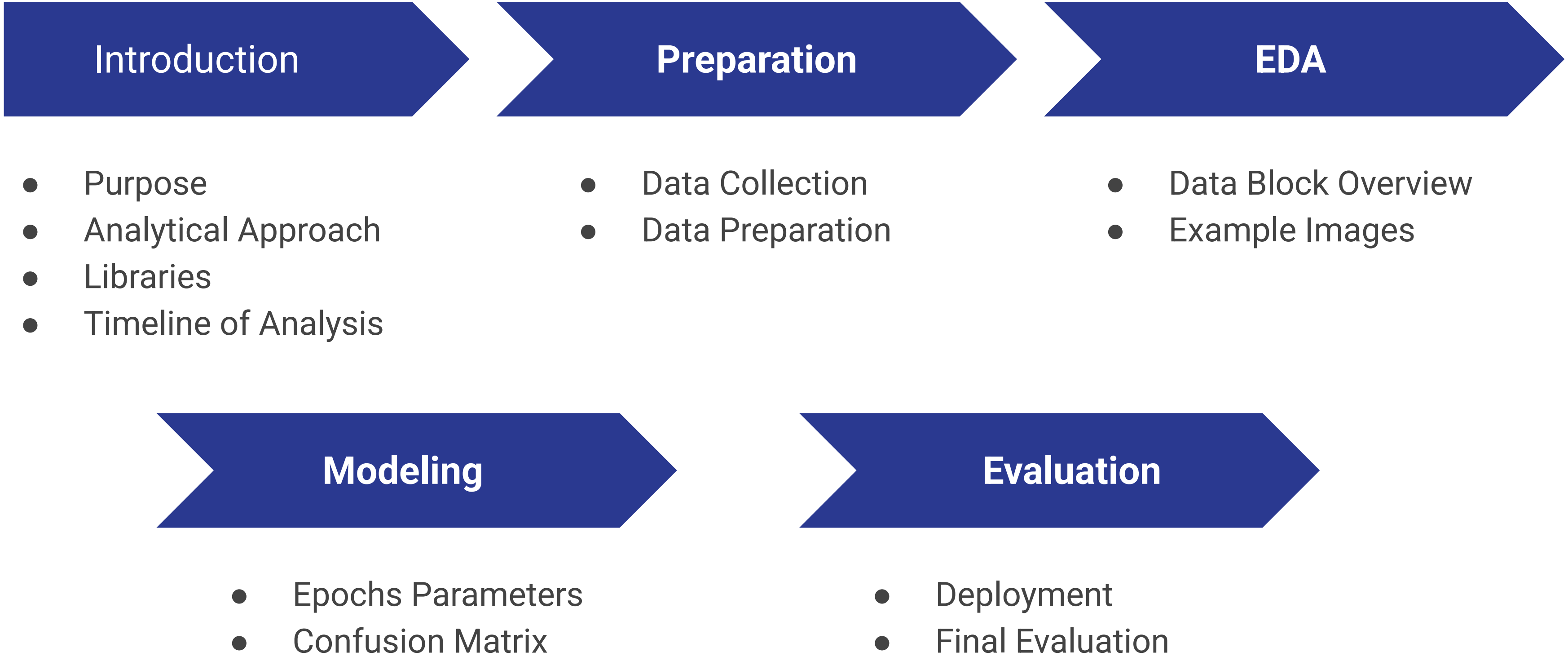
Bike Riding Style Classification

DATA 5100, Image Classification Project

By: Alyssa Zukas

Agenda



Introduction

- Purpose
- Analytical Approach
- Libraries
- Timeline of Analysis

Preparation

- Data Collection
- Data Preparation

EDA

- Data Block Overview
- Example Images

Modeling

- Epochs Parameters
- Confusion Matrix

Evaluation

- Deployment
- Final Evaluation

Introduction

- Purpose
- Domain Problem
- Analytic Approach

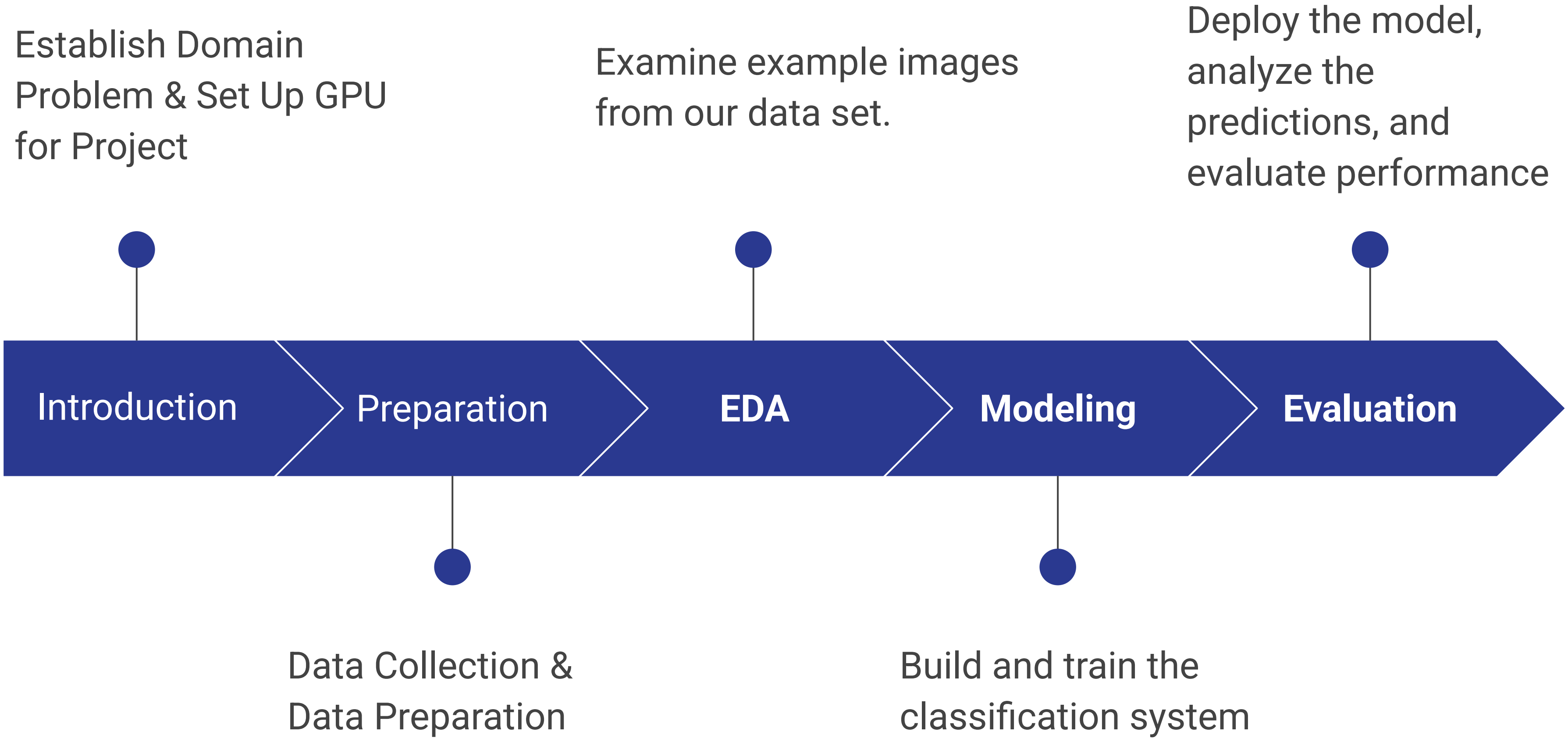
Introduction

Domain Problem

We want to develop a model to classify a bike based off of what it can see an image, into one of the following riding styles: **Cross Country**, **Downhill**, **Enduro**, **Trail** or **Dirt Jumper**.

Libraries

- **Kagglehub:** For downloading datasets directly from Kaggle into Google Colab.
- **Json:** For parsing and processing structured data from JSON files.
- **Pandas**: For cleaning, merging, and manipulating tabular data.
- **Matplotlib.pyplot**: For creating static visualizations and plotting model results.
- **Seaborn**: For statistical data visualization and probability plots.
- **Os**: For file and directory operations.
- **Requests**: For retrieving and downloading image data from URLs.
- **Tqdm**: For displaying progress bars during data download and preprocessing loops.
- **FastAI**: For training and testing the model - mainly the vision_learner and Classification Interpretation features



Establish Domain Problem & Set Up GPU for Project

Introduction

Preparation

Data Collection & Data Preparation

Examine example images from our data set.

EDA

Modeling

Build and train the classification system

Deploy the model, analyze the predictions, and evaluate performance

Evaluation

Preparation

- Data Collection
- Data Preparation

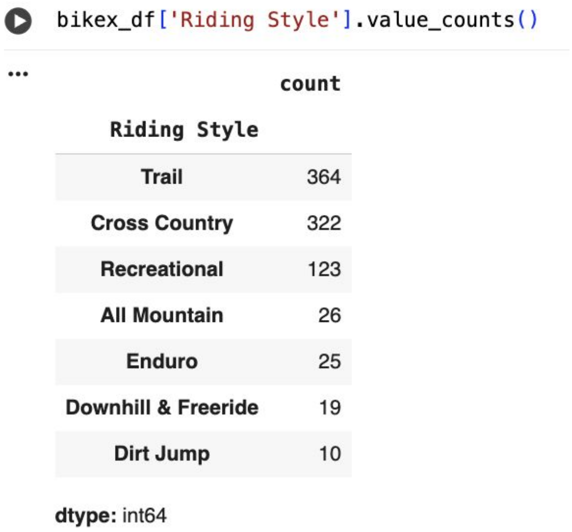
Data Collection

Bike Ads (images, prices, specifications)

- 10,052 bicycle advertisements from June 2020.
- 4 main data folders
  - `combined_price-only.csv`
  - `data_bike_exchange.json`
  - `data_ebay.json`
  - `images`

Data Preparation: Riding Style Classification

- Original Riding Styles

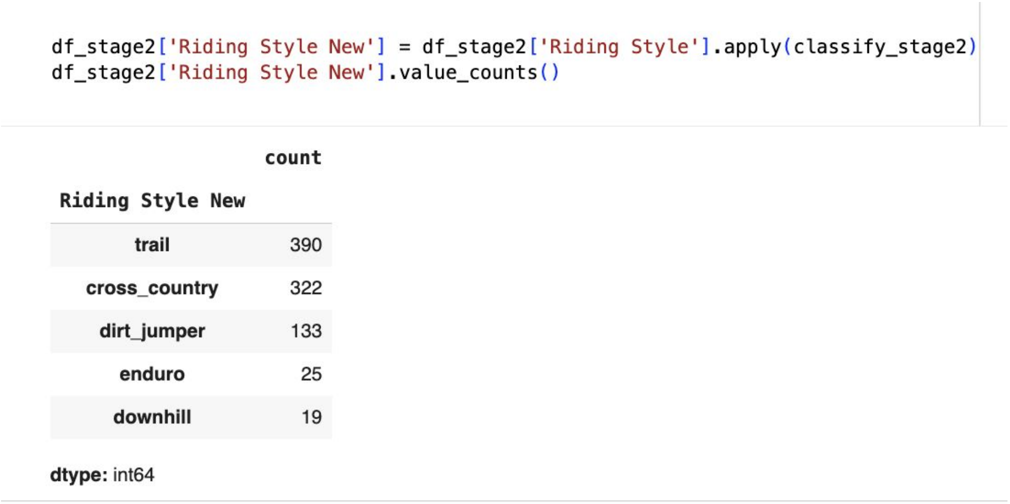


```
bikex_df['Riding Style'].value_counts()
```

Riding Style	count
Trail	364
Cross Country	322
Recreational	123
All Mountain	26
Enduro	25
Downhill & Freeride	19
Dirt Jump	10

dtype: int64

- New Riding Styles



```
df_stage2['Riding Style New'] = df_stage2['Riding Style'].apply(classify_stage2)
df_stage2['Riding Style New'].value_counts()
```

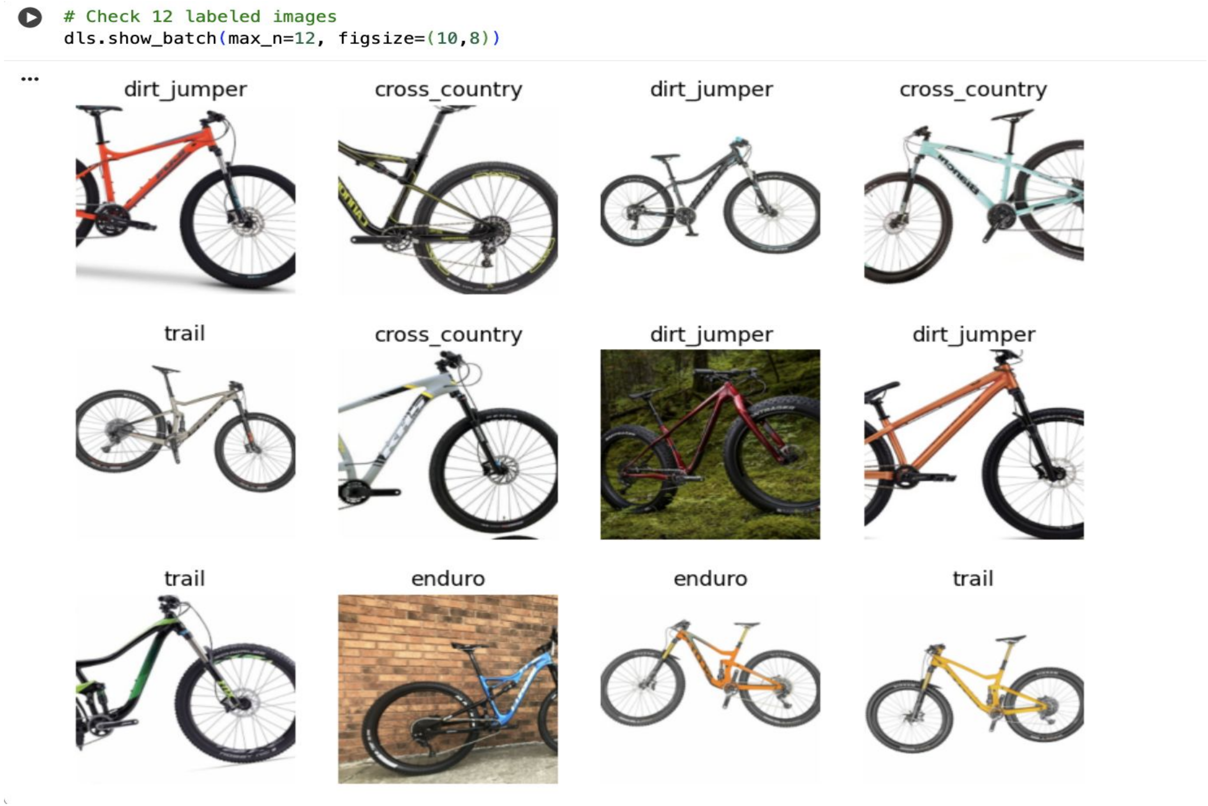
Riding Style New	count
trail	390
cross_country	322
dirt_jumper	133
enduro	25
downhill	19

dtype: int64

Exploratory Data Analysis

- Data Block Overview
- Example Images

Example Images

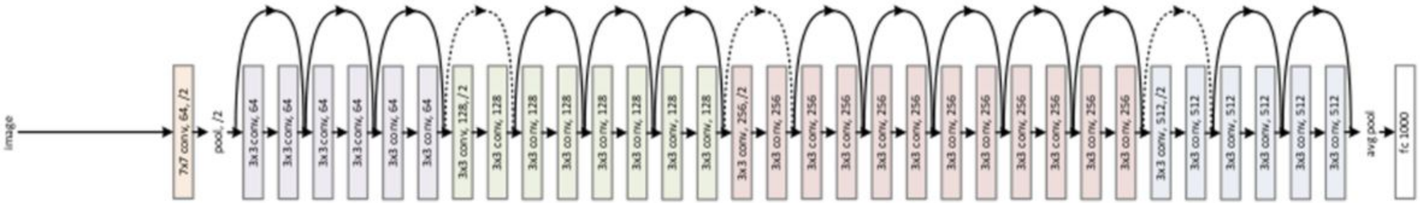


```
# Check 12 labeled images
dls.show_batch(max_n=12, figsize=(10,8))
```

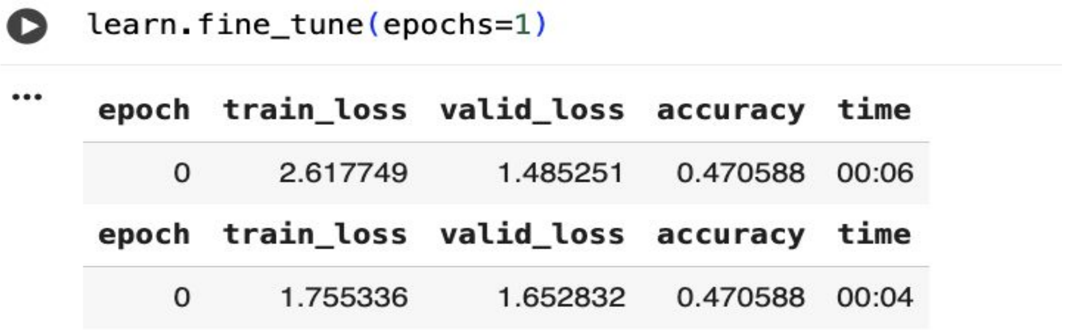
Modeling

- Epochs Parameters
- Confusion Matrix

ResNet-34 Modeling

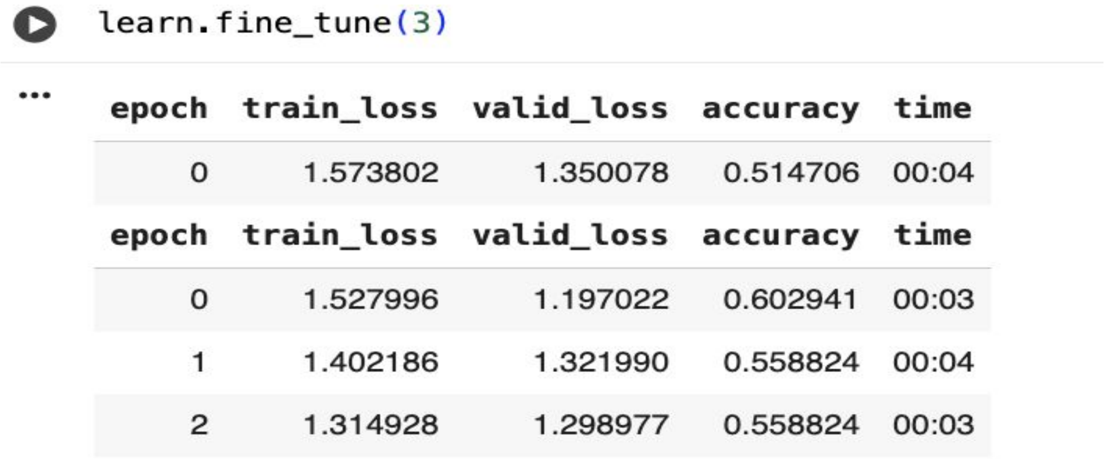


Training & Testing



```
learn.fine_tune(epochs=1)
```

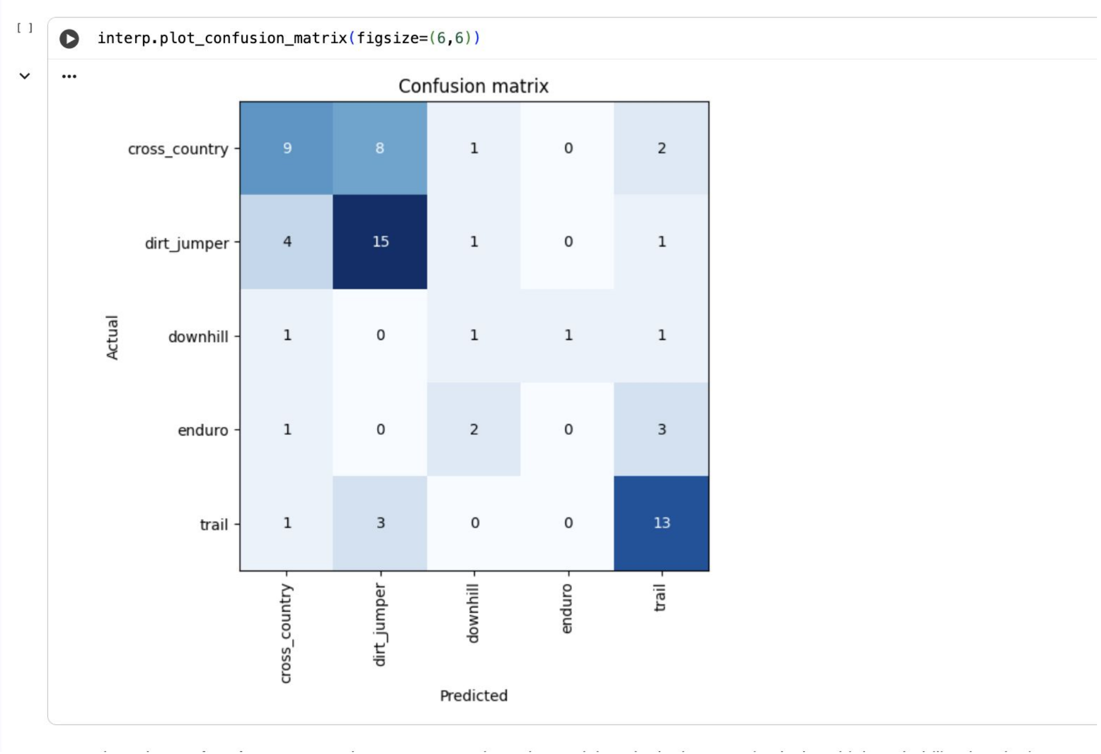
```
...
 epoch  train_loss  valid_loss  accuracy  time
     0    2.617749    1.485251  0.470588  00:06
 epoch  train_loss  valid_loss  accuracy  time
     0    1.755336    1.652832  0.470588  00:04
```



```
learn.fine_tune(3)
```

```
...
 epoch  train_loss  valid_loss  accuracy  time
     0    1.573802    1.350078  0.514706  00:04
 epoch  train_loss  valid_loss  accuracy  time
     0    1.527996    1.197022  0.602941  00:03
     1    1.402186    1.321990  0.558824  00:04
     2    1.314928    1.298977  0.558824  00:03
```

Confusion Matrix



```
interp.plot_confusion_matrix(figsize=(6,6))
```

Confusion matrix

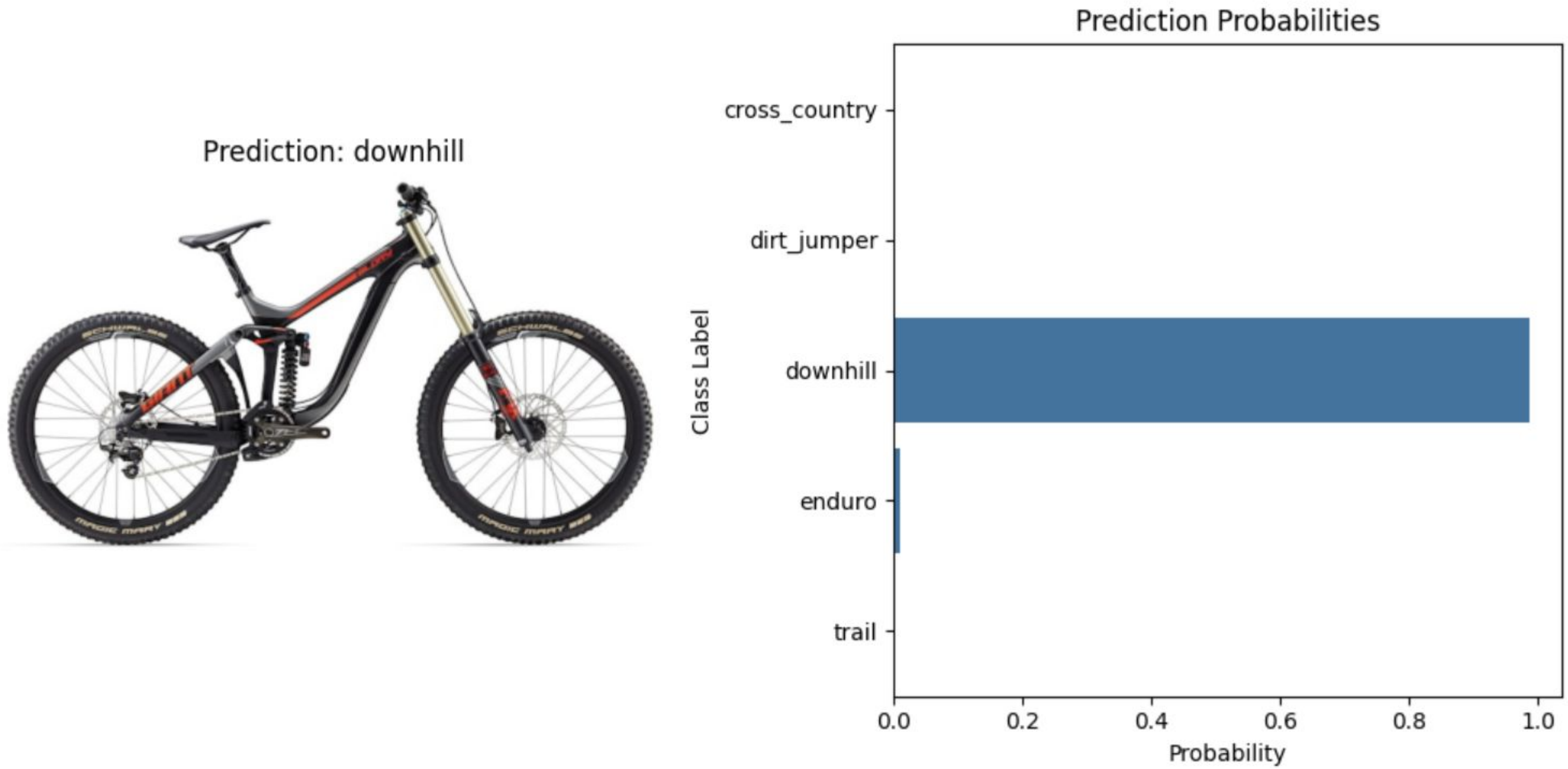
	cross_country	dirt_jumper	downhill	enduro	trail
cross_country	9	8	1	0	2
dirt_jumper	4	15	1	0	1
downhill	1	0	1	1	1
enduro	1	0	2	0	3
trail	1	3	0	0	13

Evaluation

- Deployment
- Final Evaluation

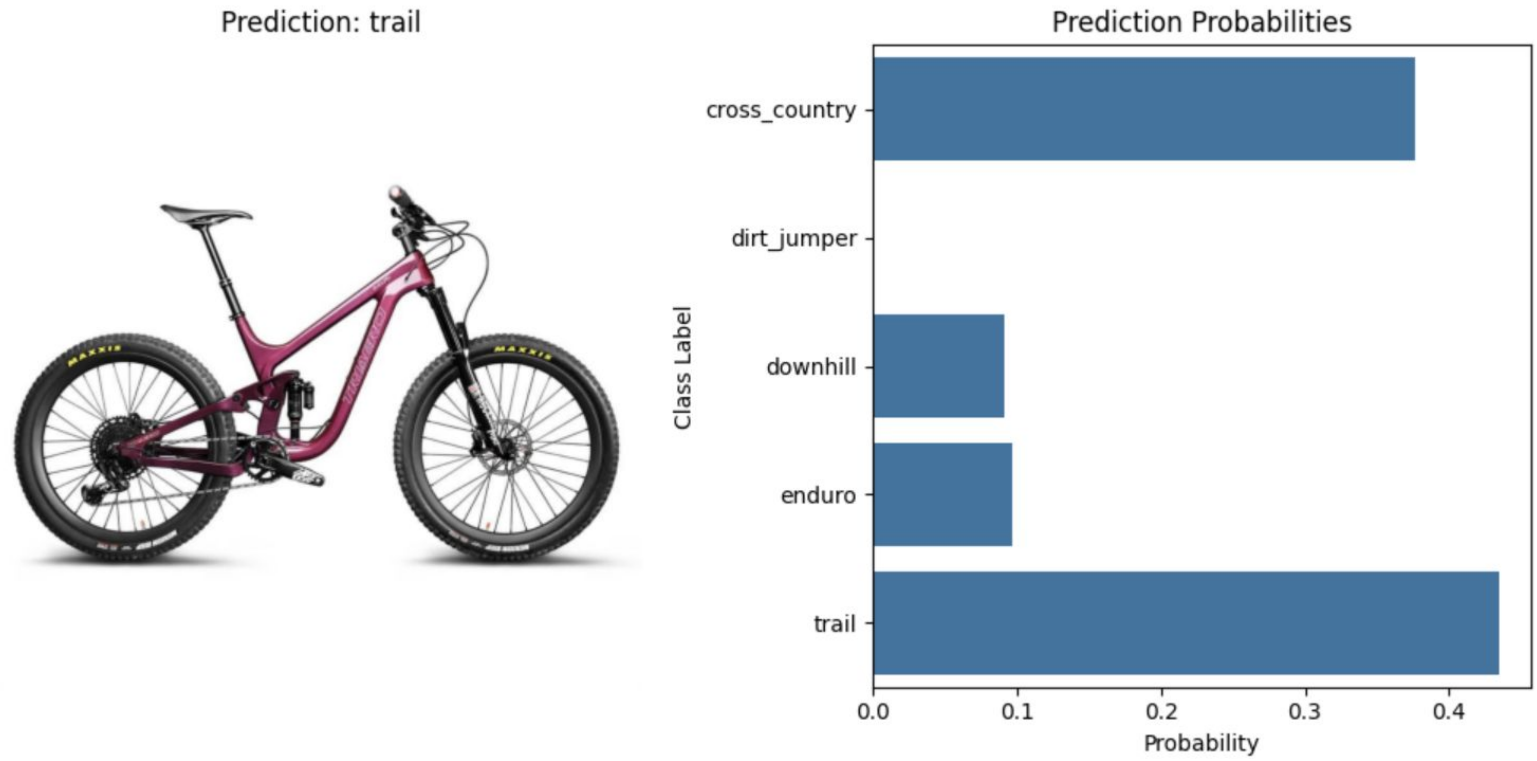
Deployment, First Test

...



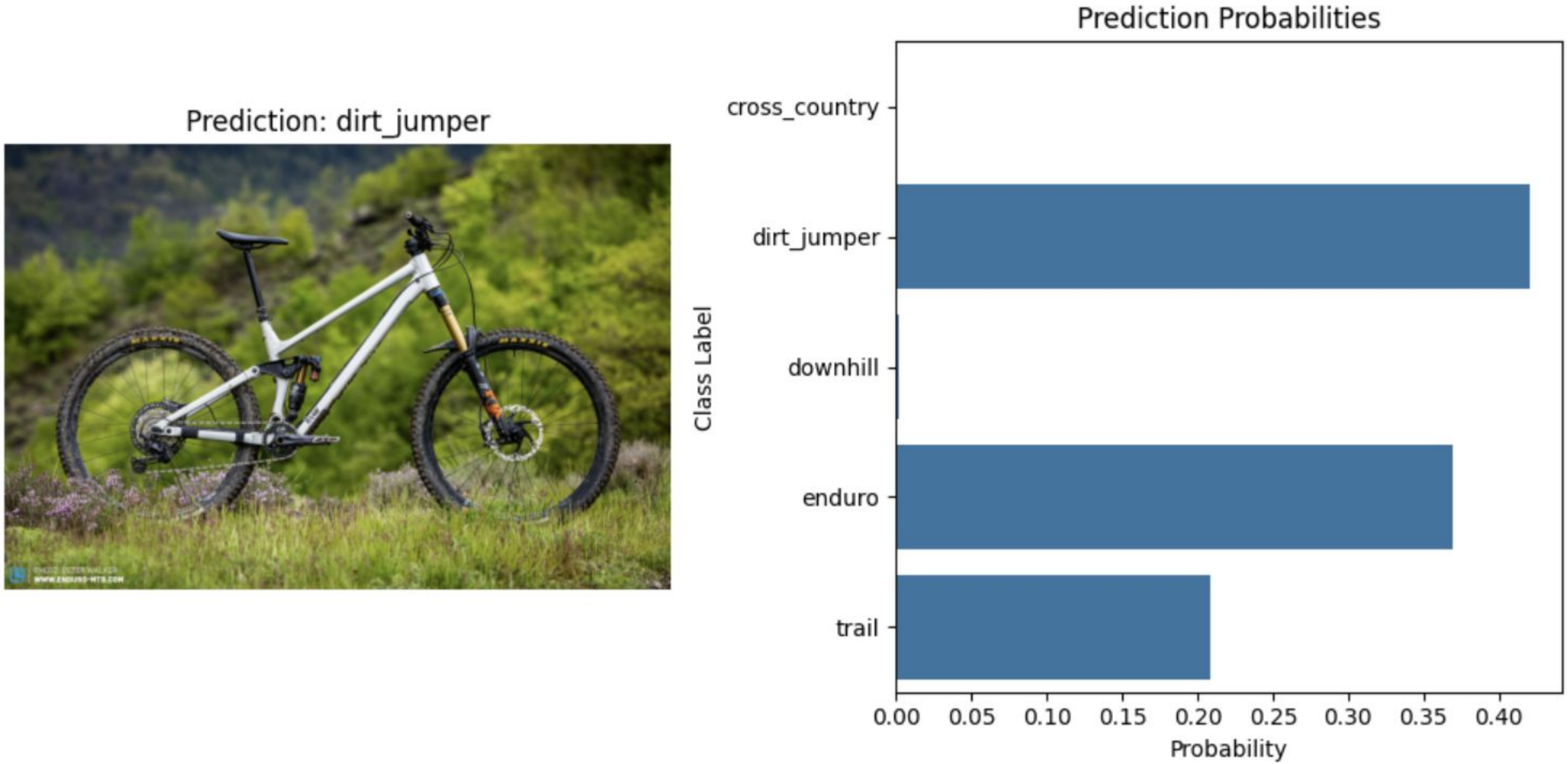
Deployment, Image 1

...



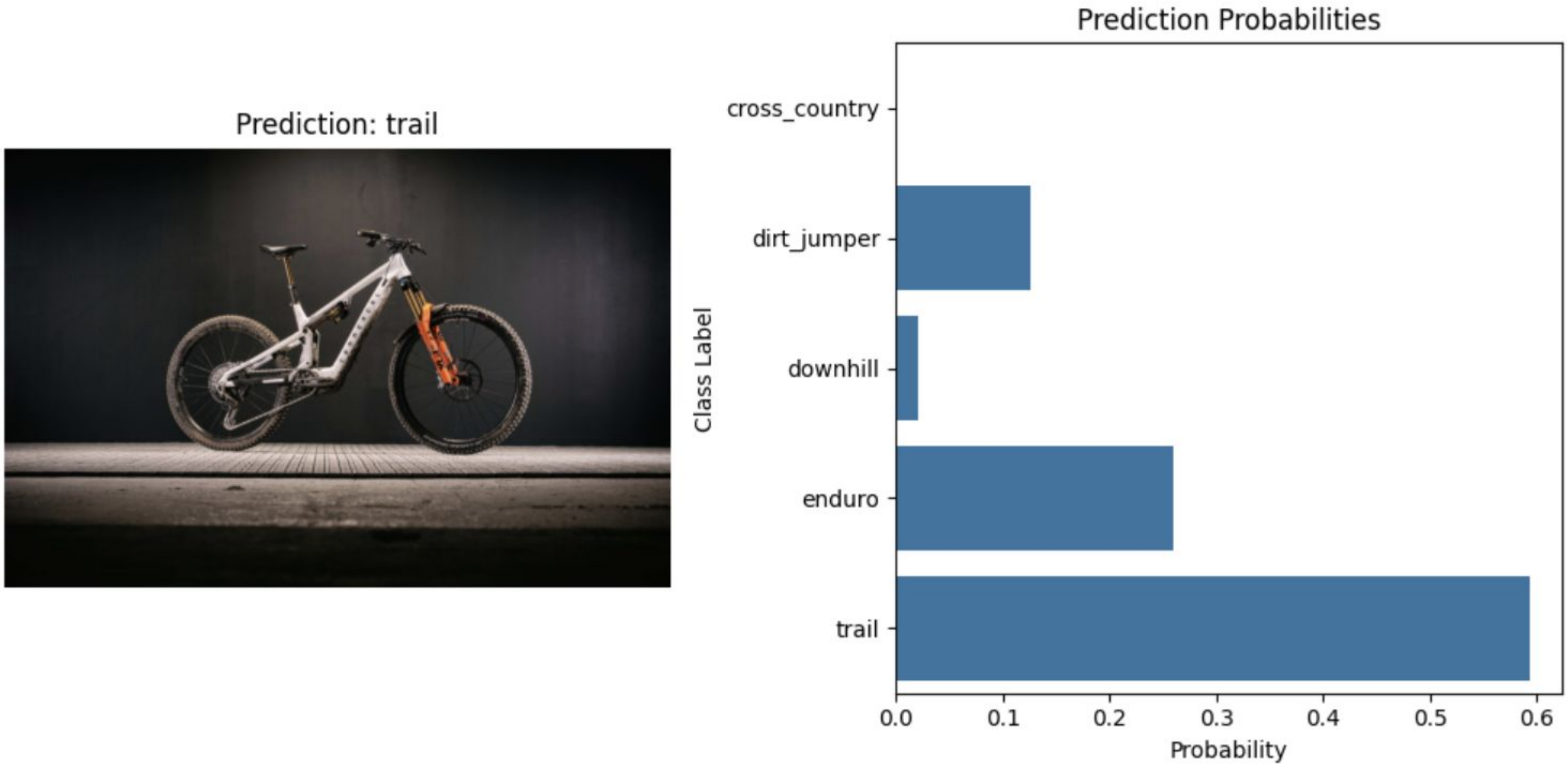
Deployment, Image 2

...



Deployment, Image 3

...

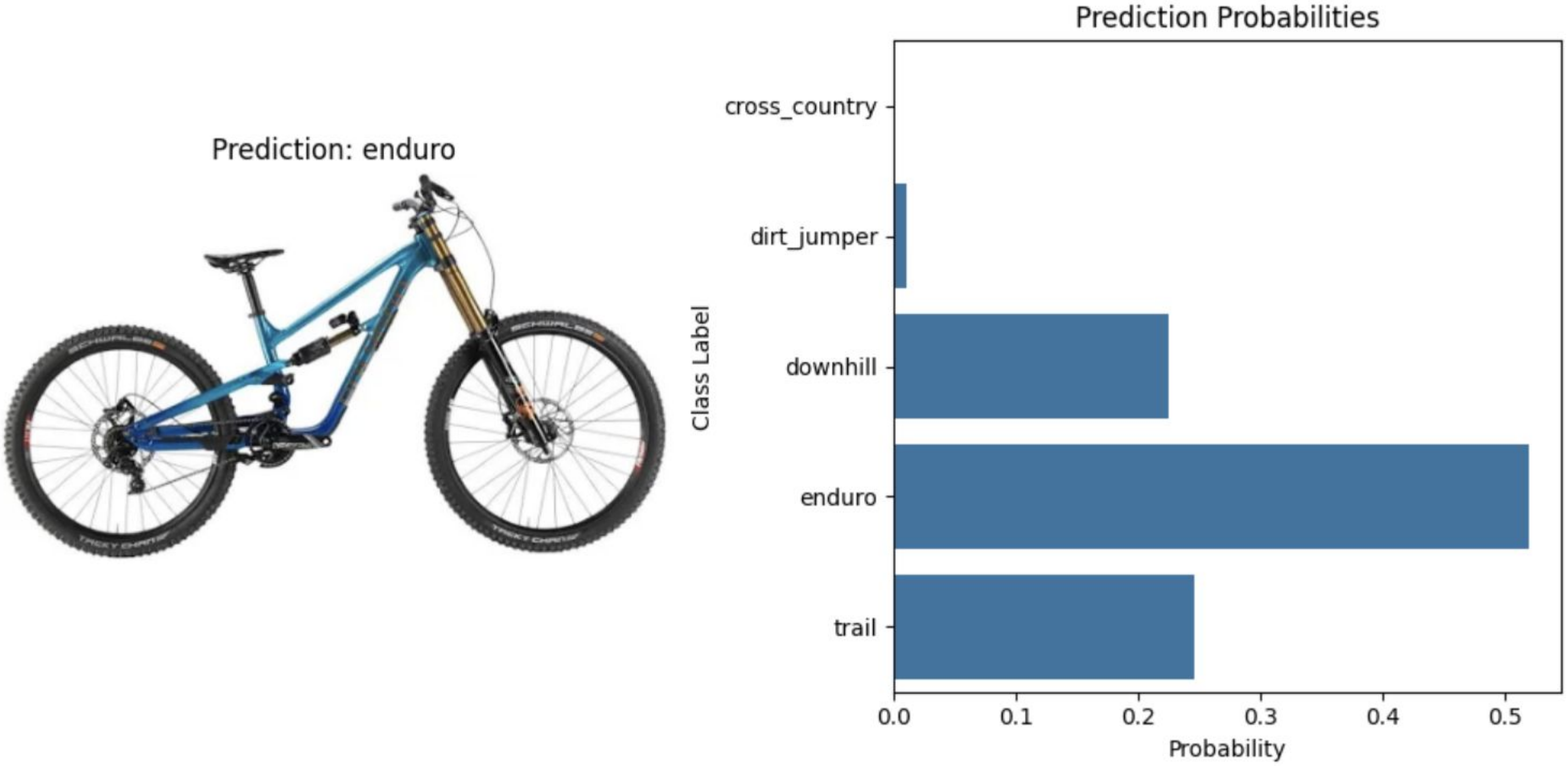


Prediction: trail

Prediction Probabilities

Deployment, Image 4

...



Final Evaluation

Domain Problem:

We want to develop a model to classify a bike based off of what it can see an image, into one of the following riding styles: **Cross Country**, **Downhill**, **Enduro**, **Trail** or **Dirt Jumper**.

Expansions for Improved Accuracy:

I believe the biggest things that could help make the model more accurate would be **limiting the image amount for each category.**