DevOps in Machine Learning

From Dev to Prod in 90 Days

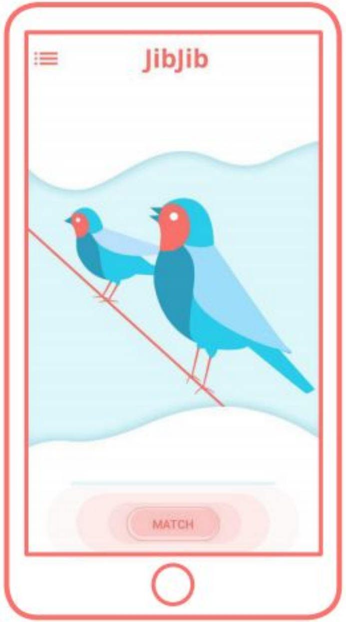
DEVOPSDAYS Berlin 2018

Alexander Knipping | DevOps Days Berlin 2018

JibJib

- Entry for Coding Davinci OST 2018 Hackathon
- "Shazam for Birds"
- Android App ←→ Backend ←→ TensorFlow Model
- Google Play Store: JibJib
- https://github.com/gojibjib

JibJib



JibJib
MATCH

MATCHED BIRDS
Felsentaube (Columba livia) 0.96203816
Wintergoldhähnchen (Regulus regulus) 0.03078022
Waldlaubsänger (Phylloscopus sibilatrix) 0.007181617

Elster (Pica pica)
https://de.wikipedia.org/wiki/Pica_pica
Die Elster (Pica pica) ist eine Vogelart aus der Familie der Rabenvögel. Sie besiedelt weite Teile Europas und Asiens sowie das nördliche Nordafrika. In Europa ist sie vor allem im Siedlungsraum häufig. Aufgrund ihres charakteristischen schwarz-weißen Gefieders mit den auffallend langen Schwanzfedern ist sie auch für den vogelkundlichen Laien unverwechselbar.
In der germanischen Mythologie war die Elster sowohl Götterbote als auch der Vogel der Todesgöttin Hel, so dass sie in Europa den Ruf des Unheilsboten bekam. Als „diebische" Elster war sie auch im Mittelalter als Hexentier und Galgenvogel unbeliebt. Im Gegensatz dazu gilt sie in Asien traditionell als Glücksbringer und
SAVE

JibJib

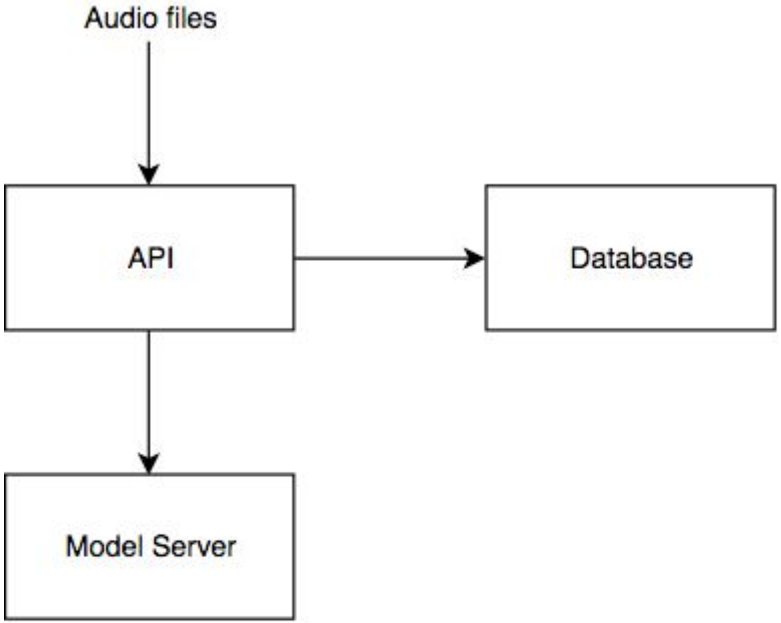
- Development & deployment process
  - Collecting & preparing data
  - Training the model
  - Building the backend
  - Serving the model
  - Deployment
- **Goal**: giving ideas for your own project!

Part 0

Overview

✓ Architecture
✓ Crash Course: ML

Architecture



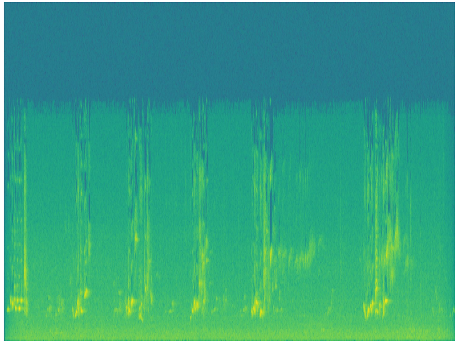

```
{
    "status": 200,
    "message": "Detection successful",
    "count": 3,
    "data": [
        {
            "accuracy": 0.7741285540736146,
            "id": 110
        },
        {
            "accuracy": 0.14705901025204263,
            "id": 7
        },
        {
            "accuracy": 0.07881243567434278,
            "id": 184
        }
    ]
}
```

Crash Course: Machine Learning

Goal: Build classifier to identify N birds by their calls

Input: Vector representing pixels of a spectrogram

Output: Vector of N probabilities (one for each bird)



[254, 243, 0, 143, 1, ...]

Mallard Cuckoo

[0.02, 0.8, 0.003, 0.1, ...]

Blackbird

Crash Course: Machine Learning

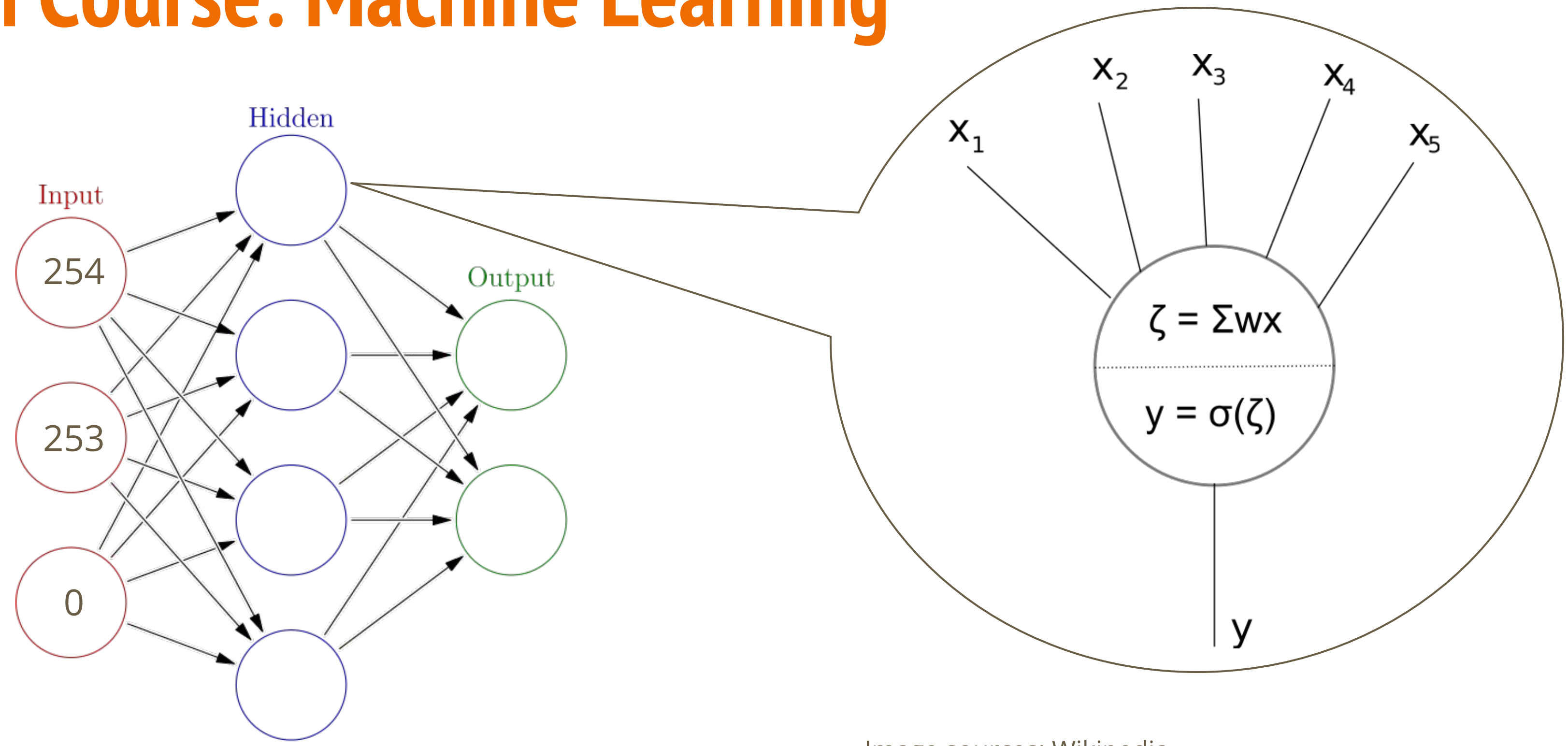
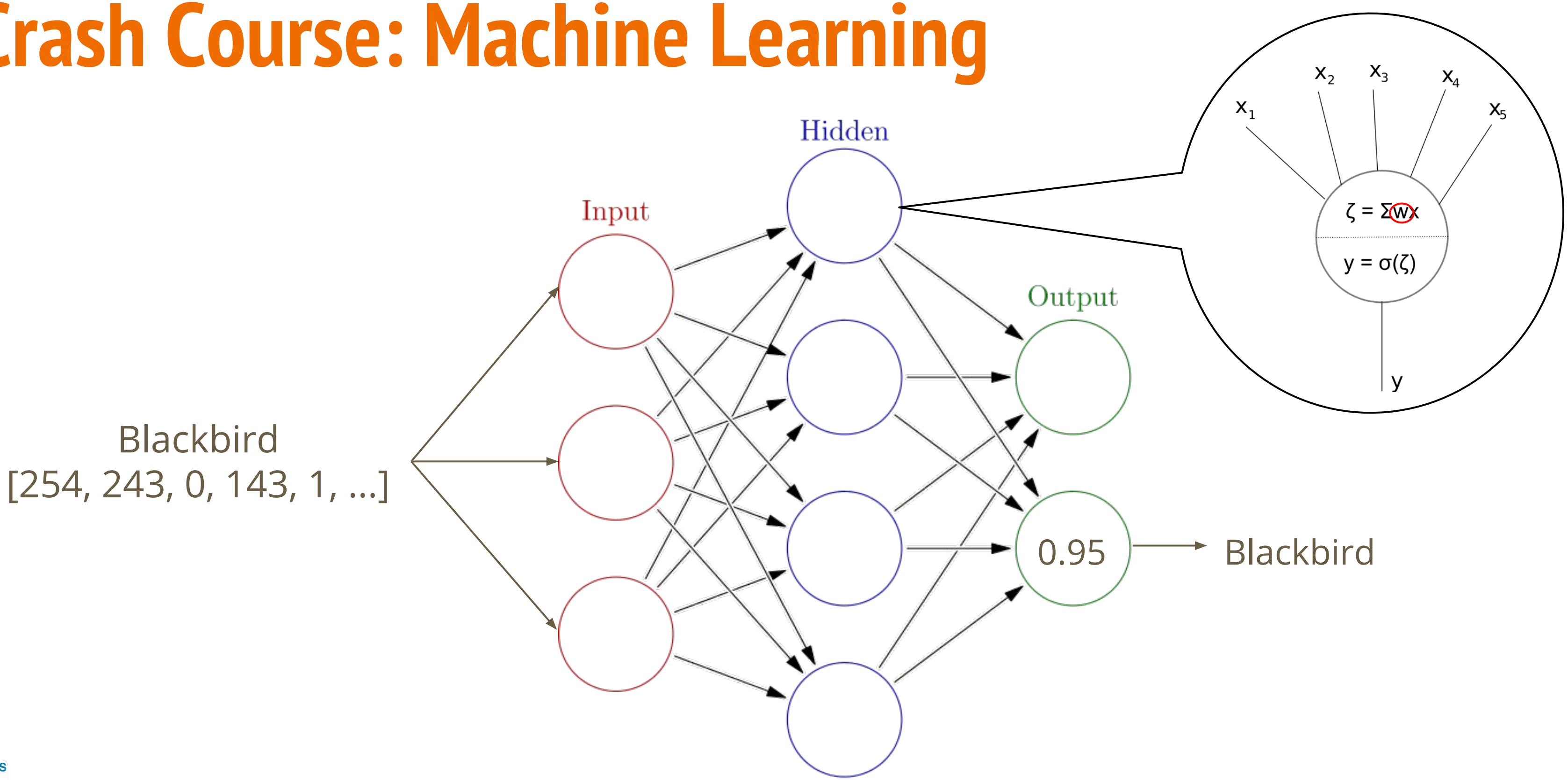


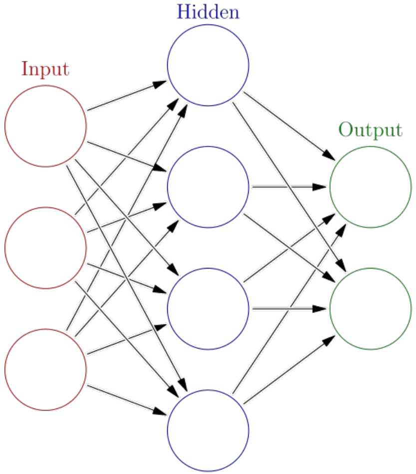
Image sources: Wikipedia

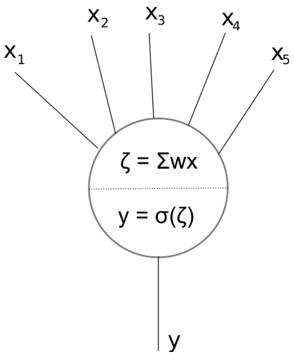
Crash Course: Machine Learning



Designing a model

- Input pre-processing?
- How many layers?
- How deep should layers be?
- What are initial weights to "start" the model?
- Which function(s) to use in the layers?
- How big/small should steps in gradient descent be?





Part 1

Data Science

- ✓ Building a Model
- ✓ Getting a Data Set

The Model

- How to design a model from scratch in ~ 30 days?

You (most likely) don't!

The Model

- Google's AudioSet (2017): dataset of 2+ mio. human-labeled 10 sec. YouTube video soundtracks of 600+ classes

{ ||||| } AudioSet

2.1 million annotated videos

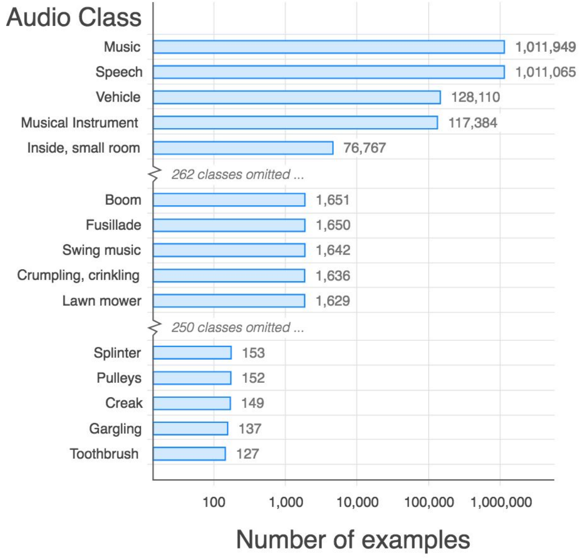
5.8 thousand hours of audio

527 classes of annotated sounds

Large-scale data collection

To collect all our data we worked with human annotators who verified the presence of sounds they heard within YouTube segments. To nominate segments for annotation, we relied on YouTube metadata and content-based search.

Our resulting dataset has excellent coverage over the audio event classes in our ontology.



https://research.google.com/audioset/

Google

Hershey et al. (2017)
https://ai.google/research/pubs/pub45611

CNN ARCHITECTURES FOR LARGE-SCALE AUDIO CLASSIFICATION

Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, Kevin Wilson

Google, Inc., New York, NY, and Mountain View, CA, USA

shershey@google.com

[cs.SD] 10 Jan 2017

ABSTRACT

Convolutional Neural Networks (CNNs) have proven very effective in image classification and show promise for audio. We use various CNN architectures to classify the soundtracks of a dataset of 70M training videos (5.24 million hours) with 30,871 video-level labels. We examine fully connected Deep Neural Networks (DNNs), AlexNet [1], VGG [2], Inception [3], and ResNet [4]. We investigate varying the size of both training set and label vocabulary, finding that analogs of the CNNs used in image classification do well on our audio classification task, and larger training and label sets help up to a point. A model using embeddings from these classifiers does much better than raw features on the *Audio Set* [5] Acoustic Event Detection (AED) classification task.

Index Terms— Acoustic Event Detection, Acoustic Scene Classification, Convolutional Neural Networks, Deep Neural Networks, Video Classification

Eghbal-Zadeh et al. [19] recently won the DCASE 2016 Acoustic Scene Classification (ASC) task, which, like soundtrack classification, involves assigning a single label to an audio clip containing many events. Their system used spectrogram features feeding a VGG classifier, similar to one of the classifiers in our work. This paper, however, compares the performance of several different architectures. To our knowledge, we are the first to publish results of Inception and ResNet networks applied to audio.

We aggregate local classifications to whole-soundtrack decisions by imitating the visual-based video classification of Ng et al. [20]. After investigating several more complex models for combining information across time, they found simple averaging of single-frame CNN classification outputs performed nearly as well. By analogy, we apply a classifier to a series of non-overlapping segments, then average all the sets of classifier outputs.

Kumar et al. [21] consider AED in a dataset with video-level labels as a Multiple Instance Learning (MIL) problem, but remark that scaling such approaches remains an open problem. By contrast, we

AUDIO SET: AN ONTOLOGY AND HUMAN-LABELED DATASET FOR AUDIO EVENTS

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, Marvin Ritter

Google, Inc., Mountain View, CA, and New York, NY, USA

{jgemmeke,dpwe,freedmand,arenjansen,wadelawrence,channingmoore,plakal,marvinritter}@google.com

ABSTRACT

Audio event recognition, the human-like ability to identify and relate sounds from audio, is a nascent problem in machine perception. Comparable problems such as object detection in images have reaped enormous benefits from comprehensive datasets – principally ImageNet. This paper describes the creation of *Audio Set*, a large-scale dataset of manually-annotated audio events that endeavors to bridge the gap in data availability between image and audio research. Using a carefully structured hierarchical ontology of 632 audio classes guided by the literature and manual curation, we collect data from human labelers to probe the presence of specific audio classes in 10 second segments of YouTube videos. Segments are proposed for labeling using searches based on metadata, context (e.g., links), and content analysis. The result is a dataset of unprecedented breadth and size that will, we hope, substantially stimulate the development of high-performance audio event recognizers.

Index Terms— Audio event detection, sound ontology, audio databases, data collection

among 50 environmental sounds. LeMaitre and Heller [8] proposed a taxonomy of sound events distinguishing objects and actions, and used identification time and priming effects to show that listeners find a “middle range” of abstraction most natural.

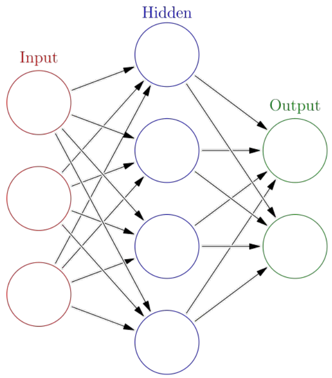
Engineering-oriented taxonomies and datasets began with Gaver [9] who used perceptual factors to guide the design of synthetic sound effects conveying different actions and materials (tapping, scraping, etc.). Nakatani & Okuno [10] devised a sound ontology to support real-world computational auditory scene analysis. Burger et al. [11] developed a set of 42 “noisemes” (by analogy with phonemes) to provide a practical basis for fine-grained manual annotation of 5.6 hours of web video soundtrack. Sharing many of the motivations of this paper, Salamon et al. [12] released a dataset of 18.5 hours of urban sound recordings selected from `freesound.org`, labeled at fine temporal resolution with 10 low-level sound categories chosen from their urban sound taxonomy of 101 categories. Most recently, Säger et al. [13] systematically constructed adjective-noun and verb-noun pairs from tags applied to entire `freesound.org` recordings to construct AudioSen-

Genmeke et al. (2017)
https://ai.google/research/pubs/pub45857

The Model

- Google's AudioSet (2017): dataset of 2+ mio. human-labeled 10 sec YouTube video soundtracks of 600+ classes
- VGGish: pre-trained VGG-like model defined in TensorFlow (https://github.com/tensorflow/models/tree/master/research/audioset)

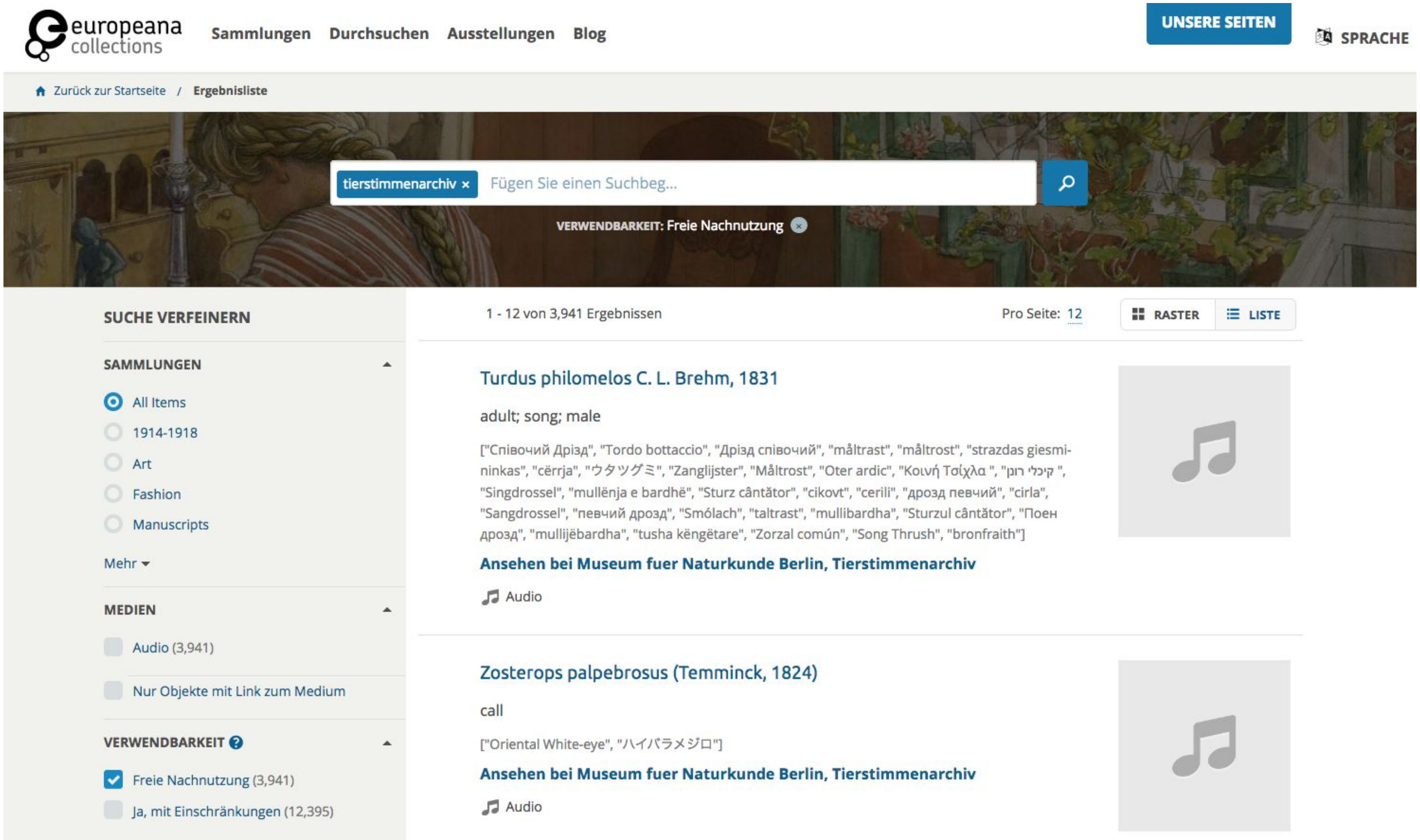
→ "warm start"



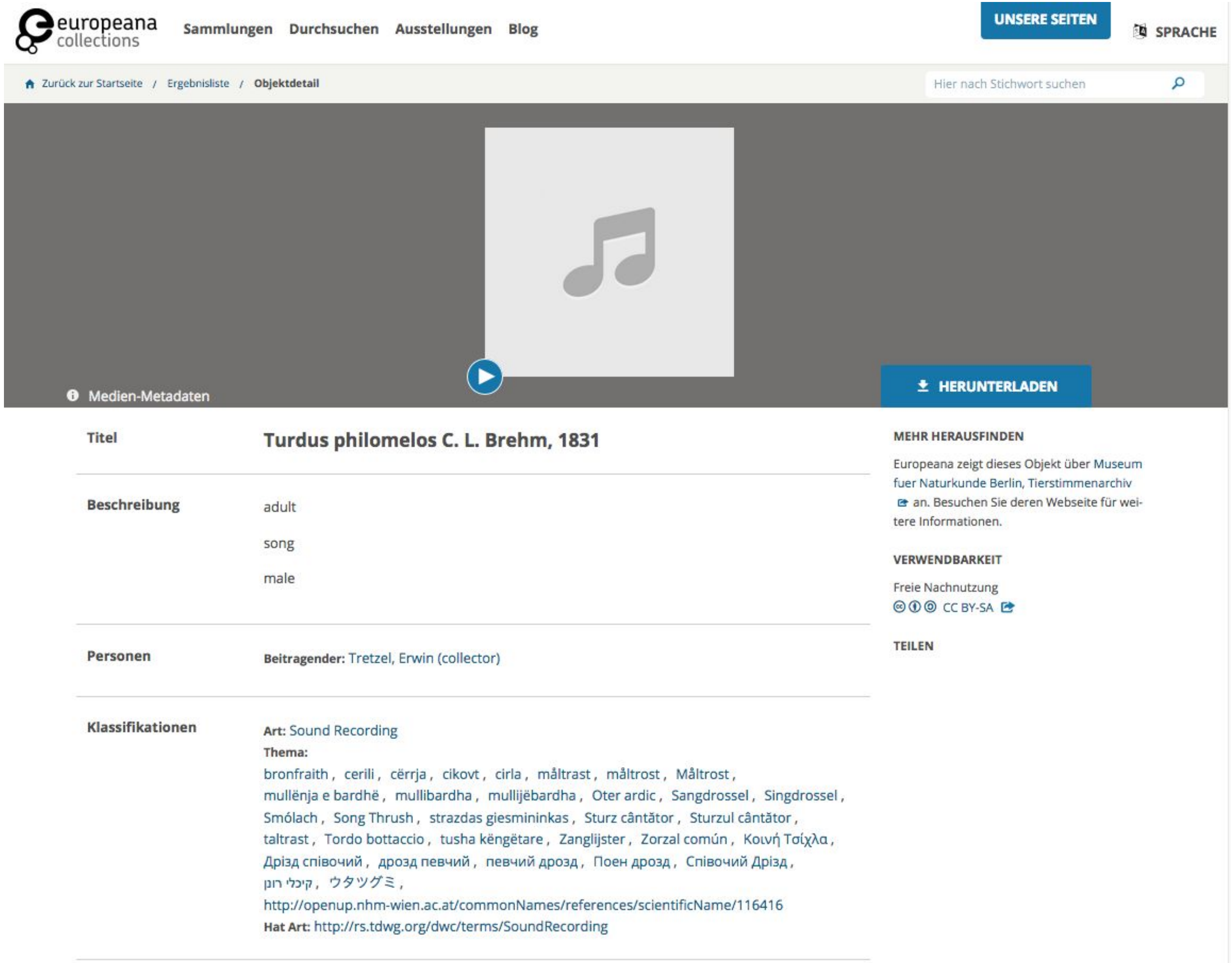
The Process

Initial Dataset

- Museum of Natural History Berlin (MFN) data set for CDV OST 2018
- ~ 4000 openly licensed out of 120k from tierstimmenarchiv.de
- Accessible through Europeana (metadata API)
- Questions:
  - How many different birds?
  - How many files per bird?
  - How to download them?



https://europeana.eu



https://europeana.eu

Initial Dataset

1. Write Europeana API Client
   - https://github.com/gojibjib/gopeana
2. Use the client to dump URLs + names into JSON
   - https://github.com/gojibjib/voice-grabber/blob/master/data_grabber/data_grabber.go
3. Iterate over JSON to download files
   - https://github.com/gojibjib/voice-grabber/blob/master/file_grabber/file_grabber.go

Initial Dataset

Evaluate/Pre-Process

Total size of dataset	6GB
Total # of files	3843
Unique classes	1189
Files / bird	3,23

Wikipedia

Idea: cross-reference list of birds with German wikipedia

1. Query Wikipedia API to retrieve descriptions for all birds
2. Throw out all classes without a German entry
3. Update our JSON

Total # of files	3400
Unique classes	800
Files / bird	4,25

xeno-canto.org

Select

Idea: use audio files from xeno-canto.org to enrich the dataset

xeno-canto
Sharing bird sounds from around the world

Search recordings... Search

Advanced Search
Tips

About | Explore | Upload Sounds | Forum | Mysteries | Articles | Log in / Register

XC371285

0:40

Sumba Myzomela (Myzomela dammermani) · song
Philippe Verbelen

In 1990 the Australian ornithologist Ron Johnstone observed a Myzomela on the island of Rote in the Lesser Sundas in Indonesia. Without pictures or sound recordings the birds were assumed to be Sumba Myzomelas. In 2009 the birds were observed again by Belgian birders Philippe Verbelen and Veerle Dossche who noticed that **the songs they recorded** were quite different from other Myzomela populations. In the intervening years the case has been properly studied and the taxon has now been described as Rote Myzomela *Myzomela irianawidodoae*. In due course it will be named as such here, for the time being it is stored under Sumba Myzomela.

Collection Statistics

418965	Recordings
9961	Species
10941	Subspecies
4712	Recordists
6575:56:01	Recording Time

More...

Latest New Species

- Blue Petrel
- Yellow-crowned Woodpecker
- Mindoro Scops Owl
- Sparkling-tailed Woodstar
- Gola Malimbe
- More...

What is xeno-canto?

xeno-canto is a website dedicated to sharing bird sounds from all over the world. Whether you are a research scientist, a birder, or simply curious about a sound that you heard out your kitchen window, we invite you to listen, download, and explore the bird sound recordings in the collection.

But xeno-canto is more than just a collection of recordings. It is also a collaborative project. We invite you to share your own bird recordings, help identify mystery recordings, or share your expertise in the forums. Welcome!

Try this!

Using Audacity

Audacity is a freely available software package that lets you analyse and visualise XC

DEVOPSDAYS Berlin 2018

http://xeno-canto.org

xeno-canto

Sharing bird sounds from around the world

blackbird Search

Advanced Search
Tips

About Explore Upload Sounds Forum Mysteries Articles Log in / Register

Recordings

4463 results from 32 species for query '**blackbird**' (foreground species only) (3.37s)

- Results format: detailed | **concise** | **codes** | **sonograms**

1 2 3 4 5 6 7 8 9 ... 149 Next >

Common name / Scientific	Length	Recordist	Date	Time	Country	Location	Elev. (m)	Type	Remarks	Actions	Cat.nr.
White-collared Blackbird (Turdus albocinctus)	4:15	**Jens Kirkeby**	2018-06-04	09:00	Bhutan	**Gangtey Palace, Paro**	2300	song	Singing from treetop near farmhouse. Recorded in mono, using Røde NTG-3/MixPre-3. bird-seen:yes playback-used:no **[also] [sono]**	A B C D E	**XC422862**
White-collared Blackbird (Turdus albocinctus)	0:27	**Rolf A. de By**	2018-03-18	17:48	India	**Mud Hut, Chopta/Chamoli, Uttarakhand**	2900	courtship subsong, male, song	Male courting around a female high up in a leafless tree. High winds making this inaudible from 15 m distance. Until I used the parabola and heard this really soft, whispered, actually intimate song. Only meant for her to hear. bird-seen:yes playback-used:no **[sono]**	A B C D E	**XC407857**
White-collared Blackbird (Turdus albocinctus)	0:37	**Rolf A. de By**	2018-03-18	07:14	India	**Mud Hut, Chopta/Chamoli, Uttarakhand**	3150	alarm call, flock calls	A group of six or so WCBs, mixed sexes actively chasing each other around the canopy of three trees. bird-seen:yes playback-used:no **[sono]**	A B C D E	**XC407654**

http://xeno-canto.org

xeno-canto.org

Idea: use audio files from xeno-canto.org to enrich the dataset

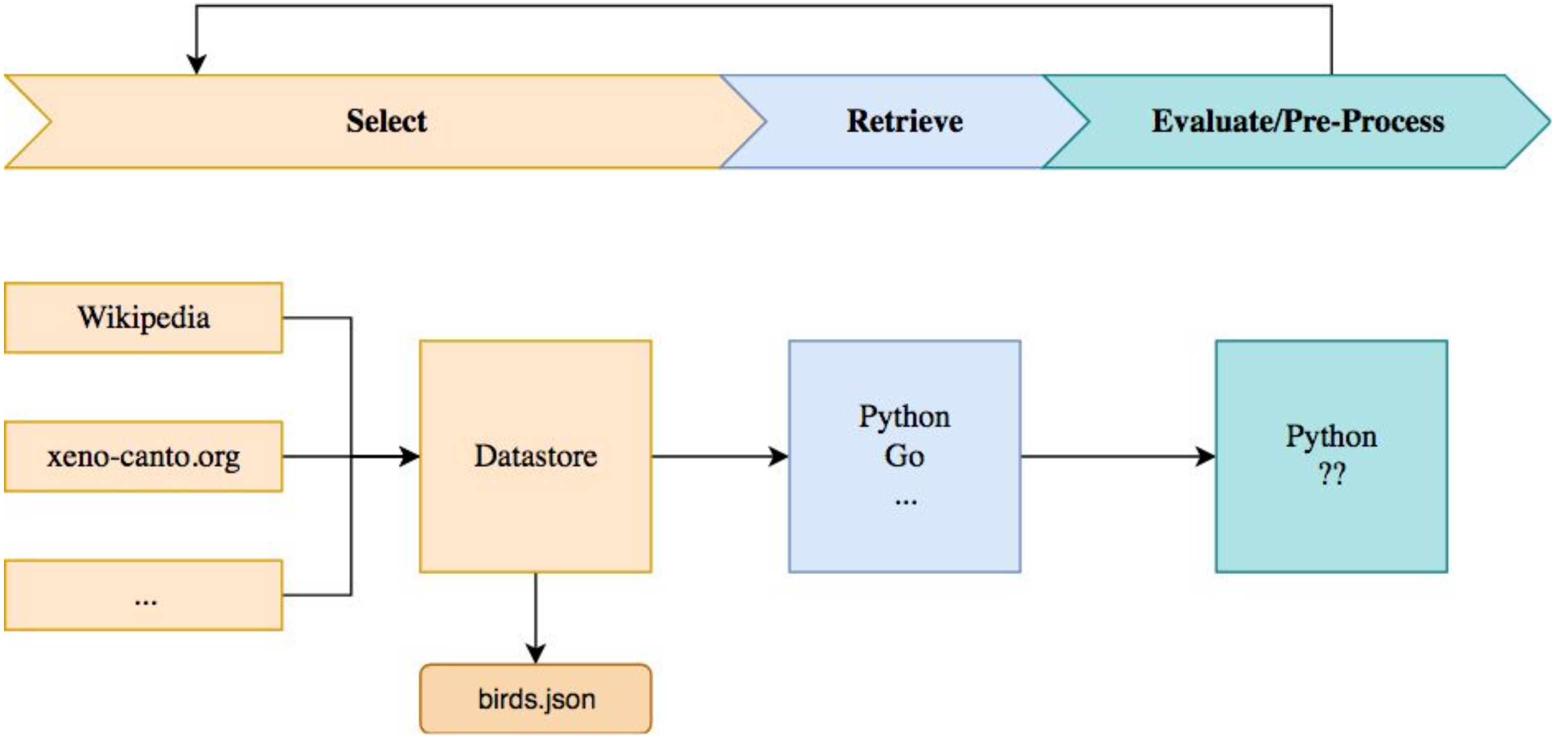
1. Get a list of birds with recordings from xento-canto.org in Germany
2. Intersect the list with our JSON file: 194 species
3. Use Python to crawl xeno-canto.org to retrieve download URLs of max. 50 audio files for each bird, save URL in JSON
4. Use Go to download the audio files from the URLs in our JSON
5. Cut audio files into 10 - 20s parts

Final dataset

Evaluate/Pre-Process

Total size of dataset	120GB
Total # of files	80 000
Unique classes	194
Files / bird	412,4

Retrieval Process



birds.json

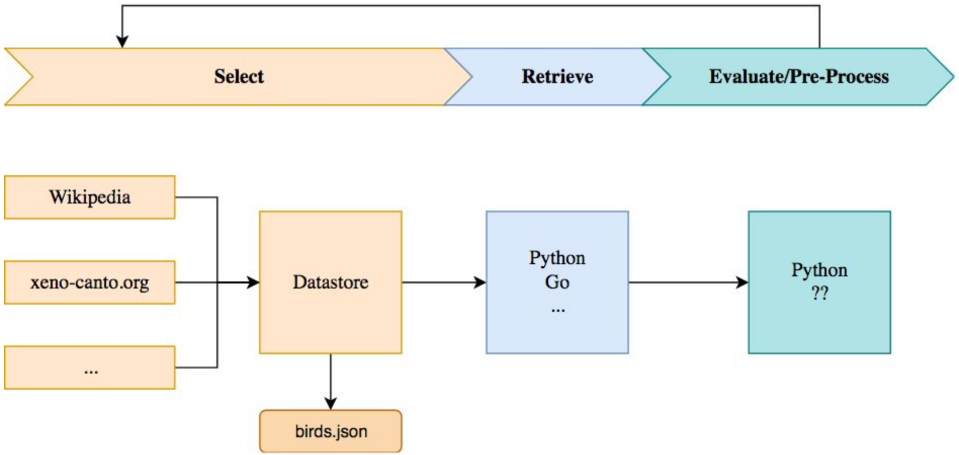

```json
[
    {
        "id": 0,
        "name": "Chlidonias niger",
        "genus": "Chlidonias",
        "species": "niger",
        "title_de": "Trauerseeschwalbe",
        "title_en": "Black tern",
        "desc_de": "Die Trauerseeschwalbe (Chlidonias niger) ist eine Vogelart aus der Familie der Seeschwalben (Sternidae), die in der Holarktis vorkommt.\\nDie Bestandssituation der Trauerseeschwalbe wurde 2016 in der Roten Liste gefährdeter Arten der IUCN als „Least Concern (LC)“ = „nicht gefährdet“ eingestuft.\\n\\n\\n== Beschreibung ==\\nDie Trauerseeschwalbe erreicht eine Körperlänge von bis zu 25 Zentimeter. Sie ist damit deutlich kleiner als die Flussseeschwalbe. Es besteht kein auffälliger Sexualdimorphismus.\\nIm Prachtkleid sind der Kopf, die Brust und der Bauch tiefschwarz und die Körperoberseite ist grau gefärbt. Die Flügelunterseite ist hellgrau, die hintere Körperunterseite ist schwarzgrau, während die Unterschwanzdecken weiß sind. Im Schlichtkleid ist die Oberseite dunkelgrau, die Unterseite ist weiß. Auf den Brustseiten befindet sich vor dem Flügelansatz jeweils ein dunkler Fleck. Ihr spitzer Schnabel ist schwarz und die Füße rotschwarz gefärbt.\\n",
        "desc_en": "The black tern (Chlidonias niger) is a small tern generally found in or near inland water in Europe and North America. As its name suggests, it has predominantly dark plumage. In some lights it can appear blue in the breeding season, hence the old English name \"blue darr\". The genus name is from Ancient Greek khelidonios, \"swallow-like\", from khelidon, \"swallow\": another old English name for the black tern is \"carr (i.e. lake) swallow\". The species name is from Latin niger \"shining black\".\\n\\n\\n",
    }
]
```


DEVOPSDAYS Berlin 2018

Retrieval Process

The requirement of...

- Interdisciplinary cooperation
- Good tooling
- Automatable steps
- Short feedback loops



Part 1.5

Model Training

Model Training

1. Setup, adjust VGGish
    - https://github.com/gojibjib/jibjib-model
2. Deploy dataset
3. Start with docker-nvidia
    - https://github.com/NVIDIA/nvidia-docker
4. Wait
5. Evaluate
6. Export

Model Training

Input:

- Dataset

Output:

- Logs, metrics
- Model.ckpt
- *mappings.pickle*

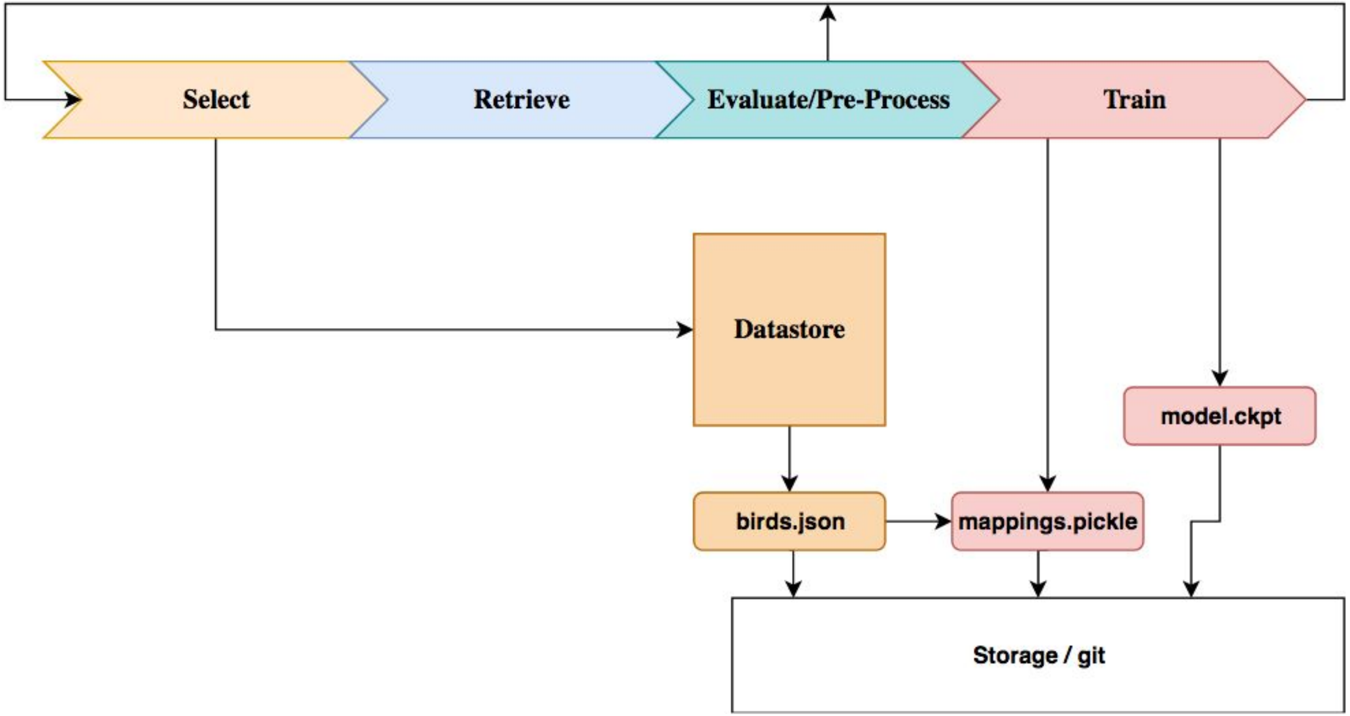
```
docker run --rm -d \
    --name jibjib-model \
    --runtime=nvidia \
    -v $(pwd)/code:/model/code \
    -v $(pwd)/input:/model/input \
    -v $(pwd)/output:/model/output \
    obitech/jibjib-model:latest-gpu \
    python vggish_train.py \
    --num_batches=60 \
    --num_mini_batches=1400 \
    --num_classes=195 \
    --validation=True \
    --test_size=0.1
```

Model Training

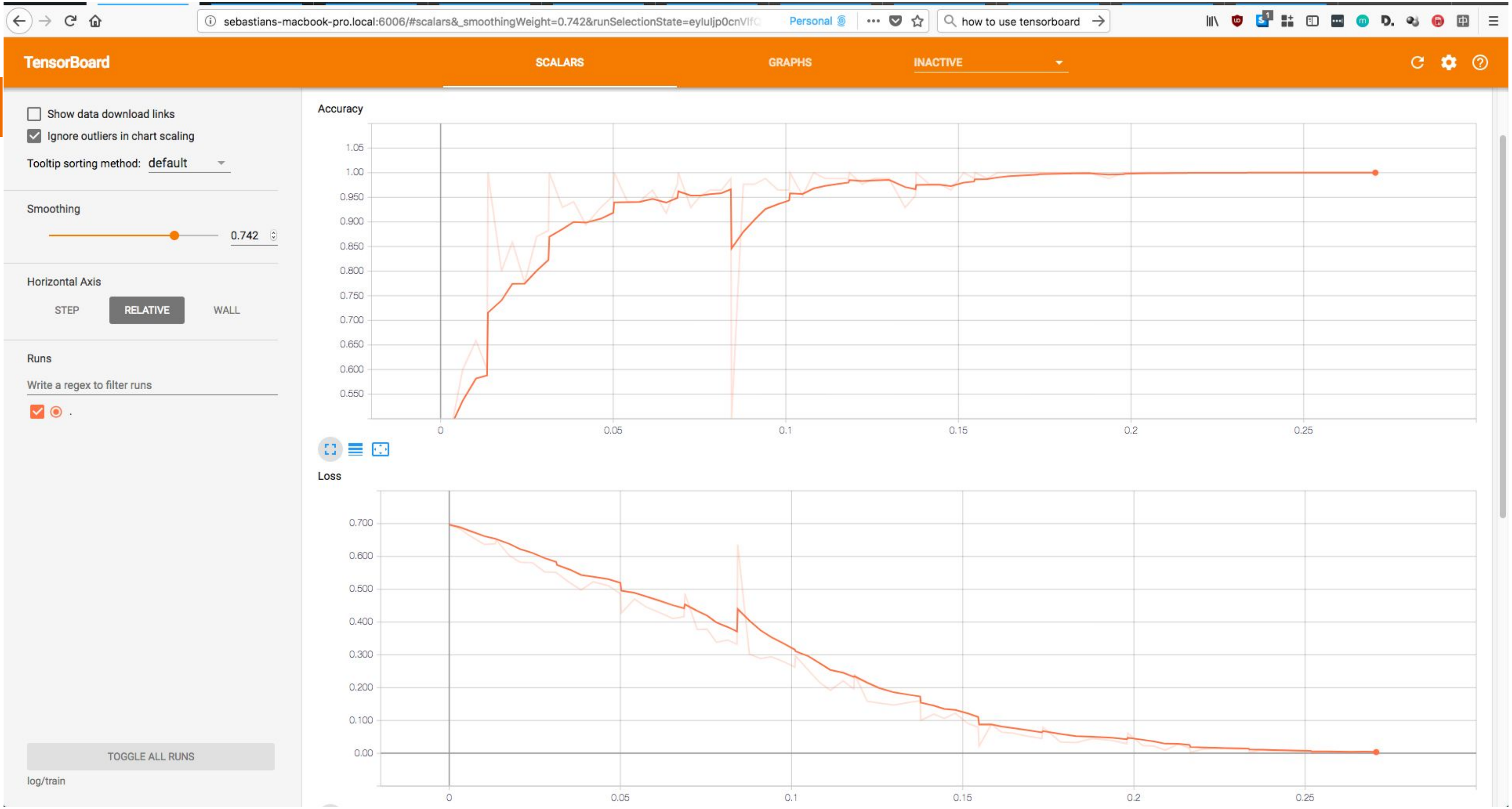
Mappings:

```
20180909-113429:root:INFO:Acrocephalus_arundinaceus -> 0
20180909-113700:root:INFO:Acrocephalus_palustris -> 1
20180909-114301:root:INFO:Acrocephalus_schoenobaenus -> 2
20180909-114557:root:INFO:Acrocephalus_scirpaceus -> 3
20180909-114824:root:INFO:Aegithalos_caudatus -> 4
20180909-114953:root:INFO:Alauda_arvensis -> 5
20180909-115136:root:INFO:Alcedo_atthis -> 6
20180909-115222:root:INFO:Anas_penelope -> 7
20180909-115323:root:INFO:Anas_platyrhynchos -> 8
20180909-115504:root:INFO:Anser_anser -> 9
20180909-115608:root:INFO:Anser_fabalis -> 10
20180909-115646:root:INFO:Anthus_spinoletta -> 11
20180909-115743:root:INFO:Anthus_trivialis -> 12
```

Data Science Workflow



M



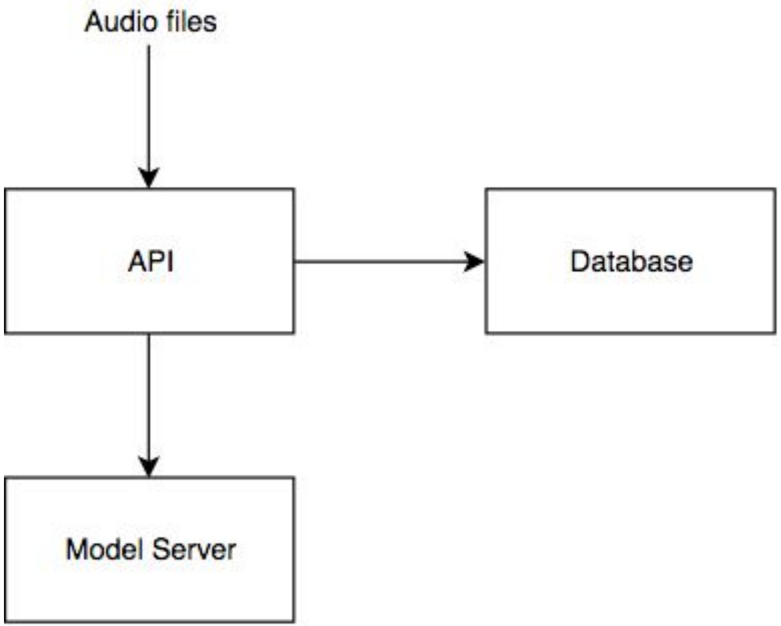
sebastians-macbook-pro.local:6006/#scalars&_smoothingWeight=0.742&runSelectionState=eyIuIjp0cnVlfQ
Personal
how to use tensorboard
TensorBoard
SCALARS
GRAPHS
INACTIVE
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.742
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
TOGGLE ALL RUNS
log/train
Accuracy
1.05
1.00
0.950
0.900
0.850
0.800
0.750
0.700
0.650
0.600
0.550
0
0.05
0.1
0.15
0.2
0.25
Loss
0.700
0.600
0.500
0.400
0.300
0.200
0.100
0.00
0
0.05
0.1
0.15
0.2
0.25

Part 2

The Backend

- ✓ Setting up an API & Database
- ✓ Setting up the Model Server
- ✓ Deploying the Model

Architecture




```json
{
    "status": 200,
    "message": "Detection successful",
    "count": 3,
    "data": [
        {
            "accuracy": 0.7741285540736146,
            "id": 110
        },
        {
            "accuracy": 0.14705901025204263,
            "id": 7
        },
        {
            "accuracy": 0.07881243567434278,
            "id": 184
        }
    ]
}
```

API & Database

API

- Written in Go
- Communicates with DB and Model Server
- Passes through binary file, exception handling
- https://github.com/gojibjib/jibjib-api

Database

- birds.json → MongoDB
- Data baked into image
- https://github.com/gojibjib/jibjib-data

Crash Course: TensorFlow

- "Open-source software library for **dataflow programming**"
- Released by Google: Nov 9, 2015 (Apache 2.0)
- Dataflow: programming model for parallel computing
  - **Nodes:** units of computation
  - **Edges:** data consumed or produced by nodes
- → https://www.tensorflow.org/guide/graphs

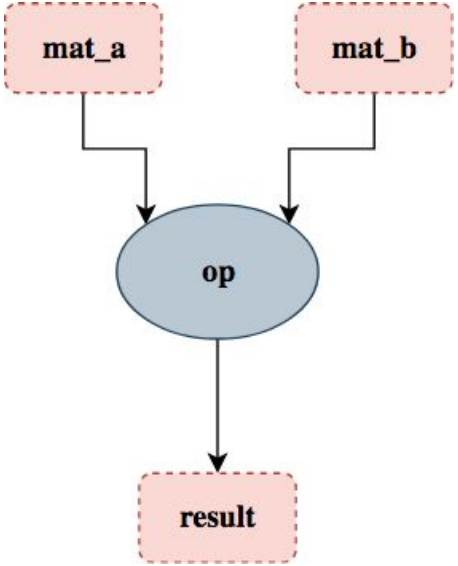
Crash Course: TensorFlow

```python
#!/usr/bin/env python
# matmul.py - Use TensorFlow to multiply two matrices
import tensorflow as tf

# Creates a graph with 2x tf.Tensor, 1x tf.Operation
mat_a = tf.constant([0, 0.123, 51.63, 42], shape=[2000, 1500], name='mat_')
mat_b = tf.constant([1, 2, 3, 23], shape=[1500, 2000], name='mat_b')
op = tf.matmul(mat_a, mat_b)

# A session evaluates Tensors & executes operations
with tf.Session() as sess:
    result = sess.run(op)

print("Result: {}".format(result))
```



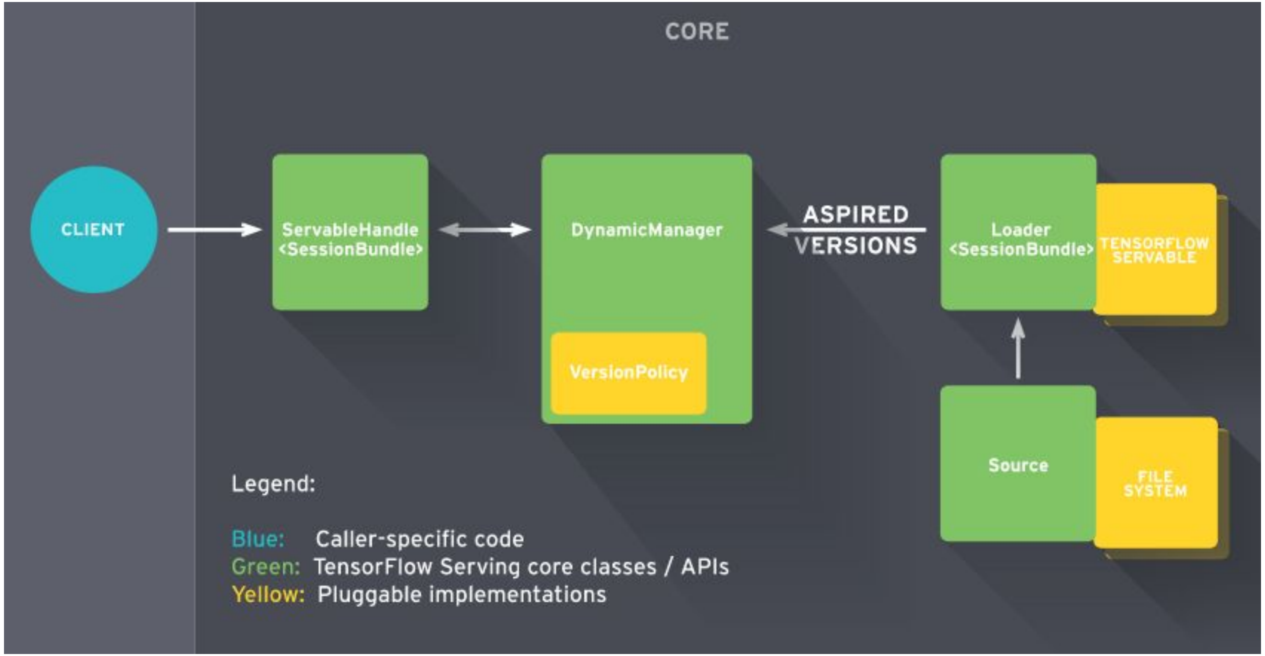
Model Server

1. Load the model (= construct the graph)
2. Accept audio file
3. Convert to .wav
4. Convert .wav to spectrogram
5. Convert spectrogram to tf.Tensor()
6. Run tf.Session()
7. Parse output Tensor
8. Send response

TensorFlow Serving

- Models represented as **servables**
- **Managers** handling the loading, serving, unloading of servables
- Lifecycle management, versioning, A/B testing
- Client-Server architecture to query model
- Can handle 100k QPS / core
- → https://arxiv.org/pdf/1712.06139.pdf

TensorFlow Serving



https://github.com/tensorflow/serving

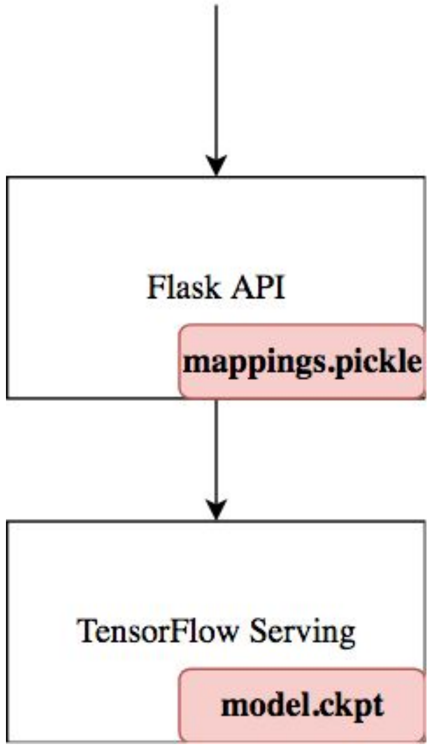
Model Server

Thin Python Flask API

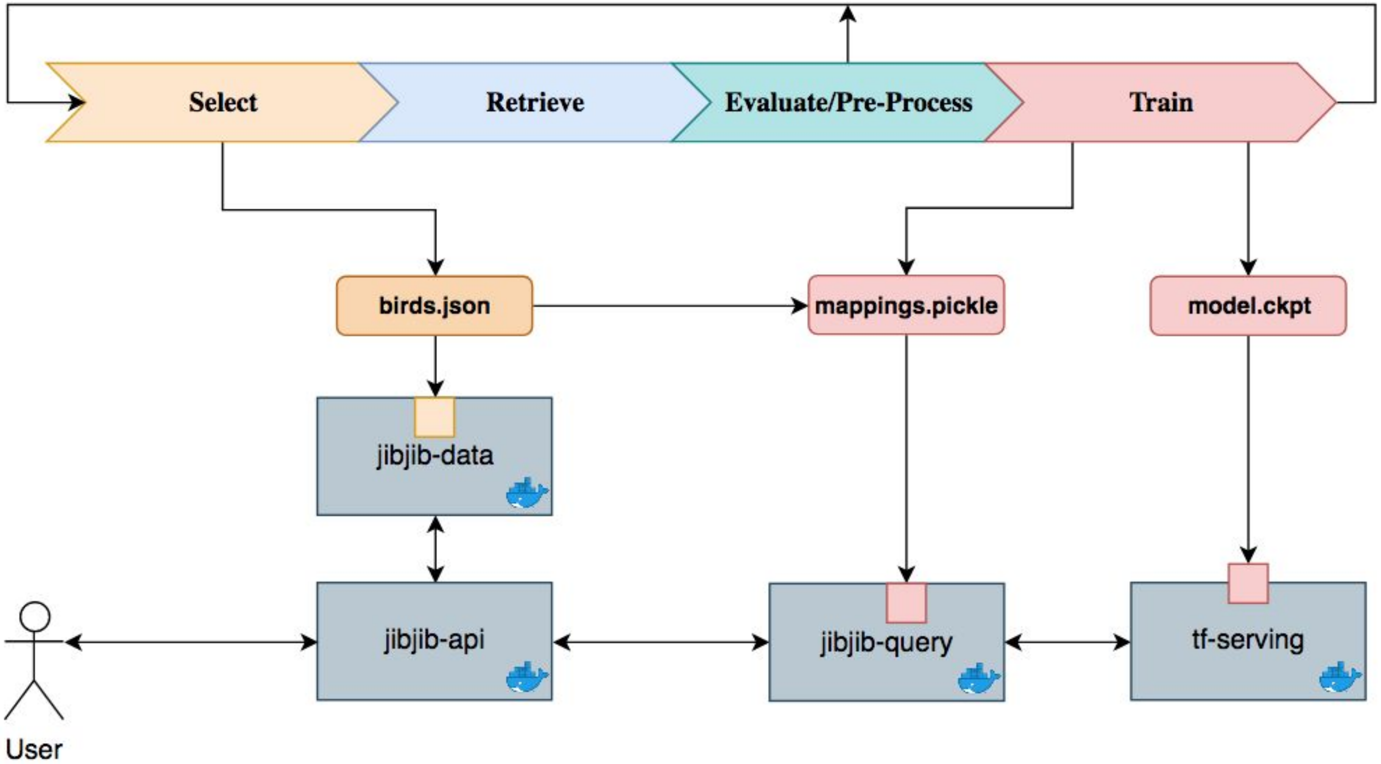
- Converts audio file
- Constructs request Tensor, converts to protobuf
- Parses response Tensor from TF Serving
- https://github.com/gojibjib/jibjib-query

TF Serving

- Hold model in protobuf format
- Handles requests



Overview


Select
Retrieve
Evaluate/Pre-Process
Train
birds.json
mappings.pickle
model.ckpt
jibjib-data
jibjib-api
jibjib-query
tf-serving
User

Part 3

Deployment

https://github.com/gojibjib/deploy

✓ Locally
✓ Remotely

Locally

- Clone the repo
- Run init.sh
- Run start.sh
- Done!

Remotely

- Clone the repo
- Create a keypair
- Set your Terraform variables
- $ terraform apply
- Deploy via SaltStack

Summary

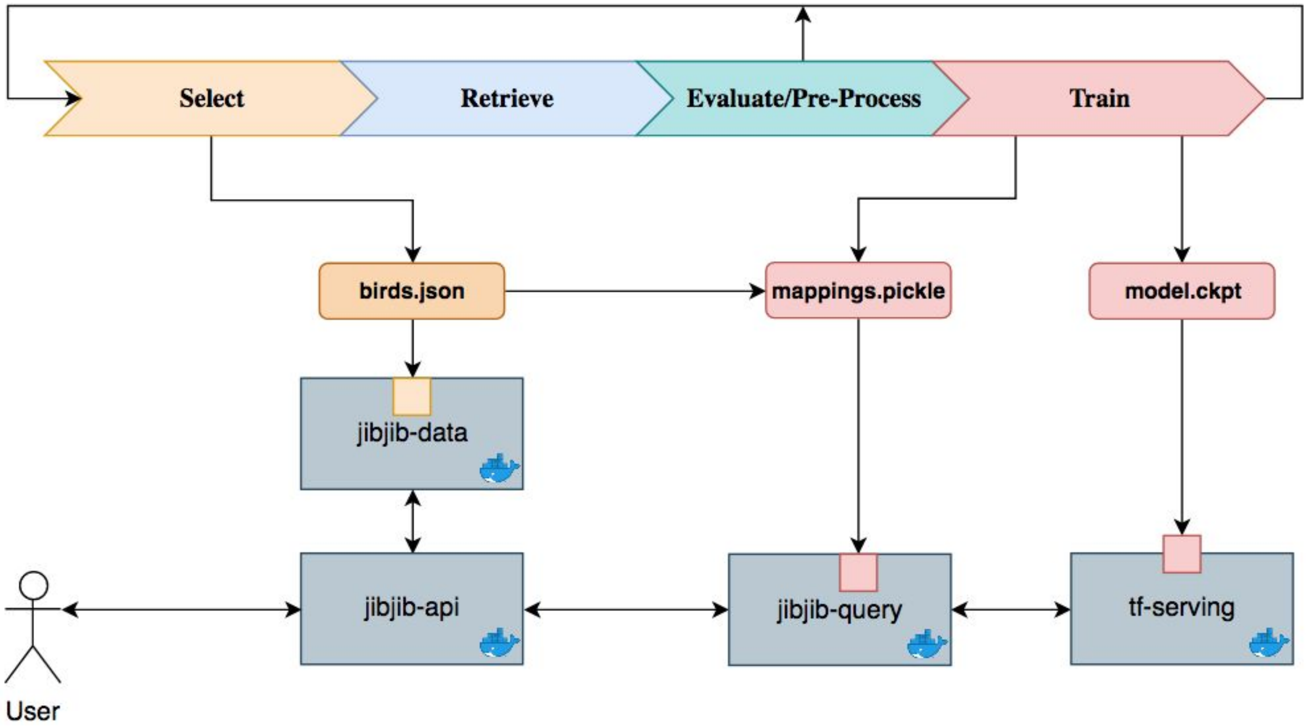
Architecture

Benefits

- Modular
- Flexible
- Scalable

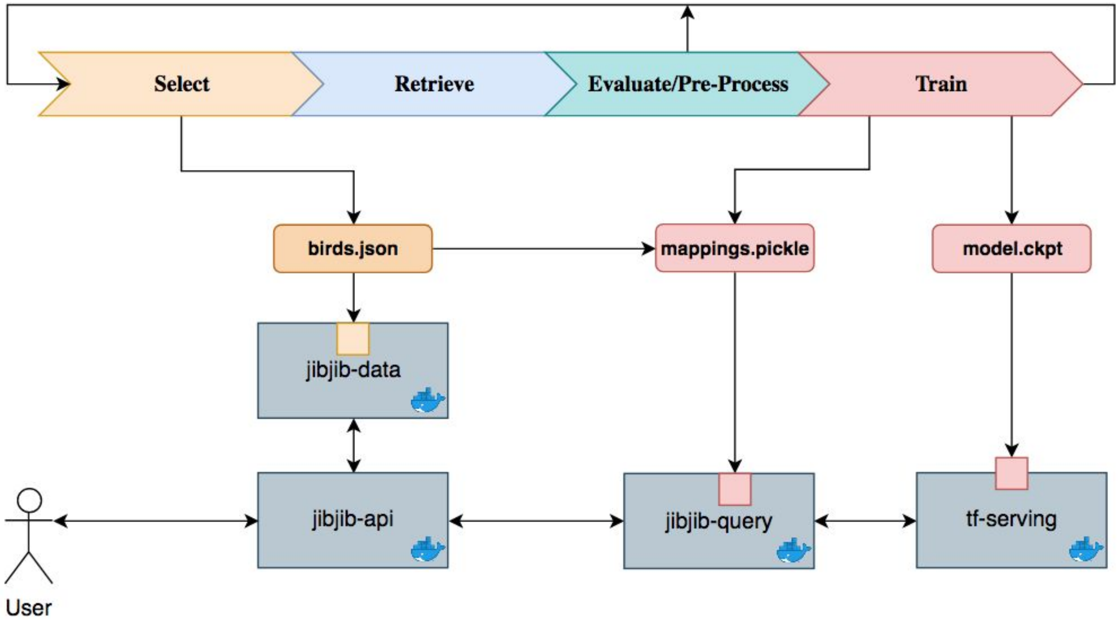
Challenges

- Complexity
- Resources
- Data versioning



Summary

- Data science takes time, trial & error
- Get domain experts
- You're always training
- Difficult to test



Thank you!

About JibJib

https://github.com/gojibjib

Google Play Store: JibJib

gojibjib@gmail.com

About myself

https://github.com/obitech

https://linkedin.com/in/ajknipping

@obit3ch