ZOODENTIFICATION

Prepared by Ahmet Ozmus 2023719081

Advisor: Berk Gokberk

Project Overview

Objective

- Develop a Convolutional Neural Network (CNN) model to classify 14 zoo animal species from images.

Dataset

- Source: OpenImages dataset (custom subset for 14 animal classes).

Input and Output

- Input: Images resized to 128×128×3.
- Output: Predicted class labels for animal species.

Dataset and Preprocessing

Classes

- 14 Classes: Elephant, Giraffe, Lion, Tiger, Bear, Red Panda, Kangaroo, Panda, Crocodile, Penguin, Jaguar, Rhinoceros, Hippopotamus, Monkey.

Challenges

- Initial goal: 1000 images per class.
- Problem: Limited images for some classes in OpenImages dataset.
- Solution: Used 100 images per class (1400 images total) to ensure class diversity.

Preprocessing Steps

1. Resized images to $128 \times 128 \times 3$.
2. Normalized pixel values to [0,1].
3. Labels encoded as one-hot vectors.

Data Split

- **Training:** 80%
- **Validation:** 20%

Dataset problem!

Animal	Number
Bear	165
Crocodile	86
Elephant	316
Giraffe	518
Hippo	100
Jaguar	92
Kangaroo	139
Lion	170
Monkey	1000
Panda	107
Penguin	759
Red Panda	72
Rhino	222
Tiger	255

At first tried to get 1000 images for each animal class, yet there were not enough for some of the animals.

Later used 100 for each in the training to provide diversity. (1400 images at total)

Model Architecture

Overview

- Transfer learning model leveraging MobileNetV2 as the base.
- Designed for feature extraction and classification. All weights are fine tuned during process.

Architecture Details

- **Input Shape:** 128x128x3.
- **Output Shape:** 14 classes (softmax activation).

Architecture Sequence

- Base Model: MobileNetV2 (pre-trained on ImageNet).
- Global Average Pooling Layer.
- Dense Layer with 128 units (ReLU activation).
- Dropout Layer (0.5).
- Output Layer: Dense with softmax activation for 14 classes.

Data Augmentation

Purpose

- Increase training data diversity.
- Reduce overfitting.

Techniques

- Horizontal Flips.
- Random Rotation (±20 degrees).

Training Process

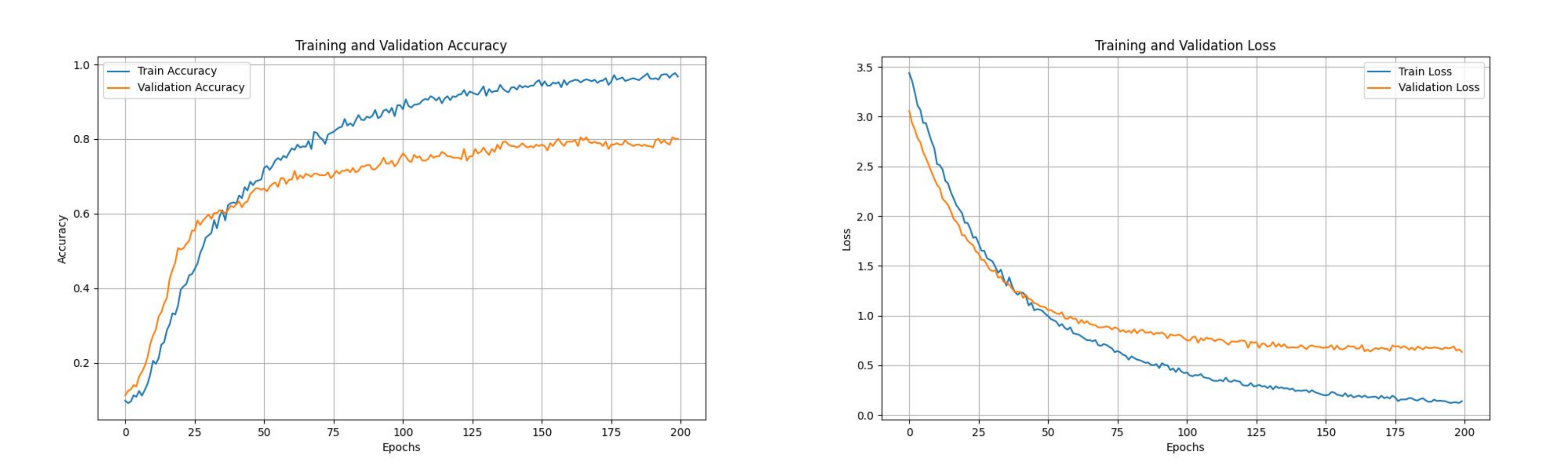
Parameters

- Batch Size: 128
- Epochs: 200
- Optimizer: Adam (learning rate: 1e-5)
- Loss Function: Categorical Cross-Entropy
- Metrics: Accuracy

Data Generators

- **Training:** Augmented data generator.
- **Validation:** No augmentation applied.

Training and Validation Accuracy and Loss



Training Results

Performance Metrics

- Validation Accuracy: **~80%**
- Validation Loss: Stable across 200 epochs.

Visualization

- Training and validation accuracy and loss plots saved as images:
  - `accuracy_plot.png`
  - `loss_plot.png`

Testing Results

Testing 98 images (7 images for each class) taken from google images used.

Epochs	Dropout	Batch Size	True Predictions	False Predictions	Accuracy (%)
30	0.5	128	64	34	65.31
50	0.5	128	85	13	86.73
100	0.5	128	90	8	91.84
200	0.5	128	93	5	94.90

Best Accuracy

- **200 Epochs: Accuracy: 94.90%** (93 out of 98 images correctly classified).

Lessons Learned

- With the lack of image amount, using transfer learning is beneficial

- Transfer learning prevents overfitting due to the fact that it uses weights of base model which trained with different dataset.

- Therefore, I could increase the number of epochs from 30 to 200 and batch size from 32 to 128 without encountering overfitting. In the previous structure where I used in progress report, overfitting was encountered after 30 epochs.

- During process I tuned dropout layers rate, learning rate, batch size and number of epochs

GitHub code repository: https://github.com/Ozmus/Zoodentification


THANK YOU