Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 220 (2023) 48–54

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 14th International Conference on Ambient Systems, Networks and Technologies (ANT)
March 15-17, 2023, Leuven, Belgium

Image Classification with Transfer Learning Using a Custom Dataset: Comparative Study

Houda Bichri[a,*], Adil Chergui[b], Mustapha Hain[a]

[a]AICSE lab, ENSAM Casablanca, Morocco
[b]ISSIEE lab, ENSAM Casablanca, Morocco

Abstract

Training a deep neural network is an expensive work because it is a time-consuming task and requires high computational power and usually a lot of dataset is needed to train a neural network which is not always available. These problems can be avoided by re-using the model weights from pre-trained models that were developed for standard computer vision benchmark datasets. In transfer learning, we basically try to exploit what has been learned in one task to improve generalization in another. We transfer the weights that a network has learned at 'task A' with a lot of available labeled training data to a new 'task B' that doesn't have much data. The knowledge of an already trained model is transferred to a different but closely linked problem throughout transfer learning; For example, if we trained a simple classifier to predict whether an image contains food, we could use the model's training knowledge to identify other objects such as drinks. This knowledge can be in various forms depending on the problem and the data. Top performing models can be downloaded and used directly, or integrated into a new model for our own computer vision problems. Thus, we speed up the training phase and improve the performance of our deep learning model even with a small dataset. In the present work, we propose to do the classification task using the three pre-trained models : MobileNet V2, ResNet50 and VGG19 and we will discuss the results using four key evaluating metrics. The dataset constituted is about a particular object to compare the performance of The models.

Keywords: Computer vision; Deep learning; Transfer learning; MobileNet V2; VGG19; ResNet50.

* Corresponding author. Tel.: +212648327553.
E-mail address: houda@bichri.ma

10.1016/j.procs.2023.03.009

1. Introduction

The use of deep neural networks on the classification task is the ideal choice, because of their better performance than other classification algorithms. But training a deep neural network may take days or even weeks to train on very large dataset.

Transfer learning has the benefit of decreasing time for a neural network model. It's a subfield of machine learning and artificial intelligence; it aims to apply the knowledge gained from one task to another different but similar task. Transfer learning boosts the performance of the new model even when it is trained on a small dataset.

In the literature, image classification with pre-trained models has been introduced to do different tasks such as identifying the display of daunting pictures on the internet [1], classifying non-carcinoma and carcinoma histopathology images of breast cancer [2], classifying melanoma images into benign and malignant [3] and classifying the chest X-ray images into normal, pneumonia and COVID-19 [4]. In [5], the authors suggested two approaches for the classification of papaya maturity status and the paper [6] proposes a new method for sports video scene classification with the particular intention of video summarization. The authors of [7] paper used the transfer learning to fine-tune pre-trained networks VGG16 and Wide Residual Networks for land use and land cover classification using the red-green-blue version of the EuroSAT dataset.

In the proposed work, the classification is achieved through three pre-trained models: MobileNet V2, ResNet50 and VGG19. The results are then discussed by using four evaluating metrics: accuracy, precision, recall and f1-score. The dataset is formed by 1200 color images collected from internet and divided into two folders: class_0 and class_1, class_0 contains 600 images of the object we want to classify and class_1 contains 600 images of objects that can be confused to the object in class_0 and other different objects, We used the Data Augmentation with the ImageDataGenerator tool to increase the amount of data.

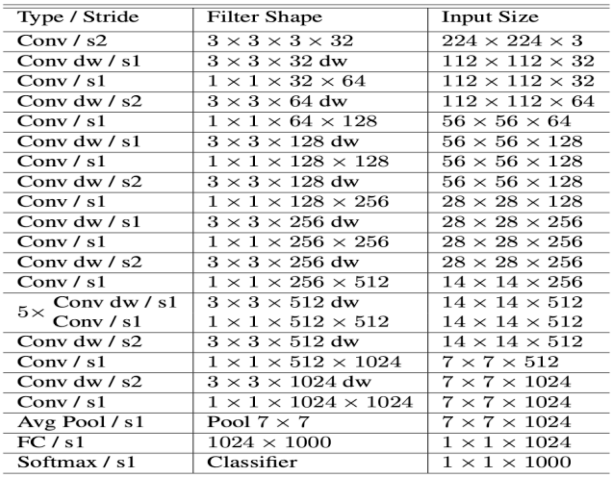
This paper is organized as follows: Section 2 provides a brief definition of MobileNet V1, MobileNet V2, VGG19 and ResNet50; Section 3 presents our custom dataset which divided into two classes; Section 4 describes on two subsections the preprocessing phase and the evaluation metrics; In section 5 we discuss and compare the results of the three pre-trained models and Section 6 is our conclusion.

2. Pre-trained models

2.1. MobileNet V1

Mobilenet V1 [8] is a class of CNN model (Convolutional Neural Network) which is designed to be used in mobile and embedded vision applications. The big idea behind MobileNet V1 is that convolutional layers, which are essential to computer vision tasks but are quite expensive to compute, can be replaced by so-called depthwise separable convolutions which reduces significantly the number of parameters when compared to the network with regular convolutions with the same depth in the nets.

A depthwise separable convolution is made from two operations: Depthwise convolution and Pointwise convolution. The depthwise convolution filters the input while while the pointwise convolution combines these filtered values to create new features (pointwise convolution is a 1×1 convolution layer). The MobileNet V1 body architecture is shown in Fig. 1 [8].



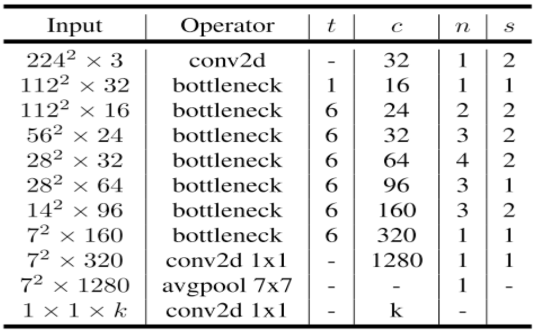
Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Fig. 1. MobileNet V1 body architecture.

2.2. MobileNet V2

MobileNetV2 [9] is the improved version of MobileNet V1, it uses inverted residual blocks with bottlenecking features. It has a drastically lower parameter count than the original MobileNet. MobileNetV2 is much faster than MobileNetV1 and it requires about 2 times fewer operations and has higher accuracy than the MobileNetV1.

Instead of using depthwise separable convolution as efficient building blocks, MobileNet V2 introduces two new features to the architecture: linear bottlenecks between the layers and shortcut connections between the bottlenecks. The MobileNet V2 body architecture is shown in Fig. 2 [9].



Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$28^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times k$	conv2d 1x1	-	k	-	

Fig. 2. MobileNet V2 body architecture.

2.3. ResNet50

ResNet50 [10] is a variant of ResNet model which has 48 Convolutional layers along with 1 MaxPool and 1 Average Pool layer. ResNet50 is a convolutional neural network that has 50 layers deep. ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers. It is an innovative neural network that was first introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their 2015 computer vision research paper titled 'Deep Residual Learning for Image Recognition'. The ResNet50 architecture is shown in Fig. 3[11].

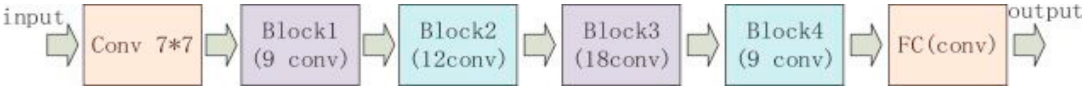


Fig. 3. ResNet50 architecture.

2.4. VGG19

VGG19 [12] proposed by Simonyan and Zisserman (2014) is a convolutional neural network that comprises 19 layers with 16 convolution layers and 3 fully connected to classify the images into 1000 object categories. VGG19 is trained on the ImageNet database that contains a million images of 1000 categories. It is a very popular method for image classification due to the use of multiple 3 × 3 filters in each convolutional layer. The architecture of VGG19 is shown in Fig. 4[13]. This shows that 16 convolutional layers are used for feature extraction and the next 3 layers work for classification. The layers used for feature extraction are segregated into 5 groups where each group is followed by a max-pooling layer.

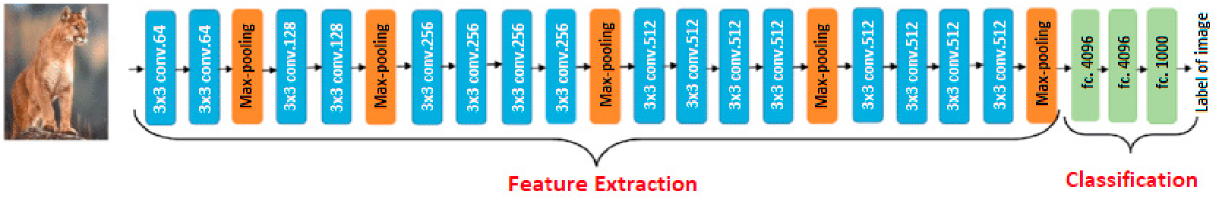


Fig. 4. VGG19 architecture.

3. Dataset

In this study, we have used a dataset containing 1200 of color images collected from internet. Our dataset is divided into 600 images classified "positive" and 600 images classified "negative", this second class contains images that may cause confusion with those of the first class and other images different from those of "positive" . Sample images of our dataset is presented on Fig. 5.

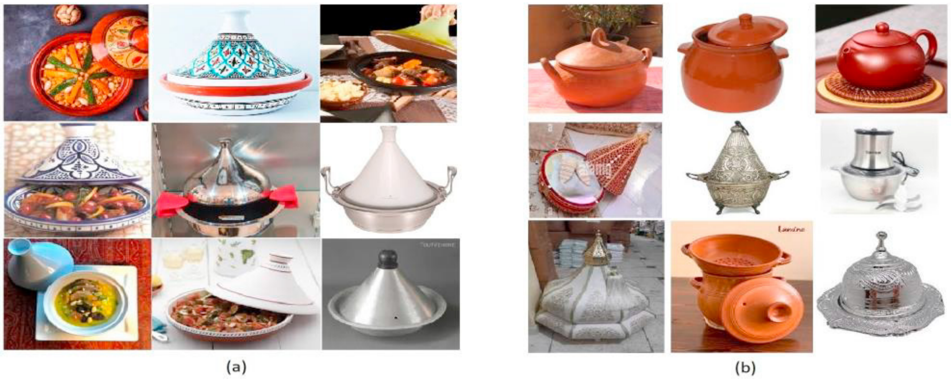


Fig. 5. Some images of our dataset: (a) class_0; (b) class_1.

4. Preprocessing and evaluation metrics

4.1. Preprocessing

When working with deep learning models, we must have a huge data which is not an easy task. For that, we have used the data augmentation process with the Keras ImageDataGenerator class to expand the size of our dataset. The images were rescaled to 224x224 pixel resolution to make images compatible with the pre-trained models.

The dataset is splitted into 80% for training phase and 20% for the testing one.

Adam function is the simple and time-efficient optimizer for deep neural networks. Thus, we have utilized it for the compilation process.

4.2. Evaluation metrics

There are several measures to exhibit the performance of classification results [14], we have considered the four following ones:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

TP: True Positive, TN: True Negative, FP: False Positive and FN: False Negative.

$$f_{1-score} = 2\left(\frac{Precision \times Recall}{Precision + Recall}\right) \tag{4}$$

Accuracy, calculated by Equation(1), is the base metric used for model evaluation, it describes the number of correct predictions over all predictions. In general, it’s good to obtain the accuracy more than 90%.

Precision is a measure of how the positive predictions made are correct (True Positives). The formula for it is obtained by the Equation(2).

Recall is a measure of how many of the positive cases the classifier correctly predicted, over all the positive cases in the data; it’s defined by the Equation(3).

The $f_{1\text{-score}}$ metric weights the two ratios: *Precision* and *Recall* in a balanced way. It’s calculated by the Equation(4). $f_{1\text{-score}}$ can range from 0 to 1, with 0 being the worst possible and 1 the best value. 1 represents that the model has perfectly classifies each observation into the correct class.

5. Results and discussion

We trained our model using Adam optimizer, on an Intel(R) Core™ i7 CPU with 16Go RAM, with a learning rate of 10^{-3} for 30 epochs and 20 as a value for the batch size hyperparameter. Precision, recall and f1-score obtained by the three models are shown in Fig. 6, The training loss and accuracy are represented in Fig. 7. The confusion matrix for the three models is shown in Fig. 8.

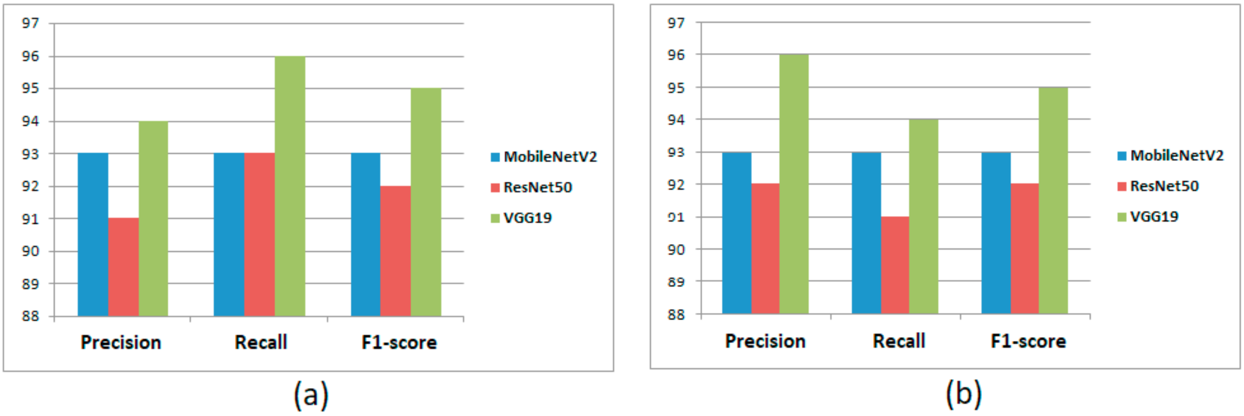


Fig. 6. Precision, Recall and f1-score obtained by the three models for: (a) positive; (b) negative.

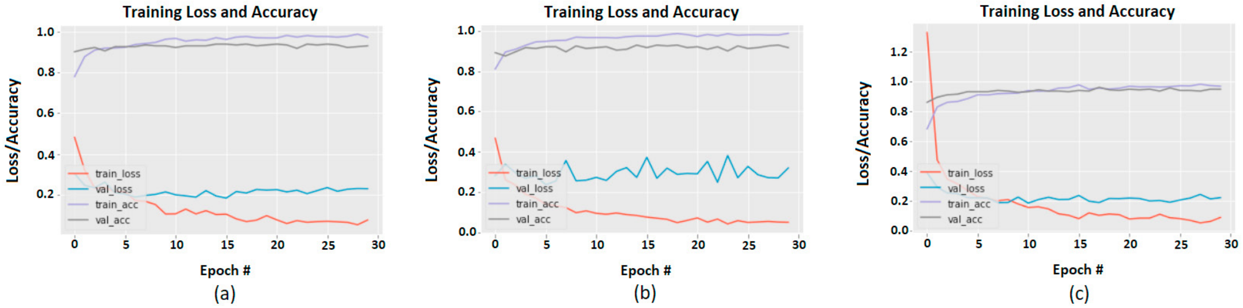


Fig. 7. Accuracy and Loss for: (a) MobileNet V2; (b) ResNet50, (c): VGG19.

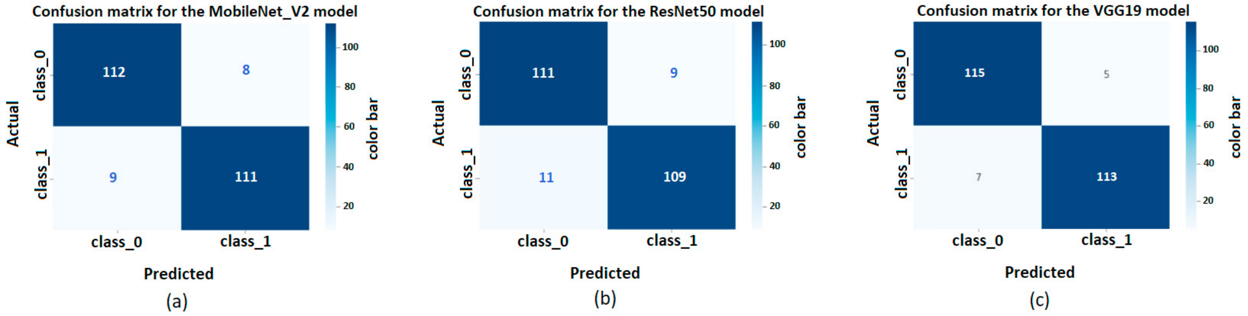


Fig. 8. Confusion matrices for: (a) MobileNet V2; (b) ResNet50; (c): VGG19.

For our dataset never seen by the three pre-trained models, we achieved the best accuracy (95%) by using the VGG19 network (7h05min52s), the Resnet50 and MobileNet V2 have realized successively 92% (2h03min04s) and 93% (19min35s). It can be observed from Fig. 6 that for the two different classes 'positive' and 'negative', the best precision, recall and f1-score are also obtained by VGG19.

If we consider the long execution time taken by VGG19 as an inconvenient, the second best accuracy, recall, precision and f1-score are obtained by using the MobileNet V2 network which takes 19min35s as an execution time. Looking at the plot of confusion matrix (Fig. 8), we can see that MobileNet V2 model accurately predicted 223 out of 240 total samples while ResNet50 achieved 220 exact predictions and VGG19 had 228 correct predictions.

6. Conclusion

In this paper, we make use of transfer learning to do classification task using the pre-trained models: MobileNet V2, ResNet50 and VGG19. For our specific dataset never seen by the three models, the VGG19 network have reached the highest classification accuracy (95%) and the highest f1-score although the execution time is 7h05min52s, while we obtained the accuracy of 92% using the Resnet50 in 2h03min04s and 93% while working with the MobileNet V2 network in just 19min35s.

In future works, we would explore the classification task by transfer learning using the same three pre-trained models : MobileNet V2, VGG19 and ResNet50 but with other datasets to compare the results and have some generalizations if possible.

In addition of binary classification treated in this study, we intend to do the multiple classification using our custom dataset to observe and analyze some specific metrics and see if the model classify our specific object by the same performance or there will be some changes.

References

[1] Gupta, J., Pathak, S., & Kumar, G. (2022). Bare skin image classification using convolution neural netowrk. International Journal of Emerging Technology and Advanced Engineering, 12(01).

[2] Hameed, Z., Zahia, S., Garcia-Zapirain, B., Javier Aguirre, J., & María Vanegas, A. (2020). Breast cancer histopathology image classification using an ensemble of deep learning models. *Sensors*, *20*(16), 4373.

[3] Indraswari, R., Rokhana, R., & Herulambang, W. (2022). Melanoma image classification based on MobileNetV2 network. *Procedia Computer Science*, *197*, 198-207.

[4] Rajpal, S., Lakhyani, N., Singh, A. K., Kohli, R., & Kumar, N. (2021). Using handpicked features in conjunction with ResNet-50 for improved detection of COVID-19 from chest X-ray images. *Chaos, Solitons & Fractals*, *145*, 110749.

[5] Behera, S. K., Rath, A. K., & Sethy, P. K. (2021). Maturity status classification of papaya fruits based on machine learning and transfer learning approach. *Information Processing in Agriculture*, *8*(2), 244-250.

[6] Rafiq, M., Rafiq, G., Agyeman, R., Choi, G. S., & Jin, S. I. (2020). Scene classification for sports video summarization using transfer learning. *Sensors*, *20*(6), 1702.

[7] Naushad, R., Kaur, T., & Ghaderpour, E. (2021). Deep transfer learning for land use and land cover classification: A comparative study. *Sensors*, *21*(23), 8083.

[8] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[9] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

[10] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[11] Li, B., & Lima, D. (2021). Facial expression recognition via ResNet-50. International Journal of Cognitive Computing in Engineering, 2, 57-64.

[12] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[13] Bansal, M., Kumar, M., Sachdeva, M., & Mittal, A. (2021). Transfer learning for image classification using VGG19: Caltech-101 image data set. Journal of Ambient Intelligence and Humanized Computing, 1-12.

[14] Ikechukwu, A. V., Murali, S., Deepu, R., & Shivamurthy, R. C. (2021). ResNet-50 vs VGG-19 vs training from scratch: a comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images. Global Transitions Proceedings, 2(2), 375-381.