A new convolutional neural network based on a sparse convolutional layer for animal face detection

Islem Jarraya1 · Fatma BenSaid1 · Wael Ouarda1,2 · Umapada Pal3 · Adel M. Alimi1,4

Received: 4 June 2021 / Revised: 7 January 2022 / Accepted: 9 February 2022 / Published online: 3 June 2022 © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
This paper focuses on the face detection problem of three popular animal categories that need control such as horses, cats and dogs. Existing detectors are generally based on Convolutional Neural Networks (CNNs) as backbones. CNNs are strong and fascinating classification tools but present some weak points such as the big number of layers and parameters, require a huge dataset and ignore the relationship between image parts. To be precise, to deal with these problems, this paper contributes to present a new Convolutional Neural Network for Animal Face Detection (CNNAFD), a new backbone CNNAFD-MobileNetV2 for animal face detection and a new Tunisian Horse Detection Database (THDD). CNNAFD used a processed filters based on gradient features and applied with a new way. A new sparse convolutional layer ANOFS-Conv is proposed through a sparse feature selection method known as Automated Negotiation-based Online Feature Selection (ANOFS). The ANOFS method is used as a training optimizer for the new ANOFS-Conv layer. CNNAFD ends by stacked fully connected layers which represent a
strong classifier. The fusion of CNNAFD and MobileNetV2 constructs the new network CNNAFD-MobileNetV2 which improves the classification results and gives better detection decisions. The proposed detector with the new CNNAFD-MobileNetV2 network provides effective results and proves to be competitive with the detectors of the related works with an Average Precision equal to 98.28%, 99.78%, 99.00% and 92.86% on the THDD, Cat Database, Stanford Dogs Dataset and Oxford-IIIT Pet Dataset respectively.

**Keywords** Animal face detection · ANOFS · Convolution neural network · MobileNetV2

### 1 Introduction

In a natural scene, face detection is a long-term objective for remote control and security needs. Using facial features, animal monitoring does not require a direct contact with the sensor and the animal will thus be at ease. Animal face detection can be used for many security applications that identify animal faces, gender/age detection and visual monitoring [49]. Ensuring safety for animals is an important task, particularly for breeders in many commercial fields such as horse race, livestock buying and selling as well as cow breeding. Face detection helps to fight against fraud and animal theft and to enforce health monitoring and traceability. Regretfully, it is still too difficult to detect animal faces given that face textures and shapes are grossly diverse. This is probably the reason for the small number of approaches.

The face detection task is still extremely difficult mainly because of the wide intra-class variation, illumination change, variable pose, complex background and partial occlusion. Despite these difficulties, recent research has achieved significant progress to resolve the interesting detection problems. The detection rate has reached nearly 90% of the face using boosting-based [40] and CNN (Convolutional Neural Network) based [48] approaches. Traditional human face detectors adopting handcrafted features have been replaced in many works by deep convolutional neural networks with the ability to extract discriminative facial features.

In the literature, the detection procedure usually includes three steps: block generation (multi-scale sliding windows or region proposals), face classification (in the backbone of the detector) and post-processing (non-maximum suppression and bounding box regression). In fact, the performance of face detectors is mainly influenced by the face classification network also known as the backbone. Duan et al. [7] discovered that the detector and the classifier of the general object detection have comparable performances using the same backbone. These explain that the designed backbone for the classification dataset is applied easily to the general object detection which gets an excellent mAP (mean Average Precision) score. Existing detectors, especially those for humans, have already taken on known CNN architectures as backbones. Convolutional neural networks (CNNs) are strong and fascinating classification tools. This is one of the reasons why Deep Learning is immensely popular and widely used for computer vision tasks. Given the rapid development, the questions that must arise are the following: Are CNNs flawless? Are they the best? In fact, there are different challenges during CNN training:

- Most network optimization algorithms (such as SGD and Adam) use the backpropagation method to set the layer weights. Backpropagation has yielded good training results in recent years but it is not a very effective way to learn as it requires a huge dataset for CNNs [15].
According to Hinton [35], pooling layers eliminate a great deal of information and ignore the relationship between image parts. In face detectors, for instance, just combining some features (mouth, eyes, face oval and nose) makes a face.

Face detectors use CNNs which represent a big number of parameters and layers. This leads to much training time and high computational complexity.

According to the above weak points of CNN, the main problems to be addressed in this paper are as follows:

- How to maintain the relationship between image parts in a pooling layer.
- How to achieve a good detection results using a small dataset for CNN training.
- If it is possible to create a new efficient CNN which represents a small number of parameters and layers.

The new challenging issue consists in creating a new convolutional neural network that effectively exploits the animal face characterization maintaining the relationship between image parts and using the smallest number parameters without the need for a huge dataset in order to obtain a robust and fast detector. To deal with the above-listed problems, a new CNN was proposed based on the ANOFS method for sparse feature selection which was introduced by BenSaid et al. [3, 5]. The different contributions in this paper are as follows:

- A new convolutional neural network CNNAFD (Convolutional Neural Network for Animal Face Detection) was proposed for binary classification (face/non-face) based on new sparse convolutional layer.
- The ANOFS method for sparse feature selection which has been adapted only on pattern recognition applications was employed in this paper for face detection.
- The ANOFS method for sparse feature selection was used as a training optimizer for a new sparse convolutional layer ANOFS-Conv instead of the traditional algorithms such as Adam and SGD.
- A new backbone CNNAFD-MobileNetV2 (fusion of CNNAFD and MobileNetV2 [36]) for the purpose of an efficient animal face detection.
- Making a new horse database called Tunisian Horse Detection Database (THDD). This database can contribute for the research community of the animal biometrics. To the best of our knowledge, this is the only dataset of public face image that is available for research on horse detection
- Extensive experimental studies showed that our proposed CNNAFD -MobileNetV2 backbone could get better performance than the backbones of the traditional detectors. In the experiments, we proved the efficiency of CNNAFD network compared to the other Convolutional Neural Networks with maintaining the relationship between image parts and using the smallest number of parameters without the need for a huge dataset.

The rest of this paper is organized as follows. Section 2 presents the related works of animal face detection. However, the proposed CNNAFD network and its layers are described in Section 3. While Section 4 is devoted to presenting CNNAFD training methodology. Section 5 shows the final proposed backbone CNNAFD-MobileNetV2 for animal face detection. Section 6 is devoted to the experimental test bed. Section 7 focuses on evaluation methodology and metrics. Section 8, however, presents the experimental results. Eventually the main conclusion of this paper which presents some possible future work is drawn in Section 7. All Abbreviations and Symbols are shown in Tables 1, 2, 3 and 4.
### Table 1  List of abbreviations

| Abbreviation          | Full words                                                                 |
|-----------------------|-----------------------------------------------------------------------------|
| Adam                  | Adaptive Moment Estimation                                                  |
| ANOFS                 | Automated Negotiation-based Online Feature Selection                       |
| ANOFS-Conv            | the ANOFS convolutional layer                                               |
| Anch                  | Anchors                                                                     |
| AP                    | Average Precision                                                          |
| BS                    | Batch Size                                                                  |
| CNN                   | Convolutional Neural Network                                                |
| CNNs                  | Convolutional Neural Networks                                               |
| CNNAFD                | Convolutional Neural Network for Animal Face Detection                      |
| CNNAFD-MobileNetV2    | The fusion of CNNAFD and MobileNetV2                                        |
| CSPNet                | Cross Stage Partial Network                                                 |
| DarkNet-53            | Dark Network with 53 layers                                                 |
| DenseNet-190          | Densely Connected Convolution Networks with 190 layers.                     |
| Ep                    | Epochs                                                                      |
| FC                    | Fully Connected layer                                                       |
| faster R-CNN          | faster Region-based Convolution Neural Networks                              |
| GB                    | GigaByte                                                                    |
| GPU                   | Graphic Processing Unit                                                     |
| Gradient-Conv         | gradient convolution layer                                                  |
| HOG                   | Histogram Oriented Gradient                                                 |
| HOOG                  | Haar of Oriented Gradients                                                  |
| Inception-v3          | Inception Network version 3                                                 |
| ImageNet              | Image database                                                              |
| IS                    | Input Size                                                                  |
| loss-F                | loss Function                                                               |
| MobileNetV2           | Mobile Network Version 2                                                    |
| mAP                   | mean Average Precision                                                      |
| Non-zero Pool         | Non-zero Pool (Pooling) layer                                               |
| NPV                   | Negative Predictive Value                                                   |
| OP                    | OutPut layer                                                                |
| Opt                   | Optimizer                                                                   |
| Pascal VOC            | PASCAL Visual Object Classes                                                |
| $PE_{train}$          | the modified perceptron                                                     |
| Pool                  | Pooling                                                                     |
| PPV                   | Positive Predictive Value                                                   |
### Table 2  List of abbreviations (continuation)

| Abbreviation | Full words |
|--------------|------------|
| RAND | the randomized feature selection algorithm |
| Relu | Relu layer |
| ReLU | Rectified Linear Unit |
| ResNet-50 | Residual Network with 50 layers |
| ResNet-101 | Residual Network with 101 layers |
| ResNext-101 | The next dimension of the Residual Network with 101 layers |
| ROC | Receiver Operating Characteristic |
| SGD | Stochastic Gradient Descent with momentum |
| SPC | Specificity |
| SSD | Single Shot multiBox Detector |
| SVM | Support Vector Machines |
| THDD | TunisianHorse Detection Database |
| TPR | True Positive Rate |
| VGG-16 | very deep convolutional networks with 16 layers |
| VGG-19 | very deep convolutional networks with 19 layers |
| YOLOv2 | You Only Look Once version 2 |
| YOLOv3 | You Only Look Once version 3 |
| YOLOv5 | You Only Look Once version 5 |

### Table 3  List of symbols

| Symbol | Definition |
|--------|------------|
| ACC    | Accuracy classification |
| ANOFSM | ANOFS Maps or the output map of ANOFS-Conv layer |
| AP     | Average Precision |
| B      | The number of the non-zero elements of the W vector |
| C      | The output vector of the gradient-Conv layer |
| CNNAFD | The resulting score from the CNNAFD |
| Fn     | False negative |
| Fp     | False positive |
| FPR    | False positive rate |
| F1     | F1-score |
| f      | Function |
| I      | The image intensity |
| i      | The filter number or the output map number of ANOFS-Conv layer |
| input  | The input value of a FC neuron |
| inputFusion | The input value of the fusion neuron |
Table 4  List of symbols (continuation)

| Symbol     | Definition                                                                 |
|------------|-----------------------------------------------------------------------------|
| IoU        | Intersection over Union                                                     |
| j          | The $C$ AND $ANOFSM$ feature number                                         |
| k          | The $PoolM$ feature number                                                 |
| Mag        | The gradient magnitude                                                     |
| MobileNetV2Output | The resulting score from the MobileNetV2                                   |
| N          | The total images number of a dataset                                        |
| n          | Number of ANOFS kernels (Weight vectors)                                   |
| output     | the output value of a FC neuron                                            |
| P          | The precision                                                              |
| PoolM      | The Non-zero Pool layer output                                             |
| p          | The number of feature in $C$ and in $ANOFSM$                                |
| R          | Recall, sensitivity or true-positive rate                                   |
| r          | The $PoolM$ features numbers                                                |
| t          | the iteration number                                                       |
| T          | The total number of iterations                                              |
| $T_n$      | True negative                                                              |
| TNR        | True negative rate                                                         |
| TPR        | Recall, sensitivity or true-positive rate                                   |
| $T_p$      | True positive                                                              |
| tan        | Tangent formula                                                            |
| W          | The weight vector or filter or kernel of the ANOFS-Conv layer               |
| w          | The neuron weight value of the OP layer                                     |
| $w_z$      | The weight value of the $z$ FC neuron                                       |
| X          | The coordinate value in the x axis of the image                             |
| Y          | The coordinate value in the y axis of the image                             |
| y          | The desired output                                                         |
| z          | The FC neuron number                                                       |
| $\theta$  | The gradient angle                                                         |
| $\sum$    | The sum function                                                           |
| $\sqrt{}$ | Square Root                                                                |

2  Related works

The number of works in this area is very limited due to the complication of the animal face detection task. The existing related works in this field are as follows:

Zhang et al. [50] proposed a set of Haar of Oriented Gradients (HOOG) to capture the texture and shape features on the animal head (such as cats, tigers, pandas, foxes and cheetahs). They used the SVM for classification and decision calculation. Using the Cat Database, they found a precision ($P$) equal to 95% and a recall ($R$) equivalent to 99.8% (Table 5).

Yamada et al. [46] proposed detecting dog and cat heads using edge-based features. They selected four directional features (Horizontal, Vertical, Upper Right and Upper Left).
Multimedia Tools and Applications (2023) 82:91–124

Table 5 Summary of different related works

| Ref       | Year | Method          | Database       | Results  |
|-----------|------|-----------------|----------------|----------|
| Zhang et al. [50] | 2011 | HOOG + SVM      | Cat Database   | P: 95% R: 99.8% |
| Yamada et al. [46] | 2011 | Edges + Multi-Layer classifier | Web images | Cat R: 85% Dog R: 90% |
| Mukai et al. [22] | 2018 | Viola-Jones HAAR + HOG | Cat Database Stanford Dogs Dataset | P: 75.7% R: 96.6% P: 90.8% R: 98.3% |
| Vlachynska et al. [42] | 2019 | Faster R-CNN (ResNet-101) | Columbia Dogs Dataset | AP: 98% |
| Tureckova et al. [41] | 2020 | YOLOv3 (DarkNet-53) | Columbia Dogs Dataset + Oxford-IIIT Pet Dataset | AP: 92% |

to detect the facial characteristics. They used a Multi-Layer Classifier for features classification. Yamada et al. performed their method on a set of cat and dog images from the web. The recall rate (R) was equal to 85% on the cat set and 90% on the dog set (Table 5).

Mukai et al. [22] focused on cat and dog face detection. They used the same Viola-Jones method and employed both the Haar and the HOG descriptors for feature extraction. Using 58 images from the Cat Database for the test, they found a recall (R) equal to 96.6% and a precision (P) equivalent to 75.7%. However, they achieved a recall (R) equal to 98.3% and a precision (P) equivalent to 90.8% using 60 images from the Stanford Dogs Dataset.

These traditional animal face detectors, adopting handcrafted features, have been replaced in the recent works by deep convolutional neural networks with the ability to extract discriminating face features (Table 5).

Vlachynska et al. [42] used the faster R-CNN proposed in [34] network with ResNet-101 for dog face detection. They found an Average Precision (AP) equal to 98% on the Columbia Dogs Dataset (Table 5).

Tureckova et al. [41] who used the YOLOv3 detector with DarkNet-53 for dog face detection noticed an Average Precision (AP) equivalent to 92% on the Columbia Dogs Dataset and the Oxford-IIIT Pet Dataset (Table 5).

The traditional approaches [22, 46, 50] adopted two distinct stages: handcrafted features and feature classification. However, theses methods are not effective and have been replaced in many works by deep convolutional neural networks (CNN) able to extract discriminative facial features. The proposed detectors [41, 42] have already taken on known CNN architectures (ResNet and DarkNet) as backbones. In addition, other detectors for animal detection [17] based on CNN were proposed. However, Convolutional Neural Networks (CNNs) ignore the relationship between image parts, represent a big number of parameters and layers and require a huge dataset for training. This leads to much training time and high computational complexity.
In the last decades, the development of facial recognition systems has been achieved using manually-noted databases in order to locate the facial area in the image. Overall, facial recognition systems have not been automated by facial detection systems [12, 13, 17, 21, 24–28, 31]. However, these methods allow high recognition rates but their systems lack automatic face detection, which is why the animal face detection system is important to ensure safety and security.

To deal with the above-listed problems, this paper introduces the Convolutional Neural Network for Animal Face Detection CNNAFD. The proposed network effectively exploits the animal face characterization to obtain a robust and fast detector with maintaining the relationship between image parts and using the smallest number of operations and parameters without the need for a huge dataset.

3 CNNAFD: Convolutional neural network for animal face detection

A small convolutional network and a fast training with a small database were taken into consideration in order to overcome the previously-mentioned challenges. Despite the large diversity of animal head textures, each animal species has a distinctive head form with a similar shape. Consequently, the gradient features were considered because they are invariant to photometric and geometric transformations. Furthermore, as Dalal and Triggs [6] discovered, fine orientation sampling, coarse spatial sampling and strong local photometric normalization make it possible to ignore the object movement as long as it maintains a roughly vertical position without a big transformation as is the case for animal face. The gradient features could thus be suited for animal face detection in images.

The Automated Negotiation-based Online Feature Selection (ANOFS) is a sparse online learning method introduced by BenSaid and Alimi [1–5]. The aim of this method is to select a small number of features for binary classification on small databases and thereby replace the traditional optimizers (such as SGD and Adam) by the ANOFS, which decreases the number of layer parameters and operations. Moreover, using ANOFS helps to find the most expressive features and extract the best representation of the animal face by keeping the relationship between the face parts during the training. In fact, this paper presents the proposed convolutional neural network CNNAFD.

As shown in Fig. 1, the CNNAFD network included five types of layers as follows:

– **INPUT**: This layer was used to keep the raw pixel values of the image.

![Fig. 1 CNNAFD: Convolutional Neural Network for Animal Face Detection](image)
– **Gradient-Conv**: The gradient convolution features that were connected to local regions in the input were computed by the gradient convolution (gradient-Conv) layer.

– **ANOFS-Conv**: The ANOFS convolution features were computed by the ANOFS convolution (ANOFS-Conv) layer connected to the previous convolution features.

– **Non-zero Pool**: A pooling layer (Non-zero Pool) was used to perform the downsampling operation along the produced ANOFS-Conv output.

– **FC**: The scores were computed through a Fully Connected (FC) layer using an OutPut layer (OP).

In this network, the layers had the shape of vectors instead of matrices.

### 3.1 Gradient-Conv layer

The gradient-Conv layer incorporated some constraints and achieved some deformations using local receptive fields, gradient features and spatial subsampling. Each unit in the output vector was connected to local regions of neighborhood pixels in the input image as usual (Fig. 1). The output vector $C$ was considered as a feature map produced by a local window of size $16 \times 16 \times 1$ which scanned over the plane of the image with a stride of 8 pixels. The same principle of the HOG descriptor proposed by Triggs and Dalal [6] was applied in this layer for feature calculation. Each window was divided into four sub-windows of $8 \times 8$. The gradient magnitude $Mag$ and the gradient angle $\theta$ were calculated for each sub-window in order to construct a normalized $36 \times 1$ vector over the whole window. The produced gradient vector represented a unit in the output feature map of the gradient convolution layer. The proposed measures of this layer produced the best results in our experiments. In fact, the improvement was insignificant whether the window size was smaller or bigger. Following Triggs and Dalal [6], both the gradient magnitude and the gradient angle were calculated using the intensity $I$ of each pixel through the following expressions ((1-4)). The gradient-Conv layer extracted each feature vector automatically and then sent it to the second convolutional layer.

\[
\begin{align*}
  f(Y) &= I(X, Y - 1) I(X, Y + 1) \\
  f(X) &= I(X - 1, Y) I(X + 1, Y) \\
  Mag(X, Y) &= \sqrt{f(Y)^2 + f(X)^2} \\
  \theta(X, Y) &= \tan^{-1}\left(\frac{f(X)}{f(Y)}\right)
\end{align*}
\]

### 3.2 ANOFS-Conv layer

In other research works, feature selection field has been well adapted resorting to many learning methods for pattern recognition. However, these are not devoted to object detection applications. Despite the efficiency of the feature selection methods, they are not accurate enough to process real world data using a small number of features. The ANOFS [1–5] method is a sparse feature selection method for binary classification that treats this limitation and integrates an automated negotiation process by a simple truncation algorithm (PEtrun) and the randomized feature selection algorithm (RAND) [2, 3, 5]. The ANOFS method which has been well adapted only on pattern recognition applications was employed in this paper for face detection. Instead of the traditional training optimizers, the ANOFS plays their role in this convolutional layer. The input sequence of the ANOFS is $(C_t, y_t)$ where $t = 1, \ldots, T$, $C_t$ is the gradient vector of $d$ dimension and $y_t$ refers to the desired
output. The ANOFS requires a classifier $W_t$ which represents the weight vector (kernel) of the ANOFS-Conv layer. $W_t$ contains at most $B$ non-zero elements (with $B > 0$ is a pre-defined constant). Thus, the classification of $C_t$ depends only on $B$ features and is made by a linear function. The weight vector $W_t$ would be updated in each trial and the learner would classify the instance $C_t$ using the automated negotiation process between the PEtrun and RAND. This scenario is repeated until $t = T$ when the learner is provided with full inputs of every training instance and resulting the final kernel $W = W_T$. Using different $n$ kernels $W_i$ with $i = 1..n$ as convolutional filters, various maps (vectors) $ANOFSM_i$ (ANOFS Maps) were constructed in this layer and thus relevant features could be selected. The ANOFS-Conv layer had a size equal to that of the gradient-Conv layer. Each neuron in the gradient-Conv layer had a unique relationship with the opposite neuron in the ANOFS-Conv layer, which reduced the number of parameters. The sparse ANOFS-Conv maps were produced when the ANOFS-Conv weights $W_i$ were multiplied by the gradient-Conv output $C$ with a linear activation function as shown in the following equation with $j = 1..p$ and $p$ is the number of features:

$$ANOFSM_j^i = f(C_j * W_j^i) = C_j^i * W_j^i$$ (5)

### 3.3 Non-zero pool layer

Each ANOFS-Conv kernel weight contained a big number of zeros. The pooling layer summed up the sparse convolutional output by eliminating all the values corresponding to zero and keeping only the relevant features that represent the 10% of the ANOFSM vector. Indeed, this elimination considerably reduces the number of features. Equation (6) presents the output ($PoolM$) of the Non-zero Pool layer calculation where $k = j = 1$ in the beginning, $j = 1..p$ and $p$ represents the number of features. Unlike the max pooling, Non-zero Pool maintained the relationship between image parts by keeping the same arrangement of values.

$$PoolM^k_i = \begin{cases} 
  \text{Nothing} & \text{if } ANOFSM_j^i = 0; \ j = j + 1 \\
  \text{ANOFSM}_j^i & \text{if } ANOFSM_j^i > 0; \ k = k + 1; \ j = j + 1 
\end{cases}$$ (6)

### 3.4 Stacked fully connected (FC) layers

The proposed CNNAFD was completed with stacked Fully Connected (FC) layers for classification. Once one FC layer classified a region (Window) as non-face, the region was rejected without going through the rest of the FC layers. In fact, each FC layer was connected to a Non-zero Pool vector and ended with an OutPut layer (OP). The stacked FC layers were applied using the tangent sigmoid transfer function and the stochastic gradient descent with momentum (SGD). Assuming that the Pool map $PoolM_i$ is composed of $p$ features, each neuron $k$ of the FC layer would have an input calculated as shown in (7) and an output as illustrated in Eq. (8) where $k = 1..r$ and $w_z$ refers to the weight value of the neuron $k$ on the FC layer. Each FC layer represented a weak classifier algorithm. Nonetheless, when combining their decisions, the stacked FC layers represented a strong classifier.

$$input_z = \sum_{k=1}^{p} PoolM^k_i * w_z$$ (7)

$$output_z = \frac{1}{1 + e^{-input_z}}$$ (8)
4 Proposed training methodology of CNNAFD

The training database was divided into $n$ overlapped sub-sets. The gradient-Conv vectors of all images were extracted from each sub-set to produce $n$ new sparse convolutional filters. In fact, Fig. 2 shows that each ANOFS-Conv filter was trained separately on a specific set of data using the Automated Negotiation-based Online Feature Selection (ANOFS) method.

**Fig. 3** Flow chart of the CNNAFD training process
This sparse method has been accurate enough to create a weight vector for binary classification. The same fraction of the selected features (10% of all dimensions and the rest of the weights were zeros) used by [3–5] were chosen for use. BenSaid et al. [3, 5] proved the effectiveness of the prediction performance of this fraction on several public large-scale benchmark datasets and thereby each pooling filter summarized the sparse ANOFS-Conv map by 10% of its real size. Both Algorithm 1 and Fig. 3 detail all the training process.

Algorithm 1 CNNAFD Training.

1: for Every sub-set do
2:   Anchor boxes extraction and annotation
3:   for Each anchor box do
4:     Calculate gradient-Conv map through the gradient-Conv layer
5:     Calculate and improve ANOFS-Conv filter $W$ through the ANOFS-Conv layer
6:   for Each gradient-Conv map do
7:     multiply the gradient-Conv map by the ANOFS-Conv filter $W$ (5)
8:     Calculate the ANOFS-Conv map using a linear activation function
9:   Move to the FC layer and the OP layer to improve their weights using the SGD optimizer and the sigmoid trasnfer fuction

5 CNNAFD-MobileNetV2 backbone: fusion of CNNAFD and MobileNetV2

Table 6 presents the CNNs which have appeared in the last 10 years. In order to keep a small number of parameters, the decisions of MobileNetV2 [36] which presents fewer parameters were merged with CNNAFD. For more efficiency, the addition of CNNAFD to MobileNetV2 was proposed to strengthen the detection process. This fusion gives the new backbone CNNAFD-MobileNetV2. As shown in Fig. 4, the fusion was applied using a neuron with the tangent sigmoid transfer function and the stochastic gradient descent with momentum (SGD). This neuron that represented the final FC layer of the network triggered the final detection decision. The input $\text{input}_{\text{Fusion}}$ and the decision result $y$ of the final FC layer were calculated as shown in Eqs. (9) and (10) based on the CNNAFD

| Network            | Parameter   |
|--------------------|-------------|
| VGG-16 [37]        | 138 million |
| VGG-19 [37]        | 143 million |
| DenseNet-190 [11]  | 40 million  |
| ResNet-50 [47]     | 25 million  |
| Inception-v3 [38]  | (> 23 million) |
| GoogLeNet [39]     | 5 million   |
| MobileNetV2 [36]   | 3.4 million |
| CNNAFD              | 1 million   |
| CNNAFD-MobileNetV2  | 4.4 million |
The detection system was based on the YOLOv2 [32] strategy as indicated in Fig. 5.

\[
\text{input}_{\text{Fusion}} = \text{CNNAFD}_{\text{Output}} \ast w + \text{MobileNetV2}_{\text{Output}} \ast w \tag{9}
\]

\[
y = \frac{1}{1 + e^{-\text{input}_{\text{Fusion}}}} \tag{10}
\]

The proposed regions were applied on the original image and used as inputs by CNNAFD. As stated above, the resulting decisions of MobileNetV2 and CNNAFD were merged by the
final FC layer to obtain the proposed detections. The detection process with the fusion is detailed on Algorithm 2 and Fig. 6. The final detection decision is given after applying the Non-Maximum Suppression algorithm (NMS).

**Algorithm 2** Fusion of CNNAFD and MobileNetV2 (CNNAFD-MobileNetV2) for detection.

1: for Each anchor box do  
2:     Calculate gradient-Conv map through the gradient-Conv layer  
3:     Calculate n ANOFS-Conv maps through the ANOFS-Conv layer  
4: for Each ANOFS-Conv map do  
5:     Calculate the FC output through the corresponding FC layer  
6:     Calculate the decision through the OP layer  
7:     if decision=0 then  
8:         Exit and and move to the next anchor box  
9:     end if  
10:    Consider the anchor box as a face  
11:   Consider the decision of the final FC as the anchor box score
6 Experimental test bed

In the experiments, the proposed CNNAFD-MobilNetV2 backbone was performed on four real-world databases:

- The THDD\(^1\) (Tunisian Horse Detection Database) included a set of horse images which were taken at different distances ranging from 1 to 2 meters relative to the horses. The collected database consisted of 703 horse images for the training and 400 images for the testing (see Fig. 7). The testing set contained 415 horse faces.
- The Cat Database \(^2\) involved 10,000 cat head photos. The photos were mainly downloaded from Flickr and were paired with data files that specified the position of each cat ears, eyes and mouth. Some examples of Cat Database are shown in Fig. 8.
- The Stanford Dogs Dataset \(^1\) included over 20,580 annotated images of 120 dog breeds. This dataset has been built using images from ImageNet for the fine-grained image categorization task (Fig. 9). Each image was annotated with an object class and a bounding box label. For accurate evaluation, manual face annotations were made as they did not exist for the whole face area in the Cat Database and in the Stanford Dogs Dataset.
- The Oxford-IIIT Pet\(^2\) Dataset proposed by Parkhi et al. \(^2\) contains 37 different breeds of cats and dogs with roughly 200 labeled images for each breed. Only the cat categories were used in this study. The total number of labeled cat images was 1188. Figure 10 presents some examples of dog images from the Oxford-IIIT Pet Dataset.

7 Evaluation methodology and metrics

In order to evaluate the animal face classification and detection, outputs were extracted from the images of the testing dataset. The classification rates, the Receiver Operating Characteristic (ROC) and the precision-recall curves were recorded using different metrics such as accuracy, precision, average precision, recall, sensitivity, specificity, negative Predictive

\(^1\)https://ieee-dataport.org/open-access/thdd?fbclid=IwAR3usKGJ8Mffq8tg8zgPTr_vft15qog5oQAEXYsDBe41HcBx11eUmnRRwGJ
\(^2\)https://www.robots.ox.ac.uk/~vgg/data/pets/
value and F1 Score. The accuracy ($ACC$) is calculated by dividing the total number of two correct predictions ($T_p + T_n$) by the total images number of a dataset ($N$):

$$ACC = \frac{T_p + T_n}{N} \quad (11)$$

The precision ($P$ or PPV) calculates the percentage of the detector predictions which are correct. Precision is determined when dividing the number of true positives ($T_p$) by the number of true positives plus the number of false positives ($F_p$):

$$P = \frac{T_p}{T_p + F_p} \quad (12)$$

Recall or sensitivity ($R$, $TPR$) measures how good the detector finds all the positives. Recall is calculated by dividing the number of true positives ($T_p$) by the number of true positives plus the number of false negatives ($F_n$):

$$R = \frac{T_p}{T_p + F_n} \quad (13)$$

Specificity is also called true negative rate ($TNR$ or SPC). It is calculated by dividing the number of correct negative predictions by the total number of negatives:

$$TNR = \frac{T_n}{T_n + F_p} \quad (14)$$
The negative predictive values (NPV) refer to the proportions of negative results in statistical tests which are true negative results. A strong result can be interpreted using this statistic. The NPV is defined as follows:

\[
    NPV = \frac{T_n}{T_n + F_n}
\]  

(15)

F1-score \((F_1)\) can be useful, but it is less frequently used than the other basic measures. F1-score is a harmonic mean of precision and recall:

\[
    F1 = \frac{2 \times P \times R}{P + R}
\]  

(16)

The Intersection over Union \(IoU\) ratio is computed as a ratio between the intersection and the union of the predicted bounding box and the ground-truth bounding boxes:

\[
    IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}}
\]  

(17)

Following the Pascal VOC challenge [8], every true positive detection has an \(IoU\) ratio equal or larger than 50%.

The precision-recall curve introduces the relation between the precision and the recall calculated for different detection thresholds. Consequently, the area under the precision recall curve presents the average precision \((AP)\) of the detector.

A Receiver Operating Characteristic curve (ROC curve) illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. A ROC curve plots the relationship between the true-positive rate \((TPR)\) of detection or \(T_P\) rate and the false positive rate \((FPR)\) of error or \(F_p\) rate at various threshold settings. The ROC curve is a graph with:

\[
    \text{The x axis showing } FPR = \frac{F_p}{F_p + T_n}
\]  

(18)

\[
    \text{The y axis showing } TPR = R
\]  

(19)

8 Experimental results

The experiments were performed on Nvidia GeForce 920MX GPU and a 20 GB memory. In order to get an accurate assessment, the proposed backbone CNNAFD-MobileNetV2 was tested for animal face classification and detection.
Motivated by face classification, our work began with face proposals whereby 1124 annotated faces and 10,000 non-face samples were collected from the testing part of THDD, 3000 annotated faces and 10,000 non-face samples were gathered from the testing part of the Cat Database, 1000 annotated faces and 10,000 non-face samples from the testing part of the Stanford Dogs Dataset as well as 1000 annotated faces and 10,000 non-face samples from the cat images of the Oxford-IIIT Pet Dataset. These databases were constructed for the evaluation of the binary classification.

Different Convolutional Neural Networks were evaluated and compared. The proposed CNNAFD-MobileNetV2 and CNNAFD were compared with other CNNs such as MobileNetV2 and GoogLeNet which have represented the smallest number of parameters and ResNet-50 which is widely used for human face detection \[ 20, 48\]. A transfer learning of the four pre-trained CNNs was made on the training set of the THDD, Cat Database and Stanford Dogs Dataset. The last FC layer of these CNNs was replaced with a new FC layer having two outputs (face/non-face). As shown in Table 6, the CNNAFD and CNNAFD-MobileNetV2 present a small number of parameters.

The field of face detection has been dominated by generic object detection methods. A slight difference was actually witnessed between the face detection and the generic object detection. Consequently, it was necessary to discuss and compare the proposed detector with object detection methods such as Faster R-CNN \[ 34\], SSD \[ 19\], YOLOv3 \[ 33\], YOLOv5 \[ 14\] which have also been used for human and animal face detection. The Detectron2 \[ 44\] for object detection has become one of the most widely adopted open source projects by Facebook Artificial Intelligence Research (FAIR). The SSD-MobileNetV2 \[ 23\] is a Single-Shot multibox Detection (SSD) network designed to perform real-time object detection on mobile devices. YOLOv5 \[ 14\] is the last improved version of the You Only Look Once (YOLO) detector using CSPNet \[ 43\]. Thus, there was a reason for comparing our detector with Detectron2 \[ 44\] using Faster R-CNN \[ 34\] with ResNext-101 \[ 45\] , YOLOv3-tiny \[ 41\] with Darknet-53 \[ 33\] , YOLOv5 \[ 14\] with Cross Stage Partial Network (CSPNet) \[ 43\] and SSD \[ 19\] with MobileNetV2\(^3\) presented by TensorFlow \[ 23\].

Tables 7 and 8 represent the experimental configuration of the different CNNs and detectors. YOLOv2 (CNNAFD-MobileNetV2) was made using the SGD optimizer (Opt), 9 anchors (Anch) , 90 epochs (Ep) and a Batch Size (BS) equal to 16. As shown in Table 8, most of the detectors use SGD and select a Batch Size of 16. Indeed, to make a fairer comparison between them, the same values of these parameters were chosen. According to the experiment, 9 anchors and 90 epochs were sufficient to detect animal faces using four datasets for training (HDD, Cat Database, Stanford Dogs Dataset and Oxford-IIIT Pet Dataset). The faces in these datasets are not very big and not very small and the disparity

\(^3\)https://blog.roboflow.com/training-a-tensorflow-object-detection-model-with-a-custom-dataset/
Table 8  Experimental configurations of different Detectors

| Detector     | IS  | Anch | Ep  | BS  | Opt  | GPU         |
|--------------|-----|------|-----|-----|------|-------------|
| SSD [23]     | 416 | -    | 80  | 16  | SGD  | Tesla K80   |
| YOLOv2 [32]  | 224 | 9    | 80  | 16  | SGD  | GeForce 920MX |
| YOLOv3 [33]  | 416 | 9    | 90  | 32  | Adam | Tesla K80   |
| YOLOv5 [14]  | 416 | 9    | 100 | 16  | SGD  | Tesla K80   |
| Detectron2 [44] | 416 | 9    | >1400 | 16  | SGD  | Tesla K80   |
| YOLOv2 (CNNAFD-MobileNetV2) | 224 | 9    | 90 | 16  | SGD  | GeForce 920MX |

between them is not great. For this reason, we do not need a large number which do not give a better result. A smaller number decreases the detection accuracy by losing the detection of some faces (Table 9). The same for the number of epochs, the system converges in epoch 90 and the accuracy of detection doesn’t improve even by adding more epochs (Table 10).

The selected feature vector for anchor generation on the MobileNetV2 was the ReLU layer (Bloc-13-expand). During the training of YOLOv2 with MobileNetV2, three types of data augmentation (horizontal flipping, scaling and jitter image color) were used. The YOLOv3, YOLOv5 and Detectron2 used other transformations such as jitter image color, translation, change scale, flip left-right, flip up-down, mosaic transformation, image shear and image rotation.

8.1 Results on THDD

Classification evaluation: CNNAFD-MobileNetV2 presented precisely the maximum accuracy of classification as illustrated in Fig. 11. Figure 12 shows the ROC curves of the different CNNs. CNNAFD-MobileNetV2 outperformed all the other CNNs since it obtained the biggest critical region. Table 11 presents competitive statistic rates of classification with the other CNNs. Thus, it could be concluded that the fusion of the two networks gets encouraging results.

Detection evaluations Animals and mainly horses have many face texture variations that led to more difficulties in detection. Table 12 shows comparative results with different research studies. It is noted that our detector was a competitor to the other detectors and achieved a high performance. Figure 13 displays some detection examples while Fig. 14 represents the performance of our detector in terms of precision and recall. The figure also reports the new detector had a big critical region which indicates effective results.

Table 9  A comparative study of the proposed detector F1 with different number of anchors

| Database                  | 6 Anch | 9 Anch | 12 Anch |
|---------------------------|--------|--------|---------|
| THDD                      | 98.78% | 98.79% | 98.78%  |
| Cat Database              | 99.44% | 99.66% | 99.41%  |
| Stanford Dogs Dataset     | 98.01% | 99.49% | 98.52%  |
| Oxford-IIIT Pet Dataset (Cat set) | 94.34% | 95.22% | 94.65%  |
Table 10  A comparative study of the proposed detector F1 with different Epochs

| Database                        | 80 Ep   | 90 EP  | 100 EP |
|---------------------------------|---------|--------|--------|
| THDD                            | 98.66%  | 98.79% | 98.54% |
| Cat Database                    | 99.27%  | 99.66% | 99.66% |
| Stanford Dogs Dataset           | 98.21%  | 99.49% | 99.30% |
| Oxford-IIIT Pet Dataset (Cat set)| 94.99%  | 95.22% | 95.08% |

Fig. 11  Classification results using GoogLeNet, ResNet-50, MobileNetV2, CNNAFD and CNNAFD-MobileNetV2 on THDD

Fig. 12  Comparison of ROC curves on THDD
### Table 11  Classification Results of different CNNs on THDD

| CNN           | ACC    | PPV    | TPR    | SPC    | NPV    | F1     |
|---------------|--------|--------|--------|--------|--------|--------|
| ResNet-50     | 96.25% | 95.61% | 65.93% | 99.66% | 96.30% | 78.04% |
| GoogLeNet     | 95.89% | 97.99% | 60.59% | 99.86% | 95.75% | 74.88% |
| MobileNetV2   | 98.11% | 87.52% | 94.84% | 98.48% | 99.41% | 91.03% |
| CNNAFD        | 98.09% | 86.19% | 96.62% | 98.26% | 99.61% | 91.11% |
| CNNAFD-MobileNetV2 | 98.34% | 92.72% | 90.66% | 99.20% | 98.95% | 91.68% |

### Table 12  A comparative study of horse face detection on THDD

| Method                           | AP    | Recall | Precision | F1    |
|----------------------------------|-------|--------|-----------|-------|
| Viola-Jones [30]                 | 70%   | 40.12% | 73.00%    | 51.78%|
| SSD (MobileNetV2) [23]           | 50.00%| -      | -         | -     |
| Detectron2 (ResNext-101) [44]    | 98.89%| 99.60% | 88.24%    | 93.57%|
| YOLOv5 (CSPNet) [14]             | 99.49%| 99.51% | 97.82%    | 98.65%|
| YOLOv3 (DarkNet-53) [33]         | 98.13%| 98.55% | 99.51%    | 99.02%|
| YOLOv2 (MobileNetV2)             | 97.81%| 97.83% | 99.75%    | 98.78%|
| YOLOv2 (MobileNetV2 +CNNAFD)     | 98.28%| 98.31% | 99.27%    | 98.79%|

**Fig. 13**  Some examples of horse face detection results
8.2 Results on cat database

Classification evaluation In Table 13 and Fig. 15, CNNAFD-MobileNetV2 had adequate statistical classification rates compared to other CNNs. CNNAFD-MobileNetV2 exhibited competitive results with the biggest critical region of ROC curves, outperforming all the other CNNs (Fig. 16).

Detection evaluation The proposed detector was evaluated on the challenging Cat Database. Newly-published methods were compared to our results as illustrated in Table 14. It was suggested that the same database partition of the related work could be taken. 5000 randomly-chosen images were used for training and 3000 ones for testing. CNNAFD-MobileNetV2 achieved a competitive performance compared with the other approaches by a recall rate of 99.80% and a precision rate of 99.53%. Figure 17 shows the precision-recall curves on the Cat Database. Our detector presented the biggest critical region which indicates the performance of the proposed detector. Figure 18, however, displays some detection examples which show the efficiency of our detector.

Table 13  Classification Results of different CNNs on Cat Database

| CNN         | ACC   | PPV   | TPR   | SPC   | NPV   | F1   |
|-------------|-------|-------|-------|-------|-------|------|
| ResNet-50   | 99.69%| 100%  | 98.67%| 100%  | 99.60%| 99.33%|
| GoogLeNet   | 99.94%| 99.90%| 99.83%| 99.97%| 99.95%| 99.87%|
| MobileNetV2 | 99.87%| 99.73%| 99.70%| 99.92%| 99.91%| 99.72%|
| CNNAFD      | 99.42%| 98.60%| 98.87%| 99.58%| 99.66%| 98.74%|
| CNNAFD-MobileNetV2 | 99.97%| 99.87%| 100%  | 99.96%| 100%  | 99.93%|
8.3 Results on stanford dogs dataset

**Classification evaluation** CNNAFD-MobileNetV2 competed with the other CNNs as it had a big critical region of ROC curves in Fig. 19. In precise detail, CNNAFD-MobileNetV2 presented effective and competitive statistic rates of classification precisely as illustrated in Fig. 20 and Table 15.

**Detection evaluation** The proposed detector on the challenging Stanford Dogs Dataset was evaluated (Fig. 21). Taking 3000 randomly chosen images for training and 100 images (as in the related work) for testing has been suggested in our work. Our detector achieved a recall rate of 99% and a precision rate of 99.98%, outperforming all recently detection methods (Table 16).
Table 14  A comparative study of cat face detection on Cat Database

| Method                          | AP   | Recall      | Precision | F1     |
|--------------------------------|------|-------------|-----------|--------|
| HOOG                           | -    | 99.80%      | 95.00%    | 97.34% |
| + SVM [50]                      |      |             |           |        |
| Edge features and contrast     | -    | 85.00%      | -         | -      |
| + M-L Classifier [46]          |      |             |           |        |
| HAAR and HOG features          | -    | 96.60%      | 75.70%    | 84.88% |
| + Cascade classifiers [22]     |      |             |           |        |
| HAAR-cascade [9]               | 74.00% | 73.86%   | 80.36%    | 76.97% |
| HAAR-cascade [10]              | 74.00% | 75.34%   | 85.94%    | 80.29% |
| SSD (MobileNetV2) [23]         | 63.00% | -         | -         | -      |
| Detectron2 (ResNext-101) [44]  | 93.70% | 93.88%   | 97.63%    | 95.71% |
| YOLOv5 (CSPNet) [14]           | 99.71% | 99.75%   | 99.54%    | 99.64% |
| YOLOv3 (DarkNet-53) [33]       | 99.31% | 99.63%   | 99.47%    | 99.54% |
| YOLOv2 (MobileNetV2)           | 99.72% | 99.77%   | 99.24%    | 99.50% |
| YOLOv2 (MobileNetV2 +CNNAFD)   | 99.78% | 99.80%   | 99.53%    | 99.66% |

Figure 22 shows a comparison between CNNAFD-MobileNetV2 and the related work detectors using recall-precision curves. The CNNAFD-MobileNetV2 presented the biggest critical region which indicates the validity of the proposed detector for the dog face.
Fig. 18  Some examples of cat face detection results on Cat Database

Fig. 19  Comparison of ROC curves on Stanford Dogs Dataset

Fig. 20  Classification results using GoogLeNet, ResNet-50, MobileNetV2, CNNAFD and CNNAFD-MobileNetV2 on Stanford Dogs Dataset
**Table 15** Classification Results of different CNNs on Dog Stanford Dataset

| CNN            | ACC    | PPV    | TPR    | SPC    | NPV    | F1     |
|----------------|--------|--------|--------|--------|--------|--------|
| ResNet-50 [47] | 99.69% | 100%   | 96.60% | 100%   | 99.66% | 98.27% |
| GoogleNet [39] | 99.75% | 99.39% | 97.90% | 99.94% | 99.79% | 98.64% |
| MobileNetV2    | 99.80% | 98.80% | 99.00% | 99.88% | 99.90% | 98.90% |
| CNNAFD         | 99.99% | 99.91% | 100%   | 99.99% | 100%   | 99.96% |
| CNNAFD-M       | 99.99% | 99.91% | 100%   | 99.99% | 100%   | 99.96% |
| MobileNetV2    |        |        |        |        |        |        |

**Fig. 21** Some examples of dog face detection results on Stanford Dogs Dataset

**Table 16** A comparative study of dog face detection on Stanford Dogs Dataset

| Approach                        | AP     | Recall | Precision | F1    |
|---------------------------------|--------|--------|-----------|-------|
| Edge features and contrast      | -      | 90.00% | -         | -     |
| + M-L Classifier [46]           |        |        |           |       |
| HAAR and HOG features           | -      | 98.30% | 90.80%    | 94.40%|
| + Cascading classifiers [22]    |        |        |           |       |
| SSD (MobileNetV2) [23]          | 79.00% | -      | -         | -     |
| Detectron2                      | 98.53% | 99.00% | 93.46%    | 96.15%|
| (ResNext-101) [44]              |        |        |           |       |
| YOLOv5 (CSPNet) [14]            | 97.98% | 98.00% | 98.99%    | 98.49%|
| YOLOv3 (DarkNet-53) [33]        | 97.98% | 98.00% | 98.00%    | 98.00%|
| YOLOv2 (MobileNetV2)            | 98.96% | 99.00% | 99.00%    | 99.00%|
| YOLOv2 (MobileNetV2 +CNNAFD)    | 99.00% | 99.00% | 99.98%    | 99.49%|

© Springer
Fig. 22  Comparison of precision-recall curves on Stanford Dogs Dataset

8.4 Results on Oxford-IIIT pet dataset

Classification evaluation: CNNAFD-MobileNetV2 outperformed all the other CNNs on Oxford-IIIT Pet Dataset (cat part). The proposed CNN had a big critical region of ROC curves in Fig. 23 and proved again its effectiveness for cat face classification. Moreover, CNNAFD-MobileNetV2 presented competitive statistic classification rates as illustrated in Fig. 24 and Table 17.

Detection evaluation  The proposed detector was evaluated on the cat part of the Oxford-IIIT Pet Dataset (Fig. 25). In this work, It has been suggested that the same training model
Fig. 24  Classification results using GoogLeNet, ResNet-50, MobileNetV2, CNNAFD and CNNAFD-MobileNetV2 on Oxford-IIIT Pet Dataset

Table 17  Classification Results of different CNNs on Oxford-IIIT Pet Dataset

| CNN            | ACC     | PPV     | TPR     | SPC     | NPV     | F1     |
|----------------|---------|---------|---------|---------|---------|--------|
| ResNet-50 [47] | 99.64%  | 96.57%  | 100%    | 99.60%  | 100%    | 98.26% |
| GoogleNet [39] | 99.95%  | 99.56%  | 99.91%  | 99.95%  | 99.99%  | 99.73% |
| MobileNetV2    | 99.93%  | 100%    | 99.29%  | 100%    | 99.92%  | 99.64% |
| CNNAFD         | 98.09%  | 85.93%  | 96.98%  | 98.21%  | 99.64%  | 91.12% |
| CNNAFD-MobileNetV2 | 99.97% | 100%    | 99.73%  | 100%    | 99.97%  | 99.87% |

Fig. 25  Some examples of cat face detection results on Oxford-IIIT Pet Dataset
of the Cat Database could be used for cat face detection on the Oxford-IIIT Pet Dataset. All labeled images were used for evaluation. In fact, there are 1188 images for the test. Our detector achieved a recall rate of 93.10% and a precision rate of 97.44%, competing with recent detection methods (Table 18 and Fig. 26).

8.5 Discussion

The proposed CNNAFD-MobileNetV2 backbone proved its performance on the last experimental part for classification and detection. Figure 27 presents the classification accuracy on the three databases (THDD, Cat Database and Stanford Dogs Dataset). It is very obvious that the accuracy of MobileNetV2 was almost equal to that of CNNAFD on THDD. Moreover, it is plain to see that the accuracy of MobileNetV2 was very large compared to the accuracy of CNNAFD for cats on the Cat Database and Oxford-IIIT Pet Dataset. On the other hand, it is very noticeable that the accuracy of CNNAFD was larger than that of MobileNetV2 on the Stanford Dogs Dataset. Therefore, according to the horse, cat and dog

---

Table 18  A comparative study of cat face detection on Oxford-IIIT Pet Dataset

| Method                        | AP   | Recall  | Precision | F1    |
|-------------------------------|------|---------|-----------|-------|
| Edge features and contrast   | -    | 85.00%  | -         | -     |
| + M-L Classifier [46]         |      |         |           |       |
| Detectron2 (ResNext-101) [44] | 95.78% | 83.10%  | 95.35%    | 88.80%|
| YOLOv5 (CSPNet) [14]          | 91.00% | 90.28%  | 97.71%    | 93.84%|
| YOLOv3 (DarkNet-53) [33]      | 90.81% | 90.10%  | 95.47%    | 92.70%|
| YOLOv2 (MobileNetV2)          | 90.32% | 90.48%  | 98.98%    | 94.53%|
| YOLOv2 (MobileNetV2 +CNNAFD)  | 92.86% | 93.10%  | 97.44%    | 95.22%|

---

Fig. 26  Comparison of precision-recall curves on Oxford-IIIT Pet Dataset
face results, it could not be concluded that one of the two networks was better than the other. However, it is apparent that the CNNAFD-MobileNetV2 network achieved success on the four datasets with the best accuracy. In fact, the CNNAFD-MobileNetV2 overcame the accuracy of the other CNNs by about 1.29% on the THDD with an accuracy equal to 98.38%, 0.22% on the Cat Database with an accuracy equal to 99.95%, 0.21% on Stanford Dogs Dataset with an accuracy equal to 99.99% and 0.57% on the Oxford-IIIT Pet Dataset with an accuracy equal to 99.97% (Tables 11, 13, 15 and 17). Indeed, it could be stated that the fusion of the two networks led to a cooperation between them. The two networks were thus complementary since the fusion reinforced this coherence. The last FC layer of fusion adjusted the output decisions of the two networks and took a weight value for each of them to produce the final decision.

The same thing is noticed for the detection process. The fusion of CNNAFD and MobileNetV2 improved the Recall and the Precision results by maximizing true detections and minimizing false detections. In fact, the addition of CNNAFD to MobileNetV2 reinforced the YOLOv2 detector and overcame the F1 of the other related works (Detectron2,YOLOv5, YOLOv3, YOLOv2) by about 5.14% on THDD with an Average Precision equal to 98.28%, 1.06% on Cat Database with an Average Precision equal to 99.66%, 1.58% on the Stanford Dogs Dataset with an Average Precision equal to 99.49% and 2.75% on the Oxford-IIIT Pet Dataset with an Average Precision equal to 95.22% (Tables 12, 14, 16 and 18).

The use of another learning algorithm through a sparse feature selection method (ANOFS) enhanced the information transmitted to the FC layers. In fact, the proposed sparse ANOFS-Conv layer and training methodology contributed to proper distinction between true and false detections (face/non-face). CNNAFD extracted the relevant features using ANOFS-Conv layer and then classified the candidate bloc using the stacked Fully Connected (FC) layers. However, unlike the other CNNs, maintaining the relationship between image parts on the Non-zero Pool layer kept as much information as possible by minimizing the number of operations and parameters. Consequently, the new sparse ANOFS-Conv layer and Non-zero Pool layer positively influenced decisions and brought detections closer to reality.
The addition of CNNAFD was proposed on the YOLOv2 detector with MobileNetV2 to reinforce the animal face detection process. In fact, the fusion of CNNAFD with the MobileNetV2 helped to increase the Precision rate and to decrease the number of false detections.

9 Limitations

Owing to the photos taken close to the pets in the used databases, the faces of the animals are not very small and the system easily detected them. Indeed, our detector cannot detect very small faces when the animal present in the photo is very far away. Figure 28 shows the detection results on some images loaded from the Oxford-IIIT Pet Dataset and from the web. These images contain cats far away.

This was due to the poor resolution of the facial area as well as the lack of the important details and information. However, the performance of the ANOFS method decreased as the information was reduced. The more information there is, the fairer the ANOFS does the classification. The problem of detecting small faces is actually a challenge in the backbone of the most popular detectors.

10 Conclusion

Traditional approaches based on handcrafted features are not effective. They have been replaced by many recent approaches that use deep convolutional neural networks (CNNs) with the ability to extract discriminative facial features. However, CNNs present different weak points such as ignoring the relationship between image parts, representing a large number of parameters and layers and requiring a huge set of data for training.

To avoid these problems, the CNNAFD was proposed in this work. In fact, the traditional training optimizer (such as ADAM and SGD) was replaced with the ANOFS method for sparse feature selection so as to create a new convolutional layer ANOFS-Conv. The ANOFS-Conv layer was connected to the Non-zero Pool layer to remove all null features while maintaining the relationship between the image parts. Indeed, this elimination reduced the number of features. CNNAFD ended by stacked fully connected layers that represented a strong classifier. The proposed CNNAFD succeeded to do the training with a small database and a small number of parameters that were equal to 1 million and layers equivalent to 5.

The detection system was based on the YOLOv2 strategy. The addition of CNNAFD to MobileNetV2 was proposed to strengthen the detection process. This fusion resulted in

Fig. 28  Some examples of images that are difficult to detect
the new backbone CNNAFD-MobileNetV2. The fusion was applied using a neuron that represented the final FC layer of the network and triggered the final detection decision. The resulting decisions of MobileNetV2 and CNNAFD were merged by the final FC layer in order to obtain the final detection decision. Despite this fusion, the proposed backbone kept the minimum parameters all the time compared to other CNNs, improved the classification results and gave better detection decisions.

The proposed system was evaluated on three known datasets such as Cat Database, Stanford Dogs Dataset and Oxford-IIIT Pet Dataset. Furthermore, our paper introduced a new Tunisian Horse Detection Database (THDD). The performance of the proposed CNNAFD network on three real-world databases has been demonstrated. It was very noticeable in the experimental part that the CNNAFD-MobileNetV2 outperformed the other CNNs by about 1.29% on the THDD with a classification accuracy equal to 98.38%, 0.22% on the Cat Database with an accuracy equal to 99.95%, 0.21% on Stanford Dogs Dataset with an accuracy equal to 99.99%, 0.57% on the Oxford-IIIT Pet Dataset with an accuracy equal to 99.97%.

Using the CNNAFD-MobileNetV2 backbone, the proposed detector outperformed the state of the art detectors by about 5.14% on the THDD with an Average Precision equal to 98.28%, 1.06% on Cat Database with an Average Precision equivalent to 99.66%, 1.58% on the Stanford Dogs Dataset with an Average Precision equal to 99.49% and 2.75% on the Oxford-IIIT Pet Dataset with an Average Precision equivalent to 95.22%.

In our future work, our objective will be to improve the CNNAFD-MobileNetV2 performance by exploring more discriminant filters and also to extend our proposed detector to other animals as well as to humans.

Acknowledgment  The authors would like to acknowledge that the THDD was created with the help of four Riding Clubs in Sfax, Tunisia: Equestrian Clubs of road Mahdia, road Tunis in Sakiet Ezzit, road Ain Km 17, and road Saltinia Km 17.

Funding  The research leading to these results has received funding from the Tunisian Ministry of Higher Education and Scientific Research under the grant agreement number LR11ES48.

Declarations

Financial interests  The authors declare they have no financial interests.

Conflict of Interests  The authors declare that they have no conflict of interest.

References

1. BenSaid F, Alimi AM (2016) A new online feature selection method for decision making problems with ultra-high dimension and massive training data. Journal of Information Assurance and Security, pp 293–301
2. BenSaid F, Alimi AsM (2016) Ofsf-bc: Online feature selection framework for binary classification. In: International conference of computer systems and applications, pp 1–8
3. BenSaid F, Alimi AM (2015) Anofs: Automated negotiation based online feature selection method. In: International conference on intelligent systems design and applications, pp 225–230
4. BenSaid F, Alimi AM (2016) Multi agent-learner based online feature selection system. In: IEEE international conference on systems, man, and cybernetics (SMC), pp 003652–003657
5. BenSaid F, Alimi AM (2021) Online feature selection system for big data classification based on multi-objective automated negotiation. Pattern Recogn 110:107–629
6. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision and pattern recognition (CVPR), pp 886–893
7. Duan J, Liao S, Zhou S, ZLi S (2016) Face classification: A specialized benchmark study. In: Chinese conference on biometric recognition, pp 22–29
8. Everingham M, Gool LV, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis (IJCV) 88:303–338
9. Howse J, Alekhin A (2020) Haarcascade frontal cat face. https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalcatface.xml. Accessed 13 April 2020
10. Howse J, Alekhin A (2020) Haarcascade frontal cat face extended. OpenCV, https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalcatface_extended.xml. Accessed 13 April 2020
11. Huang G, Liu Z, Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 2261–2269
12. Jarrahi I, Ouarda W, Alimi AM (2015) A preliminary investigation on horses recognition using facial texture features. In: IEEE International Conference on Systems, Man, and Cybernetics, pp 2803–2808
13. Jarrahi I, Ouarda W, Alimi AM (2017) Deep neural network features for horses identity recognition using multiview horses? face pattern. In: International conference on machine vision (ICMV), vol 10341, pp 52–56
14. Jocher G, Stoken A, Borovec J et al (2020) ultralytics/yolov5: v3.0. https://zenodo.org/record/3983579#.YKUgT6HjLIU. Accessed 13 August 2020
15. Joshi S, Verma DK, Saxena G, Paraye A (2019) Issues in training a convolutional neural network model for image classification. In: International conference on advances in computing and data sciences, pp 282–293
16. Khosla A, Jayadevaprakash N, Yao B, Li F-F (2011) Novel dataset for fine-grained image categorization : Stanford dogs. In: First workshop on fine-grained visual categorization, CVPR
17. Kishore T, Jha A, Kumar S, Bhattacharya S, Sultana M (2021) Deep cnn based automatic detection and identification of bengal tigers. In: Computational intelligence in communications and business analytics, pp 503–507
18. Korte T (2014) 10,000 cat pictures (for science). Center For Data Innovation. https://datainnovation.org/2014/08/10000-cat-pictures-for-science/. Accessed 21 August 2014
19. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: Single shot multibox detector. European Conference on Computer Vision 9905:21–37
20. Liu W, Hasan I, Liao S (2020) Center and scale prediction: A box-free approach for pedestrian and face detection. In: arXiv:1904.02948v3
21. Manoj S, Rakshith S, Kanchana V (2021) Identification of cattle breed using the convolutional neural network. In: International conference on signal processing and communication, pp 503–507. https://doi.org/10.1016/ICSPCS1531.2021.9451706
22. Mukai N, Zhang Y, Chang Y (2018) Pet face detection. In: Nicograph International (NicoInt), pp 52–57. https://doi.org/10.1016/NICOINT.2018.00018
23. Nelson J (2020) Training a tensorflow mobilenet object detection model with a custom dataset. Roboflow. https://blog.roboflow.com/training-a-tensorflow-object-detection-model-with-a-custom-dataset/. Accessed 09 February 2020
24. Ouarda W, Trichili H, Alimi AM, Solaiman B (2013) Combined local features selection for face recognition based on naive bayesian classification. In: International conference on hybrid intelligent systems, pp 240–245
25. Ouarda W, Trichili H, Alimi AM, Solaiman B (2014) Face recognition based on geometric features using support vector machines. In: International conference of soft computing and pattern recognition, pp 89–95
26. Ouarda W, Trichili H, Alimi AM, Solaiman B (2014) Mlp neural network for face recognition based on gabor features and dimensionality reduction techniques. In: International conference on multimedia computing and systems, pp 127–134. https://doi.org/10.1109/ICMCSC.2014.6911265
27. Ouarda W, Trichili H, Alimi AM, Solaiman B (2015) Bag of face recognition systems based on holistic approaches. In: International conference on intelligent systems design and applications (ISDA), pp 201–206
28. Ouarda W, Trichili H, Alimi AM, Solaiman B (2016) Towards a novel biometric system for smart riding club. Journal of information assurance and security (JIAS) 11:201–213
29. Parkhi OM, Vedaldi A, Zisserman A, Jawahar CV (2012) Cats and dogs. In: IEEE conference on computer vision and pattern recognition
30. Paul V, Jones MJ (2004) Robust real-time face detection. Int J Comput Vis 57:137?154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb
31. Rathor S, Kumari S, Singh R, Gupta P (2021) Two layers machine learning architecture for animal classification using hog and lbp. In: Proceedings of international conference on communication and artificial intelligence, pp 445–453
32. Redmon J, Farhadi A (2016) Yolo9000: Better, faster, stronger. arXiv:1612.08242
33. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv:1804.02767v1
34. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, vol 28, pp 91–99
35. Sabour S, Frosst N, Hinton G (2017) Dynamic routing between capsules. In: Proceedings of the 31st international conference on neural information processing systems, p 3859?3869
36. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: IEEE conference on computer vision and pattern recognition, pp 4510–4520
37. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations
38. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 2818–2826
39. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 2818–2826
40. Taunk P, Jayasri G, Priya J P, Kumar NS (2020) Face detection using viola jones with haar cascade. Test Engineering and Management, 83
41. Tureckova A, Holik T, Oplatkova ZK (2020) Dog face detection using yolo network. MENDEL Soft Computing Journal 26:2571–3701
42. Vlachynska A, Oplatkova ZK, Turecek T (2019) Dog face detection and localization of dogface’s landmarks. Advances in Intelligent Systems and Computing 764:465–476
43. Wang C-Y, Liao H-YM, Yeh I-H, Wu Y-H, Chen P-Y, Hsieh J-W (2019) Cspnet: A new backbone that can enhance learning capability of cnn. In: ArXiv: Comput Vis Pattern Recognit
44. Wu Y, Kirillov A, Massa F, Lo W-Y, Girshick R (2019) Detectron2. https://github.com/facebookresearch/detectron2
45. Xie S, Girshick R, Dollár P, Tu Z, He K (2016) Aggregated residual transformations for deep neural networks. arXiv:1611.05431
46. Yamada A, Kojima K, Kiyama J, Okamoto M, Murata H (2011) Directional edge-based dog and cat face detection method for digital camera. In: IEEE international conference on consumer electronics (ICCE), pp 87–88
47. Zagoruyko S, Komodakis N (2017) Wide residual networks. arXiv:1605.07146v4
48. Zhang B, Li J, Wang Y, Tai Y, Wang C, Li J, Huang F, Xia Y, Pei W, Ji R (2020) Asfd: Automatic and scalable face detector. arXiv:2003.11228
49. Zhang C, Zhang Z (2010) A survey of recent advances in face detection. Microsoft, https://www.microsoft.com/en-us/research/publication/a-survey-of-recent-advances-in-face-detection/. Accessed June 2010
50. Zhang W, Sun J, Tang X (2011) From tiger to panda: Animal head detection. IEEE Trans Image Process 20:1696–1708

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.