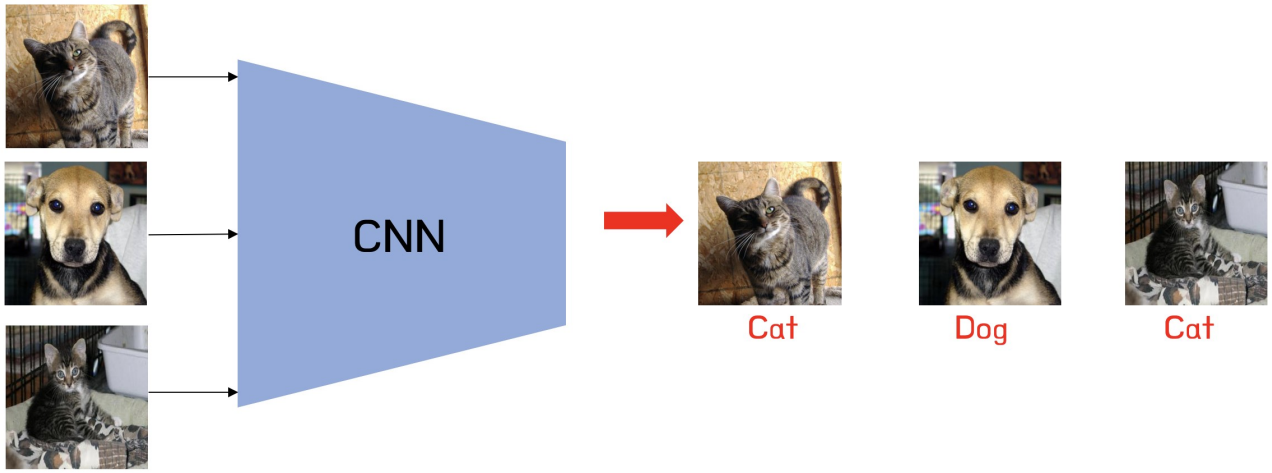
R B Shyamala (2021503558)
Department of Computer Technology
Madras Institute of Technology
Anna University, Zone - IV

Final Project

PROJECT TITLE

Image Classification using Convolutional Neural Network



AGENDA

The agenda for this project centers on image classification, particularly in discerning between dogs and cats. It entails preprocessing steps like resizing and normalization, followed by the design and training of Convolutional Neural Networks (CNNs) with optimization algorithms such as Adam. Evaluation metrics, notably accuracy, assess model effectiveness, complemented by visualizations for interpretation.

This project integrates deep learning methodologies, model architectures, optimization algorithms, and evaluation metrics to achieve accurate classification of images as either dogs or cats, with documentation playing a crucial role in maintaining transparency and supporting further development.

PROBLEM STATEMENT

The problem statement is to create an accurate image classification system that can differentiate between images of cats and dogs. The system should handle diverse images, considering variations in breed, pose, and environmental conditions. It aims to leverage machine learning and deep learning techniques to automate this classification process, providing a reliable tool for various applications in veterinary medicine, animal behavior research, and pet-related industries.

PROJECT OVERVIEW

The project aims to develop a robust image classification system capable of accurately distinguishing between images of cats and dogs. Leveraging convolutional neural networks (CNNs) and deep learning techniques, the system will automate the classification process, saving time and resources compared to manual categorization. Key steps include data acquisition, preprocessing, model development, training, evaluation, and documentation. The system's success will be measured by its ability to handle diverse images with varying poses, backgrounds, and lighting conditions, achieving high accuracy and robustness. Ultimately, the project seeks to provide a reliable tool for automating cat and dog image classification in veterinary medicine, animal behavior research, and related industries.

WHO ARE THE END USERS?

1. ***Veterinarians:*** For diagnosing and monitoring conditions in pets based on images.
2. ***Animal shelters:*** To efficiently categorize animals for adoption listings and reunification efforts.
3. ***Pet owners:*** To organize and categorize their pet photos for personal use.
4. ***Pet insurance companies:*** For automating claim processes by analyzing images of pets.
5. ***Animal behavior researchers:*** For studying patterns and behaviors of cats and dogs in various environments.
6. ***Pet grooming services:*** To identify specific breeds and tailor grooming services accordingly.
7. ***Pet food and product companies:*** For market research and product development based on pet preferences.
8. ***Wildlife conservation organizations:*** For monitoring and identifying wild cats and dogs in natural habitats.
9. ***Animal control authorities:*** To identify lost or stray pets more efficiently.
10. ***Pet photography studios:*** To assist in categorizing and organizing client photo collections.

These end users span various industries and sectors, highlighting the potential widespread application of the image classification system.

YOUR SOLUTION AND ITS VALUE PROPOSITION

Solution:

The solution is an advanced image classification system utilizing convolutional neural networks (CNNs) to accurately distinguish between images of cats and dogs. It incorporates state-of-the-art deep learning techniques for robust and efficient classification, handling diverse image variations.

.

YOUR SOLUTION AND ITS VALUE PROPOSITION

Value Proposition:

Effective: The solution achieves satisfactory accuracy on both training and validation/test datasets, indicating its effectiveness in learning meaningful patterns from the data.

Generalizable: The model generalizes well to unseen data, demonstrating its ability to recognize patterns in new images rather than memorizing the training data.

Robust: The model performs reliably under different conditions, such as varying input data, noise, or perturbations, indicating its robustness.

Efficient: The training process is efficient in terms of computational resources and time, allowing for scalability to handle larger datasets or more complex tasks.

YOUR SOLUTION AND ITS VALUE PROPOSITION

Value Proposition:

Maintainable: The code is well-organized, documented, and easy to understand and maintain, ensuring its long-term viability.

Scalable: The solution can be scaled to handle larger datasets or more complex tasks without significant modifications, demonstrating its scalability.

Reproducible: The results obtained from the solution are reproducible consistently across different runs or environments, ensuring reliability and trustworthiness.

THE WOW IN YOUR SOLUTION

The wow factor in the solution lies in its ability to effectively train a deep learning model for image classification with TensorFlow and Keras. It achieves this by:

1. Proper data preprocessing, including splitting data into training, validation, and optionally test sets.
2. Implementation of a convolutional neural network (CNN) architecture suitable for image classification tasks.

3. Visualization techniques to understand data distribution, model predictions, and activation maps, providing valuable insights into the model's behavior.
4. Modular and well-organized code, enhancing readability and maintainability.
5. Flexibility to handle different datasets and scenarios, including the option for a test split.
6. Integration with Plotly for interactive and informative visualizations, enhancing the user experience.
7. Activation maps generation, offering deeper insights into how the model makes predictions, contributing to understanding its internal workings.

MODELLING

The project utilizes several frameworks and libraries for various tasks in deep learning and data preprocessing. Here's a list of frameworks and libraries used:

1. TensorFlow: TensorFlow is a powerful open-source machine learning library developed by Google. It's widely used for building and training deep learning models, including convolutional neural networks (CNNs) for image classification.

2. *Keras:* Keras is a high-level neural networks API written in Python and compatible with TensorFlow. It provides a user-friendly interface for building and training deep learning models, making it easier to prototype and experiment with different architectures.

3. *NumPy:* NumPy is a fundamental package for scientific computing with Python. It provides support for multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays, making it essential for data manipulation and numerical computations.

MODELLING

4. *Pandas:* Pandas is a fast, powerful, and flexible open-source data analysis and manipulation library built on top of NumPy. It provides data structures like DataFrame for handling structured data and tools for data cleaning, reshaping, and analysis.

5. *Matplotlib:* Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It's commonly used for plotting graphs, histograms, scatter plots, etc., to visualize data and model performance.

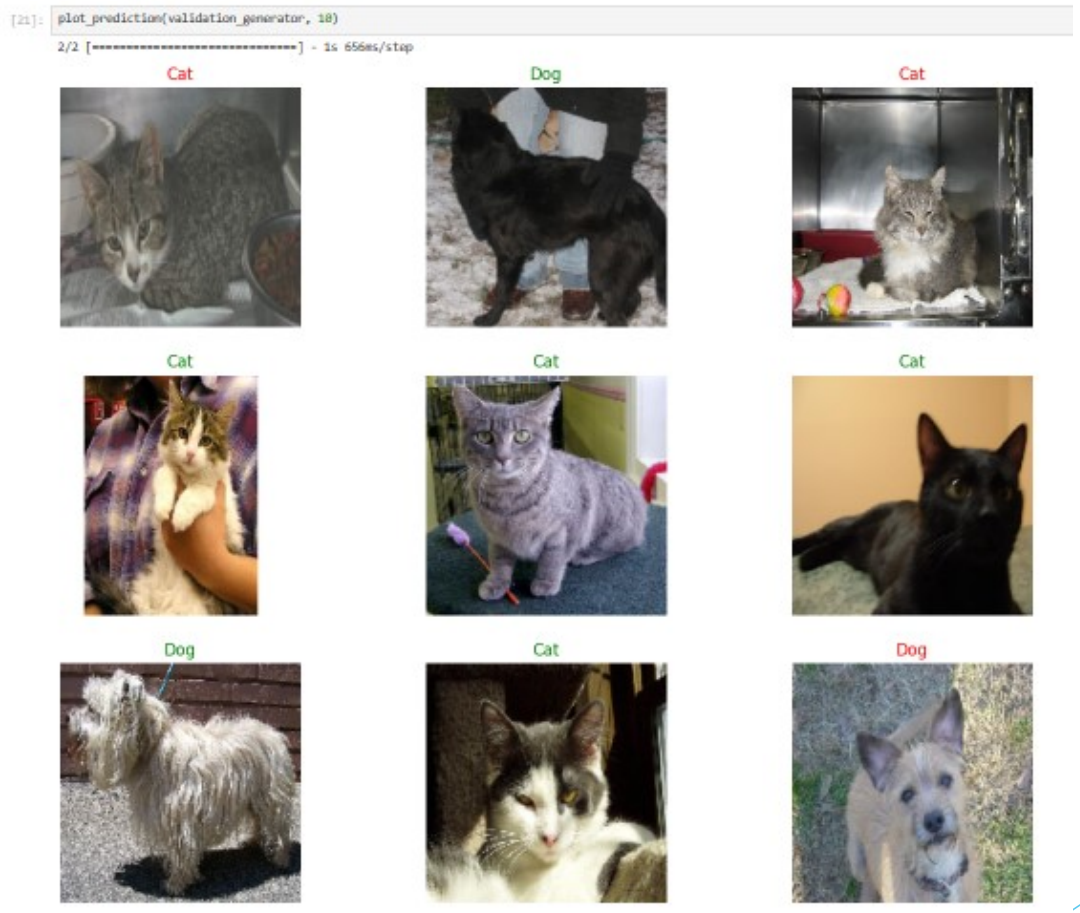
6. *Plotly Express:* Plotly Express is a high-level interface for creating expressive and interactive visualizations using Plotly. It provides easy-to-use functions for generating various types of plots, including pie charts, line plots, scatter plots, etc.

7. *SciPy:* SciPy is a library used for scientific and technical computing in Python. It provides modules for optimization, integration, interpolation, linear algebra, and more, making it useful for various numerical computations and statistical analysis.

These frameworks and libraries collectively provide a robust ecosystem for building, training, and analyzing deep learning models for image classification tasks.

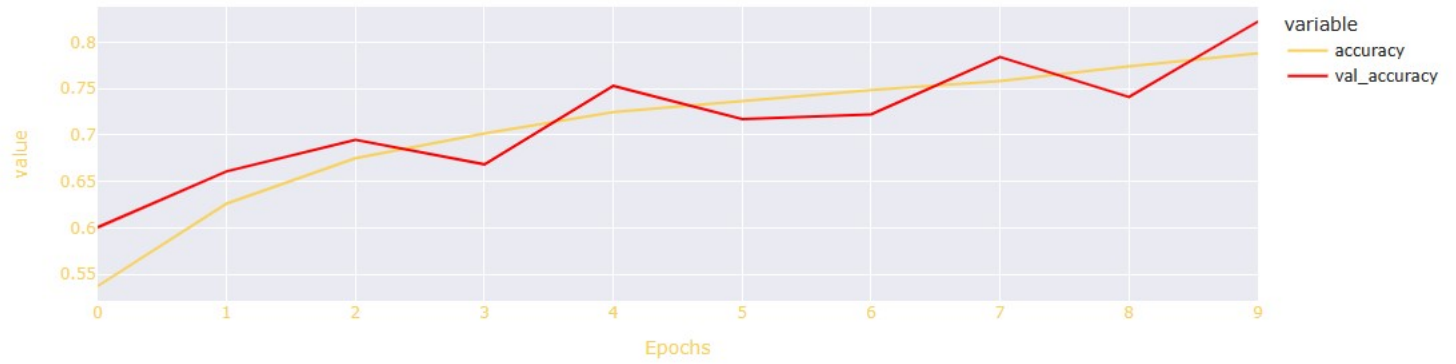
RESULTS

Results of testing on Random Samples. The labels in green have a probability factor of above 90%, while the red labels have a probability lesser than 90%



RESULTS

```
fig = px.line(results,y=[results['accuracy'],results['val_accuracy']],template="seaborn",color_discrete_sequence=['#fad25a','red'])
fig.update_layout(
    title_font_color="#fad25a",
    xaxis=dict(color="#fad25a",title='Epochs'),
    yaxis=dict(color="#fad25a")
 )
fig.show()
```



A graph on epochs vs value to evaluate the accuracy of the model

Thank you!