DOG BREEDS CLASSIFICATION THROUGH MACHINE AND DEEP LEARNING

Micol Cozzarizza - Carlotta Polidoro

October 2023

Abstract

Classification of different dog breeds is a challenging task, due to strong similarities and dissimilarities that can be observed among existing breeds. This project explores different methods to perform classification of dog images in one of 120 different breeds. The purpose of this is to analyze two different approaches to carry out such classification, based respectively on Machine Learning with Feature Descriptors and on Convolutional Neural Networks, with the end goal of comparing the accuracy of the two identified models.

The project follows a well-structured approach for both the selected models, by covering all the steps of the Machine Learning pipeline:

- data acquisition
- data processing
- feature extraction
- model
- prediction

The Machine learning approach consists on training three different classifiers (SVM, RF, AdaBoost) with the images data obtained applying two feature descriptors and three different image sizes, and comparing the results obtained through accuracy.

On the other hand, the Convolutional Neural Network model has been built by using the MobileNet CNN pre-trained on the Imagenet dataset, and Tensorflow method for its final evaluation.

The final results obtained demonstrate how the two considered approaches give significantly different results, with the CNN model outperforming the Machine Learning one.

1 Introduction

According to the most popular groups that govern the registration of canine breeds, there exist between 195 and 500 different dog breeds in the world. These are spread in most countries and regions around the world, and many of these regions are home to their own special breeds that originated there, and showcase unique characteristics inherent to their place of origin. On the other side, many dog breeds present remarkable similarities, but they are considered as separate breeds due to some subtle details. Due to very strong diversities and similarities, classification of dog breeds can be an intricate task in the absence of appropriate expertise Nonetheless, dog breed identification is essential for many reasons, particularly for understanding individual breeds' conditions, health concerns, interaction behavior, and natural instinct.

This project has the purpose of building a model that is able to automatically learn how to classify fine-grained dog images into different breeds, and to apply the knowledge to new samples. Classification has been carried out by exploiting two different techniques involving Machine Learning and Feature Descriptors on one side, and Deep Learning with Convolutional Neural Networks on the other. The difficulty of such task has been eased through the utilization of readily available tools and libraries, such as the Scikit-learn or the TensorFlow libraries, and pre-trained architectures.

2 Related work

2.1 Dog Breed Classification Using Deep Learning (2021)- Akash Varshney, Abhay Katiyar, Aman Kumar Singh, Surendra Singh Chauhan

This study proposes two classification approaches based on Deep Learning, performed on the Stanford Dogs Dataset. In particular, the researchers worked on the concept of transfer learning, which deals with data augmentation technique with its properties to increase the size of dataset. They propose two pre-trained models based on two different kinds of Neural Networks: Inception V3 and VGG16. In the end, the model that provided the best result was the one based on the Inception V3 CNN, with an accuracy score of 0.85. [1]

2.2 Deep Residual Learning for Image Recognition (2015)- Kaiming He

This paper focuses on the training of neural networks that are deeper than those used previously, by presenting a residual learning framework to ease the training of such networks. By reformulating the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions, the authors were able to create networks able to gain accuracy from increasing depth. Compared to the VGG nets, they were able to create residual networks that are 8 times deeper but achieve incredible error values such as 3.57% on the ImageNet test set. The ability to represent depth is crucial for many visual recognition tasks such as fine-grained image classification. [2]

2.3 Dog Breed Identification Using Deep Learning (2019) - Zalan Raduly

The paper present a multi-class classification problem, aimed to determine the dog breed from a given image. Two models based on two different CNNs are proposed, both trained on the Stanford Dogs Dataset. The paper is focused on the fine-tuning of hyperparameters, which is carried out through 5-fold cross-validation, producing 5 different training and validation subsets. The authors propose a model based on Inception-Resnet V2 CNN, and a model based on NASNet-a architecture. During model evaluation, Inception Resnet V2 proved to be the most appropriate model, with an accuracy of 93.66%. The authors point out that the superior performance is due to the greater depth of this CNN. [3]

3 Proposed Methods

3.1 Data Acquisition and Exploration

For our project, we decided to collect data considering publicly available datasets. After conducting some research on Kaggle, we found the Standford

Dogs Dataset, which contains images of 120 breeds of dogs from around the world, and we were able to download it for free. This dataset has been built using images and annotation from the ImageNet database, originally for fine-grain image categorization, a challenging problem as certain dog breeds have near identical features or differ in colour and age. The dataset is already divided into the different 120 dog breeds. Each class is composed of approximately 200 images, for a total number of 20,580 images.

As a first step, we loaded the dataset into our data analysis environment: for this we relied on the Google Colab service. Thanks to this, we could better understand the structure of the data and gain initial insights into their characteristics and potential challenges we may encounter.

3.2 Model 1: Machine Learning and Feature Descriptors

Machine Learning systems are able to automatically learn from the provided training data, and to generalize the knowledge on new and different testing data.

With high-dimensionality data like images, the learning phase might result extremely complex: this is when the introduction of a feature extraction step comes handy.

Feature extraction involves transforming raw data into a more emblematic and compact representation. Among the most common existing methods for feature extraction in computer vision, we used Histogram of Oriented Gradients(HOG) and Local Binary Patterns (LBP).

- HOG: HOG is a widely used visual feature descriptor generally useful to describe the shape of an object. It captures local gradient information by dividing an image into smaller cells, computing gradient orientations within each cell, and constructing histograms to store these orientations.
- LBP: LBP is a texture descriptor widely used for face recognition, texture classification, and object detection. It encodes the local texture information by comparing the intensity values of pixels in a neighborhood to a central pixel and creating from this a binary pattern.

The feature extraction step has been implemented through a function that requires, as parameters, the specification of both the desired feature descriptor and image size to use during computation.

The new more compact data obtained after applying such feature descriptors have been used to populate, in a random way, the corresponding training (60% of the data), validation (10% of the data) and testing set(30% of the data).

Once terminated the feature extraction step, we started with the definition and training, validation and testing of the different classifiers we aim to analyze, through a combination of these together with different feature descriptors and image sizes, to eventually identify the approach that result in a superior accuracy. Specifically, classifiers considered are:

- Support Vector Machines: SVM works by finding the optimal hyperplane, meaning the decision surface most able to maximally separate different classes in a high-dimensional feature space. The hyperplane is determined by identifying support vectors, which are the data points closest to the decision boundary. SVM can be used for both linear and nonlinear classification tasks, in this second case by utilizing different kernel functions able to map the data into a higher-dimensional space. In our case, an implementation of SVM specifically designed for classification task has been used: SVC. It utilizes the same principles and mathematical formulation as SVM but is specifically tailored for classification purposes.

- AdaBoost: it is a multi classifier approach based on boosting. Through AdaBoost, we try to build a strong classifier by combining several weak classifiers, since the mistakes of previous classifiers are learned by their successors.

- Random Forest: it is a multi classifier approach based on bagging. In Random Forest, a collection of decision trees is created: each tree is fitted on a random sample with replacement of the training set and in each decision node, the choice of the best split feature is made considering only a random subset of these. During the training, each decision tree independently makes predictions, and the final decision is determined by aggregating them through a majority voting approach.

As a final step, the accuracy of the various combinations of different feature descriptors, image sizes and classifiers is compared. Results will be further analyzed in a following dedicated section (Section 4).

3.3 Model 2: Deep Learning and Convolutional Neural Networks

Deep learning models are based on Neural Networks, which are stacked layers of artificial neurons. In particular, Convolutional Neural Networks (CNNs) are a type of neural networks specifically designed to process images, by automatically learning and extracting relevant features from visual data.

In a traditional architecture, CNN consists of multiple layers stacked one on top of the other, including convolutional layers, pooling layers and fully connected layers. The convolutional layers perform local operations by applying kernels(filters) to the input data to capture spatial patterns and features, gathered in a feature map. The pooling layers downsample the spatial dimensions, reducing the complexity of the network while preserving important features. The fully connected layers connect all neurons in one layer to those in the next layer, enabling high-level feature learning and classification.

To implement our classification model based on CNN, we relied on the use of TensorFlow and Keras as Deep Learning frameworks. The data were loaded using already developed methods. We split the dataset with 80% of data in the training and the remaining 20% in the validation set.

As a next step, the architecture of our Convolutional Neural Network (CNN) was defined. In particular, we decided to use one proposed in the literature : MobileNet CNN. We used such architecture pre-trained on the Imagenet dataset, and therefore downloaded the weights of the network from the official storage. For it, we had to provide some model adjustment since it has 1000 classes, i.e. the final dense layer has 1000 neurons, while in our classification task we have only 120 classes: we adapted the architecture by removing the last layer and creating a new dense layer with only 120 neurons. Specifically, we were able to do so by relying on the sequential model provided in TensorFlow: through it, we can define a model as a sequence of layers that we can access, and every layer has an input and output attribute.

After creating the model architecture, we had to define also:

- number of epoch: after several trials, we decided to set the number of epochs to 30
- model saving: after every epoch
- optimizer: Stochastic Gradient Descent
- loss function: Categorical Cross Entropy
- callbacks: an object (method) to perform some actions at various stages of training (e.g. at the start or end of an epoch, before or after a single batch, etc).

Before the training phase, the model has to be compiled. As a final step, we relied on the Tensorflow method to easily evaluate the trained network on a test dataset and on the TensorBoard module (provided with Tensorflow) to visualize the learning curves.

4 Results

4.1 Machine Learning Model

The tables below illustrate the accuracies values resulting from the different combinations of feature descriptors, image sizes and classifiers.

	hog + 32	hog + 64	hog + 128
SVM	0.021	0.032	0.041
RF	0.020	0.018	0.018
AdaBoost	0.012	0.015	0.015

	lbp + 32	lbp + 64	lbp + 128
SVM	0.016	0.016	0.018
RF	0.021	0.015	0.016
AdaBoost	0.014	0.011	0.011

As we can see from the provided results, the machine learning approach is not suitable for our case. In addition to this, also the computing time is inefficient, resulting for some of the considered combination in hours.

We hypothesize this is due to:

- extremely high dimensionality of the many input data
- very similar images
- images highly detailed

Indeed, the implemented Machine Learning models completely fails in classifying the dog images in their corresponding breed: the feature extractors are not powerful enough to efficiently extract relevant features that enables images distinction, the dimensionality of the data is too high for the approaches to work correctly, the computing time significantly increases with the number of samples and their sizes and may be impractical beyond tens of thousands of them.

4.2 Deep Learning Model

The deep learning approach based on Convolution Neural Networks adopted demonstrated the superiority of such models with significantly greater accuracy results than the machine learning model: the accuracy obtained with the selected configuration is of 0.9307.

The reason for this are to be found in the better capability of Convolutional Neural Networks in extracting and reducing the most relevant information from the input images by applying filters and pooling operations, making them able to learn increasingly complex features that eventually provides for a more precise image classification

5 Conclusions

The project aimed at analyzing and comparing two different approaches for the classification of dog breeds, a very challenging task due to strong similarities and dissimilarities that can be observed among them. Indeed, the machine learning approaches analyzed, based on a combination of different classifiers and feature extractors, has resulted too weak and inadequate to support the classification of such images. On the other hand, the approach based on Convolutional Neural Networks has proven to be a considerably better alternative in a context characterized by high dimensionality and detailed data like dog images.

5.1 Peer Review

For the overall project, the work was divided equally so as to everyone giving its contribute. The code was developed together but some running session and parameter testing where done individually. For the paper drafting, we divided the development of each part, eventually reviewed by each other to double check them.

References

[1] Aman Kumar Singh Surendra Singh Chauhan Akash Varshney, Abhay Katiyar. Dog breed classification using deep learning. *International Conference on Intelligent Technologies (CONIT)*, 2021.

[2] Kaiming He. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] Zalan Raduly. Dog breed identification using deep learning. *Conference: 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, 2018.