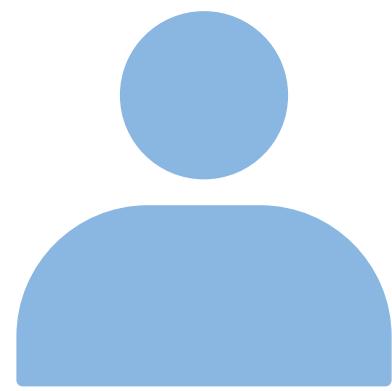


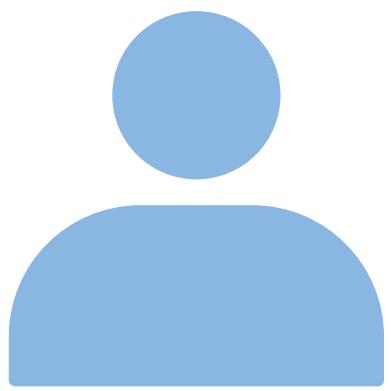
# Automated Image Tagging System using AWS



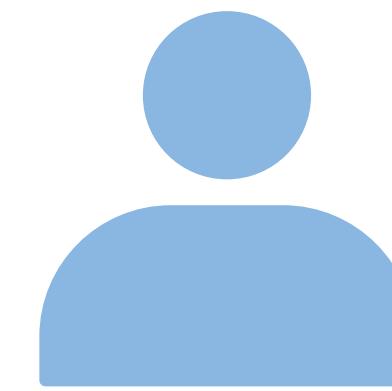
# Our Team



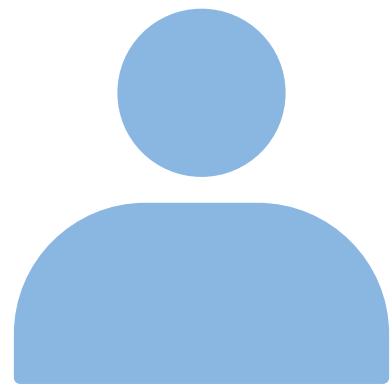
Ahmed Waheed



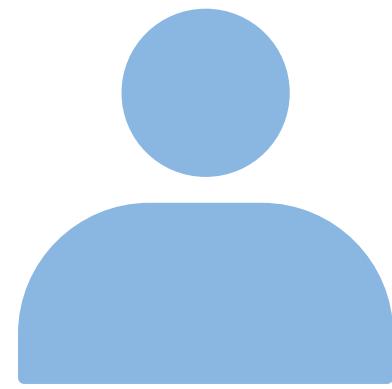
Bedour Fouad



Menna Osama



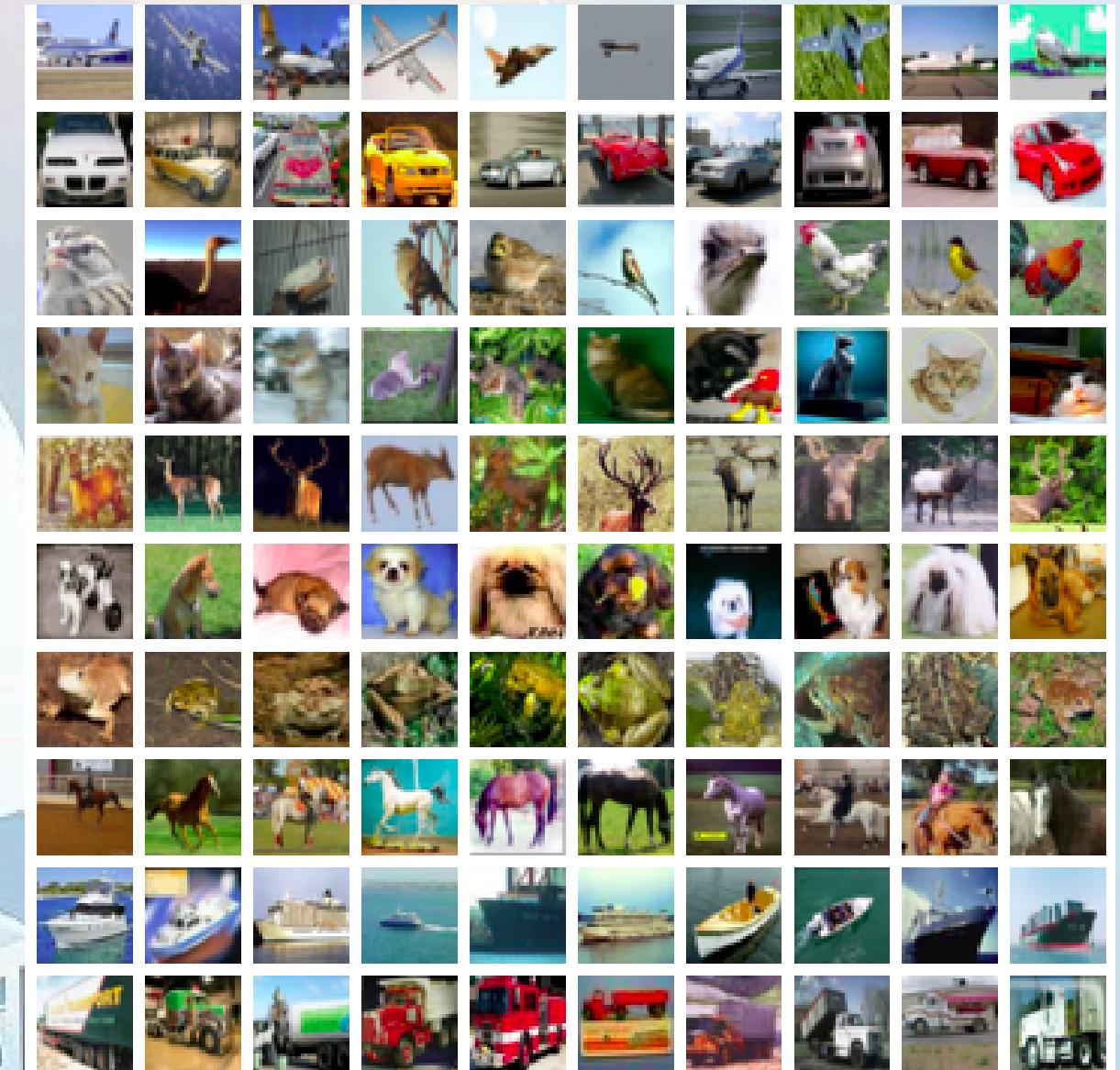
Norhan El-Sayed



Mina Farid

# Introduction

- **Objective:** To develop a machine learning system for automatic image tagging.
- **Problem:** Manual image tagging is inefficient and time-consuming, especially for large datasets with multiple objects per image.
- **Solution:** Use deep learning with ResNet50 for automated image tagging and deploy it on AWS SageMaker for scalable inference.



# Project Overview

- **Dataset:** Subset of the COCO dataset with 80 classes and 100,000 images.
- **Model:** Transfer learning using ResNet50.
- **Training:** Conducted on AWS SageMaker.
- **Deployment:** Deployed using SageMaker Inference Endpoints and API Gateway.

# Process

**01**

Data Processing

**02**

Model Development

**03**

Training and  
Evaluation

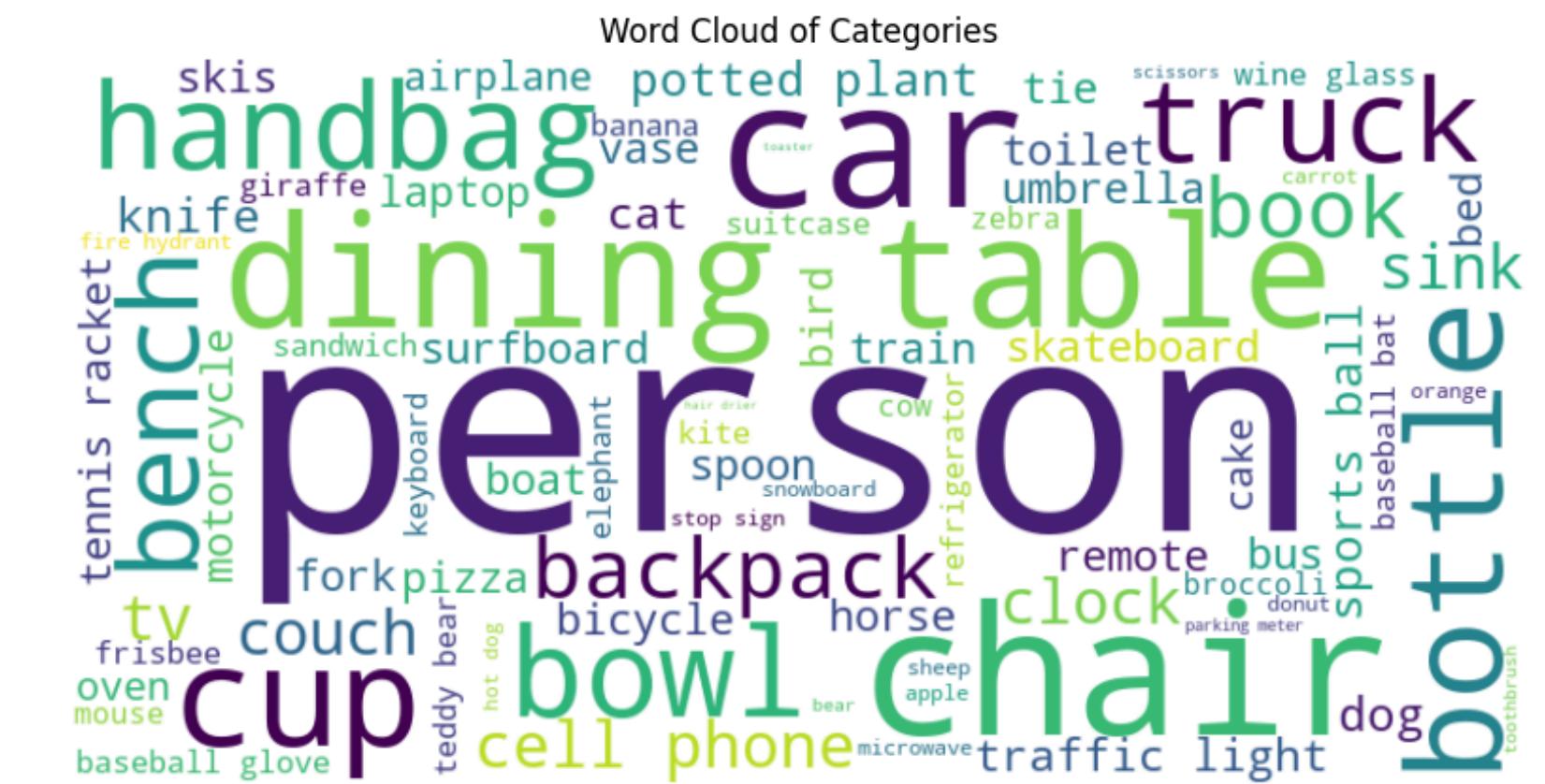
# Process

**04**  
Deployment

**05**  
Application Development

# Data

- Dataset with 80 classes and 100,000 training images.
  - The most frequently occurring label is person.
  - The least frequently occurring labels like hair drier and toaster
  - The main challenge in working with this dataset is the class imbalance



# Data Preparation

- Collected and prepared the dataset.
- Uploaded dataset to AWS S3.



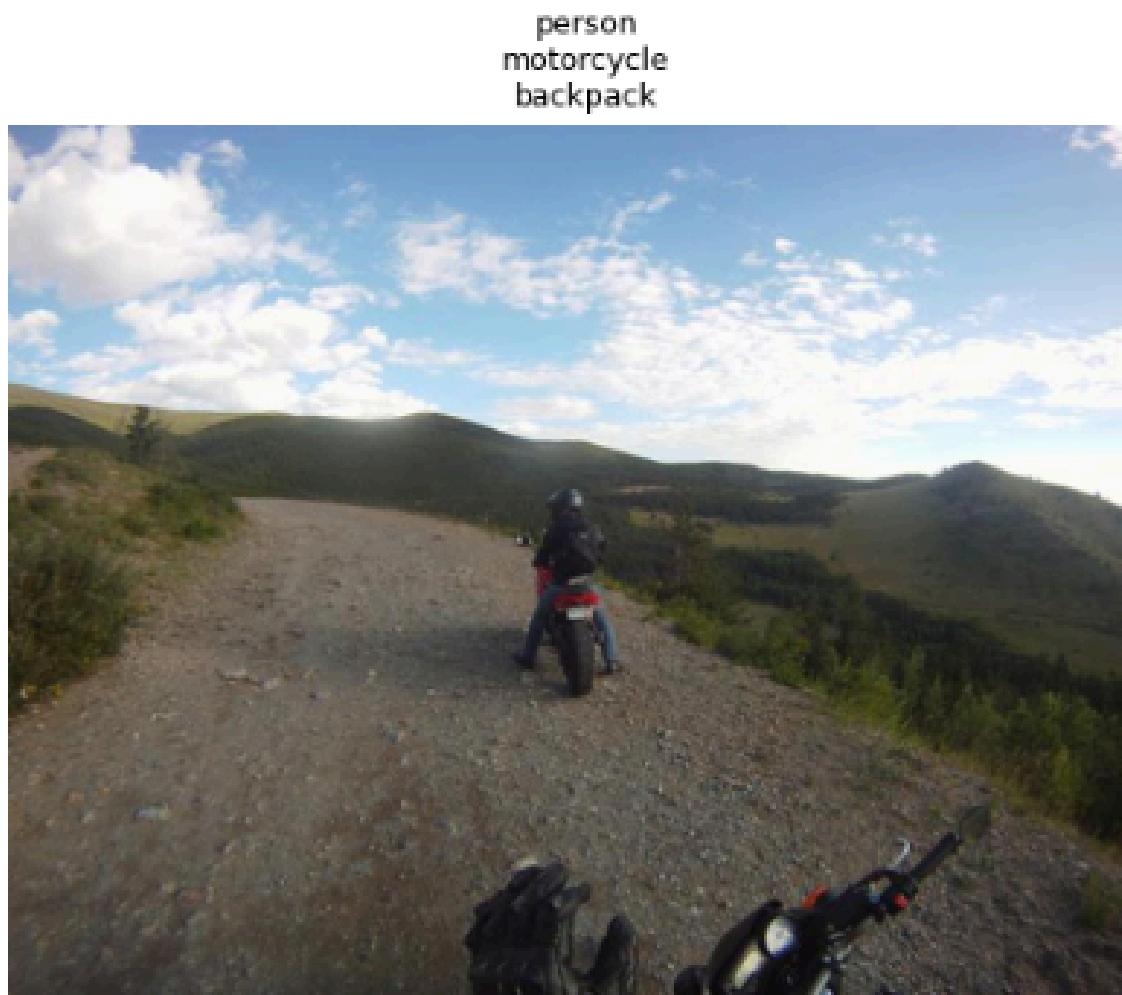
S3 Buckets

# Data Challenges

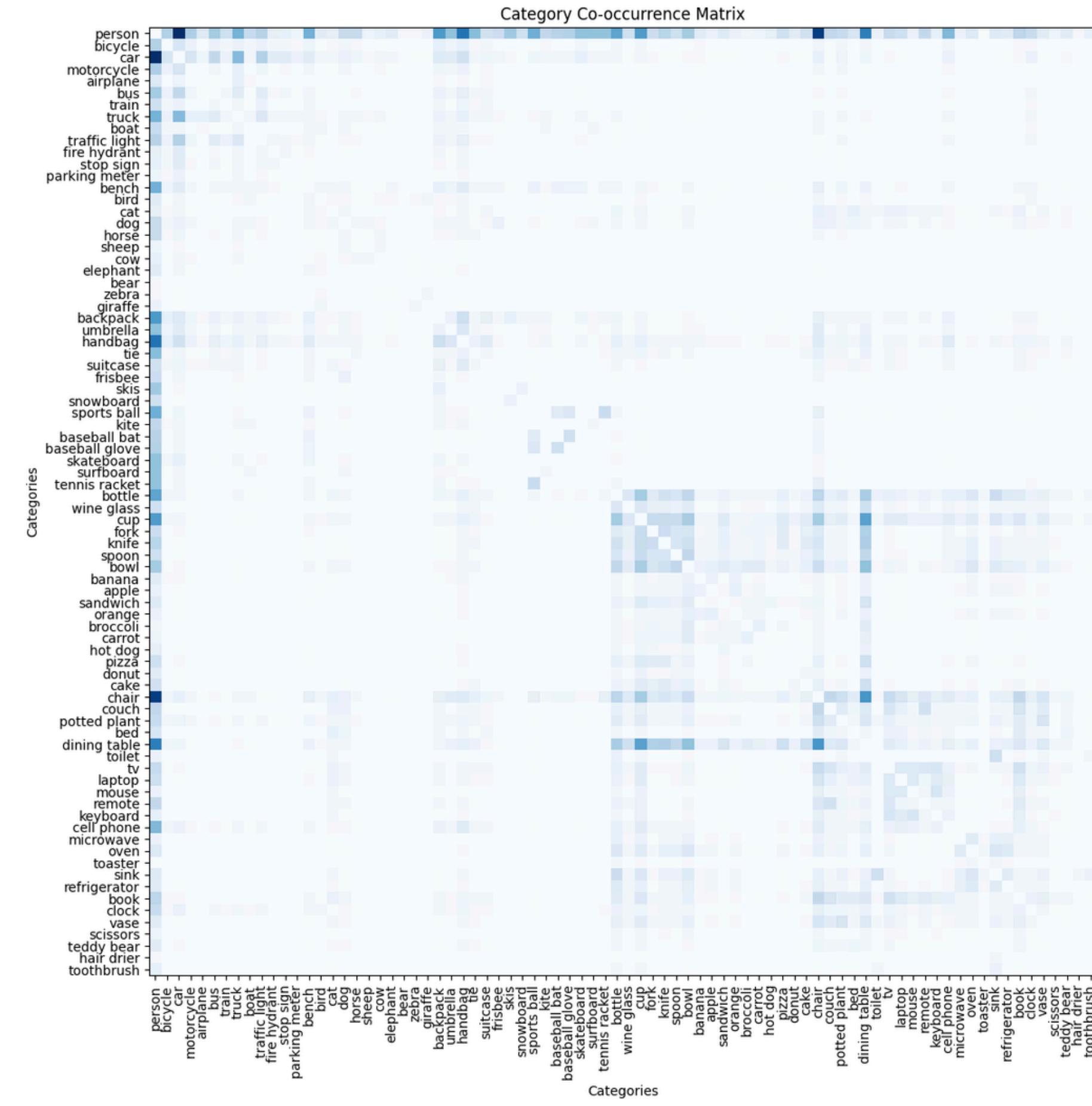
- **Class Imbalance:** Addressed using weighted sampling and focal loss.
- **Image resizing and normalization** for model input.
- **Data Leakage:** Resolved by carefully separating training and validation sets.

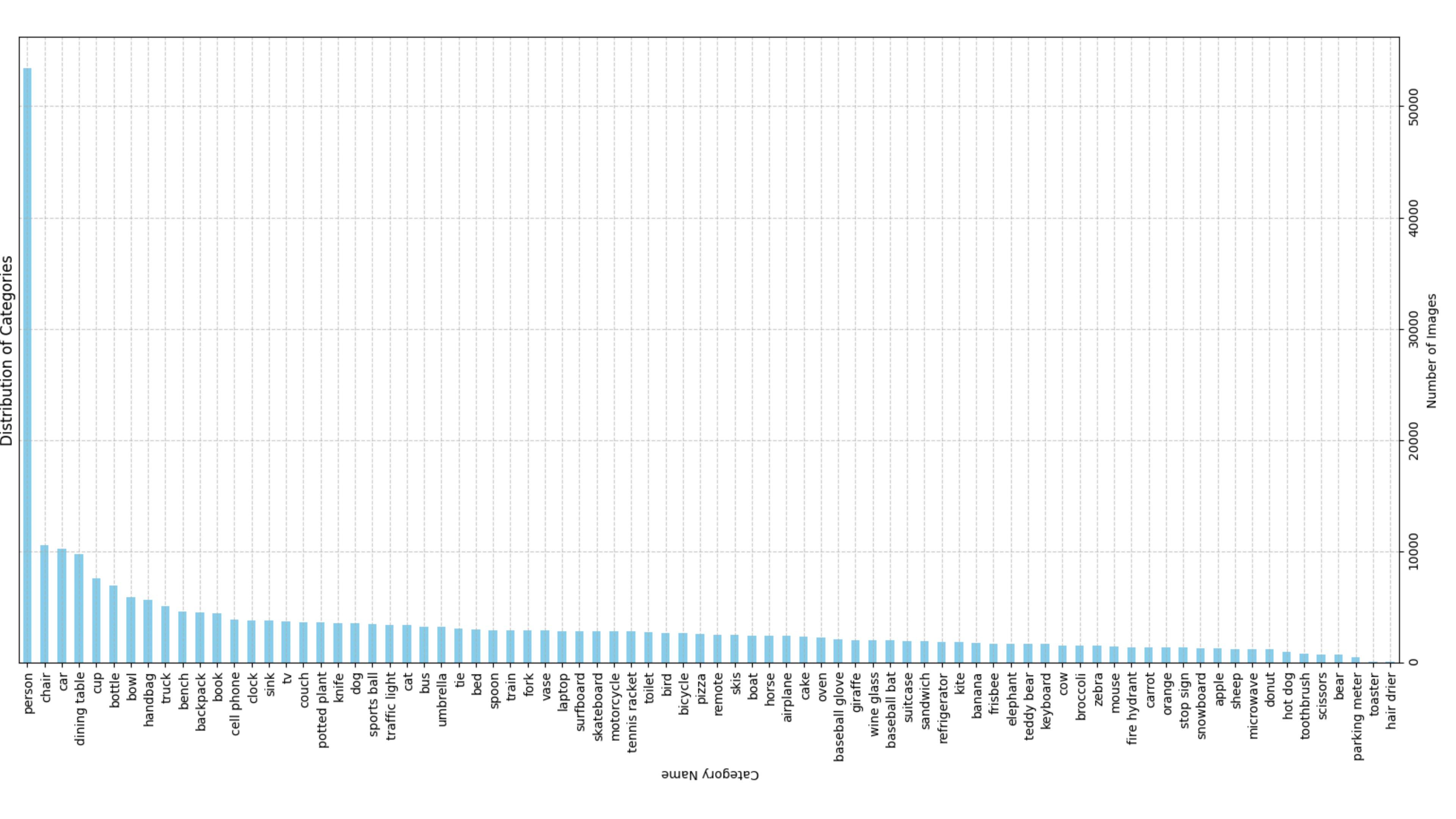
# Data Analysis

Ensured data consistency, integrity, and quality.



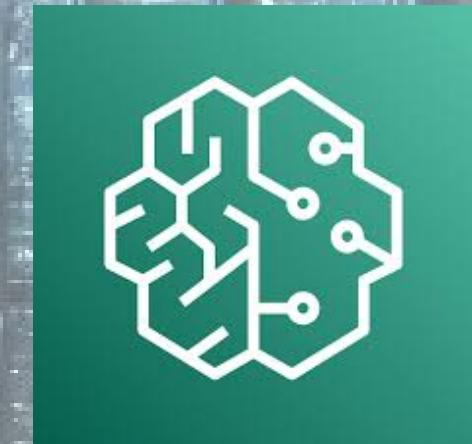
# Co-occurrence Matrix





# Preprocessing

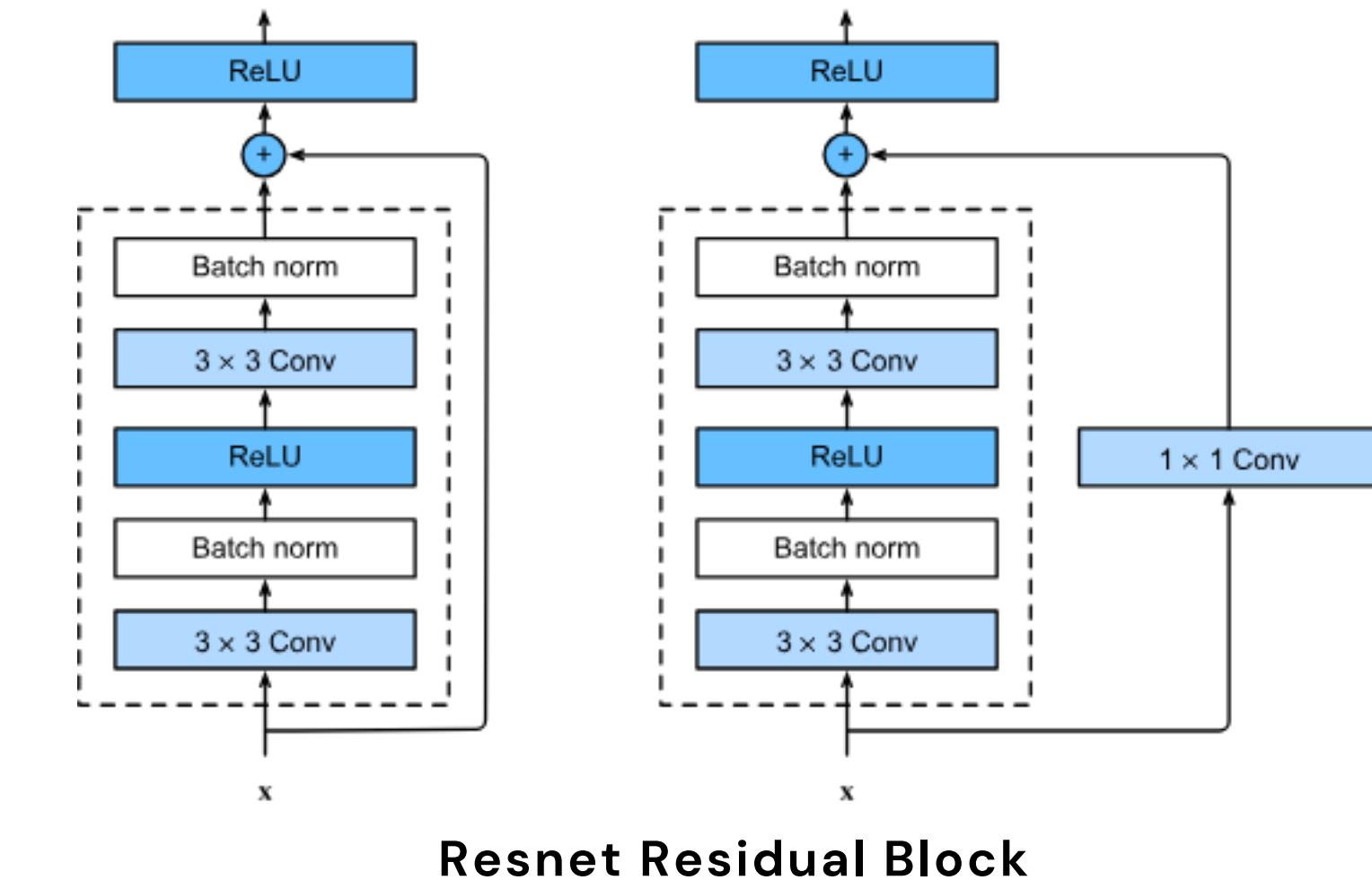
- Preprocessed data using AWS SageMaker notebooks.
- Applied data augmentation techniques (e.g., flipping, rotation)
- Applied weighted sampling to handle class imbalance



Sagemaker Notebook

# Model Architecture and Transfer Learning

- ResNet50: Pre-trained model used for feature extraction.
- Final layers adapted for 80-class classification.
- Trained and optimized using transfer learning and SageMaker.



# Training

- Model: Transfer learning using ResNet50.
- Platform: AWS SageMaker
- Data Split:
  - Training: 80%
  - Validation: 10%
  - Testing: 10%
- Optimization: Adam optimizer, learning rate = 0.001, batch size = 32.
- Training Epochs: 10 epochs.



# Model Evaluation and Optimization

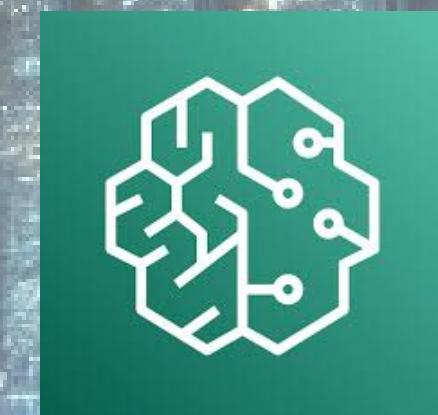
- Evaluated model performance and fine-tuned trainable layers.
- Applied Focal Loss and weighted sampling to handle class imbalance and improve recall.

# Results

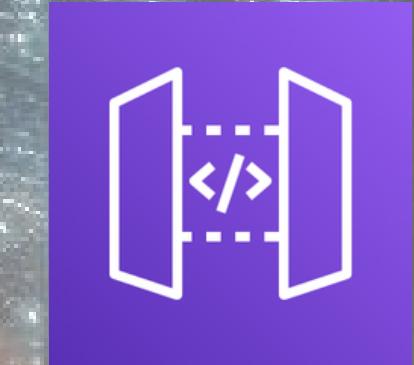
- Achieved a testing set per label accuracy of 97.85%
- Manually tested number of samples to ensure model quality.
- Saved the best model to S3 after optimization.

# Model Deployment

- Deployed the model using AWS SageMaker Serverless Inference Endpoints.
- Integrated the model with API Gateway for secure API access.
- Built a mobile and web app to interact with the deployed model.



SageMaker Serverless



API Gateway

# API Integration and Mobile/Web Apps

- Mobile and Web Apps: Developed for image tagging using API.
- Users can upload images and get tag predictions.
- Built using Flutter (mobile) and HTML/CSS (web) and web app hosted on an EC2 instance.

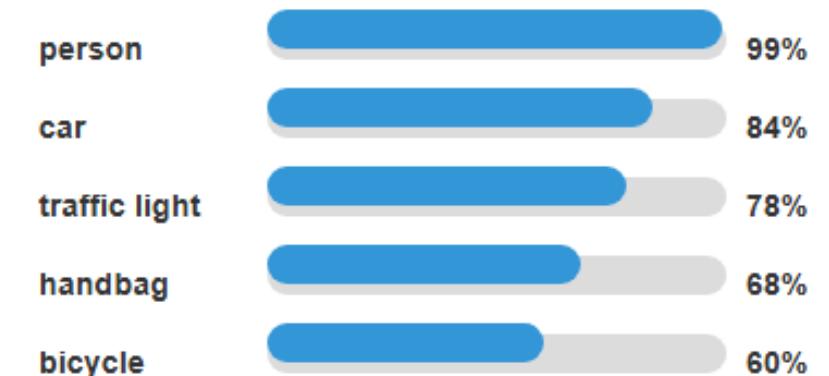
Upload Image for Tagging

Choose an Image



Upload and Tag

Tagging Result



# Monitoring and Debugging

- Used AWS CloudWatch for real-time logging and performance monitoring.
- Debugged inference requests, errors, and monitored API performance.



Cloudwatch Logs

# Problems Faced

- **Class Imbalance:** Addressed using weighted sampling and focal loss.
- **Model checkpointing:** Implemented to save the best model and prevent overfitting during longer training.
- **Endpoint Performance:** Initial issues with inference timeouts, fixed with optimized API and monitoring.

# Conclusion

The project effectively utilized cloud-based machine learning techniques to develop a scalable, real-time image tagging system with a 97.85% accuracy. AWS provided the necessary infrastructure for smooth model deployment and inference.

# Demo and Resources

**Live Demo:**

- [Web app Link](#)
- [Windows app link](#)
- [Mobile app link](#)

**GitHub Repository:**

- [Link](#)

**Dataset Link:**

- [Link](#)

**Documentation Link:**

- [Link](#)



ANY QUESTION?

•THANK YOU•