report

by Report Report

Submission date: 19-Jul-2022 08:22PM (UTC-0700)
Submission ID: 1872865757
File name: DEEP_LEARNING_DOG_BREED_CLASSIFIER.docx (20.18K)
Word count: 1486
Character count: 7646

DEEP LEARNING DOG BREED CLASSIFIER

1.1 Problem Statement

The focus of this project is to write an algorithm that can be deployed in an application for dog classification. A dog or a human can be spotted in an image, and if so, an estimate of the dog's breed or the dog breed most similar to the human can be provided. The algorithm will be implemented within the program to address this problem of dog identification. This algorithm will potentially be based on a convolutional neural network classifier that will perform the classification.

1.2 Project Overview

1.2.0 Domain Background

The concept of computer vision is used in attempting to solve the problem of classifying the dog and human images. Computer vision aims to characterize and recreate the world's features, such as shape, illumination, and color distributions, from a set of photographs. [1] In particular, we explore the concept of convolutional neural networks that use a hierarchical structure to' learn' image features. As you go deeper into the layers, we will start with lower-level features and build them up into larger, higher-level features. [2]

1.2.1 Datasets and Inputs

This project uses two datasets to build, train and test the algorithm that we will be developed, namely:-

- Dog dataset is obtained from Udacity AI; it contains 8351 images of dogs. The dogs' dataset will be used to train and test the model that will be built.
- Human dataset is also obtained from the Udacity AI. It contains 13233 images of humans. The human dataset will be used to test to guide how the final algorithm is designed and also be used to test it.

1.2.2 Solution Statement

The solution is to build a convolutional neural network model trained on the data specified above to perform the dog breed classification within the dog application. Given the small size of the dataset, transfer learning is to be used to attain web/mobile application usable accuracy. Transfer learning involves using the lower hidden layers of an already existing pre-trained model that solves a similar task. This speeds up training considerably and also works with limited data. [2]

1.3 Metrics

Given that this is a classification problem, the evaluation is based on the model's accuracy on the test set. The model should reach more than 60% accuracy on the test set.

2.0 Analysis

2.1 Data Exploration

As described in the Sections above, the project uses two datasets, i.e., the human faces dataset and the dog dataset.

- The human dataset contains 13233 images, 100 of which are used to assess the performance of the chosen face detector model.

- Dog dataset is the reference dataset for this project as it is used to train the dog breed classifier. The dataset contains 8351 color images of dogs, representing 133 different breeds of dogs. The dataset comprises truncated images; the solution involves case handling for when that occurs.

- The images in the dog dataset are of different sizes, which requires that they are resized to a uniform shape when used as input to the model.

2.2 Algorithms and Techniques

- Haar feature-based cascade classifiers: this is a technique for object detection including face recognition. [3] The face detector part of the algorithm is built using this model. The model provides a pre-trained face detector with acceptable performance i.e. detecting 100% of faces in the sampled human images and 12% on the sampled dog dataset.
- VGG16 - Dog Detector [4] to Implements the dog breed classifier. The model has been pre-trained on a much bigger dog inclusive dataset, hence providing a robust starting point for training a custom dog breed classifier. Using a pre-trained model also allows us to use a small dataset and achieve acceptable performance as per our evaluation targets.

3.0 Methodology

4.0 3.1 Data Preprocessing

The dog dataset comes structured in a format suitable for a classification task and split across the train, validation and test sets. No work was done on this effort. The train set contains 6680 images, the test set contains 836 images and the validation contains 835 images. Specific preprocessing steps followed are: The dataset contains truncated images, as such a solution was written to handle these cases when loading the images. The images are then resized to a uniform size, and center cropped under the assumption that the subject is generally in the center of image, these steps reduces the noise in a particular image. The dataset is then augmented with a random rotation and random horizontal flip, data augmentation improves the generalization of the model to data not seen before. Finally, the dataset is converted to a tensor, and normalized.

3.2 Implementation

3.2.1 Data Preprocessing

The dog dataset is loaded and or made ready for training, first the locations of the data for each set train, test and validation are initialized. The data is then loaded and the defined transformations and augmentations as discussed in Section 3.1 are implemented.

3.2.2 Modeling

3.2.2.1 Modeling without transfer learning

Model: A convolutional neural network is defined from scratch; the architecture includes five convolutional layers each with a filter of size 3x3. Three fully connected layers are then added defining the classifier part of the model. Pooling layer follows each layer of convolution which reduces the spatial dimension by a factor, this ensures that the number of parameters doesn’t explode affecting memory usage and computational load. Two dropout layers are added to reduce the risk of overfitting.

Loss function: of Cross entropy was used, a suitable loss function for classification tasks.

Optimizer: Adam optimizer [5] was used with a learning rate of 0.001, albeit a small learning rate which is not ideal, the accuracy metric was met.

3.2.2.2 Modeling using transfer learning

Model: A VGG16 [4], is defined using its pre-trained weights. The hidden layers of the model are frozen to preserve the pre-trained weights. The classifier of the model was updated to fit this particular use case i.e. the output of the last fully connected layer is set to 133 given the 133 dog classes in the dog dataset. The same loss function and optimizer as defined in Section 3.2.2.1 are reused for this model training.

3.2.2.3 Training

Both models are trained the same way, with a difference in the number of epochs, the model from scratch is trained for 50 epochs, given the small learning rate and the exploratory nature of the training process. The model with the lowest loss on the validation set is saved and used for inference later on.
The transfer learning is trained for 10 epochs as it is expected that it converges easily, the best model just like in the process before is saved. The batch size for the model from scratch is 32, while for the transfer learning model is 64. The batch size is chosen solely on the available computing resources.

3.2.3 Algorithm Implementation

The solution is implemented as three if clauses:

1. The first if clause has a human face detector based on OpenCV implemented, it checks if an input image contains a human face, this returns a true or false response. If true, then the dog breed classifier model, trained above predicts the most likely breed that the human resembles.

2. The second if clause checks if an image contains a dog using the pretrainedVGG16 model, returns true or false. If true, the dog breed classifier predicts the most likely breed that the dog is.

3. Finally, the last clause handles the scenario in which the image doesn't belong to either human or dog, returns an error message.

3.3 Refinement

The initial model is the model built from without use of transfer learning however its performance does not meet the accuracy requirements that could solve our problem. Which necessitates the training of an alternative model using transfer learning, giving the final model used, VGG16 which achieves the set accuracy requirements.

4.0 Results

4.1 Model Evaluation and Validation

The final model chosen for use, is the VGG16 model based on transfer learning, this model achieves 73% accuracy on the test set. The model contains a classifier with three layers fully connected each with the RELU activation function apart from the output layer and two dropout layers. With this accuracy, our project adequately solves the set problems in the problem statement as per defined requirements.

4.3 Algorithm Results

Included below is a sample output from the implementation of the algorithm.

References

[1] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag London Limited, 2011.

[2] Aur´elien G´eron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2019.

[3] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

report

ORIGINALITY REPORT

2%	1%	1%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Shuo Tian, Kai Hu, Shasha Guo, Shiming Li, Lei Wang, Weixia Xu. "Evolutionary Recurrent Neural Architecture Search", IEEE Embedded Systems Letters, 2021 Publication	1%
2	annalsofrscb.ro Internet Source	1%
3	publications.aston.ac.uk Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

report

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5