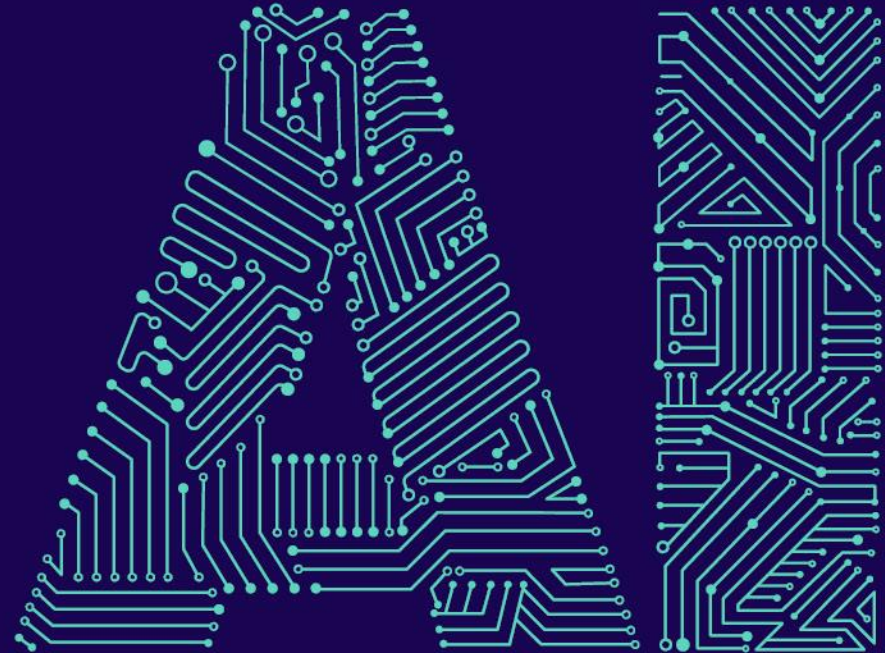


cnvrg.io

OCR + Text Detection Workshop

AI Blueprints



Agenda

- What are you going to get out of this workshop?
- cnvrg.io Overview
- cnvrg.io AI Blueprints
- What is OCR/Text Detection?
- OCR Inference Example
- Using YOLO for Text Detection
- Preparing your Dataset for Yolov5 Model
- Bonus: Train your Dataset with cnvrg.io Blueprints

What are you going to get out of this workshop?

Knowledge

- Introduction to the cnvrg.io platform and AI Blueprints
- Understand the different types of OCR and Text Detection
- Overview of Yolov5 and Labelling



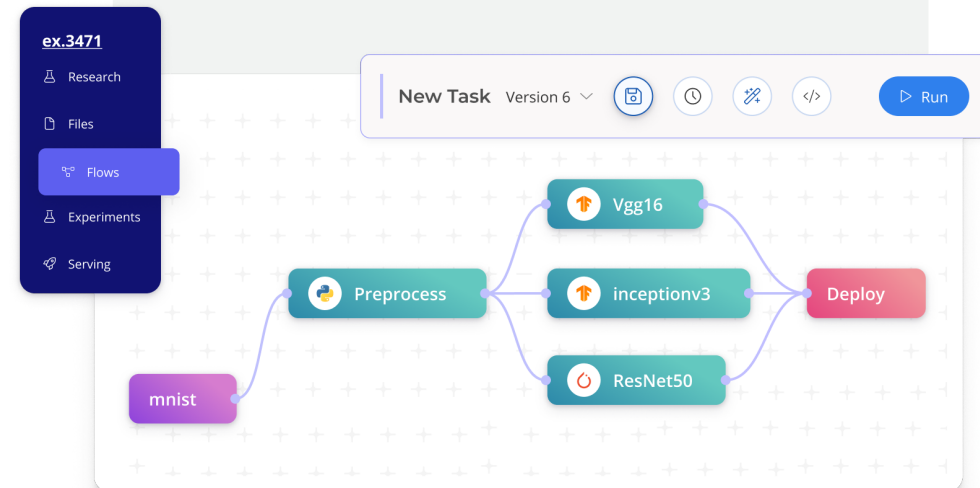
Practical Experience

- Prepare your dataset using a labelling tool
- Use cnvrg.io Blueprints to convert images to machine readable text
- Deploy and monitor your model using cnvrg.io

cnvrg.io Overview

Built by data scientists for developers of AI applications

- A platform to automate the continuous **training and deployment** of AI and ML models.
- Manages the **entire lifecycle**: data preprocessing, experimentation, training, testing, versioning, deployment, monitoring, and automatic retraining.
- Enables developers to train and deploy on **any infrastructure at scale**
- cnvrg.io **Metacloud** is the cnvrg.io platform offered as a **managed service**



Benefits

- Up to 10x increase in productivity
- Up to 5x faster model training
- Up to 50% increase in compute utilization

cnvrg.io

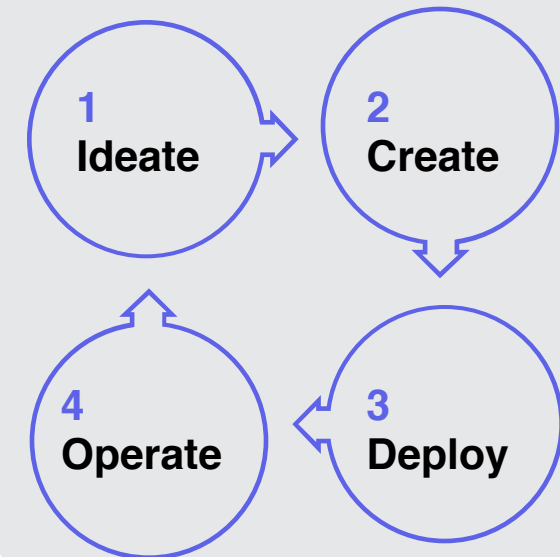
AI Blueprints

cnvrg.io AI Blueprints

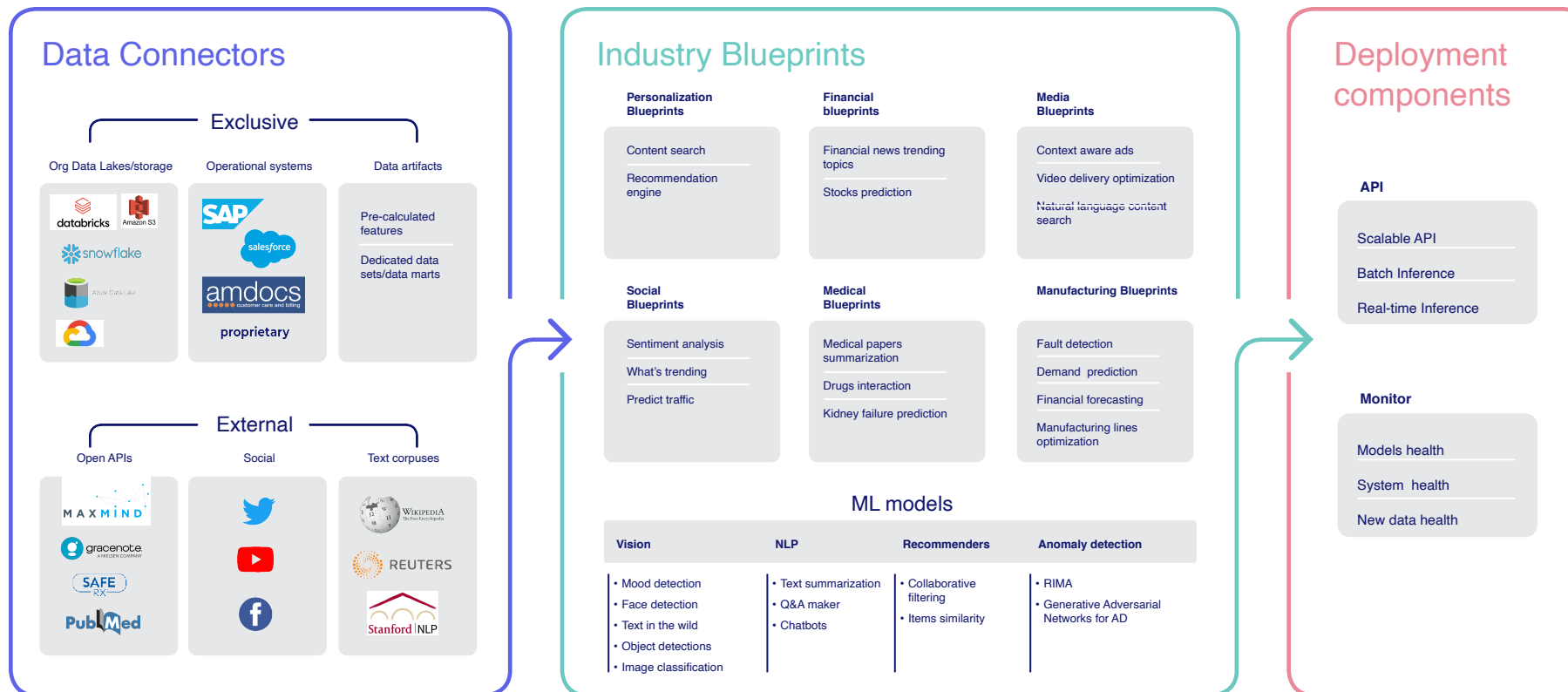
are ready to use, low-code
and open-source ML pipelines.

Built by data scientists, for
everyone.

AI Blueprints



Rich Marketplace of Blueprints and Components



AI Blueprints

From ideation to production in a few clicks



Inference blueprints

Pull updated data from a variety of data sources and run it through a pre-trained model for immediate predictions

data

>

processing

>

inference



Training blueprints

Fine tune a model or train it on your own exclusive data and deploy it on your selected infrastructure

data

>

train

>

deploy



Blueprints components

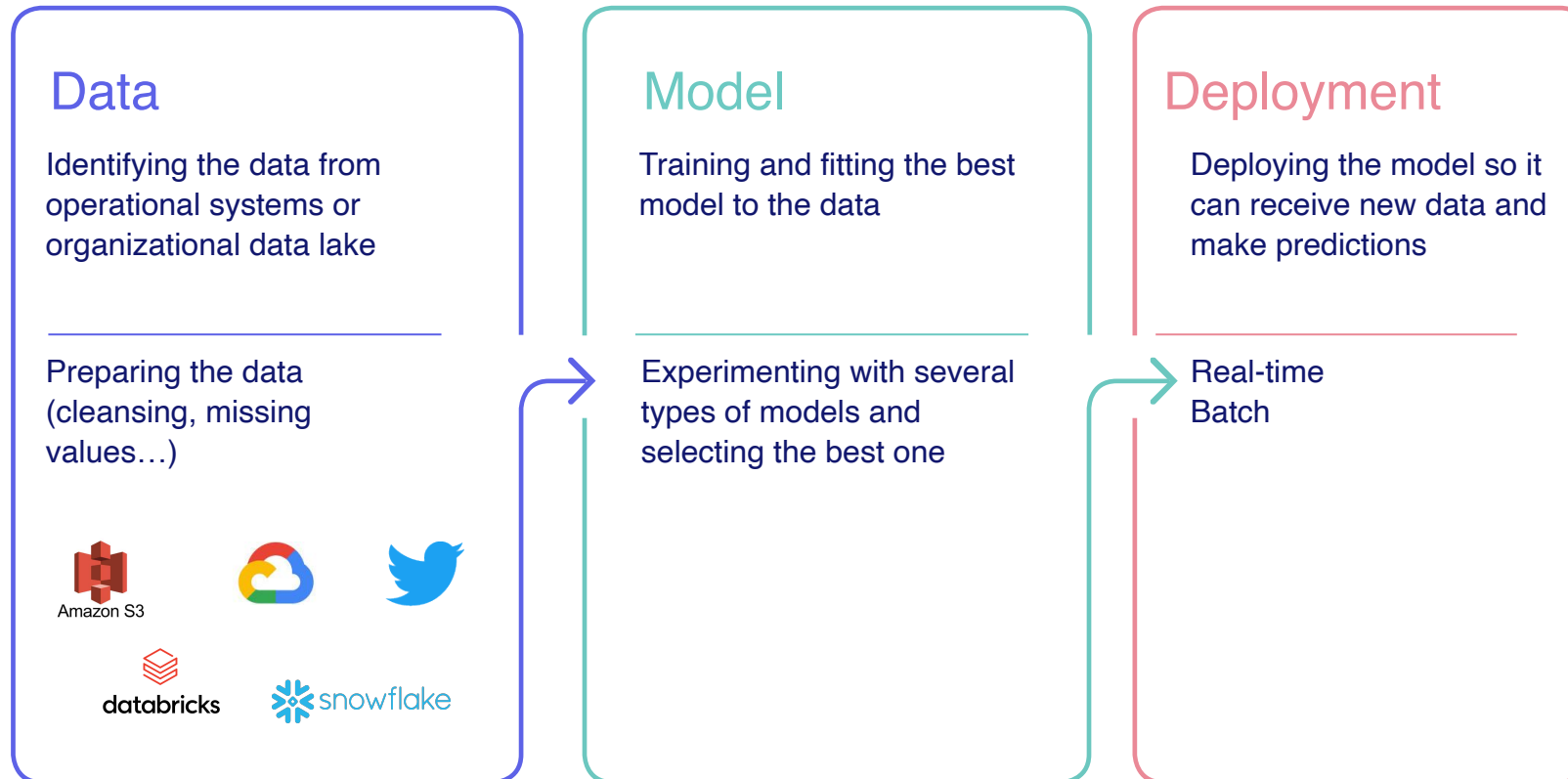
Get access to a rich library of ready components and create your own story !

Scene detection

Text in the wild

Q&A maker

The Basic Process



So what is
OCR?



OCR - Optical Character Recognition

- Used to read text from images to extract information
- Examples: Street Signs, License Plate Numbers, Drug Labels, Receipts
- Types of OCR:
 1. Text Detection
 2. Text Recognition



Text Detection

- **Region-Based detectors -**

- Finds the regions which have the objects (bounding box) and the class
- 2 step process
- Examples: Faster R-CNN and R-FCN

- **Single Shot detectors -**


- Predict the bounding box and the class at the same time
- Faster processing time
- Examples: SSD and YOLO



Text Detection: Inference

Try it live

Upload files

 DriverLicense.png (271.35 KB) x

Execute →

Result

[JSON](#)

```
{
  "1": [
    {
      "bounding_box": [
        235,
        89,
        258,
        72
      ],
      "confidence": 0.9988463454314217,
      "text": "HAWAII"
    },
    {
      "bounding_box": [
        276,
        74,
        88,
        30
      ]
    }
  ]
}
```

Text Detection: Inference

Input



Output

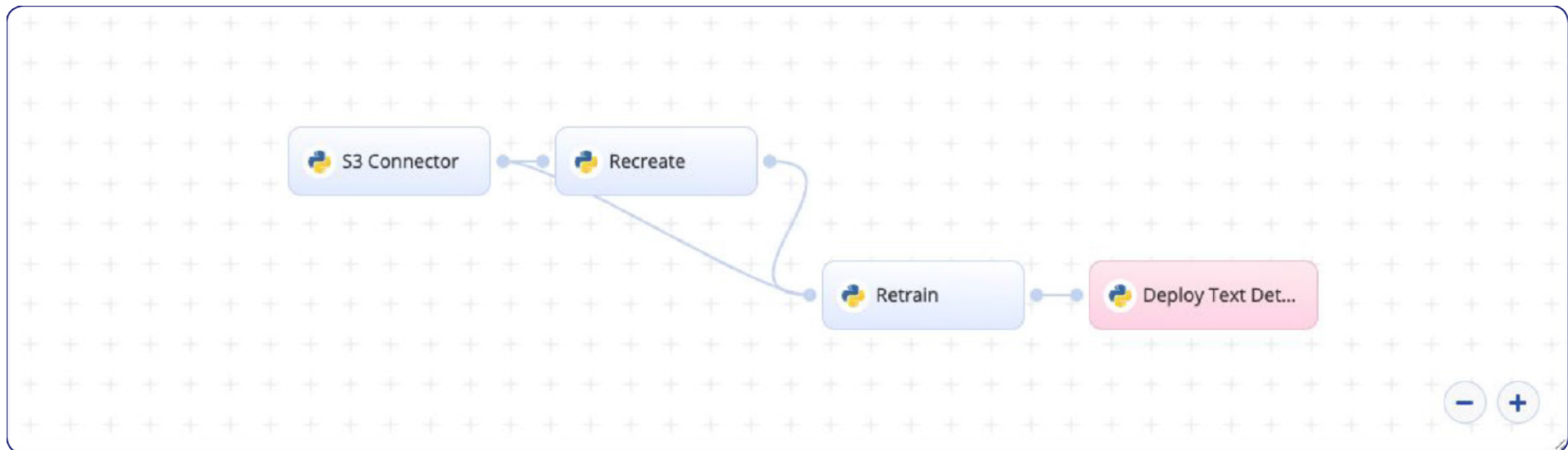
center_x
center_y
Width
Height



```
{  
  "prediction":{  
    "1":[  
      {  
        "bounding_box":[  
          166,  
          222,  
          83,  
          36  
        ],  
        "confidence":0.9999963045120239,  
        "text":"2470"  
      }  
    ]  
  }  
}
```

Text Detection: Training

Blueprint Deep dive



YOLO

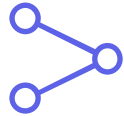
“You Only Look Once”



Nano

YOLOv5n

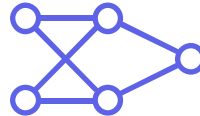
4 MB_{FP16}
6.3 ms_{V100}
28.4 mAP_{COCO}



Small

YOLOv5s

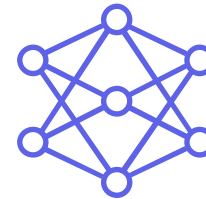
14 MB_{FP16}
6.4 ms_{V100}
37.2 mAP_{COCO}



Medium

YOLOv5m

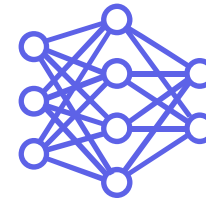
41 MB_{FP16}
8.2 ms_{V100}
45.2 mAP_{COCO}



Large

YOLOv5l

89 MB_{FP16}
10.1 ms_{V100}
48.8 mAP_{COCO}

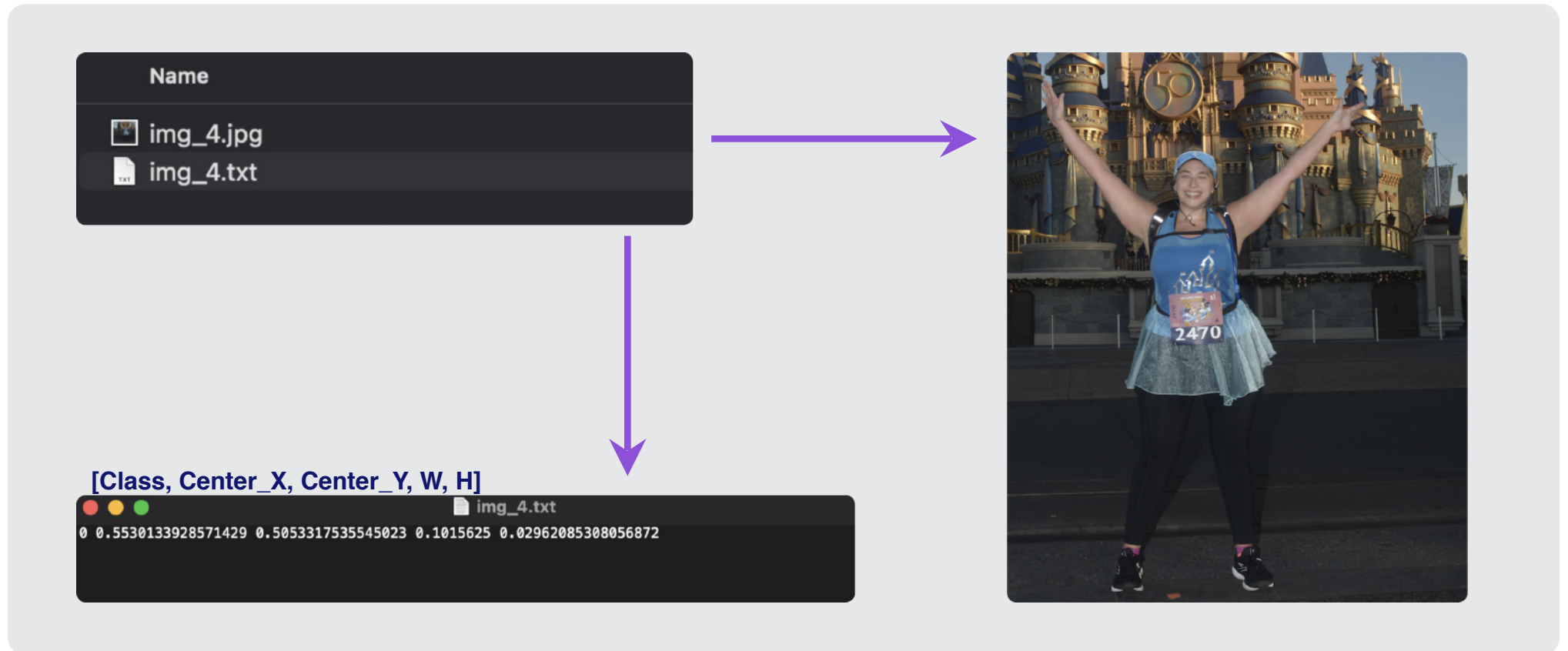


XLarge

YOLOv5x

166 MB_{FP16}
12.1 ms_{V100}
50.7 mAP_{COCO}

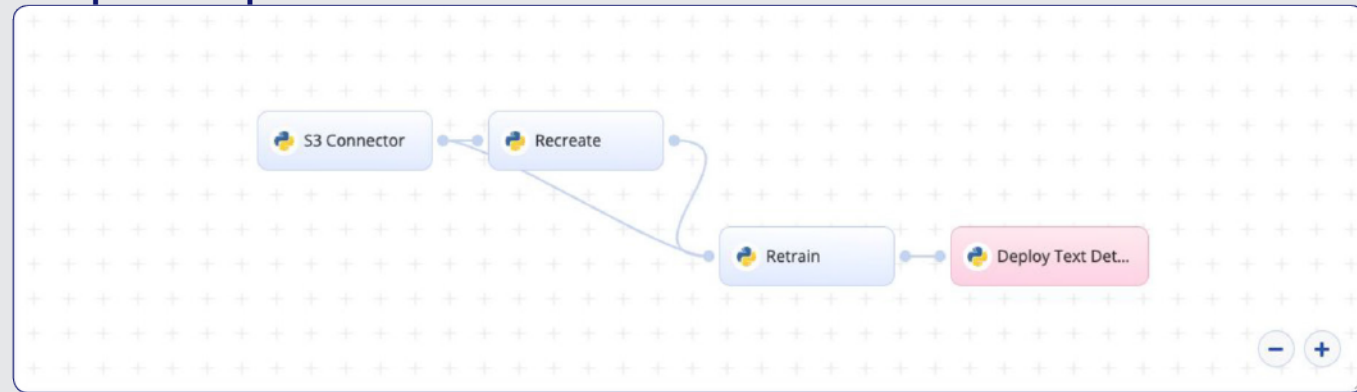
Data Format for YOLOv5 (Input)



Outputs

```
| - images
|   | - train
|   |   | - img13.jpg
|   |   | - img24.jpg
|   |   | ..
|   | - val
|   |   | - img73.jpg
|   |   | - img30.jpg
|   |   | ..
|   | - test
|   |   | - img2.jpg
|   |   | - img50.jpg
|   |   | ..
| - labels
|   | - train
|   |   | - img13.txt
|   |   | - img24.txt
|   |   | ..
|   | - val
|   |   | - img73.txt
|   |   | - img30.txt
|   |   | ..
|   | - test
|   |   | - img2.txt
|   |   | - img50.txt
|   |   | ..
```

Blueprint Deep dive



**Before we start building, any questions on OCR
and Text Detection?**

So What Are
We Building?



Let's build!

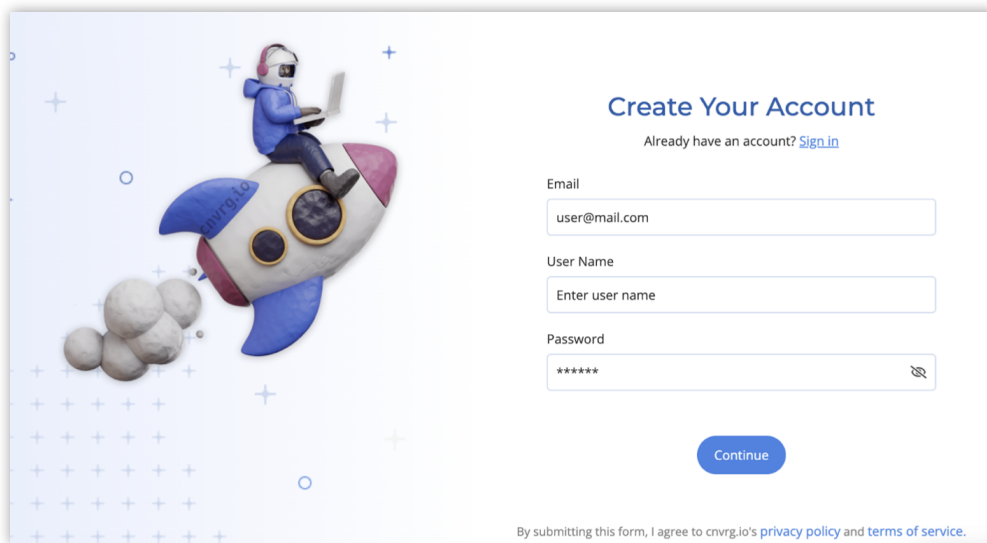
Summary

What we have seen:

- Imported and used our own data
- Created and trained a text detection system
- Deployed the system for inference on Cnvrg
- Used the created Inference to give predictions

Next

You are welcome to sign-up to Metacloud and recreate what you tried here as well as other blueprints



The sign-up form is titled "Create Your Account" and includes a link for users who already have an account. It features four input fields: Email (pre-filled with "user@mail.com"), User Name (placeholder "Enter user name"), Password (masked with "*****"), and a "Continue" button. A small illustration of a person in a blue hoodie sitting on a rocket is on the left. At the bottom, there is a disclaimer: "By submitting this form, I agree to cnvrg.io's [privacy policy](#) and [terms of service](#)."

Create Your Account

Already have an account? [Sign in](#)

Email

user@mail.com

User Name

Enter user name

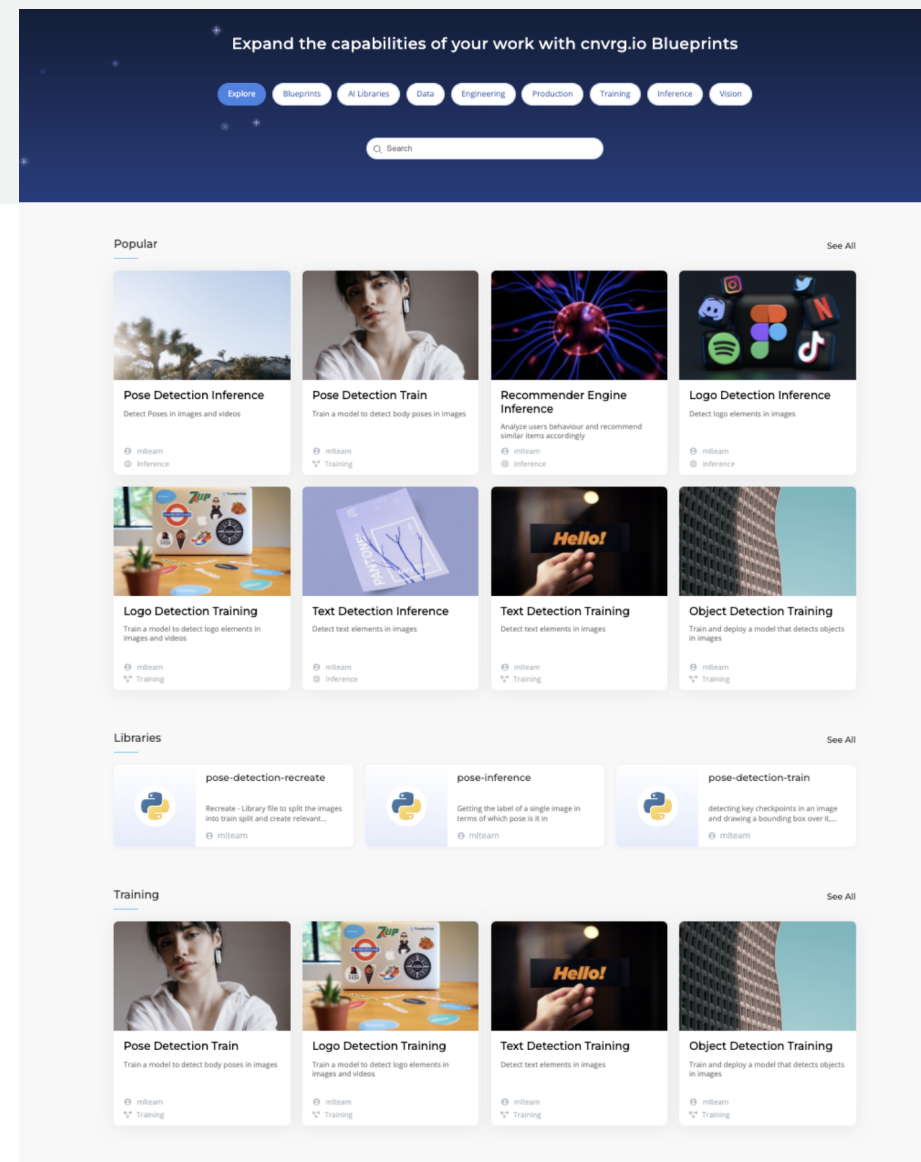
Password

Continue

By submitting this form, I agree to cnvrg.io's [privacy policy](#) and [terms of service](#).

<https://metacloud.cloud.cnvrg.io/sign-up>

cnvrg.io



The dashboard has a dark blue header with the text "Expand the capabilities of your work with cnvrg.io Blueprints". Below this is a navigation bar with buttons for Explore, Blueprints, AI Libraries, Data, Engineering, Production, Training, Inference, and Tools. A search bar is also present. The main content area is divided into two sections: "Popular" and "Libraries". The "Popular" section displays eight blueprint cards, each with a thumbnail image, a title, a description, and a category icon (e.g., Inference, Training). The "Libraries" section displays three blueprint cards, each with a Python logo icon, a title, a description, and a category icon. The "Training" section displays four blueprint cards, each with a thumbnail image, a title, a description, and a category icon.

Expand the capabilities of your work with cnvrg.io Blueprints

Explore Blueprints AI Libraries Data Engineering Production Training Inference Tools

Search

Popular

See All

Pose Detection Inference

Detect Poses in Images and Videos

mi3team Inference

Pose Detection Train

Train a model to detect body poses in images

mi3team Training

Recommender Engine Inference

Analyze users behaviour and recommend similar items accordingly

mi3team Inference

Logo Detection Inference

Detect logo elements in images

mi3team Inference

Logo Detection Training

Train a model to detect logo elements in images and videos

mi3team Training

Text Detection Inference

Detect text elements in images

mi3team Inference

Text Detection Training

Detect text elements in images

mi3team Training

Object Detection Training

Train and deploy a model that detects objects in images

mi3team Training

Libraries

See All

pose-detection-recreate

Recreate - Library file to split the images into train split and create relevant...

mi3team

pose-inference

Getting the label of a single image in terms of which pose it is in

mi3team

pose-detection-train

Detecting key checkpoints in an image and drawing a bounding box over it...

mi3team

Training

See All

Pose Detection Train

Train a model to detect body poses in images

mi3team Training

Logo Detection Training

Train a model to detect logo elements in images and videos

mi3team Training

Text Detection Training

Detect text elements in images

mi3team Training

Object Detection Training

Train and deploy a model that detects objects in images

mi3team Training

Thank you
cnvrg.io