horizontal line

Dog vs Cat Classification Using Transfer Learning

Monday, 08.01.2024

**─**

# Mohamad Jarad

215060048

Software Engineering

Image Processing

# Abstract

The Dog vs Cat Classification project is a machine learning endeavor that aims to automate the identification of animal images, categorizing them accurately into dogs and cats. This report outlines the project's objectives, its significance in the broader context of AI, its practical applications, and the resources utilized throughout the project's development.

# Introduction

In the digital age, the sheer volume of image data available presents both a challenge and an opportunity for automation. The classification of images, particularly for recognizing pets like dogs and cats, is a task that has seen significant demand across various sectors.

# Purpose

The project's purpose is to develop a machine learning model capable of distinguishing between images of dogs and cats with high accuracy. This is achieved through the application of Transfer Learning, leveraging pre-trained neural networks to capitalize on existing knowledge

# Purpose

The ability to classify images with precision is crucial for numerous applications, from organizing personal photo libraries to enhancing search engine capabilities. In scientific research, such technology can aid in biodiversity studies, and in social media platforms, it can improve content filtering and user experience.

# Area of Use

## The use cases of such a classification system are diverse, including:

1. Digital photo organization tools
2. Content moderation in social media
3. Animal welfare and surveillance
4. Veterinary research and diagnostics
5. Automated pet detection in security footage

# Materials Used

## The project was developed using the following resources

1. **Data Source**: A large dataset from Kaggle's "Dogs vs Cats" competition.
2. **Programming Tools**: Python, TensorFlow, Keras, NumPy, and OpenCV for image processing.
3. **Hardware**: High-performance computing resources, potentially using GPU acceleration for model training.
4. **Software**: Jupyter Notebooks on Google Colab for interactive development and testing.
5. **Machine Learning Model**: The MobileNet V2 neural network pre-trained on ImageNet.

# Methodology

The methodology encompassed data acquisition, preprocessing (resizing and normalization), dataset splitting, model training and evaluation, and the development of a predictive system for real-world testing.

# Results

The model demonstrated a high degree of accuracy, outperforming baseline models trained from scratch. The use of Transfer Learning proved to be efficient in terms of training time and resource usage.

# Conclusion

This project underscores the effectiveness of Transfer Learning in image classification tasks. The success of the Dog vs Cat Classification model opens the door to further research and application development in automated image recognition.

# Future Work

Future enhancements could include expanding the model's capabilities to recognize additional animal species, improving real-time classification performance, and integrating the model into mobile applications for wider accessibility.

# Acknowledgments

We extend our gratitude to the online machine learning communities, dataset contributors, and the developers of the open-source tools that made this project possible.

# Appendix

## The use cases of such a classification system are diverse, including:

### Appendix A: Image Preprocessing

* Resizing and Format Conversion

|  |
| --- |
| for i in range(2000):  img = Image.open('/content/train/' + filenames[i])  img = img.resize((224, 224))  img = img.convert('RGB')  img.save('/content/image resized/' + filenames[i]) |

This code resizes and converts the images to RGB format, standardizing them for the model.

## 

### Appendix B: Visualization of Preprocessed Images

* Displaying Sample Images

|  |
| --- |
| img = mpimg.imread('/content/train/dog.8298.jpg') plt.imshow(img) plt.show()  img = mpimg.imread('/content/train/cat.4352.jpg') plt.imshow(img) plt.show() |

### Appendix C: Image Preprocessing

* Resizing and Normalizing Images

|  |
| --- |
| *# Resizing the images to 224x224 pixels* img = img.resize((224, 224)) img = img.convert('RGB') |

### Appendix D: Label Assignment

* Assigning Labels to Images:

|  |
| --- |
| *# Assigning labels: 0 for cats and 1 for dogs* labels = [1 if filename.startswith('dog') else 0 for filename in file\_names] |