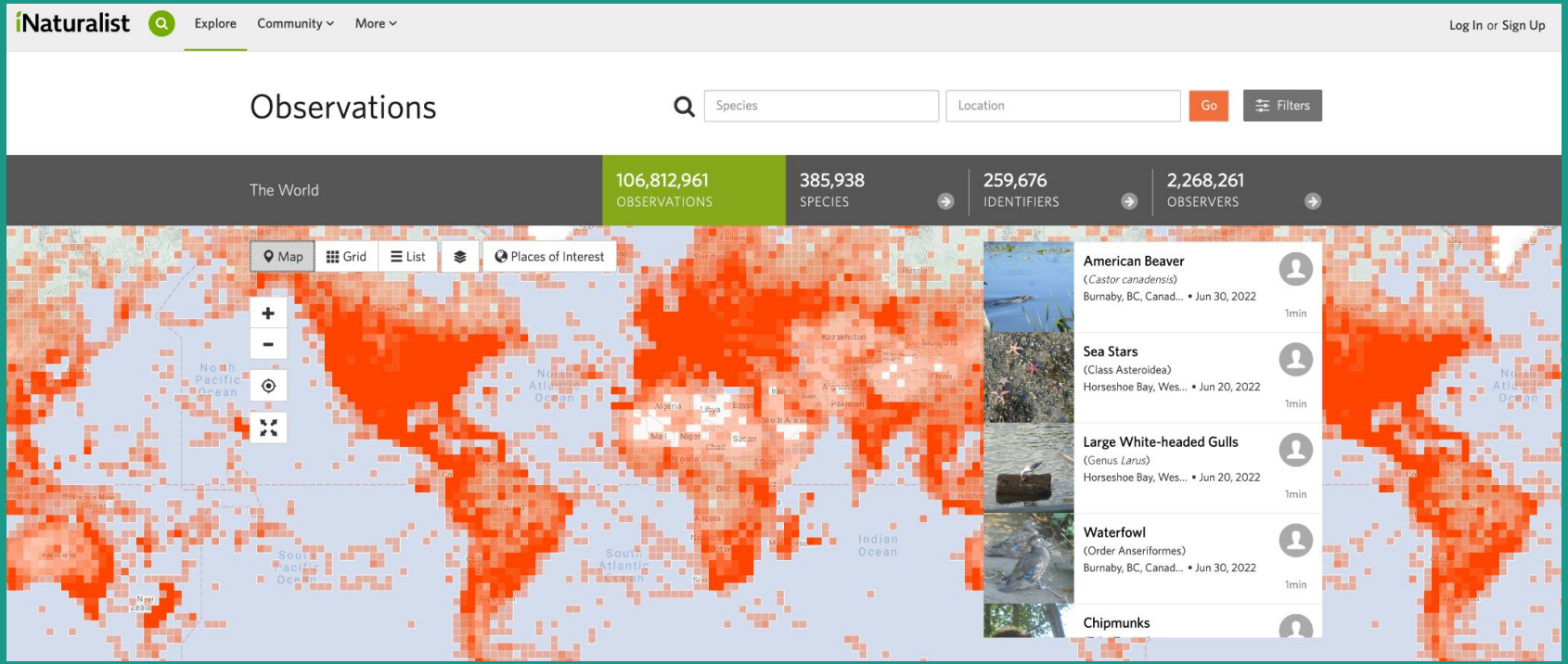
Categorizing Wildlife Habitat Types Using Image Recognition

Metis Deep Learning Project 2022

Goal: Automated classification of habitat types to allow for insight into wildlife ecology



Goal: Automated classification of habitat types to allow for insight into wildlife ecology



Approach

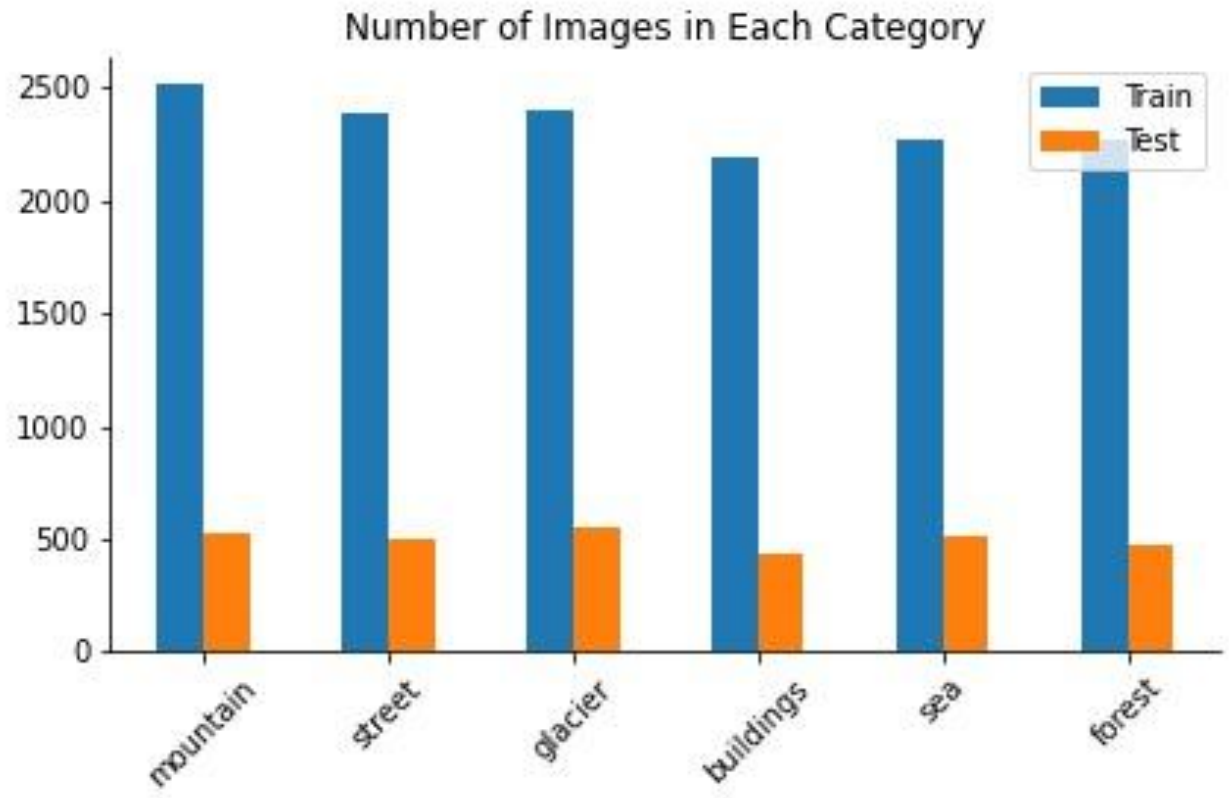
- Leverage **pre-labeled dataset**
- Develop a **deep learning classification model**
- Test model accuracy on validation data
- Test model functionality on unlabeled images

Tools

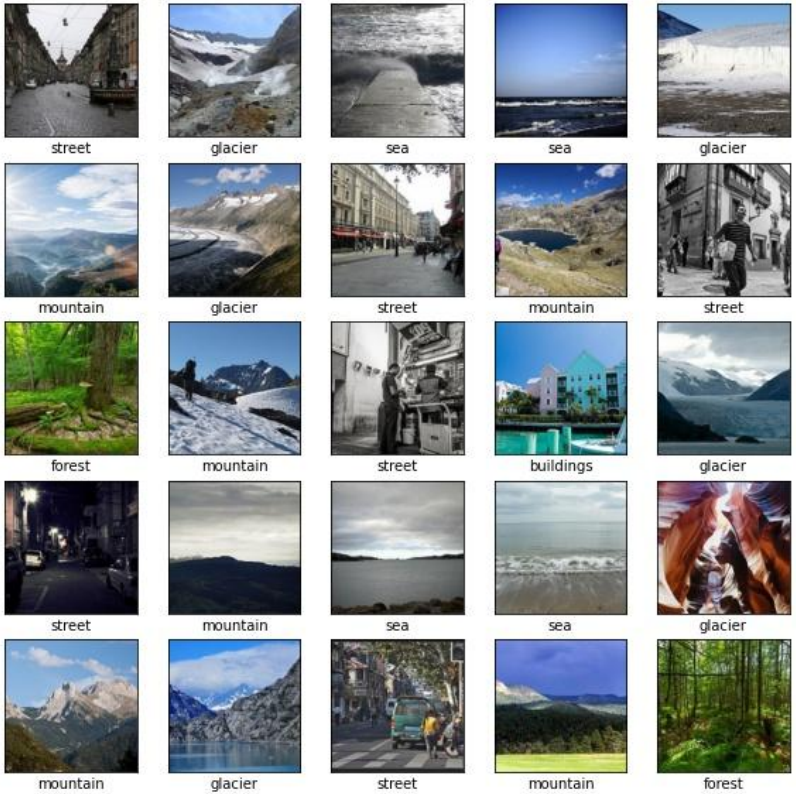
- **Tensorflow** and **Keras** for deep learning and model development
- **Numpy** and **Pandas** for data manipulation
- **Matplotlib** for visualization
- **Scikit-learn** for model score reporting

Dataset

- 14,034 training images
- 3,000 validation images
- https://www.kaggle.com/datasets/puneet6060/intel-image-classification



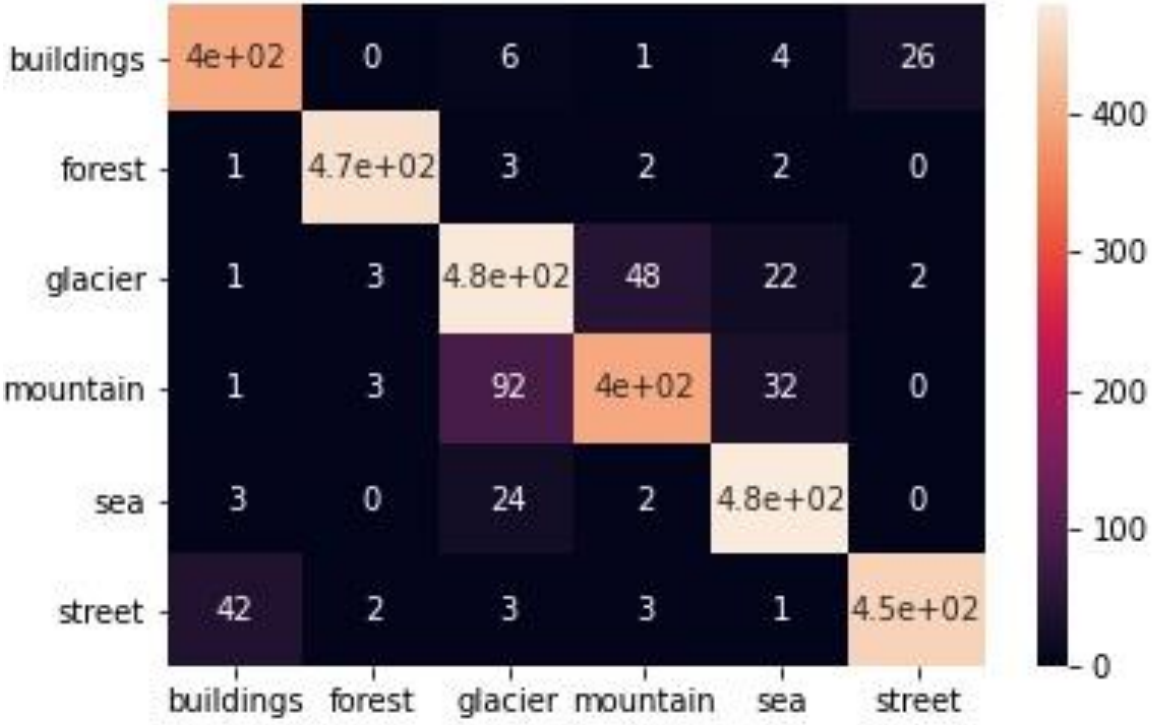
Sample images from the dataset



Model Comparison

Model Name	Accuracy	Model Description
Baseline CNN	0.735	CNN with 7 layers
MobileNetV2 Transfer	0.885	Freeze MobileNetV2 weights, flatten the output, add and train three dense layers
MobileNetV2 Tuned	0.897	Unfreeze base model weights and train with low learning rate
MobileNetV2 w/ Image Augmentation	0.890	Apply shear, zoom, and rotation to the images

Final Model Confusion Matrix



Final Model Confusion Matrix

Sample Predictions

Key Takeaways

- Labels matter
- Transfer learning is effective for this problem

Next Steps

- How to scrape relevant images and labels?
- Apply similar process to satellite imagery for broad habitat categorization?
- Revisit and explore image augmentation

Thank you!

https://github.com/ngoodby/Metis-Deep-Learning-Project/

Appendix

Baseline CNN Model

```
baseline_model = keras.Sequential([
    layers.Conv2D(10, (3, 3), activation = 'relu', padding = 'same', input_shape = (224,224,3)),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(10, (3, 3), activation = 'relu', padding = 'same'),
    layers.MaxPooling2D(2,2),
    layers.Flatten(),
    layers.Dense(20, activation='relu'),
    layers.Dense(num_classes, activation='softmax')
])
```

Baseline Confusion Matrix