ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/282790234

Where is my puppy? Retrieving lost dogs by facial features

Article *in* Multimedia Tools and Applications · July 2017

DOI: 10.1007/s11042-016-3824-1 · Source: arXiv

CITATIONS
8

READS
1,656

4 authors, including:

Thierry Moreira
São Paulo State University
14 PUBLICATIONS **93** CITATIONS
SEE PROFILE

Mauricio Perez
Agency for Science, Technology and Research (A*STAR)
18 PUBLICATIONS **396** CITATIONS
SEE PROFILE

Rafael de Oliveira Werneck
University of Campinas
11 PUBLICATIONS **215** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Sensitive Media Analysis View project

Action Recognition View project

All content following this page was uploaded by Mauricio Perez on 22 September 2016.

The user has requested enhancement of the downloaded file.

arXiv:1510.02781v2 [cs.CV] 1 Aug 2016

Where Is My Puppy? Retrieving Lost Dogs by Facial Features

Thierry Pinheiro Moreira · Mauricio Lisboa Perez · Rafael de Oliveira Werneck · Eduardo Valle

Received: date / Accepted: date

Abstract A pet that goes missing is among many people's worst fears: a moment of distraction is enough for a dog or a cat wandering off from home. Some measures help matching lost animals to their owners; but automated visual recognition is one that — although convenient, highly available, and low-cost — is surprisingly overlooked. In this paper, we inaugurate that promising avenue by pursuing face recognition for dogs. We contrast four ready-to-use human facial recognizers (EigenFaces, FisherFaces, LBPH, and a Sparse method) to two original solutions based upon convolutional neural networks: BARK (inspired in architecture-optimized networks employed for human facial recognition) and WOOF (based upon off-the-shelf OverFeat features). Human facial recognizers perform poorly for dogs (up to 60.5% accuracy), showing that dog facial recognition is not a trivial extension of human facial recognition. The convolutional network solutions work much better, with BARK attaining up to 81.1% accuracy, and WOOF, 89.4%. The tests were conducted in two datasets: Flickr-dog, with 42 dogs of two breeds (pugs and huskies); and Snoopybook, with 18 mongrel dogs.

Keywords Face Recognition · Dog Recognition · Deep Learning · Convolutional Networks

This work was supported in part by FAPESP, CAPES, CNPq, and SAMSUNG.

T. Moreira, M. Perez, and R. Werneck contributed equally to this work.

T. Moreira
LIV Lab., Institute of Computing (IC), University of Campinas (Unicamp)
E-mail: thierrypin@liv.ic.unicamp.br

M. Perez · R. Werneck · E. Valle
RECOD Lab., Institute of Computing (IC), University of Campinas (Unicamp)

E. Valle
Department of Computer Engineering and Industrial Automation (DCA), School of Electrical and Computer Engineering (FEEC), University of Campinas (Unicamp) Av. Albert Einstein, 400, Campinas 13083-852, SP, Brazil

1 Introduction

Dogs hold a special status for humans: they were the first domesticated animals, and are beloved as pet companions. The American Pet Products Association estimates that they are present in 65% of U.S. households [1].

However, our love for dogs becomes worry when they run away. The American Society for the Prevention of Cruelty to Animals[2] reports that 15% of pet-owning households lost a dog or a cat in the last five years; and 15% of those lost animals were never recovered. Most dog owners recovered their dogs through searches in the neighborhood (49%), or through ID tags (15%). Only 6% of owners found their pets at shelters.

Owners have a handful of ways to locate lost dogs: collar tags, GPS devices, tattoos, implanted microchips/RFIDs. Collar tags are cheap, but easily lost. In addition, not all dogs tolerate wearing a collar all the time. Tattoos and implanted chips are more robust, but also more expensive and require a preventive attitude by the owners. Some first-world countries solve that problem by making tattooing or chip implant mandatory. But in other countries, including Brazil, tattoos and implants are virtually unheard-of.

Thus the main motivation of this work: why not identifying lost dogs using... the dogs themselves? Contrarily to collars and IDs, the dog appearance is intrinsic. Contrarily to chips and implants, appearance is readily available and does not require extraordinary forethought. Most, if not all, owners will spontaneously accumulate dozens of pictures of their pets.

In this paper, we inaugurate that promising avenue by pursuing face recognition for dogs. Although literature on pet biometry is — to say the least — scarce, human face recognition is a well-established domain with a vast literature [1–3, 7, 8, 18, 21, 23, 25]. That allows us to take inspiration from methods intended for humans, and to evaluate their effectiveness for dogs.

2 Literature review

Literature on animal biometry is scarce. Computer Vision is mainly interested in animals as members of large classes, not as individuals. The interest in animals as individuals come mainly from wildlife researchers, for whom the identification of specimens is critical to determine ecological patterns[3].

The art on biometry for pets is virtually nonexistent. However, some works address fine-grained classification of dogs into breeds. Parkhi et al. [17] used a model to describe both the shape and the texture of a pet in their Oxford-IIIT Pet dataset,

[1] `http://www.humanesociety.org/issues/pet_overpopulation/facts/pet_ownership_statistics.html` (December 2015).

[2] `https://www.aspca.org/about-us/press-releases/how-many-pets-are-lost-how-many-find-their-way-home-aspca-survey-has-answers` (March 2015).

[3] For example, in August 2015, the New England Aquarium and MathWorks have launched a large competition for identifying individual endangered whales: `https://www.kaggle.com/c/noaa-right-whale-recognition`.

achieving 59% of accuracy. Oxford-IIIT Pet is a public and carefully annotated dataset[4] containing 12 cat breeds, and 25 dog breeds, with 200 images for each breed, downloaded from Google Images, Flickr, Catster, and Dogster. Unfortunately, each individual appears only once in this dataset, rendering it useless for our purposes. Parkhi et al. [17] uses a deformable part model [11] with HOG filters [9] to detect edge information and represent shapes. They aim at detecting stable and distinctive components of the dog, particularly the head. Bag of visual words (from densely sampled SIFT descriptors [16]) were used for texture.

Liu et al. [15] classified dogs into breeds using models for the geometry and appearance of breeds, including facial clues. First, they detect the dog face using a sliding window, then locate the nose and eyes using consensus of exemplars [4], and finally infer four possible locations for the ears and remaining components. Next, they extract color histograms, and grayscale SIFT descriptors, from different parts of the dog face. They achieved 67% accuracy in a database created by the authors with 133 dog breeds and 8,351 images. Wang et al. [24], working in this same dataset, improved the results to 96% accuracy, representing dog shapes using 8 landmarks given by points on the Grassmann manifold. Grassmann manifolds are differential geometry structures used, among other things, for image registration.

The only vision studies we could find that consider dogs as individuals are not about Computer Vision, but about the Human Visual System. Scapinello et al. [19] studied how familiarity and orientation affects recognition of human faces, canine faces, and buildings. Diamond and Carey [10] also study how 180-degree rotations affect the recognition of pictures, including dogs, by humans. Both works find that rotations affect the recognition of faces, but Scapinello et al. [19] show that the effect is greater for human faces than for dog faces. More important to us is that Diamond and Carey [10] report only 76% accuracy for humans identifying dogs in optimal conditions. When the subjects were experts (breeders and judges from the American Kennel Club/New York) the accuracy raised to 81%. Those numbers suggest that recognizing a large number of dogs is far from trivial, even for human experts.

Human Face Recognition is a well-established topic [2, 3, 23, 25], with different paradigms: *i*) *detection*, in which the system detects whether a face is present in the image (i.e., any face, not someone's face in particular); *ii*) *verification*, in which the system decides whether two faces come from the same person; and *iii*) *identification*, where the system matches an unknown face against a labeled set [18].

Detection is useful as pre-processing for the other paradigms, and has application of its own, e.g., in camera auto-focusing. Verification is useful, for example, for biometric authentication. However, here we will focus on identification.

Sirovich and Kirby [21] first introduced *Eigenfaces* for face detection. It employs the eigenvectors of the image; hence the name. In Eigenfaces, one starts with a large collection of face images, all of the same size, and all (approximately) registered. The images are taken as vectors, whose covariance matrix is estimated. The eigenvectors of that covariance matrix are computed, and those corresponding to the highest eigenvalues are kept to form a subspace in which the faces are represented

[4] `http://www.robots.ox.ac.uk/~vgg/data/pets/`

(the whole procedure is akin to a Principal Component Analysis). Soon, Eigenfaces became a tool for verification and identification as well [23].

Fisherfaces are similar to Eigenfaces, but using Fisher Linear Discriminants [12], or, very often Linear Discriminant Analysis (although each technique gives slightly different results). A subspace is chosen in order to minimize within-class distances and maximize between-class distances [3]. While Eigenfaces emphasizes the ability to reconstruct the original image using a small number of components, Fisherfaces emphasizes the ability to discriminate faces from different people.

Contrasting with those holistic approaches, Local Binary Patterns describe small patches of the image. Surrounding pixels are thresholded against the center of the neighborhood, resulting in a compact bit array. The local binary patterns can be pooled together into a single histogram per image [2]. At first, it was used for texture classification; later, it was applied to different domains, such as face recognition.

Newer approaches use sparse coding for face recognition. Xu et al. [26] presented a two-phase method. First, the query image is represented as a linear combination of the n training faces: $y = a_1x_1 + \ldots + a_nx_n$, being y the reconstructed image, and a_i and x_i the *i*th coefficient and train image. The contribution of the *i*th face to the reconstruction is taken as a_ix_i, and its deviation to the reconstruction, $e_i = ||y - a_ix_i||^2$, is the measure of distance to select the M nearest faces.

The second phase is, again, reconstructing the query image, now using only the selected M faces. Between the selected images, there is a subset of the total classes. The contributions of all elements of the set are summed to obtain the class contribution to the reconstruction. Finally, the class with the smallest deviation to the reconstruction, computed as before, is chosen as the verdict of the system.

Convolutional Neural Networks have seized the attention of the Computer Vision community in virtually all tasks, and facial recognition is no exception.

Pinto et al. [18] evaluated networks with two to three layers, which extract features that are then fed to Support Vector Machines, in a one-versus-all configuration. They optimized their networks by trying a range of network architectures initialized with random weights. In this work the weights themselves are not optimized at all. They achieved 85~89% identification accuracy with the best network architecture on the public datasets Facebook100 and PubFig83.

Chiachia et al. [7] improved that result by tailoring the last layer of the network for each individual, and optimizing both the hyperparameters and the weights of that layer. They raised the state of the art on PubFig83 to 92% accuracy.

Those networks, although successful, are still rather shallow, with up to three layers. Deeper Convolutional Neural Networks have obtained state-of-the-art results in many image classification tasks [14, 20, 22]. Although those network are very expensive to train — both in computational resources and in training samples — it is often possible to reuse the weights learned from a task to another, in what is called transfer learning.

The ImageNet Competition is a challenging annual contest to find the best classification model for a thousand different classes, using a training set of over 1.2 million images. Since 2012, Deep Convolutional Networks got systematically the best places. In particular, Sermanet et al. [20], improving on the winner model for ImageNet 2012 [14], proposed a model who won the localization task for ImageNet

2013. They distributed a ready-to-use model, with the weights already learned, as OverFeat[5]. OverFeat became, through transfer learning, a very popular model for feature extraction in a myriad of tasks beyond ImageNet.

3 Material and methods

3.1 Datasets

We employed two datasets, described below.

3.1.1 Flickr-dog

We acquired the Flickr-dog dataset[6] by selecting dog photos from Flickr available under Creative Commons licenses. We cropped the dog faces, rotated them to align the eyes horizontally, and resized them to 250×250 pixels.

We selected dogs from two breeds: pugs and huskies. Those breeds were selected to represent the different degrees of challenge: we expected pugs to be difficult to identify, and huskies to be easy. For each breed, we found 21 individuals, each with at least 5 photos. We labeled the individuals by interpreting picture metadata (user, title, description, timestamps, etc.), and double checked with our own ability to identify the dogs.

Altogether, the Flickr-dog dataset has 42 classes and 374 photos. Figure 1 shows a typical sample.

The annotation is very laborious and is the limiting factor in expanding the dataset. Still, while keeping the data acquisition manageable, we strove to make the dataset as challenging as possible. By focusing on just pugs and huskies, we prevented the classifier from simply identifying the breeds. By cropping the faces, we reduced background information that could give unfair clues to the classifier. The choice of 21 individuals per breed was not accidental either: it makes random matches just unlikely enough for the usual significance of 95%.

3.1.2 Snoopybook

This dataset has 18 mongrel dogs — mostly puppies — with, again, at least 5 photos per individual, for a total of 251 photos. Each photo was registered to put the eyes and snout at the same position, and then resized to 200×200 pixels (Figure 2).

The Snoopybook dataset is complementary to Flickr-dog, as it offers a less controlled array of individuals.

3.2 Protocol

Figure 3 shows the basic pipeline used in the experiments of this work.

[5] `http://cilvr.nyu.edu/doku.php?id=software:overfeat:start` (August 2015)

[6] available at: To be published on article acceptance.

Fig. 1: A sample of Flickr-dog dataset. We acquired 374 photos licensed under Creative Commons from Flickr, representing 2 breeds (pugs and huskies), 21 individuals per breed, and at least 5 photos per individual. The choice of breeds intended to reflect a difficult case (pugs) and an easy one (huskies).

Fig. 2: A sample of Snoopybook dataset. This dataset has 18 mongrel dogs, with at least 5 photos per individual, for a total of 251 photos. With a less controlled array of individuals, Snoopybook is complementary to the well-controlled Flickr-dog dataset.

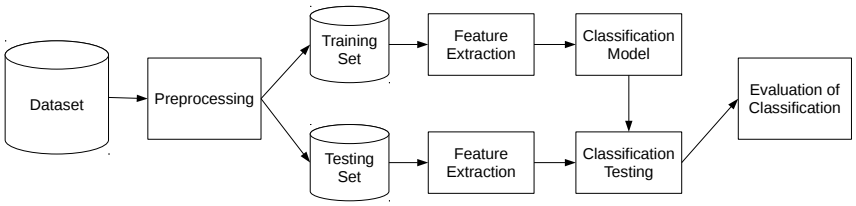

Fig. 3: General pipeline of the experiments.

The protocol was a stratified k-fold cross-validation, that splits the dataset into k folds, preserving as much as possible the class proportions among the folds. We used 10 folds, with nine folds for training, and one for testing.

Our main metric is the balanced average accuracy, which is the arithmetic mean of the accuracy for each of the classes. We also employ confusion matrices for detailed analyses of the results.

For the retrieval experiment, we employ a top-k recall, which ranks the classification scores for all classes, and counts the test as successful if the right class is among the highest k scores.

3.3 Techniques

3.3.1 Baselines

As baselines, we employed the human face recognition algorithms readily available in OpenCV [6]: LBPH, EigenFaces and FisherFaces. OpenCV offers end-to-end algorithms for a facial recognition that start with the image and end with the class label; we chose to use those algorithms without modification. EigenFaces' best results employed 80 components, while FisherFaces kept all components (corresponding, in our experiments, to the number of classes/individuals). LBPH most accurate and agile results used the standard eight points on a circle of radius one.

We also employed a sparse method for face recognition [26], This method is based in two phases, in which the first phase reconstruct a test image by the training subset of images, and the second phase also reconstructed the test image, but only using a fourth of the nearest faces, to define the contribution of class to the reconstruction.

3.3.2 Shallow CNNs — BARK

We propose Convolutional Neural Networks as the most promising solution for dog facial recognition. We follow Pinto et al. [18] by optimizing the network architecture, and employing random weights. The weights themselves are not optimized. The resulting network extracts features that are fed to a linear SVM.

The network structures explored during optimization follow a scheme defined in Cox et al. [8]. The network consists mainly of stacked layers, going from one up to three layers with the same hyperparameter space, which on the bottom of the

architecture there is a pre-processing normalization of the input and at the top a linear SVM. Each of the intermediary layers have the following components, in this order:

1. **Convolution Linear Filters**: Bank of linear filters to be applied in the input signal with convolution operation. All kernels have random values respecting a uniform distribution;
2. **Activation Function**: Function responsible for clipping the activation to a parametric range.
3. **Pooling**: Neighbors filter activations will be pooled together, spatial down sampling even further the output.
4. **Normalization** (optional): Normalization of the pooled output based on the magnitude of the response of neighbors within some region.

A detailed description on the functions' mathematics of each component of the network can be found in Cox et al. [8].

We implemented the solution with simple-hp[7], a high-level wrapper for Bergstra's Hyperopt-convnet [5]. Simple-hp employs SVM from scikit-learn 0.14.1, with linear kernel and parameter $C = 1e5$. A total of 2000 architectures were evaluated using both the Random and the Tree of Parzen Estimators (TPE) optimizers [5]. The hyperparameters space for optimizing the network architecture comprised:

- **Input size (in pixels)**: 64x64, 128x128, 256x256, original image size;
- **Number of layers in network**: 1, 2, 3;
- **Normalization after pooling**: Yes / No;
- **Filters in each layer (amount)**: 32, 64, 128, 256;
- **Size of the filters (pixels)**: 3, 5, 7, 9;
- **Exponent of pooling layers (scalar)**: 1, 2, 10;
- **Stride of pooling layers (pixels)**: 1, 2, 4, 8.

In summary, this first proposal employs rather shallow convolutional networks whose architecture/hyperparameters are trained, but whose weights are kept random. We called it BARK — Best Architecture to Retrieve K9. Figure 4 shows an example of pipeline for the BARK proposal.

3.3.3 Deep CNNs — WOOF

The second original proposal employs off-the-shelf OverFeat [20] as Convolutional Neural Network. Sermanet et al. [20], proposed the OverFeat model similar to Krizhevsky et al. [14], improving the network design and inference step. The Krizhevsky et al. [14] network's architecture comprises eight layers, divided in five convolution-based and three fully-connected layers. The authors emphasize the following as most important aspects of this model, presented in order of importance: Rectified Linear Units (ReLUs) neurons, which allow much faster learning than *tanh* units; Local Response Normalization, for aiding in generalization; Overlapping Pooling, that slightly lowers error rates by hindering overfit. Also, for reducing overfitting, data augmentation techniques and dropout [13] were used.

[7] www.github.com/giovanichiachia/simple-hp

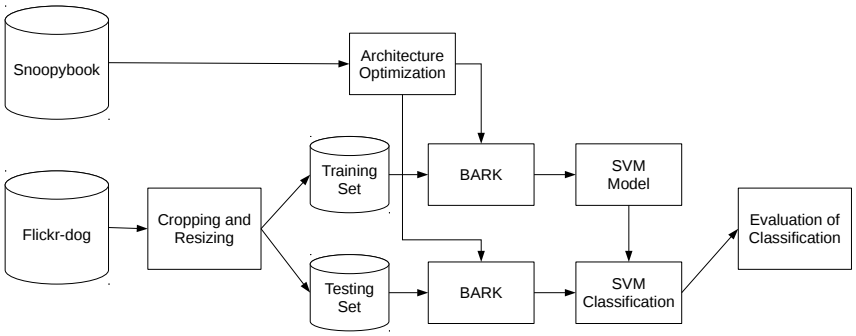

Fig. 4: An example for the BARK pipeline.

Sermanet et al. [20] network won the localization task of ImageNet 2013 challenge, and achieved 14.20% top-5 error rate on the classification task, against Krizhevsky et al. 15.30%. This error rate was obtained with an 8-layer architecture – five convolutional and three fully-connected – with some modifications. ReLUs non-linearities are still present in the convolutional layers, along with max pooling. However, there are three different aspects: Local Response Normalization is not used; there is no overlapping in pooling regions; and it was set a smaller value for stride on the 1st and 2nd layer, resulting in larger feature maps. A more accurate version of the network was also designed, improving their results in ImageNet about two percentage points. In this model, there are six convolutional and three fully-connected layers. Besides the extra convolutional stage, there are other variations, mainly on the strides and sizes of the first layer and feature maps from all convolutional stages. With these variations, the number of connections almost doubled.

As before, the network is applied extracting features that are fed to a linear SVM. We resize all input images to 221×221 pixels, as required by OverFeat. The OverFeat package offers two models: *accurate* or *fast*. We pick the former, in virtue of its better performance. We used the libSVM 3.17 with linear kernel and default parameters ($C = 1.0$).

In summary, this second proposal employs very deep, but pre-trained, convolutional networks. No training or fine tuning was performed with Deep CNNs because it would be infeasible given the handicap in the number of samples from datasets utilized. Only the SVM is trained. We called this second proposal WOOF — Wields Off-the-shelf OverFeat Features. Figure 5 shows an example of the WOOF proposal applied to the Snoopybook dataset.

4 Experiments

We understand that, in real scenarios, the dog search range may be narrowed by information readily known to the dog's owner and shelter keepers, but our experiments concern the descriptor effectiveness in general conditions.

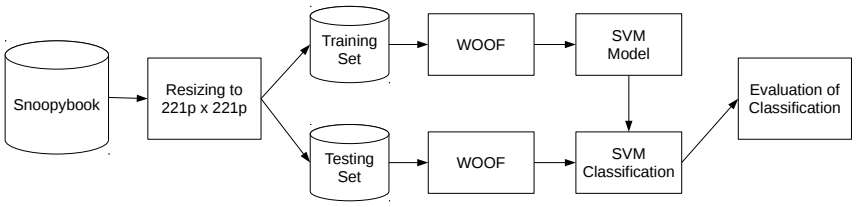

Fig. 5: Pipeline example of the WOOF proposal applied to the Snoopybook dataset.

We proposed three questions to guide our work: *Do human face recognizers generalize to canine faces?*, *Are general-purpose convolutional networks competitive with special-purpose facial recognizers?*, and *How accuracy behaves as the number of retrieved individuals vary?*.

Do human face recognizers generalize to canine faces? To answer that question, we evaluate our baselines, ready-to-use human facial recognizers available in OpenCV and the sparse method in both our datasets.

Table 1 shows the balanced average accuracy obtained with four human face recognizers: EigenFaces [21], FisherFaces [3], LBPH [2], and Sparse [26]. However, all four off-the-shelf human facial recognizers have rather poor results for dogs, while neural networks performed significantly better. The best hand crafted result, in Flickr-dog dataset, was LBPH's 43.2%, and the general purpose network yielded 66.9%. For Snoopybook, the sparse representation had the best result of 60.5% and the network, 89.4%. That answers our first question: dog facial recognition is not a trivial extension of human facial recognition.

Table 1: Balanced accuracies (in %) for all evaluated methods in both datasets. Off-the-shelf human facial recognizers (first four lines) tend to work poorly. WOOF got the best overall results, but BARK with the architecture trained in Flick-dog was a good second best. As we expected, Flickr-dog was more challenging than Snoopybook, due to presence of just two breeds.

Methods	Dataset	
	Flickr-dog	Snoopybook
EigenFaces [21]	33.9	41.6
FisherFaces [3]	22.7	55.4
LBPH [2]	43.2	56.1
Sparse [26]	39.9	60.5
$\text{BARK}_{\text{flickr}}$	**67.6**	81.1
$\text{BARK}_{\text{snoopy}}$	49.1	64.4
WOOF	**66.9**	**89.4**

Are general-purpose convolutional networks competitive with special-purpose facial recognizers? To answer that, we evaluate two original solutions based upon ex-

isting Convolutional Networks: BARK and WOOF. BARK has two training steps: one for the network architecture, and one for the SVM. The dataset used to train the architecture appears in the subscript $\text{BARK}_{dataset}$ while the dataset used for training the SVM appears at the table header in Table 1. We ensure that the test set excludes all samples used in any training.

During BARK architecture optimization, many different combination of the hyperparameters of the network were evaluated by extracting the features from the training data and feeding it to the SVM in a cross-validation manner, with different subsplits from the training dataset in train and validation. The mean accuracy of the SVM was used to determine the best architecture found, and then this architecture, and only this, was evaluated with the test data.

Table 1 shows the results in the bottom three lines. As explained, in $\text{BARK}_{\text{flickr}}$, Flickr-dog is used for training the architecture, and in $\text{BARK}_{\text{snoopy}}$, Snoopybook is employed.

BARK and WOOF performed similarly on Flickr-dog, but WOOF performed better on Snoopybook. We believe that the difference in the latter case is due to the deeper, and pre-trained, OverFeat network employed in WOOF. OverFeat is a very deep network, pre-trained with millions of images, thousands of which contain dogs.

Curiously, $\text{BARK}_{\text{flickr}}$ performed better for both datasets. This might be due to Flickr-dog having more data and with individuals more similar to each other, demanding more discriminative features, thus leading to a better optimization of the architecture.

What lead us to believe that the greatest impact on the results did not come from the parameters, and hyperparameters space, employed here, which seems to be sufficient for a near optimal performance, considering this methodology. Rather, the performance rely more on a appropriate training dataset, with challenging cases so the optimization can focus on the most discriminative features, and with distinct breeds, since different breeds may have different features that help discriminating between its individuals.

How accuracy behaves as the number of retrieved individuals vary?. To answer the third question, we evaluate the best technique's — WOOF's — top-k recall as we vary k. That corresponds to a retrieval scenario reflecting the real-world application: there is a database of labeled dog pictures, the user comes with one photo of an unknown dog, and the system provides a few best matches for the user to consider.

We present the top-k recall (explained in Section 3.2) for $k = \{1, 2, 3, 4, 5, 10, 15\}$ in Figure 6–Above. For $k = 5$, top-k recall is already over 90%. In order to evaluate the robustness of WOOF with a limited-size dataset, we made a strict comparison with random chance. The odds ratio is a standard way to compare two probabilities, fractions or rates, dividing the odds of the first by the odds of the second. For a given probability, fraction or rate p, $\text{odds}(p) = p/(1-p)$. We used $k/21$ as random chance, more stringent than the $k/42$ that would allow for confusion between the breeds in Flickr-dog. Still, WOOF performs well above it for all tested values of k, as we show in Figure 6–Below.

To complement the result above, we show two examples in a retrieval scenario in Figure 7.

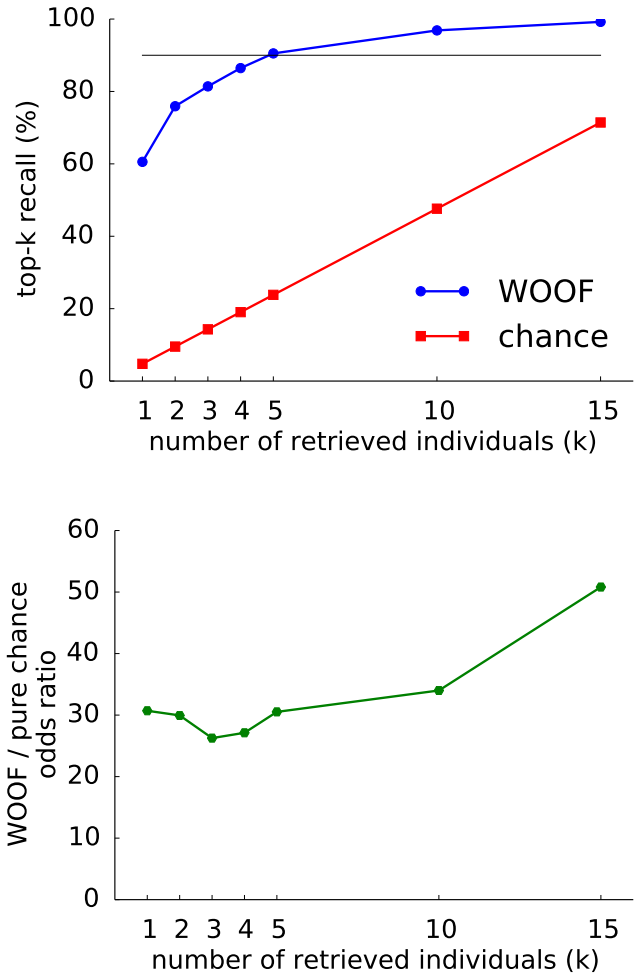

Fig. 6: Above: Top-k recall (whether the correct class was among the k most probable predicted classes) for WOOF in Flickr-dog. For $k = 5$ recall is already above 90%. The red line shows the expected (computed) recall for pure chance. Below: The recall for all k stays well above random chance, with the odds-ratio actually improving for large k.

Figure 7–Above presents an example of a correct predicting of the Oliver query in the first probable class, with a large margin for the second probable class. Figure 7–Below does not predict the correct class for the Dakota query until the fourth probable class. However, the probabilities of the first class and the fourth class are too close, showing that our approach is really robust, as shown by Figure 6–Above.

Fig. 7: Retrieval example for an example query from each one of the breeds. Above: Retrieval using a Pug dog as query, correct predicting its class. Below: Retrieval using a Husky dog, in which it predict correctly only in the fourth retrieved class.

We also provide a more in-depth profile of WOOF, with a confusion matrix (Figure 8). The matrix shows that very few mistakes mix-up dogs of different breeds. As expected, almost all mistakes are between dogs of the same breed. Considering only huskies, accuracy was 75.14%; while for pugs, accuracy was 54.38%. That confirms our expectation that pugs would be harder to identify than huskies.

Neural networks are known to have issues with overfitting, mainly with small datasets, which is the case with our sets, specially Snoopybook. This does not apply to the WOOF method, since it is a descriptor based upon a general purpose network, pre-trained over a completely different set. As for BARK, we trained networks on both datasets, so when it is optimized for a set and tested on another, it is also not

overfitted. In the last case, where network and SVM training is made in the same set, we avoid overfitting by isolating a subset for each phase.

Moreover, a form of assessing if a network is overfitted is applying it over different data. This is done for BARK, as shown in Table 1. The network optimized for Flickr-dog and tested on Snoopybook yielded good results – worse than WOOF, but considerably better than handcrafted descriptors. Snoopybook, being a very small set, produced a weaker network. Nevertheless, its results on Flickr-dog are still better than LBPH.

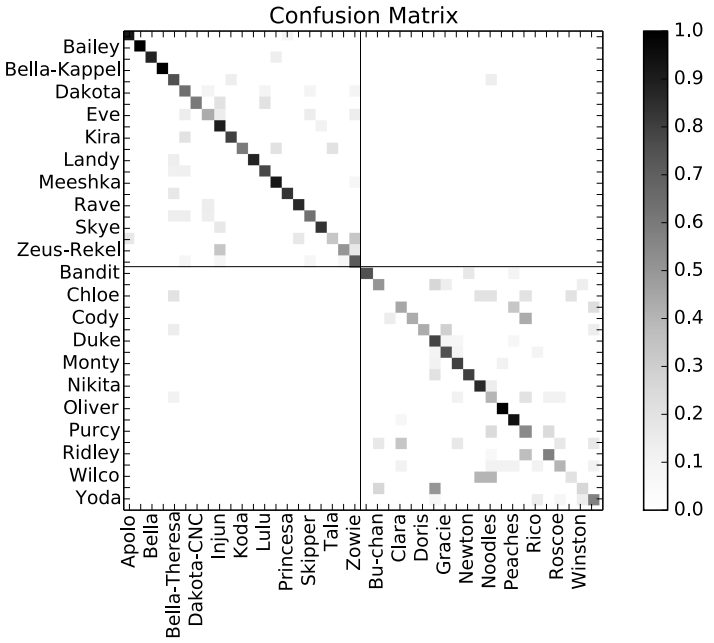

Fig. 8: Confusion matrix for WOOF in Flickr-dog: huskies at top-left (Apolo to Zowie), pugs at bottom-right (Bandit to Yoda). The two breeds appear clearly separated: almost all mistakes were between dogs of the same breeds. As expected huskies were easier to identify. Accuracy for huskies only was 75.14%; for pugs, only 54.38%.

5 Conclusions

We advanced dog identification from a new perspective, using facial features. To the best of our knowledge, our work is the first to address the problem, as existing art mostly sees animals as groups, not as individuals. Although ecologists have growing

interest in identifying individual wild animals, pet identification is a new frontier for computer vision.

Our main motivation is to evaluate if pet biometry can be useful for retrieving lost animals, complementing existing solutions like dog tags, tattoos and micro-chips, that require special forethought, or may be lost.

We introduced two new datasets annotated with individual dog labels: Flickr-dog and Snoopybook. Data acquisition was the bottleneck for growing those datasets, as the manual annotation is very laborious. However, while keeping data acquisition manageable, we have shown that facial recognition in dogs is possible with accuracies much above pure chance. We hope that this work will spark the interest of other groups, and allow more aggressive, collective, efforts of data acquisition.

The performance of off-the-shelf classical Human Facial Recognizers was rather poor for dogs, showing that dog facial recognition is not a trivial extension of human facial recognition. Convolutional Networks, both shallow and deep, proved more successful. A very deep pre-trained OverFeat network — WOOF — showed the best results, with the shallow network with special architecture optimization — BARK — achieving a very decent second best. The performance was very promising, especially when contrasted to the literature that shows that dog recognition is hard even for human dog experts, with 81% accuracy in optimal conditions [10].

Our results also suggest margin for improvement with more training samples per individual. This suggests that for the optimal operation of a real-world system, owners should be stimulated to register as many pictures as possible.

We foresee applications beyond finding lost pets. Ecologists have growing interest in identifying individual wild animals, to detect migration patterns. There is also a forensic interest, in the case of stolen horses and cattle. We would like to address those exciting use cases, extracting features from the rest of the animal, not only the face. For certain animals, the coat can provide clues even more distinctive than the face, but because animals are deformable objects, getting invariant enough features is very challenging.

Acknowledgments

Special thanks to the veterinary doctor Marjorie de Oliveira Franco, along with the Zoonoses Control Center of São José dos Campos - SP, for producing the Snoopybook dataset, and providing it for research. And thanks to Giovani Chiachia for its help on using simple-hp, tips on how to perform the experiments and helping organizing the dataset.

References

1. Ahonen T, Hadid A, Pietikinen M (2004) Face recognition with local binary patterns. In: Pajdla T, Matas J (eds) European Conference on Computer Vision, Lecture Notes in Computer Science, vol 3021, Springer Berlin Heidelberg, pp 469–481, DOI 10.1007/978-3-540-24670-1_36, URL `http://dx.doi.org/10.1007/978-3-540-24670-1_36`

2. Ahonen T, Hadid A, Pietikäinen M (2006) Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 28:2037–2041, DOI 10.1109/TPAMI.2006.244
3. Belhumeur P, Hespanha J, Kriegman D (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7):711–720, DOI 10.1109/34.598228
4. Belhumeur PN, Jacobs DW, Kriegman DJ, Kumar N (2013) Localizing parts of faces using a consensus of exemplars. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 35, pp 2930–2940
5. Bergstra J, Yamins D, Cox D (2013) Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: International Conference on Machine Learning, pp 115–123, URL `http://jmlr.org/proceedings/papers/v28/bergstra13.html`
6. Bradski G (2000) The OpenCV Library. Dr Dobb's Journal of Software Tools
7. Chiachia G, Falcão AX, Pinto N, Rocha A, Cox D (2014) Learning Person-Specific Representations from Faces in the Wild. IEEE Transactions on Information Forensics and Security 9(12):2089–2099, DOI 10.1109/TIFS.2014.2359543, URL `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6905816&tag=1`
8. Cox D, Pinto N (2011) Beyond simple features: A large-scale feature search approach to unconstrained face recognition. In: IEEE International Conference on Automatic Face and Gesture Recognition and Workshops, pp 8–15, DOI 10.1109/FG.2011.5771385
9. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, vol 1, pp 886–893 vol. 1, DOI 10.1109/CVPR.2005.177
10. Diamond R, Carey S (1986) Why faces are and are not special: an effect of expertise. Journal of experimental psychology: General 115(2):107–117
11. Felzenszwalb P, Girshick R, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(9):1627–1645, DOI 10.1109/TPAMI.2009.167
12. Fisher RA (1936) The use of multiple measurements in taxonomic problems. Annals of Eugenics 7(2):179–188, DOI 10.1111/j.1469-1809.1936.tb02137.x, URL `http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x`
13. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:12070580 DOI arXiv:1207.0580, URL `http://arxiv.org/abs/1207.0580,1207.0580`
14. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems pp 1–9
15. Liu J, Kanazawa A, Jacobs D, Belhumeur P (2012) Dog breed classification using part localization. In: European Conference on Computer Vision, Springer, pp 172–185

16. Lowe D (1999) Object recognition from local scale-invariant features. In: International Conference on Computer Vision, vol 2, pp 1150–1157 vol.2, DOI 10.1109/ICCV.1999.790410
17. Parkhi OM, Vedaldi A, Zisserman A, Jawahar C (2012) Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 3498–3505
18. Pinto N, Stone Z, Zickler T, Cox D (2011) Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp 35–42, DOI 10.1109/CVPRW.2011.5981788
19. Scapinello K, Yarmey A (1970) The role of familiarity and orientation in immediate and delayed recognition of pictorial stimuli. Psychonomic Science 21(6):329–330, DOI 10.3758/BF03335807, URL http://dx.doi.org/10.3758/BF03335807
20. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y (2014) Overfeat: Integrated recognition, localization and detection using convolutional networks. In: International Conference on Learning Representations, CBLS, URL http://openreview.net/document/d332e77d-459a-4af8-b3ed-55ba
21. Sirovich L, Kirby M (1987) Low-dimensional procedure for the characterization of human faces. Journal of the Optical Society of America A 4(3):519–524, DOI 10.1364/JOSAA.4.000519, URL http://josaa.osa.org/abstract.cfm?URI=josaa-4-3-519
22. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition
23. Turk M, Pentland A (1991) Face recognition using eigenfaces. In: IEEE Conference on Computer Vision and Pattern Recognition, pp 586–591, DOI 10.1109/CVPR.1991.139758
24. Wang X, Ly V, Sorensen S, Kambhamettu C (2014) Dog breed classification via landmarks. In: IEEE International Conference on Image Processing, IEEE, pp 5237–5241
25. Wiskott L, Fellous JM, Krüger N, Von Malsburg CD (1997) Face recognition by elastic bunch graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 19:775–779, DOI 10.1109/34.598235
26. Xu Y, Zhang D, Yang J, Yang JY (2011) A two-phase test sample sparse representation method for use with face recognition. Circuits and Systems for Video Technology 21(9):1255–1262, DOI 10.1109/TCSVT.2011.2138790

View publication stats