CNN Project Report: Dog Breed Classifier

1.0 Definition

1.1 Problem Statement

The focus of this project is to write an algorithm for a dog classification application. The algorithm will be deployed within the application to solve the problem of dog identification i.e. given a user-supplied image, if a dog is detected in the image, it provides an estimate of the dog's breed or if a human is detected, it provides an estimate of the dog breed that is most resembling. 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. In computer vision, the objective is to describe the world in one or more images and to reconstruct its properties, such as shape, illumination, and color distributions. [1]

In particular, we explore the concept of convolutional neural networks, that use a hierarchical structure to 'learn' image features, starting with low-level features in the first hidden layer, then assemble them into larger higher-level features in the next hidden layer and so on across the depth of the layers. [2]

1.2.1 Datasets and Inputs

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

- Dog dataset, the dataset is obtained from Udacity AI Nano Degree course , it contains 8351 images of dogs. The dogs dataset will be used to train and test the model that will be built.
- Human dataset, this dataset is also obtained from the Udacity AI Nano Degree, 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 so as to attain web/mobile application usable accuracy.Transfer learning involves the use of 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 this is a classification problem, the evaluation is be based on the accuracy the model achieves on the test set, particularly, the model should achieve more than 60% accuracy on the test set.

2.0 Analysis

2.1 Data Exploration

As described in Section 1.2.1, 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 choosen face detector model.
- Dog dataset, this is the reference dataset for this project as it used to train the dog breed classifier, the dataset contains 8351 color images of dogs, representing 133 different breeds of dogs. The dataset contains truncated images, as such 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

Algorithm/Technique	Purpose	Justification
Haar feature-based cascade classifiers: A machine learning based approach 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]	Implements the dog detector part of the algorithm	The model is pretrained on IMAGENET, a dataset of about 14 million images containing among other things images of dogs, the model provides acceptable performance detecting 98% of the sampled dog faces, and only detecting 1% of the sampled human faces.
VGG16 - Transfer Learning	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.
Data Augmentation	Artificially increases the size of the dataset	Data augmentation improves the generalization of the model on unseen data.

2.3 Benchmark

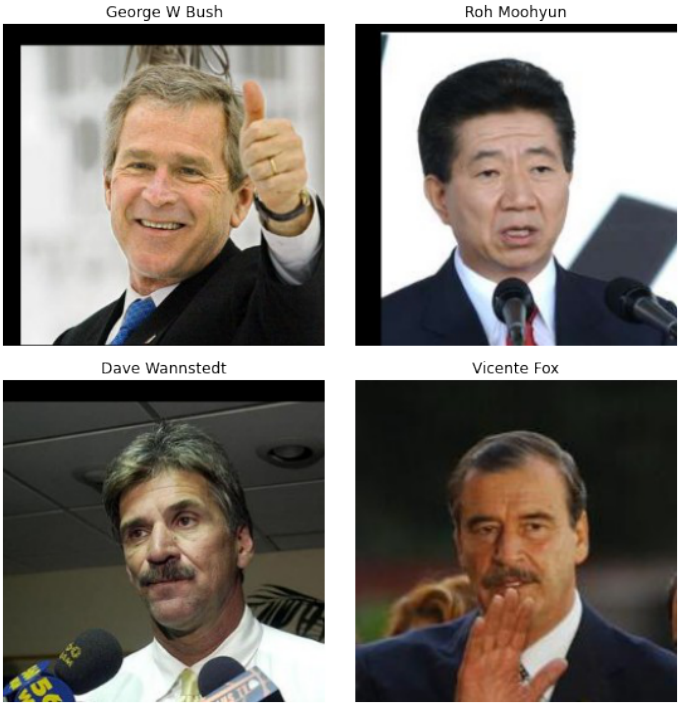
The solution is bench marked against the VGG16 state-of-the-art model. The VGG16 model achieves 92.7% top-5 test accuracy on ImageNet, a dataset containing over 14 million images belonging to 1000 classes. [4]

2.4 Exploratory Visualization

Visualizations were performed on the two datasets as shown below to understand the structure of the images in the dog dataset, the images are of different sizes as observed below.

Figure 1: Example images from the dog dataset.

Figure 2: Example images from the human dataset.



3.0 Methodology

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 are discussed below:

- 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, this 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 intialized. 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 from scratch

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. Each convolutional layer is followed by a pooling layer 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: Cross entropy loss function 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 pretrained VGG16 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 scratch however its performance does not meet the accuracy requirements as set in Section 1.3. 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 fully connected layers each with the RELU activation function apart from the output layer and two dropout layers.

4.2 Justification

While the model doesn't beat the benchmark performance cited in Section 2.3. It achieves the set accuracy requirements Section 1.3, hence the 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.

Figure 3: Dog detected.

Figure 4: Human detected

References

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

[2] Aurélien Géron. *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.