**Image classification on cats and dogs**

**Submitted for**

**Statistical Machine Learning CSET211**

Submitted by:

**(E23CSEU0627) Rohan Kumar Singh**

**(E23CSEU0607) Aryan Raj Chaudhary**

**(E23CSEU0639) Kunal Kumar Singh**

Submitted to

**Dr. Susmita Das**

**July-Dec 2024**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr.No | Content | Page No |
| 1 | **Image Classification Project: Cats vs. Dogs** |  |
| 2 | **Project Overview** |  |
| 3 | **Importance of Image Classification** |  |
| 4 | **Kaggle Cats vs. Dogs Dataset** |  |
| 5 | **Data Preprocessing and Augmentation** |  |
| 6 | **ResNet-18 Model and Fine-Tuning** |  |
| 7 | **Training the Model** |  |
| 8 | **Evaluating Model Performance** |  |
| 9 | **Predicting on Test Set** |  |
| 10 | **Deployment and Real-World Application** |  |

**Abstract**

**This project focuses on building a deep learning-based :**

image classification model using PyTorch to differentiate between cats and dogs. The ResNet-18 architecture is fine-tuned to achieve accurate predictions, leveraging Kaggle's Cats vs. Dogs dataset for training and testing. The study aims to demonstrate the effectiveness of transfer learning in image classification tasks, showcasing high accuracy while maintaining computational efficiency.

**Objective:**

**The primary objective of this project is to:**

* Develop a robust image classification model capable of differentiating between cats and dogs.
* Evaluate the effectiveness of the ResNet-18 architecture in a binary classification setting.
* Demonstrate how transfer learning can improve training efficiency and performance on a relatively small dataset.

**Method**:

**Dataset:**

* The Kaggle Cats vs. Dogs dataset is utilized, consisting of labelled images of cats and dogs.
* Images are pre processed, resized, and augmented to improve model generalization.

**Model Architecture:**

* A pre-trained ResNet-18 model is used for transfer learning.
* The final fully connected layer is replaced to output probabilities for two classes (cat and dog).

**Training Process:**

* The dataset is split into training, validation, and test sets.
* Data augmentation techniques such as rotation, flipping, and normalization are applied.
* Binary Cross-Entropy Loss is used as the loss function.
* The Adam optimizer is employed for parameter optimization, with a learning rate scheduler to improve convergence.

**Evaluation Metrics:**

* Accuracy, precision, recall, and F1-score are used to evaluate model performance on the test set.
* Training and validation loss trends are analyzed to monitor overfitting.

**Implementation Tools:**

* PyTorch for model development and training.
* CUDA for GPU acceleration, where available.
* Matplotlib and Seaborn for visualizing results.

**Key Results**:

**What were the findings or results of the project?**

* The fine-tuned ResNet-18 model achieved a test accuracy of over 95%.
* Data augmentation significantly improved model generalization, reducing overfitting.
* Transfer learning drastically reduced training time compared to training a model from scratch.

**Significance**:

**The abstract should give readers a quick understanding of what the project is about and why it matters.**

* The project demonstrates the practical utility of transfer learning for image classification tasks, even with limited datasets.
* The methodology can be generalized to other binary classification problems involving visual data.
* The results highlight the computational efficiency and high accuracy of ResNet-18, making it a suitable choice for lightweight classification tasks.

**Example**:  
This project applies deep learning techniques, specifically Convolutional Neural Networks (CNNs), to the problem of image classification. The model is trained on a dataset of labeled images and achieves high accuracy in categorizing them. Results demonstrate that CNNs are highly effective for image classification tasks, offering significant improvements over traditional machine learning approaches.

**2. Introduction**

The **introduction** provides background information on the problem you are addressing and introduces the approach you are using.

In this section, you should:

* Explain what **image classification** is and its applications in real life (e.g., facial recognition, medical imaging, self-driving cars).
* Define the problem and the dataset you're working with.
* Briefly introduce the **method** or techniques used (e.g., CNNs).
* State the **objective** of your project (e.g., to classify images with high accuracy using a CNN).

**3. Related Work**

The **related work** section reviews previous research, methodologies, or projects that are similar to yours. It helps to position your work within the broader context of what others have done in the field.

1. In June 2023, the College of Technology published a research article titled ‘**Advances in Artificial Intelligence for Image Processing’** by Sampath Boopathi, Binay Kumar Pandey, and Digvijay Pandey.
2. The paper ‘**An Unsupervised Monocular Image Depth Prediction Algorithm Using Fourier Domain Analysis**’, by Lifang Chen and Xiaojiao Tang (SPR-2021-12-0186), is dedicated to image depth estimation, which is an important method to understand the geometric structure in a scene in various artificial intelligence products such as, for example, driverless cars, and home service robots.
3. The paper ‘**An improved segmentation technique for multilevel thresholding of crop image using cuckoo search algorithm based on recursive minimum cross entropy**’, by Arun Kumar, Anil Kumar, Amit Vishwakarma, and Heung-No Lee (SPR-2022-02-0031), investigates the efficient crop segmentation widely used for the analysis of crops in agriculture.
4. The paper ‘**Multi-Graph Convolutional Clustering Network**’ by Boyue Wang, Yifan Wang, Xiaxia He, Yongli Hu, and Baocai Yin (SPR-2022-01-0004) deals with clustering, which is a fundamental topic in the ML and data mining areas, which aims to group the similar samples into the same clusters.
5. The paper ‘**The Analysis of Completely Perturbed Model Based on RIP via Orthogonal Least Squares**’ by Haifeng Li and Hao Ying (SPR-2022-01-0009) deals with compressed sensing (CS), which has a goal of recovering the original signal. In CS, we usually deal with a linear model. The linear model is replaced in real-world applications with the so-called perturbed model.

**4. Methodology**

The **methodology** section explains in detail how you implemented your solution. It should describe:

* **The dataset**: What images are you classifying? What is their structure and how are they labeled?
* **Preprocessing**: What steps did you take to prepare the data (e.g., resizing, normalization, augmentation)?
* **Model architecture**: Describe the structure of the model. For example, if you used a CNN, explain its layers (e.g., convolutional layers, pooling layers, fully connected layers).
* **Training process**: Explain how the model was trained (e.g., optimizer, loss function, number of epochs, batch size).
* **Evaluation**: Describe how you measured the model's performance (e.g., accuracy, precision, recall, confusion matrix).

The goal is to give readers enough information to understand how you built and trained your model.

**5. Hardware/Software Required**

This section lists all the resources you needed to run your project, both **hardware** and **software**.

**Hardware**:

* **CPU/GPU**: The type of processor used. A **GPU** is usually preferred for deep learning tasks because it can speed up training.
* **RAM**: The amount of memory required to handle large datasets and models.

**Software**:

* **Programming Language**: Python is commonly used for image classification projects.
* **Libraries/Frameworks**: TensorFlow/Keras, PyTorch, OpenCV, NumPy, and Matplotlib are some commonly used libraries for image classification.
* **Tools**: IDEs like Jupyter Notebook, Google Colab, or Visual Studio Code.

**6. Experimental Results**

In this section, you showcase the performance of your model. It includes:

* **Training Results**: Show how the model performed during training (e.g., accuracy vs. epoch graph, training loss curve).
* **Test Results**: Report the performance on the test dataset (e.g., final accuracy, precision, recall).
* **Visualizations**: Include visual outputs such as:
  + **Confusion Matrix**: A matrix that shows how well the model performs in classifying images across different categories.
  + **Sample Outputs**: Show some examples of correctly and incorrectly classified images.

You should interpret the results and discuss the performance of your model, such as whether it met your expectations or if improvements are needed.

**7. Conclusions**

The **conclusion** summarizes the key findings from your experiment. It answers questions like:

* How well did the model perform in the image classification task?
* What are the strengths of your approach (e.g., high accuracy, efficiency)?
* What were the limitations or challenges faced during the project?
* What conclusions can be drawn from the results?

This section should provide a brief reflection on the success of the project and its implications.

**8. Future Scope**

This section discusses potential improvements or extensions of the work. You can suggest:

* **Improvements to the model**: Perhaps using a more advanced architecture (e.g., ResNet, Inception) or incorporating transfer learning.
* **Additional datasets**: Using larger or more diverse datasets to improve generalization.
* **Deployment**: Discuss how the model could be deployed in real-world applications.
* **New techniques**: Exploring the use of semi-supervised learning or reinforcement learning for image classification.

The future scope is an opportunity to expand on how this project could evolve or lead to further research.

**9. GitHub Link of Your Complete Project**

https://github.com/Aryan-277/Image-Classification