A Deep Learning Approach for Dog Face Verification and Recognition

Pacific Rim International Conference on Artificial Intelligence

PRICAI 2019: PRICAI 2019: Trends in Artificial Intelligence pp 418-430 | Cite as

- Guillaume Mougeot (1) Email author (gmougeot@sjtu.edu.cn)
- Dewei Li (1)
- Shuai Jia (1)

1. Department of Control Science and Engineering, Shanghai Jiao Tong University, , Shanghai, China

Conference paper
First Online: 23 August 2019

Part of the Lecture Notes in Computer Science book series (LNCS, volume 11672)

Abstract

Recently, deep learning methods for biometrics identification have mainly focused on human face identification and have proven their efficiency. However, little research have been performed on animal biometrics identification. In this paper, a deep learning approach for dog face verification and recognition is proposed and evaluated. Due to the lack of available datasets and the complexity of dog face shapes this problem is harder than human identification. The first publicly available dataset is thus composed, and a deep convolutional neural network coupled with the triplet loss is trained on this dataset. The model is then evaluated on a verification problem, on a recognition problem and on clustering dog faces. For an open-set of 48 different dogs, it reaches an accuracy of 92% on a verification task and a rank-5 accuracy of 88% on a one-shot recognition task. The model can additionally cluster pictures of these unknown dogs. This work could push zoologists to further investigate these new kinds of techniques for animal identification or could help pet owners to find their lost animal. The code and the dataset of this project are publicly available (https://github.com/GuillaumeMougeot/DogFaceNet (https://github.com/GuillaumeMougeot/DogFaceNet)).

Keywords

Dog face recognition Dog face identification Pet animal identification

1 Introduction

Nowadays, the main techniques for dog identification are collars, tattoos and microchip implants [12]. As these methods can be unreliable or harmful for the animals, zoologists and computer scientists have started exploring new strategies through the use of machine learning. Machine learning techniques have now become reliable enough for human face identification [27]. They can thus potentially be used on the more complex problem of dog face identification.

This project is mainly inspired by the recent development of deep learning in face verification: does these two pictures represent the same individual?, in face recognition: who is this individual? and in face clustering: group these pictures by individuals [22, 26].

However, animal face identification needs improvement. There is no publicly available dataset and little development in used techniques. This work aims thus at developing the existing methods by using recent deep learning research in face verification and recognition on a novel dataset of dog faces.

A brief review of the last development in human and animal identification is presented in Sect. 2. In Sect. 3, we first build a dataset using pictures we took and images retrieved from the Internet. The face pictures are then aligned and grouped into triplets. In Sect. 4, a ResNet-like model [9] coupled with the triplet loss [22] is designed to be trained on this dataset. In Sect. 5, the trained model is evaluated on a verification, a recognition and a clustering task. The main applications of this work could be to find stray animals or track free-ranging animals in smart cities.

2 Related Work

So far, research on animal biometrics recognition has mainly focused on cattle identification [2, 8, 13, 15, 17, 18] or endangered species identification [5, 6]. Kumar et al. [19] present a complete review on the latest development in this field. Dog faces are mainly subject to studies in landmark detection [29] and breed classification [20, 29]. A method for dog face identification was recently proposed in [16]. In this paper they classify the most recent development in animal biometrics identification into four categories: muzzle points, iris pattern, retina vascular and face images. As human fingerprints identification, muzzle points identification is a reliable way to identify individual animals but in order to extract features, pictures have to be in a high resolution and well exposed or muzzles prints should be retrieved using an appropriate scanner. In both cases, data collection on animals is difficult and time consuming. Iris pattern and retina vascular identification suffer from the same problems. On the other hand, a lot of high resolution dog faces can now be found on the Internet. The work of Kumar et al. [16] on dog faces is inspired by classical machine learning methods for face recognition like Fisherfaces. Their image dataset is composed of one face per individual animal and data augmentation is then used to create new representations of these individual animals. Their model reached a top-1 accuracy of 82% on a *closed-set* of test data.

A more realistic goal would be to consider several actual pictures per individual animal taken in the wild. A trained model could then to be tested on an *open-set* of test data. These changes increase the difficulty of the problem significantly. The same difficulty arises for human face identification. To deal with it, researchers have created deep learning methods. As human face recognition represents a key domain in computer vision, many related research results can be found [1, 3, 4, 26, 27]. The work of Schroff et al. in [22] illustrates the importance of this new types of methods: the previously published error rate on the Labeled Faces in the Wild (LFW) dataset [11] has been reduced by 30%.

The determinant part in face identification method based on deep learning is the loss definition for which a lot of important improvements have been developed during the past few years [7, 21, 22, 28, 30]. Among these different losses, the triplet loss generates better results on a small *open-set* of data. Even though CosFace [28] or ArcFace [7] should normally work better regarding their accuracy on standard human face datasets as LFW, they rapidly overfit when trained on small datasets with a low level of regularity such as dog faces.

In order to train a network with the triplet loss, triplets of pictures have to be defined. A triplet is composed of an anchor picture x_a, a positive picture x_p and a negative picture x_n. The anchor picture is a picture of a randomly chosen individual inside the dataset. The positive picture is a different picture of the same individual. The negative picture is a picture of another individual. For an image *x* the network *f* will generate an embedding vector *f*(*x*). The goal of the model is to ensure that the Euclidean distance between the anchor embedding vector $f(x_a)$ and the positive image one $f(x_p)$ is lower than the Euclidean distance between the anchor embedding vector and the negative image one $f(x_n)$. In order to increase the margin between the different classes, a constant α is added in the previous inequality. This can be written as follows:

$$||f(x_a) - f(x_p)||^2 + \alpha < ||f(x_a) - f(x_n)||^2$$

(1)

The final objective of the network is then to minimize the following loss L defined as the triplet loss in [22] (*N* is the number of triplets per batch):

$$L = \sum_{i=0}^{N} max(||f(x_a^{(i)}) - f(x_p^{(i)})||^2 - ||f(x_a^{(i)}) - f(x_n^{(i)})||^2 + \alpha, 0)$$

(2)

Regarding the model selection, a VGG network [23] was used in Schroff et al. [22] paper. This network is a standard for image classification. Nevertheless, it has been improved by He et al. in [9]. Their created structure, called ResNet, has five time less parameters and a better accuracy on the ImageNet classification task. More complex networks have lately been developed such as InceptionNet [25], SENet [10] or NASNet [31]. These methods are the current state-of-the-art regarding image classification.

3 Dataset Creation and Pre-processing

3.1 Data Collection

As no open source dog face datasets are available, a new one is created. This dataset is a collection of dog face pictures we took ourselves and pictures found on the Internet. The main contributions on this dataset can be found on non-profit pet adoption websites: Streunerhilfe[1], Tiko[2], Pfotenhilfe[3], La SPA[4], Tieronline[5] and Animal-happyend[6].

In order to increase the network accuracy, only dogs with more than 5 pictures are selected. The final dataset contains 3148 pictures of 485 dogs.

3.2 Data Pre-processing

As the dataset is small, feeding the model with only raw images would give a bad accuracy. Three labels are thus manually added on the images: the left eye, the right eye and the muzzle, as shown on the left part of Fig. 1.

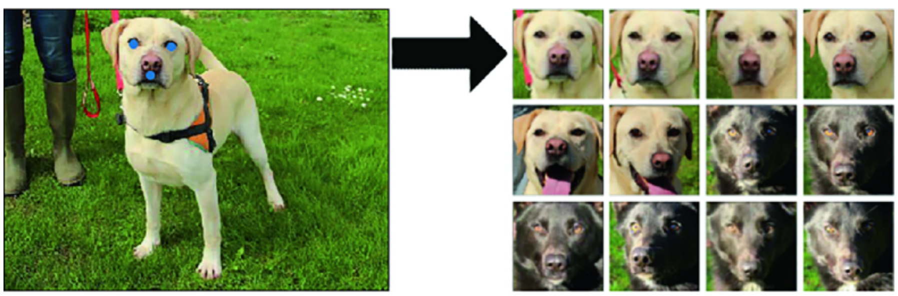

Fig. 1.

Data pre-processing. Left: a raw image with its corresponding labels, **Right:** a set of aligned images after similarity transform.

Dog faces are then aligned using the position of the eyes. Face alignment creates regularities in images and facilitates dog face parts automatic detection. A similarity transformation, i.e. a translation, a rotation and a re-sizing, is used to align the raw images. The eyes are horizontally constrained in the upper third of the picture. It creates a strong similarity between pictures and leaves enough space in the bottom part for the dog muzzle. The right eye of the dog is placed in position $(0.7/2.4 \times new\ height, 0.7/2.4 \times new\ width)$ and the left eye in position $(0.7/2.4 \times new\ height, 1.7/2.4 \times new\ width)$. It is a good compromise between reducing the background regions and ensuring enough space for long dog noses to appear on the pictures. The pictures are finally re-sized to $(new\ height, new\ width, depth) = (104 \times 104 \times 3)$ pixels. Figure 1 represents an example of different aligned dog faces.

To properly train the model, the dataset is split into a training set and a testing set. There are two main methods to define the testing set: either to create what is called a *closed-set* or to create an *open-set*. A closed-set is a set of unknown images of known dogs, which means that the network has already seen pictures of these dogs during the training stage. An open-set is a mixture of unknown dog pictures, the network sees these dogs for the first time during the testing stage. The open-set problem is a harder problem to solve and closer to a real life problem. The testing set is thus defined as an open-set. If an application has to be developed later, a network that manages to correctly identify dog faces from an open-set of images will be a necessity.

The training set is composed of 2850 pictures of 437 dogs, and the testing set contains 298 pictures of 48 dogs. As specified above, there is no intersection between these two sets of dogs.

In order to prevent overfitting, to increase the size of the dataset and to foster generalization, the training set is augmented by slightly zooming into the pictures ($zoom\ range = 0.1$), by rotating them ($rotation\ range = 8^\circ$) and by shifting their channels ($channel\ shift\ range = 0.1$).

The images are finally grouped into triplets following the procedure defined in Sect. 2. The final dataset, ready for training, contains: 10000 triplets (30000 pictures) of augmented dog faces for the training set and 1000 triplets (3000 pictures) of non-augmented dog faces for the testing set.

4 Model Definition and Training

4.1 Model Definition

Because of the novelty of the problem a new model has to be designed to solve it. This model is a deep convolutional neural network inspired by the recent development on this type of structure. Many different VGG-like [23] and ResNet-like [9] models have been trained on the dataset. The final model has the best performance on both verification and recognition task. It is described on Fig. 2. It is mainly inspired from the ResNet [9] structure to which a lot of dropout layers [24] are added. The network takes an image x of size $(104 \times 104 \times 3)$ as input and outputs an embedding vector $f(x)$ of size 32. In order to scale the Euclidean distance in the loss defined in Eq. (2), the output vector has to rely on the unit hypersphere, hence $||f(x)||^2 = 1$.

To design this model the utmost attention is paid to prevent overfitting. In order to extract a sufficient number of features from pictures, the network has to be as deep as possible. The residual layers allow to design such a deep network: it prevents the gradient from vanishing during back-propagation. The final model contains 92 layers for a total of 5.8 million of parameters. However, a too deep model would rapidly overfit to the small training set and its accuracy on the testing set would decrease. The model also contains many dropout layers to create a sparse network during training. Each of these dropout layers will set three fourths of the previous output to zero. This technique thus strongly prevents the network to specialize too much.

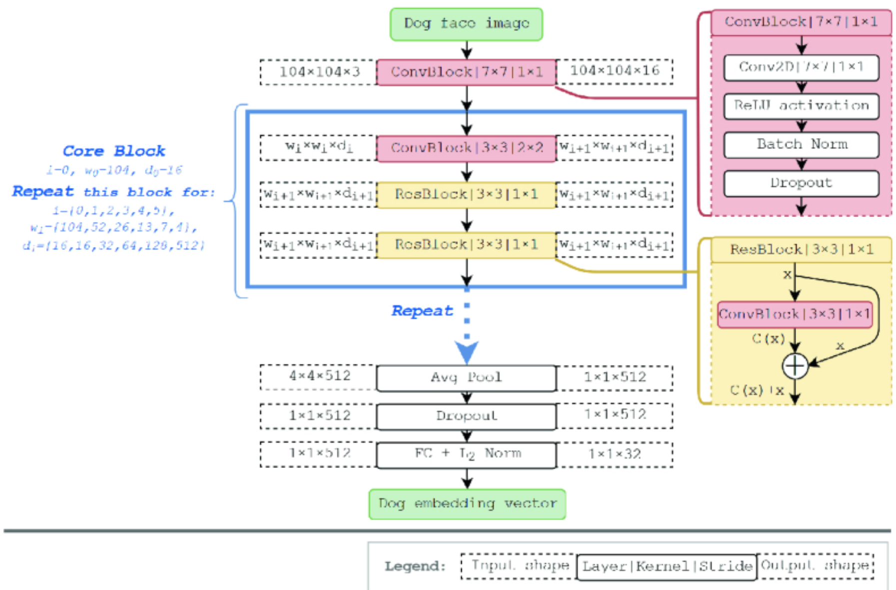

Fig. 2.

Model definition. The model takes a dog face image as input and outputs its corresponding embedding vector. The core block (inside the blue box) is sequentially repeated 6 times. The *ConvBlock* and the *ResBlock* descriptions are presented on the right side of the figure. (Color figure online)

4.2 Training

The model is trained using a GPU Tesla K80 with a margin $\alpha = 0.3$. However, the model rapidly overfits if it is only trained on the dataset defined in Sect. 3. In order to fix such a problem we could have re-generated augmented triplets. Although this solution can temporarily work, it rapidly overfits again. A better solution is described in [22]: to generate so-called *hard triplets*. It consists in taking a subset of dogs, and then for each picture x_a:

- among pictures of the same class, find the most different positive picture x_p from x_a, that is, compute $argmax_{x_p}||f(x_a) - f(x_p)||^2$
- among pictures of the other classes, find the most similar negative picture x_n from x_a, that is, compute $argmin_{x_n}||f(x_a) - f(x_n)||^2$

These hard triplets can be generated either online (during training) or offline (every n epochs). We choose to use offline training for computation power reasons: the online generation needs a minimal number of images per classes, which leads to a too big sized batch for our available computational power. Offline training consists in re-generating hard augmented triplets every $n = 3$ epochs, what will be called a *cycle*. The model is finally compiled using the Adam optimizer [14]. To improve the convergence, the learning rate is scheduled during training as shown in Table 1.

Table 1.

Learning rate scheduling.

	Cycles	Epochs	Learning rates
	13	39	0.001
	4	12	0.0005
	4	12	0.0003
	2	6	0.0001
Total	23	69	-

The accuracy is monitored on the open test set defined in Sect. 3. In this set, a triplet is considered as correct if it respects the hard condition defined by inequality (1). As shown on Fig. 3, the model loss decreases during a cycle and increases after hard augmented triplets generation. If the model is trained more than 70 epochs, it starts overfitting: the validation loss will increase as the training loss will decrease.

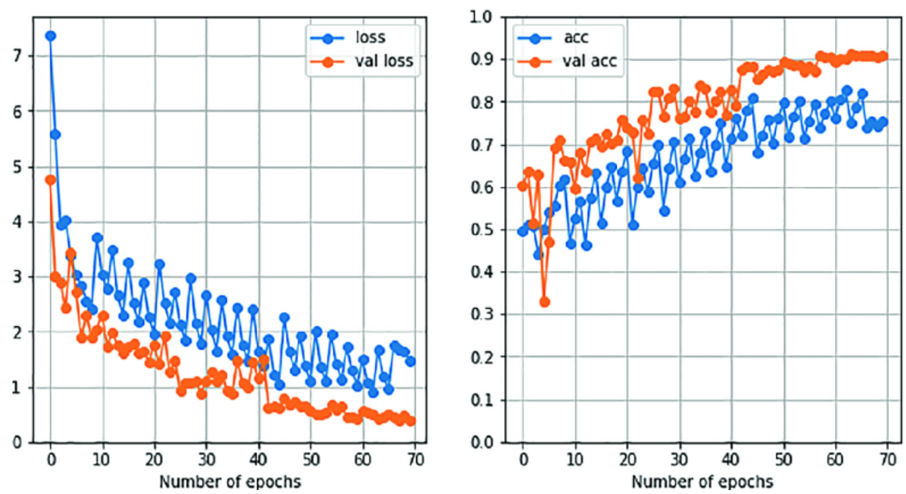

Fig. 3.

Model convergence. Left: training and validation loss. **Right:** training and validation accuracy.

5 Evaluation

As previously emphasized, the model is evaluated on an open-set of dogs. These dogs are unknown to the network during training. This set is composed of 298 pictures of 48 dogs. After feeding the network with these images, 298 embedding vectors are computed. These vectors will finally be used by a specific algorithm to evaluate the network performances on the three following tasks:

- **Verification:** given a pair of pictures, the algorithm has to say if it is the same dog or not. In term of embedding vectors, it means that the distance d between the vectors of a pair is compared with a threshold t. If $d < t$ then the algorithm considers that the pair represents the same dog. A random choice algorithm will obtain an accuracy of 50% here.

- **Recognition:** given one or several pictures per dog that are considered as *learned* by the algorithm, the algorithm has to determine for every *non-learned* picture which dog it represents. Given a newly computed vector the algorithm looks at its closest neighbors in the set of *learned* vectors and outputs the most frequent class in this set: it is the k-nearest neighbors (k-NN) algorithm. This task is harder than the previous one: a random choice algorithm will obtain an accuracy of one over the number of dogs, so 2.08% here.

- **Clustering:** given all the images, the algorithm has to group them by dogs. After computing all the embedding vectors, the clustering algorithm creates groups of vectors that are close to each other. The algorithm applied here is the classic k-means algorithm.

In order to highlight the efficiency of ResNet-like structure, the results obtained with it will be compared to the one obtained with the best VGG-like structure designed to solve this problem. The latter network is composed of 14 million parameters and just like the ResNet-like structure, it contains many dropout layers.

5.1 Face Verification

For this first test, 2500 positive pairs and 2500 negative pairs are generated. A positive pair is a pair of pictures representing the same dog. A negative pair is a pair of pictures representing two different dogs. The Table 2 sums up the main results for this task. The first row shows the models' performances using their best thresholds. The second row shows the models' performances using the value of α used for training. Additionally Fig. 4 illustrates the ROC curve of these models for this binary classification task. The best accuracy reached with the ResNet-like model is 92%. Although it could be considered as low compared to human face identification models (FaceNet methods reaches 99.63% accuracy on LFW dataset), it is high regarding the size of our dataset and the high complexity of dog faces. Indeed human face models are normally trained on datasets of millions of human faces and thanks to the strong similarity between human faces, the model can extract key features more easily.

Finally some examples of false positive and false negative pairs are presented on Fig. 5. Mistakes are mainly due to a too big difference in light exposure between the two pictures and due to a too large angle between the two dog faces. Verification problems can also sometimes come from occlusions, for instance due to the dog tongue or to a muzzle protection.

Table 2.

Verification results. First row: results obtained with the best distance threshold. **Second row:** results obtained with the α margin used for training.

ResNet-like		VGG-like	
Threshold	**Accuracy**	**Threshold**	**Accuracy**
$best = 0.63$	$\mathbf{92.0\%} \pm 0.3$	$best = 0.99$	$91.3\% \pm 0.3$
$\alpha = 0.3$	$85.8\% \pm 0.3$	$\alpha = 0.3$	$68.3\% \pm 0.3$

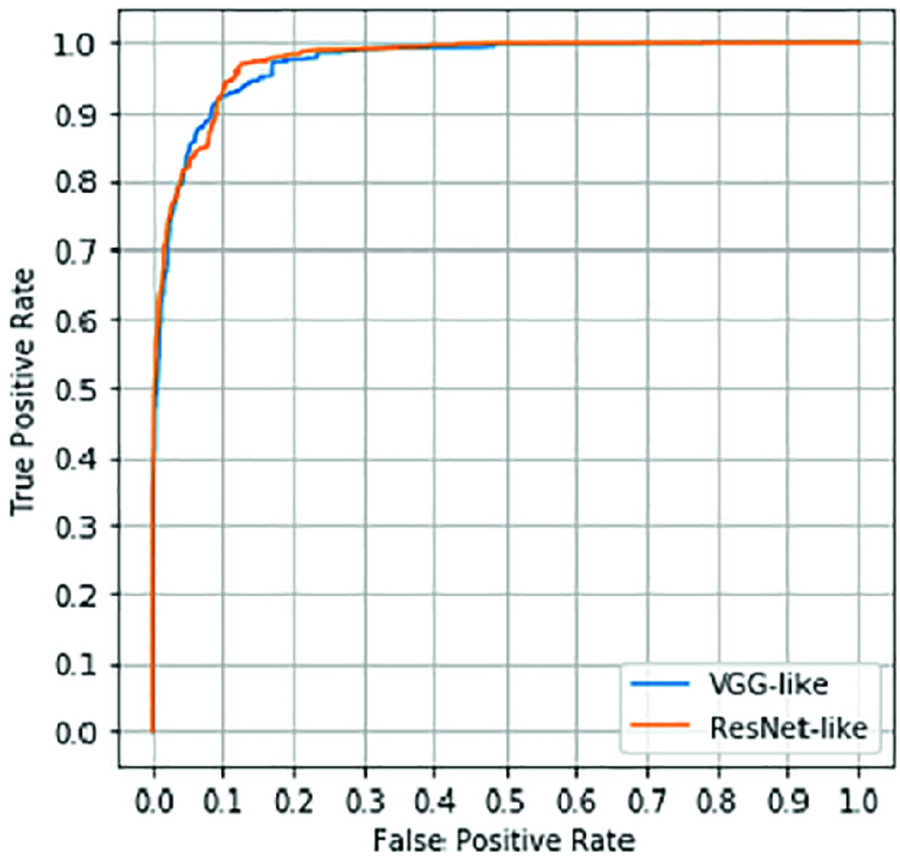

Fig. 4.

Verification task. ROC curve comparing VGG with ResNet for a face verification task.

Fig. 5.

Verification task. Left: some false positive pairs. **Right:** some false negative pairs.

5.2 Face Recognition

This is the main issue for an identification tool. As previously explained, after embedding vector computation by the model, this task is simply a k-NN problem. The testing set is divided into a sub training set and a sub testing set. The sub training set is used to train the k-NN algorithm and the sub testing set to evaluate it. In order to create the sub training set, *M* embedding vectors per dog are selected. The rest of the embedding vectors composes the sub testing set. For instance, if $M = 1$ then it means that a single vector per dog is selected (48 vector in our case) and the rest of them (250 vectors) are used for testing: it is called a one-shot recognition problem. In this case $k = 1$, which means that the class of a vector in the sub testing set is given by the class of its closest neighbor in the sub training set. For $M > 1$, taking $k = M + 1$ gives the best accuracy for this task.

As the *M* vectors are randomly selected within a class, the output accuracy on the sub testing set is different depending on this selection. In order to be more accurate on this evaluation task, for each possible value of *M*, 1000 different sub training/sub testing sets are created and evaluated. The Table 3 represents the different rank-1 results obtained for different values of *M*. The *Mean*, *Minimum* and *Maximum* columns contain the average, the minimum and the maximum accuracy of the algorithm over these 1000 sub sets.

Table 3.

Recognition results. This table presents the rank-1 accuracy on this task.

		ResNet-like			VGG-like		
M	**k**	**Mean**	**Minimum**	**Maximum**	**Mean**	**Minimum**	**Maximum**
1	1	**60.44**%	50.80%	69.60%	57.03%	44.00%	66.00%
2	3	67.49%	56.43%	79.21%	63.13%	51.98%	72.77%
3	4	73.25%	61.54%	83.33%	64.99%	54.49%	74.36%
4	5	73.92%	60.34%	86.21%	65.13%	54.72%	74.54%

The ResNet-like network reaches a rank-1 accuracy of 60.44% for the one-shot recognition task. This is lower than a normal human face recognition algorithm. However, as mentioned in the previous section, this is due to the very small amount of available data and the dog face complexity. It can also be noticed that the performances of the algorithms improve with the number of embedding vectors per class into the sub training set. Indeed the more information that are provided to the network the more accurate its predictions are.

As one of the potential applications of this project is to find a lost pet it could be interesting to look at the rank-5 accuracy. It means that the answer of the algorithm is considered as correct if the right dog is given within the 5 first suggestions. Indeed, the application could suggest to the user a list of possible dogs and the user could then select the correct animal. The obtained results are presented in Table 4. The model accuracy on this task reveals the potential of this project: for a one-shot recognition problem the model reaches 88% accuracy and given at least 3 pictures per dogs the model reaches a maximum accuracy of 100%.

Table 4.

Recognition results. This table presents the rank-5 accuracy on this task.

		ResNet-like			**VGG-like**		
M	**k**	**Mean**	**Minimum**	**Maximum**	**Mean**	**Minimum**	**Maximum**
1	1	**88.41**%	80.00%	94.40%	85.92%	78.80%	91.60%
2	3	**92.83**%	87.13%	98.83%	89.68%	85.64%	95.05%
3	4	**95.97**%	86.54%	**100**%	92.22%	87.18%	96.15%
4	5	**96.10**%	88.79%	**100**%	93.44%	87.93%	97.41%

5.3 Face Clustering

We finally try to cluster the 48 dogs' pictures after embedding vector computation using the k-means clustering algorithm. Over the 48 clustered sets created, the algorithm manages to output 20 sets of correctly clustered dogs. These results are good regarding the complexity of this task: as shown on the left side of Fig. 6, the network can cluster pictures of a dog taken with very different angles and lighting. This figure also shows one of the mistakes made by the algorithm: two badly clustered dogs from the same breed that look very similar.

This kind of application could, for example, help zoologists to automatically sort their animal photos.

Fig. 6.

Clustering task. Left: the algorithm correctly clustered 9 pictures of the same dog with different angles and lighting. **Right:** the algorithm wrongly clustered pictures of two different dogs (3 pictures for the first one and 5 for the other one).

6 Conclusion

Animal identification has until now relied on standard tagging tools and on classical computer vision and machine learning methods. Very little research on this problem has been conducted using deep learning techniques. Recent improvements in deep learning methods have created a breakthrough for human face identification and can now be employed to solve the more complex problem of animal face identification.

A novel method to identify dog faces using deep learning is presented here. Dog face identification is a more complicated task than for human faces due to the lack of available data and to the large range of texture variations in dog face pictures. A new dataset is built and a new model is designed to solve this problem. The trained network reaches a satisfying accuracy on both verification and recognition tasks. This project could thus allow researchers to pay more attention on these new kinds of techniques for their future work on animal identification.

However, there is still room for improvement. Our dataset is significantly small compared to the standard in deep learning domain and too few dogs per breed are represented. The images have to be labeled by hand which could be improved by automatic landmarks detection. Dog muzzle riddles or other dog features could also have been used to help the network with its task. The model is not accurate enough to solve a real identification problem but has its usefulness in other problems such as helping dog owners finding their lost pet. To showcase the potential of the developed method, a mobile app is currently under development to achieve the latter objective.

Footnotes

1. 1.

   https://www.streunerhilfe-bulgarien.de/ (https://www.streunerhilfe-bulgarien.de/).

2. 2.

   https://www.tiko.or.at/de/tiere/ (https://www.tiko.or.at/de/tiere/).

3. 3.

   https://www.pfotenhilfe.org/tiervermittlung/ (https://www.pfotenhilfe.org/tiervermittlung/).

4. 4.

   https://www.la-spa.fr/ (https://www.la-spa.fr/).

5. 5.

   https://www.tieronline.ch/ (https://www.tieronline.ch/).

6. 6.

   https://www.animal-happyend.ch/ (https://www.animal-happyend.ch/).

References

1. Amos, B., Ludwiczuk, B., Satyanarayanan, M.: OpenFace: a general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science (2016)
   Google Scholar (https://scholar.google.com/scholar?q=Amos%2C%20B.%2C%20Ludwiczuk%2C%20B.%2C%20Satyanarayanan%2C%20M.%3A%20OpenFace%3A%20a%20general-purpose%20face%20recognition%20library%20with%20mobile%20applications.%20Technical%20report%2C%20CMU-CS-16-118%2C%20CMU%20School%20of%20Computer%20Science%20%282016%29)

2. Awad, A.I.: From classical methods to animal biometrics: a review on cattle identification and tracking. Comput. Electron. Agric. **123**, 423–435 (2016). https://doi.org/10.1016/j.compag.2016.03.014 (https://doi.org/10.1016/j.compag.2016.03.014). http://www.sciencedirect.com/science/article/pii/S0168169916300837 (http://www.sciencedirect.com/science/article/pii/S0168169916300837)
   CrossRef (https://doi.org/10.1016/j.compag.2016.03.014)
   Google Scholar (http://scholar.google.com/scholar_lookup?title=From%20classical%20methods%20to%20animal%20biometrics%3A%20a%

20review%20on%20cattle%20identification%20and%20tracking&author=AI.%20Awad&journal=Comput.%20Electron.%20Agric.&volume=123&pages=423-435&publication_year=2016&doi=10.1016%2Fj.compag.2016.03.014)

3. Cao, K., Rong, Y., Li, C., Tang, X., Loy, C.C.: Pose-robust face recognition via deep residual equivariant mapping. CoRR abs/1803.00839 (2018). http://arxiv.org/abs/1803.00839 (http://arxiv.org/abs/1803.00839)

4. Chen, S., Liu, Y., Gao, X., Han, Z.: MobileFaceNets: efficient CNNs for accurate real-time face verification on mobile devices. CoRR abs/1804.07573 (2018). http://arxiv.org/abs/1804.07573 (http://arxiv.org/abs/1804.07573)

5. Crouse, D., et al.: LemurFaceID: a face recognition system to facilitate individual identification of lemurs. BMC Zool. **2**(1), 2 (2017). https://doi.org/10.1186/s40850-016-0011-9 (https://doi.org/10.1186/s40850-016-0011-9)
CrossRef (https://doi.org/10.1186/s40850-016-0011-9)
Google Scholar (http://scholar.google.com/scholar_lookup?title=LemurFaceID%3A%20a%20face%20recognition%20system%20to%20facilitate%20individual%20identification%20of%20lemurs&author=D.%20Crouse&journal=BMC%20Zool.&volume=2&issue=1&pages=2&publication_year=2017&doi=10.1186%2Fs40850-016-0011-9)

6. Deb, D., et al.: Face recognition: primates in the wild. CoRR abs/1804.08790 (2018). http://arxiv.org/abs/1804.08790 (http://arxiv.org/abs/1804.08790)

7. Deng, J., Guo, J., Zafeiriou, S.: ArcFace: additive angular margin loss for deep face recognition. CoRR abs/1801.07698 (2018). http://arxiv.org/abs/1801.07698 (http://arxiv.org/abs/1801.07698)

8. Hansen, M.F., et al.: Towards on-farm pig face recognition using convolutional neural networks. Comput. Ind. **98**, 145–152 (2018)
CrossRef (https://doi.org/10.1016/j.compind.2018.02.016)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Towards%20on-farm%20pig%20face%20recognition%20using%20convolutional%20neural%20networks&author=MF.%20Hansen&journal=Comput.%20Ind.&volume=98&pages=145-152&publication_year=2018)

9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015). http://arxiv.org/abs/1512.03385 (http://arxiv.org/abs/1512.03385)

10. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. CoRR abs/1709.01507 (2017). http://arxiv.org/abs/1709.01507 (http://arxiv.org/abs/1709.01507)

11. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical report 07-49, University of Massachusetts, Amherst, October 2007
Google Scholar (https://scholar.google.com/scholar?q=Huang%2C%20G.B.%2C%20Ramesh%2C%20M.%2C%20Berg%2C%20T.%2C%20Learned-Miller%2C%20E.%3A%20Labeled%20faces%20in%20the%20wild%3A%20a%20database%20for%20studying%20face%20recognition%20in%20unconstrained%20environments.%20Technical%20report%2007-

49%2C%20University%20of%20Massachusetts%2C%20Amherst%2C%20October%202007)

12. Blancou, J.: A history of the traceability of animals and animal products. Revue scientifique et technique (International Office of Epizootics) (2001). https://www.ncbi.nlm.nih.gov/pubmed/11548516 (https://www.ncbi.nlm.nih.gov/pubmed/11548516)

13. Jarraya, I., Ouarda, W., Alimi, A.M.: A preliminary investigation on horses recognition using facial texture features. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics, pp. 2803–2808, October 2015. https://doi.org/10.1109/SMC.2015.489 (https://doi.org/10.1109/SMC.2015.489)

14. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014). http://arxiv.org/abs/1412.6980 (http://arxiv.org/abs/1412.6980)

15. Kumar, S., et al.: Deep learning framework for recognition of cattle using muzzle point image pattern. Measurement **116**, 1–17 (2018). https://doi.org/10.1016/j.measurement.2017.10.064 (https://doi.org/10.1016/j.measurement.2017.10.064). http://www.sciencedirect.com/science/article/pii/S0263224117306991 (http://www.sciencedirect.com/science/article/pii/S0263224117306991) CrossRef (https://doi.org/10.1016/j.measurement.2017.10.064) Google Scholar (http://scholar.google.com/scholar_lookup?title=Deep%20learning%20framework%20for%20recognition%20of%20cattle%20using%20muzzle%20point%20image%20pattern&author=S.%20Kumar&journal=Measurement&volume=116&pages=1-17&publication_year=2018&doi=10.1016%2Fj.measurement.2017.10.064)

16. Kumar, S., Singh, S.K.: Monitoring of pet animal in smart cities using animal biometrics. Future Gener. Comput. Syst. **83**, 553–563 (2018). https://doi.org/10.1016/j.future.2016.12.006 (https://doi.org/10.1016/j.future.2016.12.006). http://www.sciencedirect.com/science/article/pii/S0167739X16307385 (http://www.sciencedirect.com/science/article/pii/S0167739X16307385) CrossRef (https://doi.org/10.1016/j.future.2016.12.006) Google Scholar (http://scholar.google.com/scholar_lookup?title=Monitoring%20of%20pet%20animal%20in%20smart%20cities%20using%20animal%20biometrics&author=S.%20Kumar&author=SK.%20Singh&journal=Future%20Gener.%20Comput.%20Syst.&volume=83&pages=553-563&publication_year=2018&doi=10.1016%2Fj.future.2016.12.006)

17. Kumar, S., Singh, S.K., Abidi, A.I., Datta, D., Sangaiah, A.K.: Group sparse representation approach for recognition of cattle on muzzle point images. Int. J. Parallel Program. **46**, 812–837 (2017) CrossRef (https://doi.org/10.1007/s10766-017-0550-x) Google Scholar (http://scholar.google.com/scholar_lookup?title=Group%20sparse%20representation%20approach%20for%20recognition%20of%20cattle%20on%20muzzle%20point%20images&author=S.%20Kumar&author=SK.%20Singh&author=AI.%20Abidi&author=D.%20Datta&author=AK.%20Sangaiah&journal=Int.%20J.%20Parallel%20Program.&volume=46&pages=812-837&publication_year=2017)

18. Kumar, S., Singh, S.K., Singh, R.S., Singh, A.K., Tiwari, S.: Real-time recognition of cattle using animal biometrics. J. Real-Time Image Process. **13**(3), 505–526 (2017). https://doi.org/10.1007/s11554-016-0645-4 (https://doi.org/10.1007/s11554-016-0645-4)
CrossRef (https://doi.org/10.1007/s11554-016-0645-4)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Real-time%20recognition%20of%20cattle%20using%20animal%20biometrics&author=S.%20Kumar&author=SK.%20Singh&author=RS.%20Singh&author=AK.%20Singh&author=S.%20Tiwari&journal=J.%20Real-Time%20Image%20Process.&volume=13&issue=3&pages=505-526&publication_year=2017&doi=10.1007%2Fs11554-016-0645-4)

19. Kumar, S., Singh, S.K., Singh, R., Singh, A.K.: Animal Biometrics: Techniques and Applications. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-7956-6 (https://doi.org/10.1007/978-981-10-7956-6). https://www.springer.com/us/book/9789811079559 (https://www.springer.com/us/book/9789811079559)
CrossRef (https://doi.org/10.1007/978-981-10-7956-6)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Animal%20Biometrics%3A%20Techniques%20and%20Applications&author=S.%20Kumar&author=SK.%20Singh&author=R.%20Singh&author=AK.%20Singh&publication_year=2017)

20. Liu, J., Kanazawa, A., Jacobs, D., Belhumeur, P.: Dog breed classification using part localization 7572, 172–185, October 2012
Google Scholar (https://scholar.google.com/scholar?q=Liu%2C%20J.%2C%20Kanazawa%2C%20A.%2C%20Jacobs%2C%20D.%2C%20Belhumeur%2C%20P.%3A%20Dog%20breed%20classification%20using%20part%20localization%207572%2C%20172%E2%80%93185%2C%20October%202012)

21. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: SphereFace: deep hypersphere embedding for face recognition. CoRR abs/1704.08063 (2017). http://arxiv.org/abs/1704.08063 (http://arxiv.org/abs/1704.08063)

22. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. CoRR abs/1503.03832 (2015). http://arxiv.org/abs/1503.03832 (http://arxiv.org/abs/1503.03832)

23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014). http://arxiv.org/abs/1409.1556 (http://arxiv.org/abs/1409.1556)

24. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014). http://jmlr.org/papers/v15/srivastava14a.html (http://jmlr.org/papers/v15/srivastava14a.html)
MathSciNet (http://www.ams.org/mathscinet-getitem?mr=3231592)
zbMATH (http://www.emis.de/MATH-item?1318.68153)
Google Scholar (http://scholar.google.com/scholar_lookup?title=Dropout%3A%20a%20simple%20way%20to%20prevent%20neural%20networks%20from%20overfitting&author=N.%20Srivastava&author=G.%20Hinton&author=A.%20Krizhevsky&author=I.%20Sutskever&author=R.%20Salakhutdin

ov&journal=J.%20Mach.%20Learn.%20Res.&volume=15&pages=1929-1958&publication_year=2014)

25. Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: AAAI (2016)
Google Scholar (https://scholar.google.com/scholar?q=Szegedy%2C%20C.%2C%20Ioffe%2C%20S.%2C%20Vanhoucke%2C%20V.%3A%20Inception-v4%2C%20Inception-ResNet%20and%20the%20impact%20of%20residual%20connections%20on%20learning.%20In%3A%20AAAI%20%282016%29)

26. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: DeepFace: closing the gap to human-level performance in face verification (2014). https://research.fb.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/ (https://research.fb.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/)

27. Trigueros, D.S., Meng, L., Hartnett, M.: Face recognition: from traditional to deep learning methods (2018)
Google Scholar (https://scholar.google.com/scholar?q=Trigueros%2C%20D.S.%2C%20Meng%2C%20L.%2C%20Hartnett%2C%20M.%3A%20Face%20recognition%3A%20from%20traditional%20to%20deep%20learning%20methods%20%282018%29)

28. Wang, H., et al.: CosFace: large margin cosine loss for deep face recognition. CoRR abs/1801.09414 (2018). http://arxiv.org/abs/1801.09414 (http://arxiv.org/abs/1801.09414)

29. Wang, X., Ly, V., Sorensen, S., Kambhamettu, C.: Dog breed classification via landmarks, pp. 5237–5241, January 2015. https://doi.org/10.1109/ICIP.2014.7026060 (https://doi.org/10.1109/ICIP.2014.7026060)

30. Zheng, Y., Pal, D.K., Savvides, M.: Ring loss: convex feature normalization for face recognition. CoRR abs/1803.00130 (2018). http://arxiv.org/abs/1803.00130 (http://arxiv.org/abs/1803.00130)

31. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. CoRR abs/1707.07012 (2017). http://arxiv.org/abs/1707.07012 (http://arxiv.org/abs/1707.07012)

Copyright information

About this paper

Cite this paper as:

Mougeot G., Li D., Jia S. (2019) A Deep Learning Approach for Dog Face Verification and Recognition. In: Nayak A., Sharma A. (eds) PRICAI 2019: Trends in Artificial Intelligence. PRICAI 2019. Lecture Notes in Computer Science, vol 11672. Springer, Cham. https://doi.org/10.1007/978-3-030-29894-4_34

- First Online 23 August 2019
- DOI https://doi.org/10.1007/978-3-030-29894-4_34
- Publisher Name Springer, Cham
- Print ISBN 978-3-030-29893-7
- Online ISBN 978-3-030-29894-4
- eBook Packages Computer Science Computer Science (R0)

Personalised recommendations