Deep Learning Project: Cat vs. Dog Image Classification using Convolutional Neural Networks (CNN)

\*\*Introduction\*\*

In this project, we explored the fascinating world of deep learning and built a Convolutional Neural Network (CNN) to classify images of cats and dogs. The objective was to train a model capable of distinguishing between these two animals with high accuracy. We used TensorFlow and Keras to create and train the CNN model.

\*\*Dataset\*\*

The dataset we utilized is the "cats\_and\_dogs\_filtered" dataset, which contains a large collection of images of cats and dogs. It was downloaded from the Google Cloud Storage using the 'wget' command and later extracted using Python's zipfile library.

\*\*Data Visualization\*\*

Before diving into building the model, we visualized a sample of images from the dataset. Using Matplotlib, we randomly selected eight images of cats and eight images of dogs to display. The images were resized to a common shape of 150x150 pixels, which is the input size for our CNN model.

\*\*The Architecture of CNN\*\*

Our CNN model consists of four pairs of Convolutional and MaxPooling layers, followed by a Flatten layer, and two fully connected Dense layers. The final layer utilizes the Sigmoid activation function to output a value between 0 and 1, representing the probability that the image is of a dog. A probability greater than 0.5 indicates a dog, while a probability less than or equal to 0.5 suggests a cat.

The structure of our CNN model is as follows:

1. Conv2D layer with 16 filters and ReLU activation, followed by MaxPooling.

2. Conv2D layer with 32 filters and ReLU activation, followed by MaxPooling.

3. Conv2D layer with 64 filters and ReLU activation, followed by MaxPooling.

4. Conv2D layer with 64 filters and ReLU activation, followed by MaxPooling.

5. Flatten layer to convert 2D feature maps to 1D vectors.

6. Dense layer with 512 units and ReLU activation.

7. Dense layer with 1 unit and Sigmoid activation.

\*\*Model Compilation and Training\*\*

Before training the model, we compiled it with the RMSprop optimizer and 'binary\_crossentropy' loss function, as it is a binary classification problem. We used accuracy as our evaluation metric. The dataset was split into training and validation sets, and we trained the model for 15 epochs.

\*\*Results\*\*

After training, we plotted the training and validation accuracy, as well as the training and validation loss. The results showed that the model achieved a high accuracy on both the training and validation sets, demonstrating the effectiveness of the CNN architecture for this image classification task.

\*\*Model Inference\*\*

To further evaluate the model, we performed inference on an external image. We uploaded an image of either a cat or a dog and fed it to the trained model. The model predicted the class of the image, and we used a threshold of 0.5 to determine whether the image was of a dog or a cat. The result was printed, indicating whether the image was identified as a cat or a dog.

\*\*Impact and Conclusion\*\*

The successful implementation of this CNN model for cat vs. dog image classification has various practical applications, such as building automated systems for pet identification in shelters or animal surveillance. Moreover, the project provided valuable hands-on experience with deep learning and reinforced the understanding of CNNs and their ability to recognize complex patterns in image data.

Overall, this project has been an exciting journey into the world of deep learning, equipping us with valuable skills to tackle various other image classification tasks in the future. As we continue our exploration of AI and machine learning, we look forward to leveraging these skills in further innovative and impactful projects.