DOG BREED CLASSIFICATION

- AUHONA CHAKRABORTY

TABLE OF CONTENTS

1

INTRO

Intro to Project

2

DATA

Data collection
and
preprocessing

3

MODEL

Model
Architecture &
training

4

GUI & CON

GUI
Implementation
& Conclusion

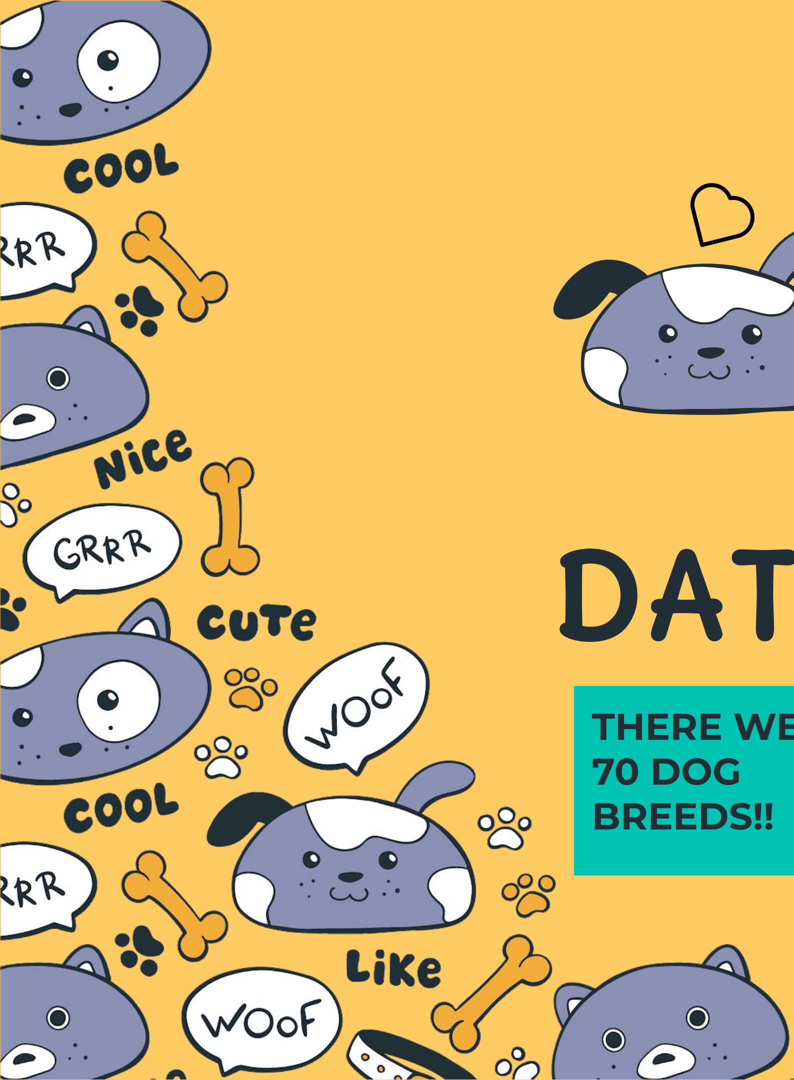
GRRR

WOOF!

HAPPINESS IS A WARM PUPPY! - Charles M. Schulz

INTRO

Dogs, diverse in breed and beloved by many, are the focus of my deep learning project. Using advanced algorithms, I aim to automate accurate dog breed classification from images. Join me on a journey through data collection, model architecture, and GUI implementation.



DATA

THERE WERE 70 DOG BREEDS!!

In the collection phase, a diverse set of images was obtained from Kaggle, ensuring a representation of various breeds. Quality checks were done to remove duplicate images. Images were standardized by resizing to a uniform dimension of 224x224 pixels. The dataset was partitioned into training, validation, and testing.

MODEL ARCHITECTURE & TRAINING

MODEL

The InceptionV3 architecture, a pre-trained convolutional neural network (CNN), was chosen for its superior performance in image classification tasks.

Transfer

Leveraging transfer learning, the pre-trained InceptionV3 model was used as a feature extractor, making efficient learning of dog breed features without the need for extensive training data.

KERAS

TensorFlow, an open-source machine learning library, and Keras, a high-level neural networks API, were employed for model implementation and training

IMPROVE

While the base layers of InceptionV3 were kept frozen to preserve the learned features, the top layers were fine-tuned during training. Val accuracy > 95%

MODEL ARCHITECTURE CONTINUED...

Training Process

The model was trained using image data generators provided by the Keras library, enabling real-time data augmentation and batch processing.

Evaluation

During training, the model's performance was evaluated using metrics such as categorical cross-entropy loss and accuracy to assess its ability to correctly classify

GUI

GUI Development

The code utilizes Tkinter, a Python library, for creating the GUI. It initializes a Tkinter window

Button Widget

The "Upload Image" button is made using Tkinter. It triggers the upload_image() function, allowing users to select an image

Label

Tkinter Label widgets are employed to display text on the GUI.

Message Box

Tkinter's built-in file dialog and message box functionalities are utilized for file selection and displaying messages

Main Loop

The main loop (root.mainloop()) ensures the continuous responsiveness of the GUI

CONCLUSION

summary

the project successfully demonstrated the effectiveness of deep learning in automating dog breed classification tasks

Future

incorporating additional dog breeds, improving model robustness, and refining the user interface for better user experience.

Impact

This project contributes to advancing field of computer vision and offers benefits to pet owners, veterinarians, and animal enthusiasts..

Achievement

Through meticulous data collection, development, and GUI implementation, I created a comprehensive solution for accurate breed prediction

THANK YOU 💙🧡