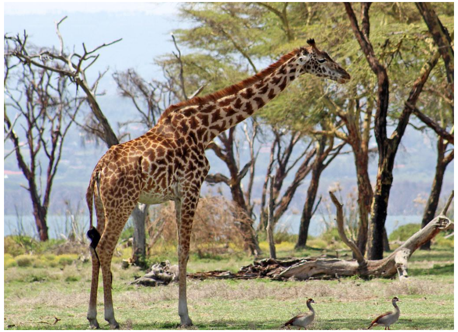
Animal Classification with Images using CNNs

By Yumeng Wang

Introduction/Motivation





Animal classes: ["zebra", "elephant", "giraffe", "lion", "hippo"]
Goal: A CNN Model that can classify animal species based on images

Label the animal!

Why is it interesting?

Real-World Application

- wildlife monitoring
- animal conservation efforts
- automated image labeling

Challenges

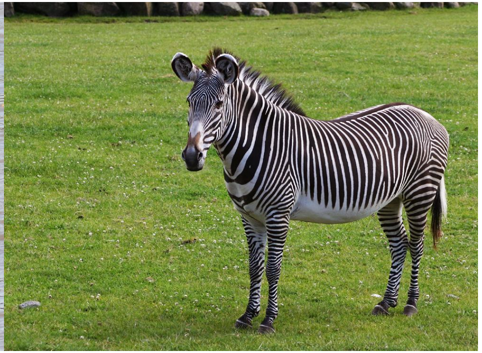
- animal appearances
- environmental factors

Initial Problem

- Classification on Animal crowd
- **Why this Approach?**
  - Real-world scenario
  - Model complexity
- **Challenges**
  - Noise and Complexity
  - Model Difficulty

Redefined Problem

- Classification on Single Animal
- **Why this Approach?**
  - Less noise, focus on animal features
  - Progressive
- **Challenges**
  - Data quality

Data

Data Preparation...

Animal Crowd Data

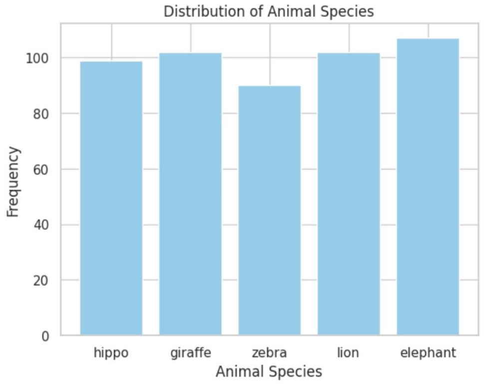
- **DALL-E-3**
  - 100 images
  - Min: 1024x1024
  - Randomization
  - Filename:
    - Count_animal_index.png
    - FIVE_elephant_55.png
  - Prompt: "A photo of exactly **THREE zebras**"
- **Challenges**
  - Expensive
  - Varying Art Styles
  - Partial Animal Bodies
  - Quality Issues
- **Data Split**:
  - Ratio: 80 - 10 -10
  - Size: 80 - 10 -10






Single Animal Data

- **DALL-E-2**
  - 500 images
  - Min: 256x256
  - Randomization
  - Filename:
    - Animal_Index.png
    - lion_16.png
  - Prompt: "A realistic photo of exactly ONE whole **giraffe** with white background"
- **Quality Control**
  - Single Animal Focus
  - Full animal body
  - Consistent Style
  - Plain Background
- **Data Split**:
  - Ratio: 80 - 10 -10
  - Size: 400 - 50 -50




```
Label distribution: Counter({'elephant': 107, 'giraffe': 102, 'lion': 102, 'hippo': 99, 'zebra': 90})
```

Method

Method

Initial Model (AnimalCountCNN)

- **Architecture**: Custom CNN with 3 Conv layers.
- **Input Features:** 16384 - 1048576
- **Image Size**: 1024x1024 - 224x224
- **Pooling**: 3 layers, 4x4 kernel
- **Result**: Less than 20% validation accuracy (worse than random predictions).
- **Issues**: Underfitting
  - Insufficient Data: Small dataset (100-500 images) can make it difficult for the model to learn meaningful patterns.
  - Inadequate Training: Increase the number of training epochs to allow the model to learn more from the data.

With Pre-trained VGG16

- **Architecture**: VGG16 with 16 Conv laygers
- **Input features**: 25088
- **Image Resize**: 224x224
- **Pooling**: 5 layers, 2x2 kernel
- **Result**: Less than 20% validation accuracy (worse than random predictions).
- **Issues**:
  - Similar issue with AnimalCountCNN Insufficient Data: Small dataset can make it difficult for the model to learn meaningful patterns.
  - Inadequate Training: Need to Increase the number of training epochs to allow the model to learn more from the data.

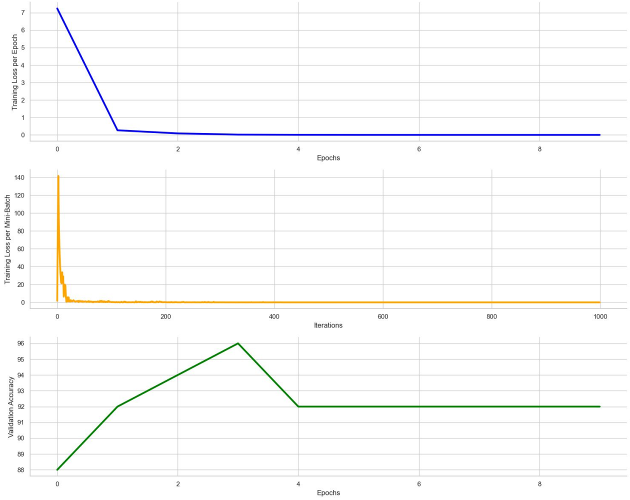
Method

Simplified Model (SimpleCNN)

- **Architecture**: Custom CNN with 1 convolutional layer.
- **Input Features:** 262144
- **Image Size**: 256x256
- **Pooling**: 1 layer, 2x2 kernel
- **Assumptions:**
  - Small Dataset Size: 100-500 images are too few for complex models but sufficient for simpler models like SimpleCNN.
  - Overfitting: More complex models (AnimalCountCNN, VGG16) overfit on small datasets.
  - Model Complexity: Simple models (like SimpleCNN) were more effective for this task.
  - Limited Data Augmentation: Data augmentation could still improve performance, but the smaller model worked well within the available data size.

Results

Results



Simplified Model (SimpleCNN)

- **Loss Function:** Cross Entropy
- **Optimizer:** Adam
- **Parameters**:
  - learning_rate = 0.01 ~ 0.002, epochs = 10, batchsize = 1 ~ 4
- **Best Model Saved:** The model with the highest validation accuracy 55% ~ 96%.
- **Validation Accuracy**: fluctuate between 86-96%, peak at epoch = 3
- **Results on Test Data:**
  - Accuracy: 50% ~ 86%
  - Precision (0.93): On average, when the model predicts a positive class, it is correct 98.6% of the time.
  - Recall (0.92): On average, the model correctly identifies 98.2% of the actual positive cases.
  - F1-score (0.92): This is a balanced measure of precision and recall, indicating overall good performance.

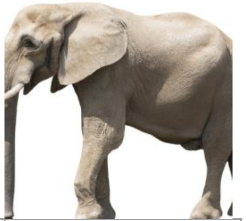
The 50 Test Results

Take a guess: which animal is the easiest/hardest to classify?

Results







Case	zebra	giraffe	hippo	lion	elephant
Precision	1	0.82	0.83	0.77	0.71
Recall	1	0.9	0.71	0.71	0.77
F1-score	1	0.86	0.77	0.74	0.74
Support	6	10	7	14	13

Relative ease of recognition by **Gemini**:

Zebra: Highly distinctive due to its striped pattern.
Giraffe: Tall neck and unique spotted pattern make it easy to identify.
Lion: Recognizable mane for males and overall body shape.
Elephant: Large size and distinctive trunk.
Hippo: Can be challenging due to its bulky shape and smooth skin.

Results on Real Data

Simplified Model (SimpleCNN)

- **Verified on 13 REAL animal pics**
- **Results on Test Data:**
  - <u>Accuracy</u>: 61.54%
  - Much better than random guess of 20%. Indicating that this model is also generalized and can be used with REAL animal pics.

```
--- Evaluation Results ---
Test Accuracy: 61.54%

Sorted Classification Report by Precision:

              precision    recall  f1-score  support
hippo               1.0  0.750000  0.857143      4.0
zebra               0.5  1.000000  0.666667      4.0
elephant            0.5  1.000000  0.666667      1.0
weighted avg        0.5  0.615385  0.520147     13.0
macro avg           0.4  0.550000  0.438095     13.0
giraffe             0.0  0.000000  0.000000      2.0
lion                0.0  0.000000  0.000000      2.0
```

Summary & Challenges

- Summary of Project
  - Goal: Classify animal species from images (zebra, elephant, giraffe, lion, hippo).
  - Approach: Shifted from animal crowds to single-animal images for better results.
  - Outcome: Achieved 90% validation accuracy and 80% test accuracy with SimpleCNN.
- Challenges
  - Animal Crowds:
    - Issues with background noise, varying art styles, and partial animal bodies.
    - Models struggled with insufficient data and complex images.
  - Complex Models (VGG16):
    - Underfitting with small datasets and large model architectures.

Discussion/Conclusion

Future Directions

- Real-World Images:
  - Standardize Backgrounds: Use image processing to isolate animals from backgrounds.
- Improving VGG16:
  - Attention Mechanisms: Add attention layers to focus on important image parts.
- Hybrid Models for Animal Crowds Classification:
  - Object detection can be used as a preprocessing step to identify and localize animals within images or videos.
  - Combine CNNs with other techniques (e.g., RNNs or transformers) to handle more complex animal interactions.
- Larger Datasets:
  - Collect more labeled images of animal crowds for better model generalization.