, including the input size with the dimensions $(224 \times 224 \times 3)$ for each CNN. We have added three other layers to each of the eight implemented CNN architectures. The first additional layer is the “GlobalAveragePooling2D”. It calculates the average output of each feature map in the previous layer, and it also reduces the dimensions of the input image, which accelerates the training duration. Besides, it prepares the model for the final classification layer. The second layer is “Dropout”, which is proposed in 2014, and it is a regularization technique for neural network models. According to [38], deep neural networks with a large number of parameters are very powerful deep learning systems. However, overfitting is a critical problem in the training of such networks. Large networks are also slow to use, making them difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. Thus, Dropout comes to address this problem by randomly dropping some units from the neural network during the training process. This operation prevents units from co-updating too much and remarkably reduces overfitting and gives great improvements over other regularization functions such as L1 and L2 regularization techniques. The third and the last additional layer is “Dense”, where it is necessary to put the number of output classes. And as long as we have performed a binary classification by using two classes for normal and emergency vehicles, thus we applied “sigmoid” activation function. However, if the number of classes exceeds two, we will use the “Softmax” function. Once the three layers are added to the implemented CNN architectures, we use “Adam” as the adaptive optimizer of the learning rate. Figure 5 displays the summary of DenseNet121 architecture with the three added layers, and Figure 6 shows the detailed DenseNet121 architecture [39]. We have built the eight CNN architectures that we have already mentioned in

the previous section. We have trained each architecture on 500 epochs, and we have used the accuracy, F1, and loss metrics to evaluate the performances of our architectures.

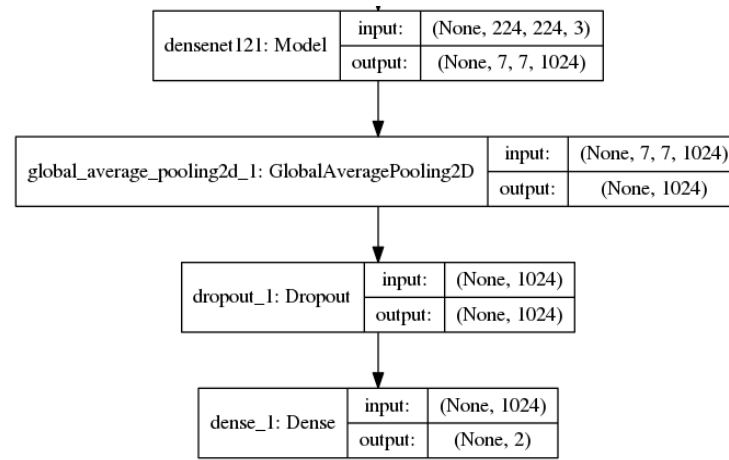


Figure 5. DenseNet121 building model summary with the three added layers

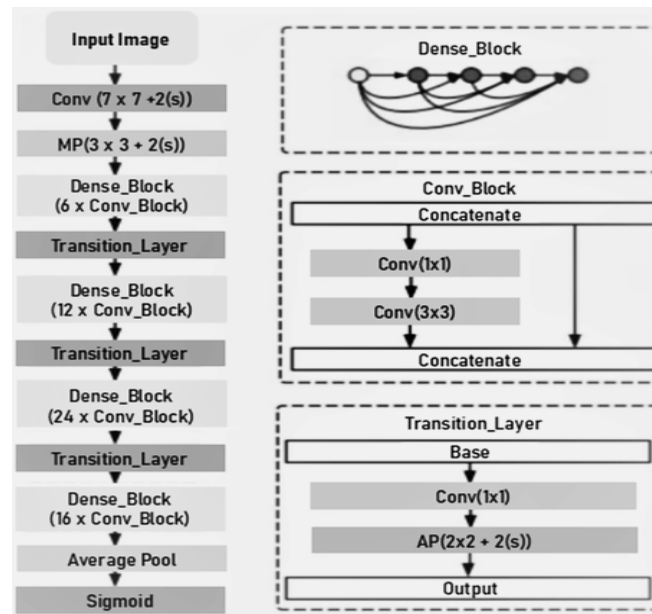


Figure 6. The used DenseNet121 architecture [37]

5. RESULTS AND DISCUSSION

In this section, we will discuss the experimental evaluation of the eight implemented CNN models that are summarized in Table 1. First, in terms of accuracy, DenseNet121 outperforms all of the other architectures with an accuracy score of 95.14%, closely followed by VGG16 with 92.3%. Simultaneously, for the F1 metric, we observe that the DenseNet121 still outperforms them, with a score of 93.87%, followed by VGG16 with 91.08%. The other models, MobileNetV2, ResNet-50, VGG19, Inception-ResNet-V2, Xception, and Inception-V3 had 91.90%, 91.90%, 91.49%, 91.09%, 90.68%, 88.25% respectively. In the end, they kept the same order of accuracy.

However, the VGG16 network needs very high computation and storage requirements, which does not always make them suitable for systems with limited resources and real-time usage. The MobileNetV2 model achieves an accuracy score of 91.9% and it does not need great requirements in terms of calculation and memory, thus, it is the most qualified model for real-time uses. Inception-ResNet-V2 offers an accuracy of 91.09%, but we observe that it surpasses all of the other models in terms of storage as well as time processing,

which makes it inappropriate for real-time applications. For the rest of the models, including ResNet-50, Xception, VGG19, and Inception-V3, they had 91.9%, 90.68%, 91.49%, and 88.25% respectively in accuracy, however, their storage and processing requirements are high.

Table 1. Simulation results of CNN models

Models	Parameters (M)	Accuracy (%)	F1 (%)	Loss (%)	Memory (MB)
Inception-V3	21, 806,882	88.25	84.84	27.11	83.69
Inception-ResNet-V2	54, 339,810	91.09	88.78	26.57	208.45
MobileNetV2	2, 260,546	91.90	89.79	19.18	8.92
DenseNet121	7, 039,554	95.14	93.87	22.76	27.5
Xception	20, 865,578	90.68	89.21	23.86	79.87
ResNet-50	23, 591,810	91.90	89.58	17.20	90.33
VGG16	14, 715,714	92.30	91.08	17.86	56.19
VGG19	20, 025,410	91.49	89.34	25.05	76.45

In regards to the loss metric, we notice that ResNet-50 had the minimum loss score of 17.20%, closely followed by VGG16 with 17.86%. However, VGG19, Inception-V3, Inception-ResNet-V2 had high loss scores, compared to the other models. Thus, they require a big amount of memory, which does not always make them suitable for a real-time use. In terms of parameters, we notice that the MobileNetV2 had the best result with the smallest number of required parameters. And the other models kept the same order as mentioned in the memory metric, because the number of parameters has a direct relationship with the size of the required memory, so the more parameters there are, the more memory there is. And this is the power of MobileNetV2 model which performs very well with a small number of parameters and low memory.

5.1. Result analysis

In this subsection, we will analyze reached results by using plots and confusion matrices. We drew plots for all the implemented models, and we have divided the plots into two parts as shown in Figure 7(a), Figure 7(b), Figure 8(a), and Figure 8(b). Part (a) of the two figures contains Xception, ResNet-50, VGG16, and VGG19, and part (b) contains DenseNet121, MobileNetV2, Inception-ResNet-V2, and Inception-V3. According to the curves of the whole models, we observe that ResNet-50 and MobileNetV2 exceeded all of the other models in terms of accuracy during the first 100 epochs, which means that they have made a good initialization as shown in Figure 7. However, DenseNet121 and VGG16, which had the best results, did not start learning well. We notice that the stability of the curves started after epoch 300 when the DenseNet121 was able to surpass all of the other models and it continues the progression, followed by VGG16. Regarding to the validation of the loss metric in Figure 8, we note a stability during the first 200 epochs. However, the loss curves of the DenseNet121 and VGG16 are clearly decreasing which means, that if we use more than 500 epochs, these last two models, will have good performances compared to the expected results of the other models.

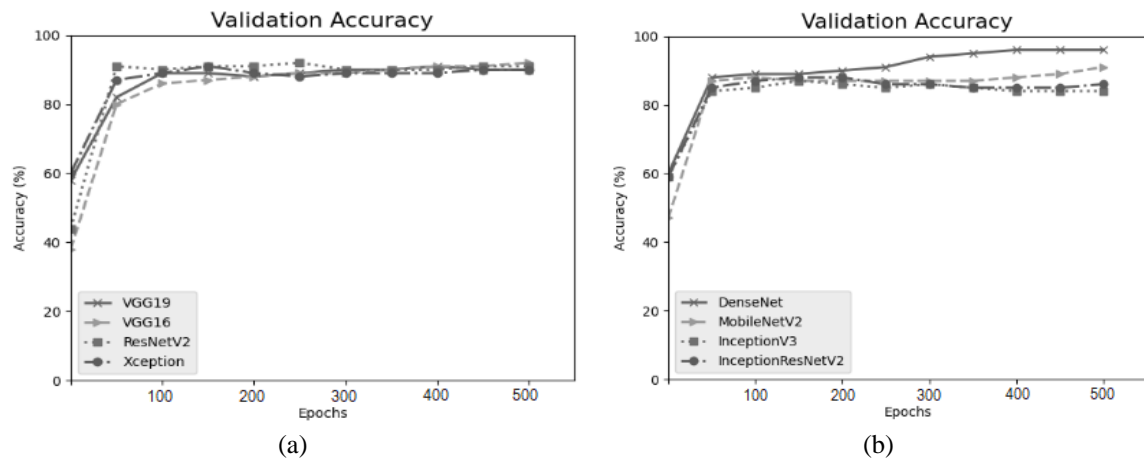


Figure 7. Accuracy of implemented CNN models; (a) Accuracy of Xception, ResNetV2, VGG16, and VGG19 CNN models; (b) Accuracy of InceptionV3, InceptionResNetV2, MobileNetV2, and DenseNet CNN models

Figure 9 shows the confusion matrices of the implemented CNN models. The confusion matrix draws some conclusions regarding the difficulty of the classification, and it also highlights some potential research opportunities. In the following, we will discuss the confusion matrix result of the DenseNet121 model. In total, we have used 247 images for validation, with 146 normal vehicles, and 101 emergency vehicles. In the “Normal Vehicle” class, we observe that DenseNet121 recognizes 143 images out of 146, in other words, it produced 143 correct predictions out of 146, with 97.94% of the entire normal vehicles. The analysis of misclassified images shows that missed vehicles contain advertisements, or have an abnormal shape compared to usual normal vehicles as shown in Figure 10. In “Emergency Vehicle” class, we observe that the correct prediction number is lower than that of “Normal Vehicle” class as illustrated in Figure 9, the DenseNet121 model produced 92 correct predictions out of 101, ie 91.08%. This is because some of the emergency vehicle images were taken from different distances, as well as different and difficult viewing angles. To avoid such cases, it is necessary to add several classes and other types of vehicle to the dataset.

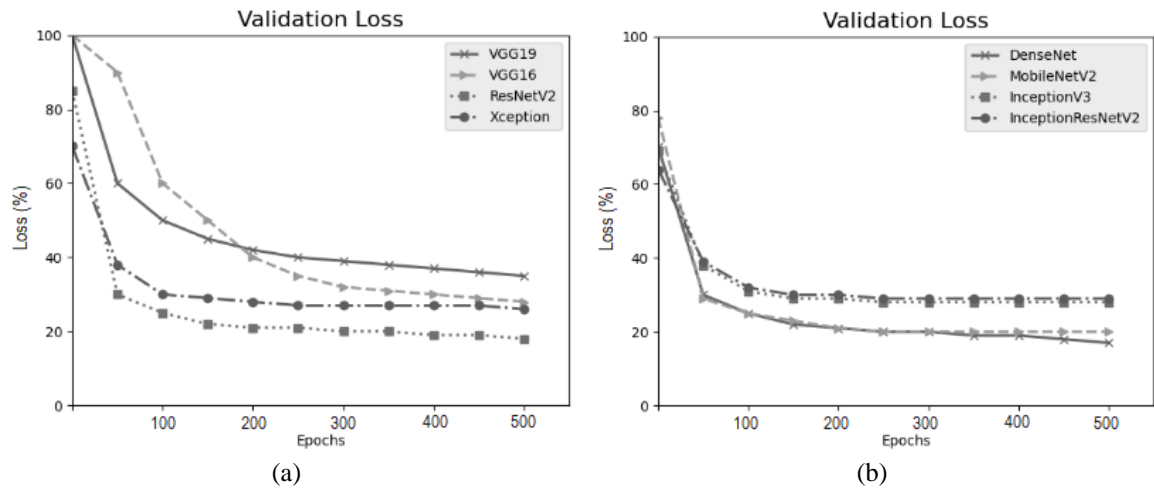


Figure 8. Loss of implemented CNN models; (a) Loss of Xception, ResNetV2, VGG16, and VGG19 CNN models; (b) Loss of InceptionV3, InceptionResNetV2, MobileNetV2, and DenseNet CNN models

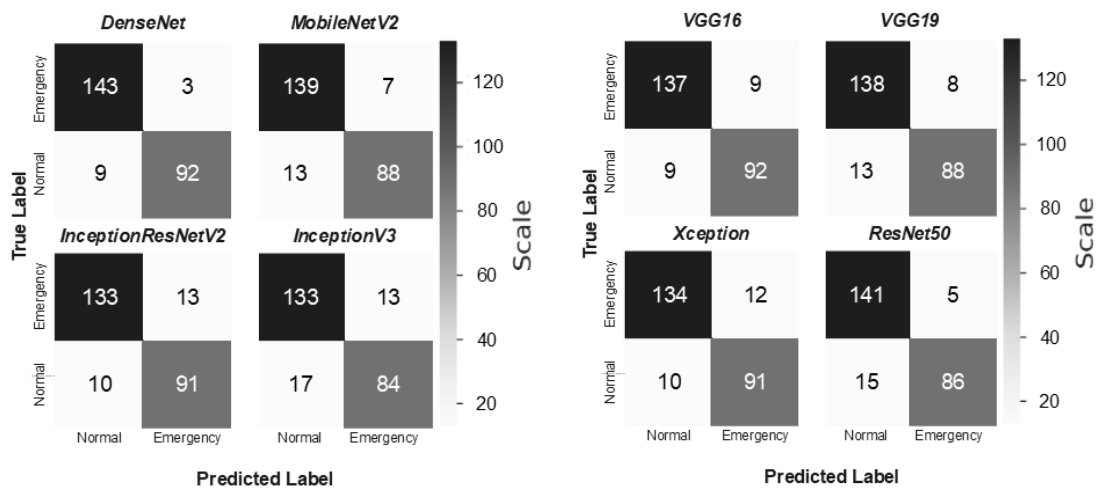


Figure 9. Confusion matrices of the implemented CNN models

5.2. Test and comparative results

The last part consists of testing all the implemented models on test images. After having trained and compiled each model on the dataset, we generated the evaluation scores for each model, we performed them on the test portion of the dataset using the TensorFlow and Keras prediction methods. Once we have a prediction, we use the Matplotlib library to display the image and its predicted class. We take as an example

the generated and trained DenseNet121 model, which had obtained the best score in terms of accuracy and F1 score. As shown in Figure 11, we have some examples of image classification into “Emergency Vehicle” and “Normal vehicle”. As we have already said, this is the first paper which makes the classification of emergency vehicle by using many CNN architectures.

There was a competition on the “Analytics Vidhya platform” on this topic, where the competitor have used ResNet-18 and ResNet-34 to classify emergency vehicles on Analytics Vidhya Emergency Vehicle dataset [37]. The average of accuracy of the two models is 94.22% [40]. Details about the used architectures are not mentioned, the only available information are the name of two architectures and the utilization of pre-trained weight models from Keras. We mention that our models are trained from scratch which needs much time for training. In addition, we emphasize that our best model performances surpass those of previous work.



Figure 10. Examples of normal vehicle images with advertisements and abnormal shapes [35]



Figure 11. Results of predicted classes using DenseNet121 model

6. CONCLUSION

In this paper, we have made a comparative study using eight famous CNN architectures to classify emergency vehicle images. The design and implementation of an efficient deep learning system, have been carried out to automatically classify emergency and normal vehicles in traffic scenes. First, we did a pre-processing on the Vidhya Emergency Vehicle dataset to unify the image sizes. We have added three layers to each CNN architecture, in particular, “GlobalAveragePooling2D” layer to reduce the dimensions of the input image and accelerate the training, “Dropout” layer to avoid overfitting, and finally, “Dense” layer where we put the number of output classes. Later, we made simulations of each architecture, and we notice that DenseNet121 is the most appropriate model in real-time emergency vehicle classification with an accuracy of 95.14%, a F1 score of 93.87%, and an average order memory of 27.5 MB. Therefore, reached results are very promising and will definitely give an important added value to applications that will use our best architecture for the classification of emergency vehicles. The experiments allow us to sort the classification architectures based on different criteria like accuracy, as well as memory, thus, researchers and developers can choose the appropriate and suitable architecture for their applications. As a perspective, we plan to improve accuracy scores and time processing, we also plan to use a hybrid approach to classify emergency vehicles based on image and siren sound.

REFERENCES




- [1] N. Buch, S. A. Velastin, and J. Orwell, “A review of computer vision techniques for the analysis of urban traffic,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Sep. 2011, doi: 10.1109/TITS.2011.2119372.

- [2] "DIRECTIVE 2010/40/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 7 July 2010 on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport (Text with EEA relevance)," 2010. Accessed: Feb. 19, 2021. [Online]. Available: <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32010L0040&qid=1613756510712&from=EN>.
- [3] M. M. Hasan, G. Saha, A. Hoque, and M. B. Majumder, "Smart traffic control system with application of image processing techniques," 2014, doi: 10.1109/ICIEV.2014.6850751.
- [4] A. Arinaldi, J. A. Pradana, and A. A. Gurusinga, "Detection and classification of vehicles for traffic video analytics," in *Procedia Computer Science*, Jan. 2018, vol. 144, pp. 259–268, doi: 10.1016/j.procs.2018.10.527.
- [5] D. Zapletal and A. Herout, "Vehicle re-identification for automatic video traffic surveillance," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Dec. 2016, pp. 1568–1574, doi: 10.1109/CVPRW.2016.195.
- [6] M. Ozuyisal, V. Lepetit, and P. Fua, "Pose estimation for category specific multiview object localization," Mar. 2010, pp. 778–785, doi: 10.1109/cvpr.2009.5206633.
- [7] S. H. Kim, J. Shi, A. Alfarrarjeh, D. Xu, Y. Tan, and C. Shahabi, "Real-time traffic video analysis using intel viewmont coprocessor," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, vol. 7813 LNCS, pp. 150–160, doi: 10.1007/978-3-642-37134-9_12.
- [8] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017, doi: 10.1109/TGRS.2016.2612821.
- [9] 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," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018, doi: 10.1109/TGRS.2017.2783902.
- [10] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 37–47, Mar. 2002, doi: 10.1109/6979.994794.
- [11] L. W. Tsai, J. W. Hsieh, and K. C. Fan, "Vehicle detection using normalized color and edge map," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 850–864, Mar. 2007, doi: 10.1109/TIP.2007.891147.
- [12] L. Zhuo, L. Jiang, Z. Zhu, J. Li, J. Zhang, and H. Long, "Vehicle classification for large-scale traffic surveillance videos using Convolutional Neural Networks," *Mach. Vis. Appl.*, vol. 28, no. 7, pp. 793–802, Oct. 2017, doi: 10.1007/s00138-017-0846-2.
- [13] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle Type Classification using a semisupervised convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2247–2256, Aug. 2015, doi: 10.1109/TITS.2015.2402438.
- [14] Z. Dong, M. Pei, Y. He, T. Liu, Y. Dong, and Y. Jia, "Vehicle type classification using unsupervised convolutional neural network," in *Proceedings - International Conference on Pattern Recognition*, Dec. 2014, pp. 172–177, doi: 10.1109/ICPR.2014.39.
- [15] C. M. Bautista, C. A. Dy, M. I. Mañalac, R. A. Orbe, and M. Cordel, "Convolutional neural network for vehicle detection in low resolution traffic videos," in *Proceedings - 2016 IEEE Region 10 Symposium, TENSYP 2016*, Jul. 2016, pp. 277–281, doi: 10.1109/TENCONSpring.2016.7519418.
- [16] H. Huttunen, F. S. Yancheshmeh, and C. Ke, "Car type recognition with deep neural networks," in *IEEE Intelligent Vehicles Symposium, Proceedings*, Aug. 2016, vol. 2016-August, pp. 1115–1120, doi: 10.1109/IVS.2016.7535529.
- [17] Y. Zhou, H. Nejati, T. T. Do, N. M. Cheung, and L. Cheah, "Image-based vehicle analysis using deep neural network: A systematic study," in *International Conference on Digital Signal Processing, DSP*, Jul. 2016, vol. 0, pp. 276–280, doi: 10.1109/ICDSP.2016.7868561.
- [18] X. Li and X. Guo, "A HOG feature and SVM based method for forward vehicle detection with single camera," in *Proceedings - 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2013*, 2013, vol. 1, pp. 263–266, doi: 10.1109/IHMSC.2013.69.
- [19] H. Razalli, R. Ramli, and M. H. Alkawaz, "Emergency vehicle recognition and classification method using HSV color segmentation," in *Proceedings - 2020 16th IEEE International Colloquium on Signal Processing and its Applications, CSPA 2020*, Feb. 2020, pp. 284–289, doi: 10.1109/CSPA48992.2020.9068695.
- [20] P. Gowtham, P. Eswari, and V. P. Arunachalam, "An Investigation approach used for pattern classification and recognition of an emergency vehicle," Dec. 2018, doi: 10.1109/ICSNS.2018.8573610.
- [21] "Convolutional Neural Networks (LeNet) - DeepLearning 0.1 Documentation | 2 D Computer Graphics | Cognitive Science." <https://www.scribd.com/document/333051443/Convolutional-Neural-Networks-LeNet-DeepLearning-0-1-Documentation> (accessed Feb. 19, 2021).
- [22] M. Boukabous and M. Azizi, "Review of learning-based techniques of sentiment analysis for security purposes," Springer, Cham, 2021, pp. 96–109.
- [23] A. Moubayed, M. Injadat, A. B. Nassif, H. Lutfiyya, and A. Shami, "E-learning: challenges and research opportunities using machine learning data analytics," *IEEE Access*, vol. 6, pp. 39117–39138, Jul. 2018, doi: 10.1109/ACCESS.2018.2851790.
- [24] "Traffic sign detection using convolutional neural network | by Sanket Doshi | Towards Data Science." <https://towardsdatascience.com/traffic-sign-detection-using-convolutional-neural-network-660fb32fe90e> (accessed Feb. 19, 2021).
- [25] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/j.patcog.2017.10.013.
- [26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, Accessed: Feb. 19, 2021. [Online]. Available: <http://www.robots.ox.ac.uk/>.
- [27] I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. El Fadili, "Toward a deep learning-based intrusion detection system for iot against botnet attacks," *IAES Int. J. Artif. Intell.*, vol. 10, no. 1, pp. 110–120, 2021, doi: 10.11591/ijai.v10.i1.pp110-120.
- [28] C. Kyrkou and T. Theodoridis, "EmergencyNet: Efficient aerial image classification for drone-based emergency monitoring using atrous convolutional feature fusion," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, pp. 1687–1699, 2020, doi: 10.1109/JSTARS.2020.2969809.
- [29] G. G. Dario *et al.*, "On the behavior of convolutional nets for feature extraction," *J. Artif. Intell. Res.*, vol. 61, pp. 563–592, 2018, doi: 10.1613/jair.5756.
- [30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Dec. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [31] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Oct. 2015, vol. 07-12-June-2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
- [32] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Nov. 2017, vol. 2017-January, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.
- [33] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv*, 2017.
- [34] M. Längkvist, L. Karlsson, and A. Loutfi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," *Pattern Recognit. Lett.*, vol. 42, no. 1, pp. 11–24, 2014, [Online]. Available: <http://arxiv.org/abs/1512.00567>.




- [35] I. Allaouzi, M. Ben Ahmed, and B. Benamrou, "An Encoder-Decoder model for visual question answering in the medical domain," *CEUR Workshop Proc.*, vol. 2380, no. September 2019, pp. 9–12, 2019.
- [36] C. Ye, C. Devaraj, M. Maynard, C. Fermüller, and Y. Aloimonos, "Evenly cascaded convolutional networks," *Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018*, pp. 4640–4647, 2019, doi: 10.1109/BigData.2018.8622196.
- [37] "JanataHack_AV_ComputerVision | Kaggle." <https://www.kaggle.com/shravankoninti/janatahack-av-computervision> (accessed Feb. 19, 2021).
- [38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res. 15 1929-1958*, vol. 299, no. 3–4, pp. 345–350, 2014, doi: 10.1016/0370-2693(93)90272-J.
- [39] Q. Ji, J. Huang, W. He, and Y. Sun, "Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images," *Algorithms*, vol. 12, no. 3, pp. 1–12, 2019, doi: 10.3390/a12030051.
- [40] "Emergency Vehicle Classification - PyTorch, ResNet | Kaggle." <https://www.kaggle.com/shravankoninti/emergency-vehicle-classification-pytorch-resnet> (accessed Feb. 19, 2021).

BIOGRAPHIES OF AUTHORS



Amine Kherraki    was born in Meknes, Morocco, in 1994. He received the B.S degree in School of Technology from Hassan First University at Berrechid, Morocco, in 2017. He received the M.S degree in Computer Science at the National School of Applied Science, Sidi Mohamed Ben Abdellah University, Fez, Morocco. Currently, He is Ph.D. candidate at Moulay Ismail University, Meknes, Morocco. His research interests include Deep Learning, Computer Vision, Business Intelligence, and Big Data. He can be contacted at email: amine.kherraki.9@gmail.com



Rajae El Ouazzani    received her Master's degree in Computer Science and Telecommunication by the Mohammed V University of Rabat (Morocco) in 2006 and the Ph.D. in Image and Video Processing by the High National School of Computer Science and Systems Analysis (Morocco) in 2010. From 2011, she is a Professor in the High School of Technology of Meknes, Moulay Ismail University in Morocco. Since 2007, she is an author of several papers in international journals and conferences. Her domains of interest include multimedia data processing and telecommunications. She can be contacted at email: elouazzanirajae@gmail.com