



Hands on Lab: Getting Started with Open Source Datasets and Deep Learning Models

In this lab, you will explore how to access and utilize open source datasets for machine learning projects. You will also get hands-on experience with pre-trained deep learning models to accelerate your development process.

Objective of Exercise 1:

- Explore the open data sets.

Objective of Exercise 2:

- Find ready-to-use deep learning models on the Model Asset Exchange.
- Explore the deep learning model trained to detect objects in an image.

It will take you approximately 15 minutes to complete the lab. Only a web browser is required to complete the tasks.

Exercise 1: Explore open source datasets

Most datasets are complemented by Python notebooks that you can use to explore, pre-process, and analyze the data. Here we will be exploring a public weather dataset taken from kaggle [LINK](#)

Step-by-Step Instructions

1. Click [HERE](#) to view the dataset.
2. Once the page loads, click the **Dataset Preview** tab at the top of the page.

This will allow you to:

- View a sample of the data directly in your browser.
- Understand the structure and fields of the dataset.
- Explore metadata and glossary terms associated with the dataset.

NOAA Weather Data – JFK Airport	
Dataset Metadata	Dataset Preview
Dataset Glossary	Notebook Preview
NOAA Weather Data – JFK Airport Dataset Metadata	
Format	CSV
License	CDLA-Sharing
Domain	Time Series
Number of Records	114,546 hourly observations
Data Split	NA
Size	3.2 MB

3. Next, click the **Notebook Preview** tab.
 - This opens a sample Python notebook associated with the dataset.
 - The notebook walks through:
 - Cleaning the weather data.
 - Performing exploratory analysis.
 - Building predictive models to help airports better manage flight schedules.

NOAA Weather Data – JFK Airport

[Dataset Metadata](#) [Dataset Preview](#) [Dataset Glossary](#) [Notebook Preview](#)

NOAA Weather Data Analysis - JFK Airport (New York)

This notebook focuses on analyzing and forecasting weather patterns using the **NOAA Weather Dataset** collected from **JFK Airport in New York**. The dataset comprises **114,546 hourly observations of 12 key climatological variables**, including temperature, wind speed, humidity, and pressure. This notebook teaches the user to extract, clean and analyze sample weather data and predict weather trends to help airports schedule better flight times.

The notebook is organized into three main parts:

Part 1: Data Cleaning

In this section, we prepare the raw data for analysis by:

- Removing unnecessary or redundant columns to retain only relevant numerical features
- Converting data types and cleaning inconsistencies
- Handling missing values with appropriate filling strategies
- Encoding categorical weather features for downstream analysis

This concludes Exercise 1 of this lab, which introduced you to the open source dataset. You may proceed to Exercise 2.

4. **[Optional]** If you are comfortable using Jupyter Notebooks, you can download and run the notebook locally by clicking [HERE](#)

Exercise 2 - Explore deep learning models

The Model Asset Exchange is a curated repository of Open Source deep learning models for a variety of domains, such as text, image, audio, and video processing.

For more details, please visit - <https://github.com/CODAIT/max-central-repo> webpage.

The curated list includes deployable models, which you can run as a microservice locally or in the cloud on Docker or Kubernetes, and trainable models where you can use your own data to train the models. Some of the models are already built for you to test. Let's test one of the models.

In this exercise, explore the **Object Detector** model hosted on CodePen platform. This model recognizes the objects present in an image. The model consists of a deep convolutional net base model for image feature extraction, together with additional convolutional layers specialized for the task of object detection, trained on the COCO data set. The input to the model is an image, and the output are extracted objects from the image, appropriately labeled.

CodePen is a social development environment. At its heart, it allows to write code in the browser and see the results of it as you build. It is a useful and liberating online code editor for developers of any skill and is particularly empowering for people learning to code.

1. Navigate to [CodePen](#) webpage.
2. Select **MAX TFJS models** as shown in the screenshot below. Here the **Image Segmenter**, divides an image into regions or categories that correspond to different objects or parts of objects. Every pixel in an image is allocated to one of a number of these categories.

CODEPEN

TRY OUR ONLINE EDITOR

Start Coding

Search Pens
Challenges
Spark

CodePen PRO

MAKE IT WITH ADOBE STOCK.
Get 10 free images >

Limited time offer: Get 10 free Adobe Stock images.
(ads via Carbon)

Search CodePen...

max-playground
CODAIT + Follow

GRID LIST SORT BY Order

MAX TFJS Models Using MAX tfjs models and visualizing the prediction

Model: Image Segmenter URL: https://cdn.jsdelivr.net/npm/@codait/max-image Webcam: Off

Select an image...

MAX Demo Sending a new file to a MAX audio model

Model Endpoint: max-image-classifier-codait-group-4f238675a9b0a23511295a7a8 Filter: Enter comma separated list of labels

See how to POST a Web audio file to the MAX Audio Classifier

The Filter input accepts a list of comma separated labels to filter the response (e.g., Bird, Thunderstorm, Storm). View the [available labels](#).

Note: The filter labels are case sensitive and subject to change depending on the model version.

Select an audio file Get predictions

MAX TFJS models
CODAIT

Send audio clip to MAX
CODAIT

3. Click on **Select Image** and upload an image. You may choose images with a person, dog, cat, truck, car, and so on, which are labels the model has been trained on.

MAX TFJS models
CODAIT + Follow

Run Settings

HTML

```
1 <header class="max-header" role="banner"
  aria-label="CODAIT MAX">
2 <a class="max-skip" href="#max-main-
  content" tabindex="0">Skip to main
  content</a>
3 <span class="max-header-logo">
4 <svg focusable="false"
  preserveAspectRatio="xMidYMid meet"
  style="will-change: transform;"
  xmlns="http://www.w3.org/2000/svg">
```

CSS

```
1 .max-subheader .max-field {
2   display: flex;
3   flex-flow: column;
4   margin-right: 1.5rem;
5 }
6 .max-subheader .max-field.endpoint-field {
7   display: flex;
8   flex-grow: 1;
9   flex-flow: column;
10 }
```

JS

```
16 const tfjsModel = {
17   imageSegmenter:
18     'https://cdn.jsdelivrivr.n
19     image-segmenter'
20 }
21 const selectModelChange
22 if (webcamStream) {
23   pauseVideo()
24 }
```

MAX TFJS Models

Using MAX tfjs models and visualizing the prediction

Model

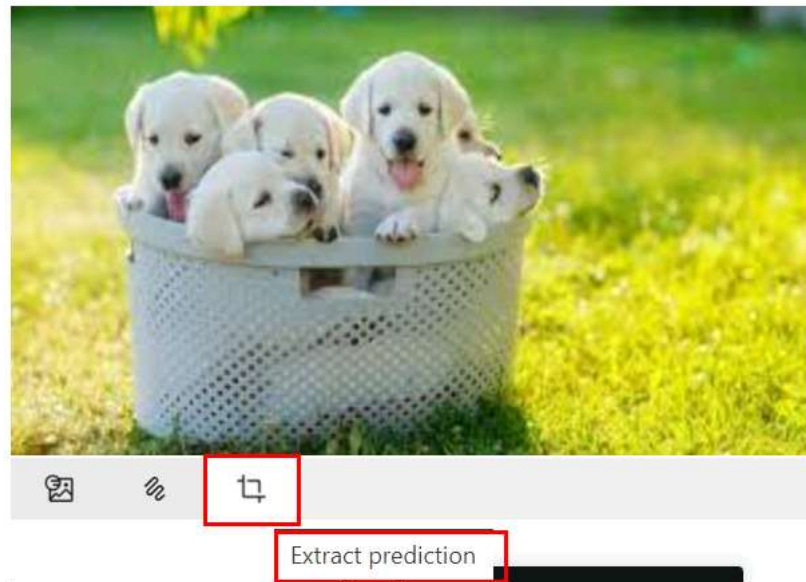
Image Segmenter

URL

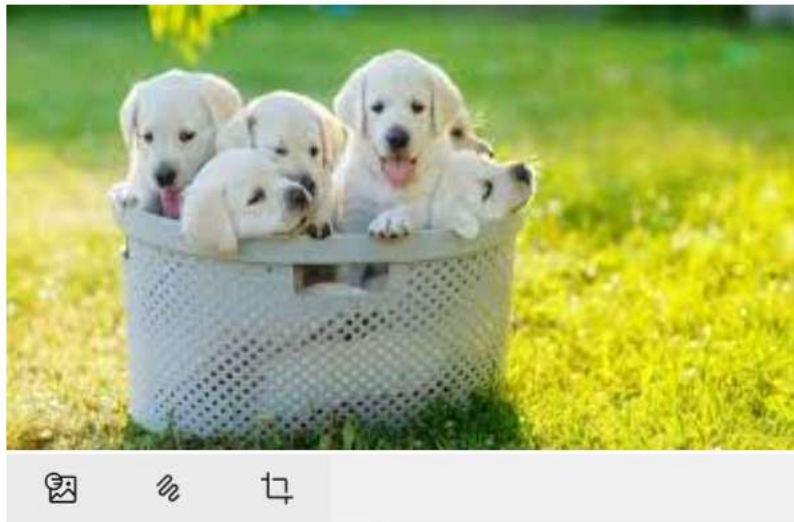
<https://cdn.jsdelivrivr.net/npm/@codait/max-image-segmenter>

Select an image...

- Click here for all the labels the model is trained on
- 4. Click the icon **Extract prediction** as shown below:



You can see the output of the prediction on the basis of the uploaded image.



Here the background and the dog image are separated, showing two different parts of the image.

You can also try the webcam option, which will show the real-time prediction by the toggle-on webcam option.

This concludes Exercise 2 of this lab, which introduced the Model Asset Exchange (MAX).

You can also watch a demo of the object detector model [here](#).

Author(s)

Joseph Santarcangelo

Other Contributor(s)

Lavanya



Skills Network