Harvard John A. Paulson
School of Engineering
and Applied Sciences

A-Eye App Project Outline

AC295
Advanced Practical Data Science, MLOps

Group Name: Pinkdrink
Anita Mahinpei, Jiahui Tang, Benjamin Liu,
Yingchen Liu, James Parker

Outline

- Problem Definition & Background
- Proposed Solution
- App Design
- Project Scope
- Project Workflow
- Process Flow
- Data
- Infrastructure
- EDA Results
- Data Pipeline
- Models

Problem Definition & Background

The World Health Organization (WHO) estimated that 314 million people have visual impairment across the world, including 269 million who have low vision, and 45 million who are blind (Ono et al 2010). Many people with visual impairments rely on screen readers in order to access the internet through audio and thus depend on image captions (Yesilada et al 2004). Therefore, **accessibility**, as well as **automatic indexing** and other goals, make accurate **image captioning** an important priority (Hossain et al 2018).

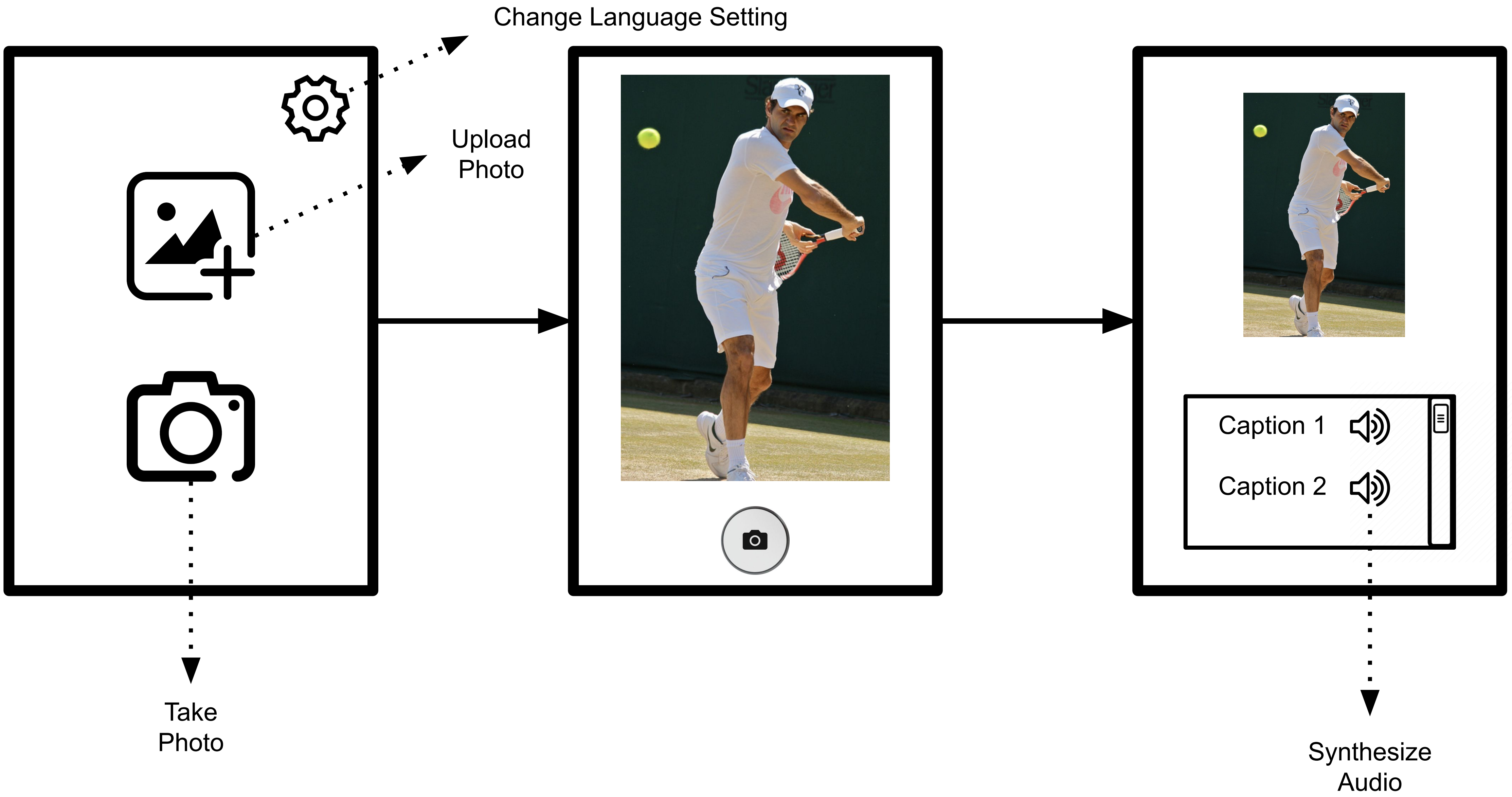
Proposed Solution

We will explore the realm of image captioning by creating an app that allows users to *upload images and have them be captioned in parallel through multiple models*. We specifically focus on creating captions for images of objects and scenery. The app will generate and display several possible captions for the image. Since the intended audiences are visually impaired individuals, the app will provide a text to audio functionality so that the generated captions can be read out loud to the user if desired. In order to support a broad range of audiences, we will allow users to select the language they wish to use for the generated captions.

Proposed Solution

Our models will be trained with a dataset of more than 330k images with 1.5 million object instances, covering more than 80 entity categories. We will explore a full-stack data science process starting from deep learning, operations to deployment and eventually distribute the project app in a contained environment to a wider range of audiences.

App Design


Change Language Setting
Upload Photo
Take Photo
Caption 1
Caption 2
Synthesize Audio

App Design

Image-to-Text:
- Trained Encoder Decoder Model
- Generate English Caption

Text-to-Text:
- Google Translate API
- Translate English to User-Specified Language

Text-to-Audio:
- Google Cloud Text-to-Speech API
- Read Caption in User-Specified Language

Project Scope (A-Eye App)

Proof Of Concept (POC)

- Download MSCOCO data
- Perform EDA to verify data
- Set up data pipeline (i.e. resize all images to a fixed size, normalize pixel values, tokenize the captions, store in tf dataset)
- Experiment with some baseline models trained on a subset of the dataset
- Validate captioning results on unseen images

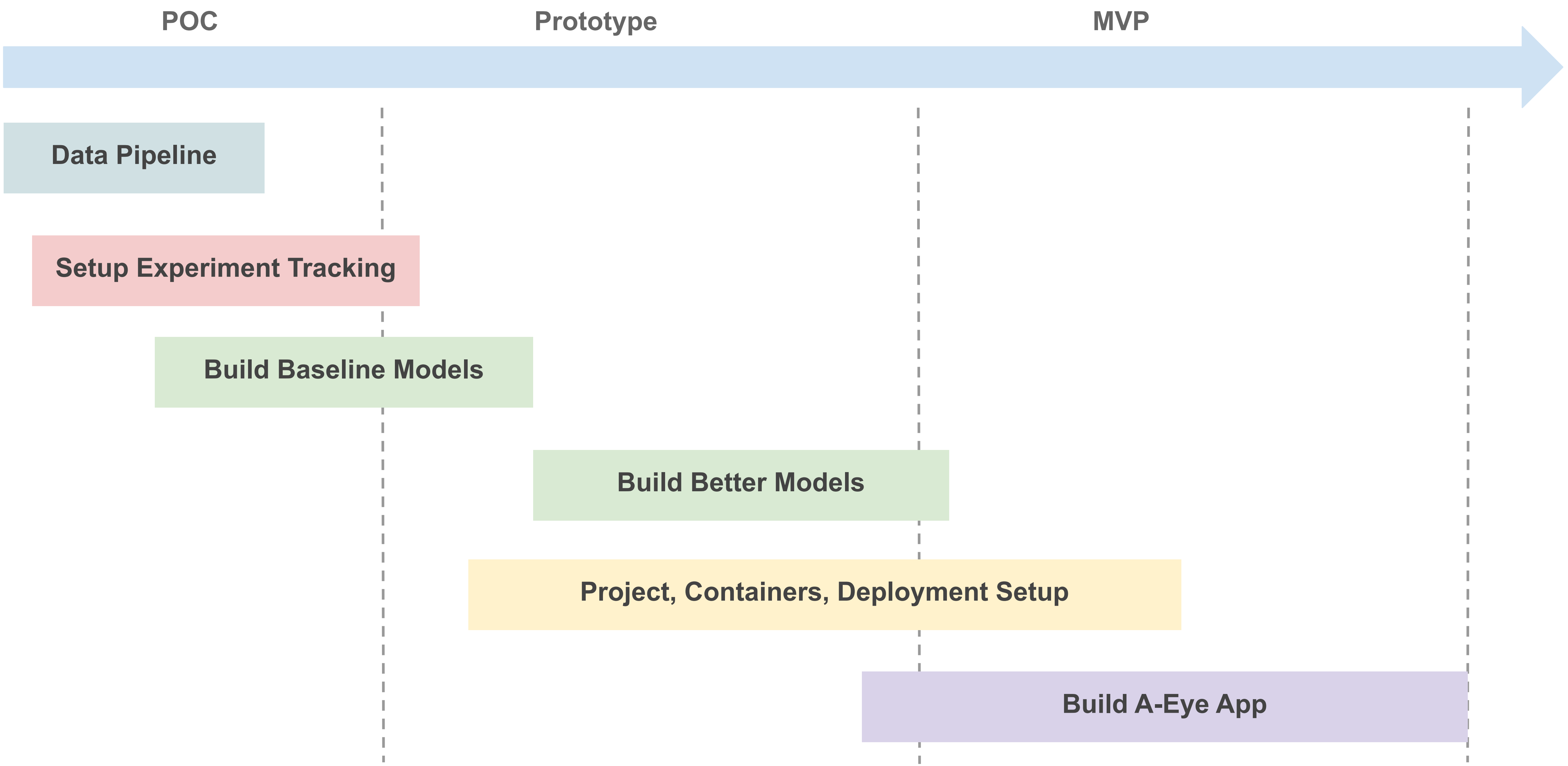
Prototype

- Create a mockup of screens to see how the app could look like
- Deploy one model to Fast API to service model predictions as an API

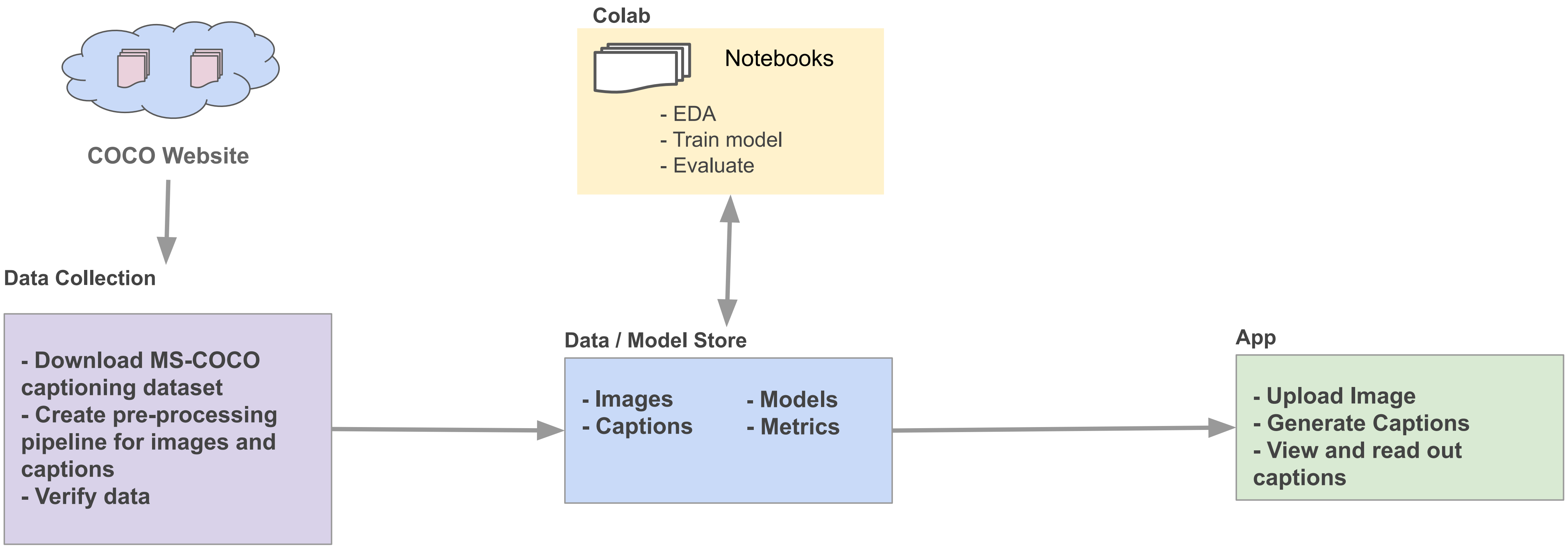
Minimum Viable Product (MVP)

- Create App to caption images
- API Server for uploading images and predicting using best model

Project Workflow



Process Flow



Data

We will use the Microsoft Common Objects in Context (COCO) data for our project. COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- Object segmentation
- Recognition in context
- Superpixel stuff segmentation
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 to 7 captions per image
- 250,000 people with keypoints

Data

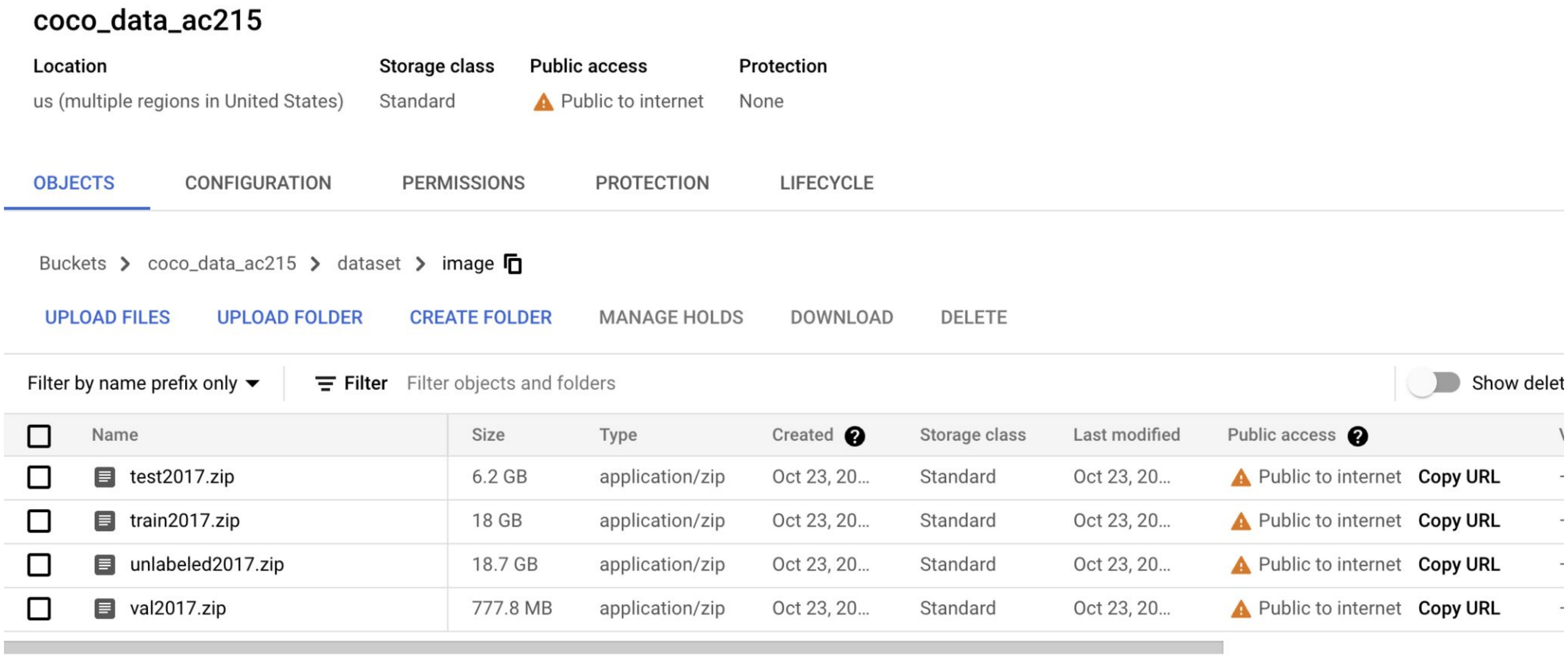
We will be using the images with the caption labels for training models.

Sample image with 5 gold captions:

A couple of people riding on top of a wave on surfboards.
A man rides a white surfboard near another person in the ocean.
one person surfing one person laying on a surfboard
The guy is riding the wave as a girl watches.
Surfers surfing in the ocean on a clear day.

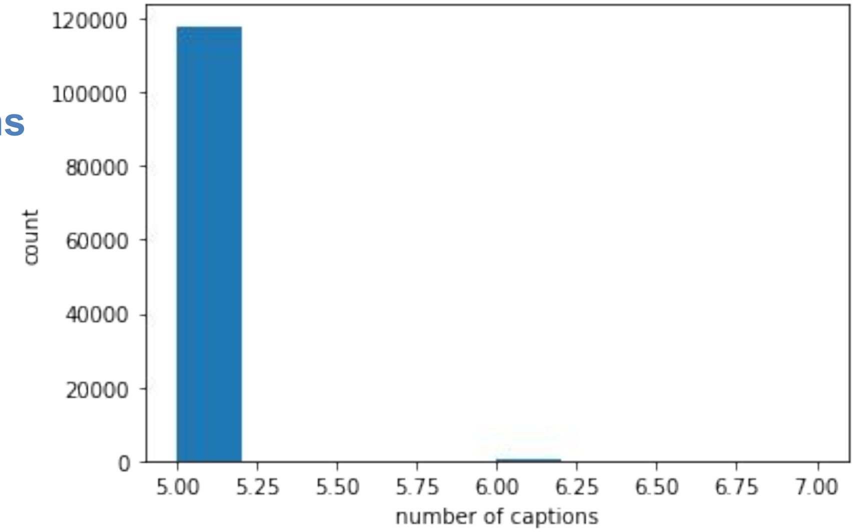
Infrastructure

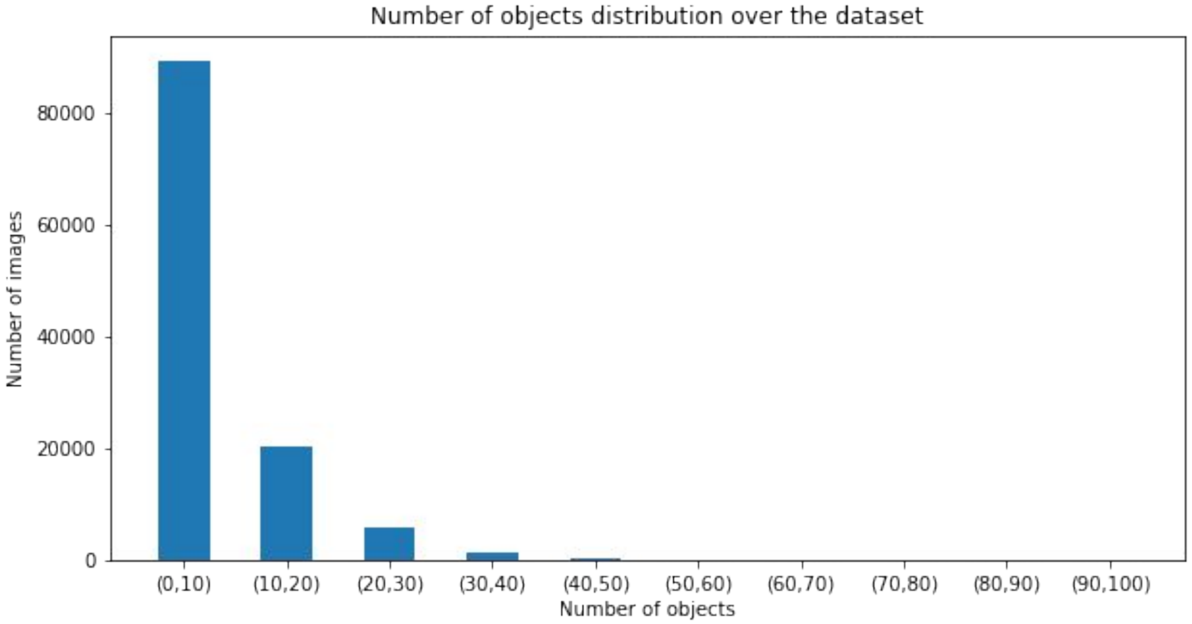
We've set up team project on **Google Cloud Platform** with raw COCO data uploaded to Google Cloud Storage and VM Instance for future deployment, scaling and automation.



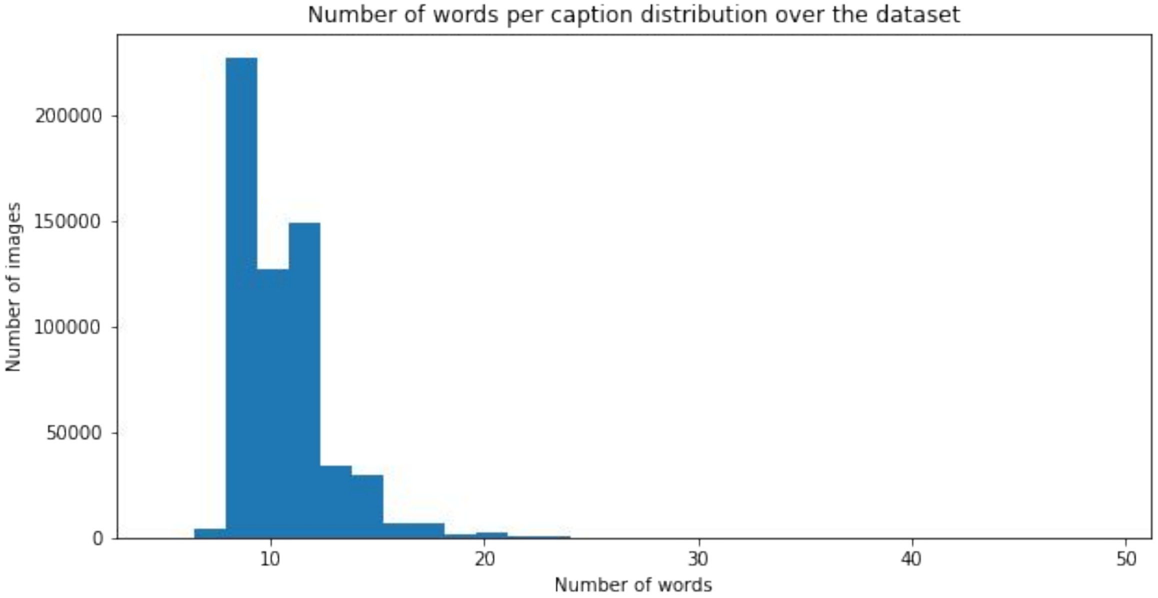
EDA Results

Most images have 5 captions



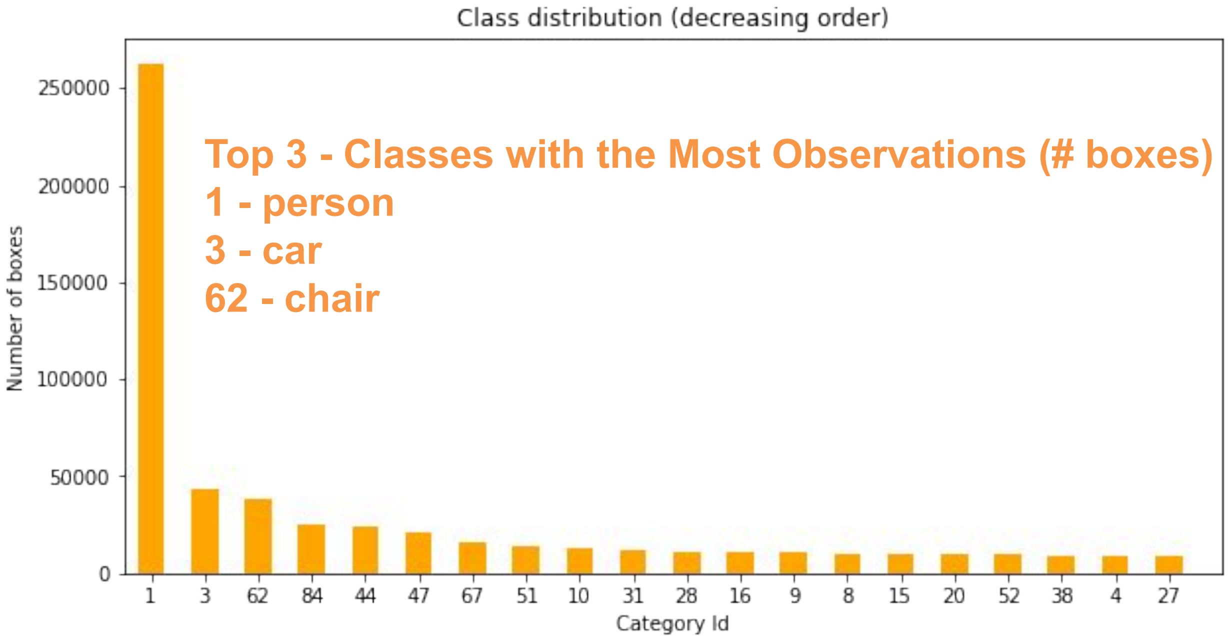


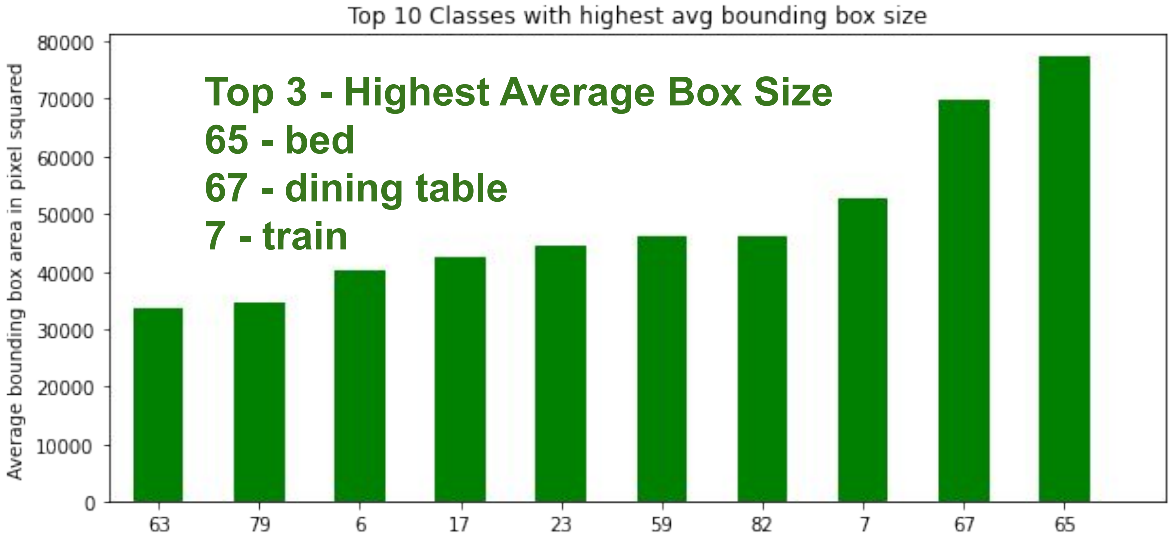
Most images contain less than 10 objects

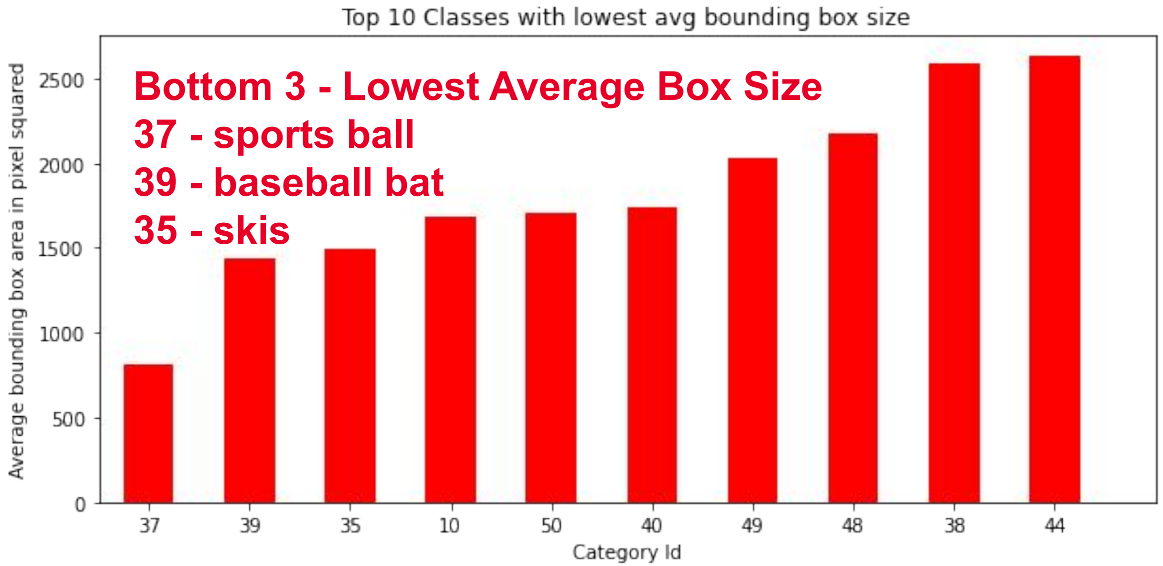


The distribution of the number of words in the captions is slightly right skewed but in general centered around 10

EDA Results


Class distribution (decreasing order)
Top 3 - Classes with the Most Observations (# boxes)
1 - person
3 - car
62 - chair
Number of boxes
250000
200000
150000
100000
50000
0
1
3
62
84
44
47
67
51
10
31
28
16
9
8
15
20
52
38
4
27
Category Id


Top 10 Classes with highest avg bounding box size
Top 3 - Highest Average Box Size
65 - bed
67 - dining table
7 - train
Average bounding box area in pixel squared
80000
70000
60000
50000
40000
30000
20000
10000
0
63
79
6
17
23
59
82
7
67
65


Top 10 Classes with lowest avg bounding box size
Bottom 3 - Lowest Average Box Size
37 - sports ball
39 - baseball bat
35 - skis
Average bounding box area in pixel squared
2500
2000
1500
1000
500
0
37
39
35
10
50
40
49
48
38
44
Category Id

Data Pipeline

The data processing pipeline is as follows:

- scale and crop images to the appropriate dimensions for the selected encoder (e.g. 299 by 299 for InceptionV3)
- normalize pixel values as needed for the selected encoder (e.g. -1 to 1 for InceptionV3)
- Process the captions by:
  - splitting at white spaces
  - adding <start> and <end> of sentence tokens
  - padding/cropping to the max allowed caption length
  - limit vocabulary size by replacing in-frequent words with the <unk> token
  - Map vocabulary tokens to integers (one-hot-encoding)

Models: Overview

In terms of deep learning models, both computer vision and natural language processing models will be used.

- **Computer Vision:** Pre-trained, frozen CNN architectures (e.g. VGG, Inception) will be used to extract features from the images. (Note: we decided to use frozen encoders trained on ImageNet because we ran a trial without freezing layers and observed a lot of over-fitting.)
- **Language:** RNN and LSTM structures
- **Attention:** As described in the paper, "Show, Attend, and Tell", models that incorporate attention to the image feature map perform, have improved performance. We will use the attention mechanism described in this paper to improve performance.

Models: Baseline

We have trained the following models on a subset of the COCO dataset with 6000 images:

1. **Inception-GRU:**
    - Extract feature map from the last CNN layer of frozen InceptionV3 with ImageNet weights
    - Attend to feature map
    - Decoder with an embedding layer (not pre-trained), a GRU layer and 2 fully-connected layers
2. **VGG16-GRU**
    - Extract feature map from the last layer CNN layer of frozen VGG16
    - Same decoder and attention as Inception-GRU model
3. **Inception-LSTM:**
    - Extract feature map from the last CNN layer of frozen InceptionV3 with ImageNet weights
    - Trained with 4800 images, 600 test images, and 600 validation images
    - Attend to feature map
    - Decoder with an embedding_layer(not pre-trained), a LSTM layer and 1 fully-connected layer

Models: Evaluation

We compare the performance of our models using the loss function and BLEU-4 scores on held-out test data. The BLEU score is a method of comparing generated texts against a set of golden text labels (in our case image captions) as described in this paper.

Models: Training Results

Currently we trained with 6000 images each with 5 captions. We can see that the VGG model is performing the best with respect to BLEU-4 score but the performances are fairly close. So one possibility is to use multiple models in our final product in order to give greater caption variety in the list of captions generated.

Model	Test Loss(10 epochs)	Test BLEU-4
Inception(frozen)-GRU	3.54	0.00142
VGG(frozen)-GRU	3.77	0.00163
Inception(frozen)-LSTM	2.33	0.00095

Note:

- loss functions are reported without teacher forcing.
- The LSTM model has a different loss function and BLEU score (the BLEU score is computed against a single caption rather than a set of captions). So we can’t compare it yet with the other two models. We’re still working on resolving this.

Models: Sample Inception(frozen)-GRU Model Captions

Example Caption: woman holding up a small boy with blue shirt at beach

Predicted Caption: a young lady is carrying a kite event

Example Caption: a man on a skateboard in on a ramp

Predicted Caption: a skateboarder is performing a trick in the air doing his <unk>

Example Caption: a tennis player is swinging at a tennis ball on a court

Predicted Caption: athlete hitting a crowd retrieving a tennis ball in a serious mound

Models: Sample Inception(frozen)-LSTM Model Captions

Example Caption: street sign that tells bicyclers not to park.

Predicted Caption: a man is parked in front of car sign

Example Caption: bathroom sink displayed under large vanity mirror.

Predicted Caption: bathroom with sink and sink

Example Caption: Two teenage girls performing chores in kitchen.

Predicted Caption: woman standing in kitchen with two people in kitchen

Example Caption: kitchen with cabinets that have glass doors

Predicted Caption: kitchen with white appliances and white cabinets and white appliances

Models: Sample VGG-GRU Model Captions

Example Caption: Snowboarders doing stunts on a ramp in the snow

Predicted Caption: A lady skiing in the snow

Example Caption: A young child is throwing a frisbee during a game

Predicted Caption: A beach area with three kite and carrying a pink frisbee

Example Caption: black and white photograph of man on skateboard carrying a surfboard

Predicted Caption: an old man rides them