Jsonl Format

To finetune paliGemma we need dataset in Jsonl format

Jsonl is similar to Json, but each line is an independent Json Object

```
[ {....}   Each Line is a seperate Json
  {....}
  {....}
```

↓

```
{ image : path to image
  prefix : prompt
  suffix : Response
}
```

Each json contain 3 keys, image, prefix, suffix.

image: holds the path to associated image
prefix : prompt we are going to send to model.
suffix : Expected model responce.

» When used for Image Captioning
prefix could be describing the image &
Suffix could be a man with a dog

```
{ image : path to man holding dog
  prefix : describing the image
  suffix : A man holding a dog }
```

>> when used for Visual Question Answering (VQA)

Prefix could be what breed of dog is it

suffix could be answer like a beagle

{

image: path to image

prefix: 'what breed of dog is it'

suffix: 'Beagle'

}

>> Object Detection:

Similar to other task, there is prefix & suffix.

For prefix we can have detect a dog, where key word detect is critical.

Suffix has specific format, 4 consequtive loc tags, followed by desired class name.

If there are multiple objects, another section appears, seperated by semicolan

{

image: path to image

prefix: detect a dog

suffix: <loc 8> <loc 9> <loc 10> <loc 12>

dog

}

If there are multiple object detected, they are displayed after semicolan

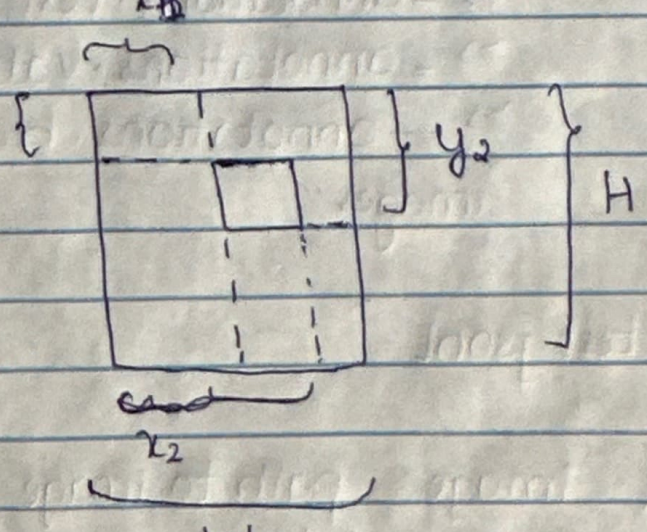
{ image : path to image
prejix : detect a dog
suffix: <loc--><loc--><loc--><loc--> dog;
<loc--><loc--><loc--><loc--> dog
}

we can detect multiple object by listing them in suffix, seperated by semicolan.

{ image: path to image
prejix: detect a dog; cat; person
suffix: <loc--><loc--><loc--><loc--> dog;
<loc--><loc--><loc--><loc--> cat
}

>> The 4 loc tag, define the location of bounding box

Imagin a image, with detection defined by coordinates x_1 x_2 y_1 y_2, the image has height h & width w



The first thing is to set coordinat in right Order
$y_1 x_1 y_2 x_2$

bbox = [$y_1 x_1 y_2 x_2$]

Normalize the image

bbox = bbox / [H, W, H, W]
bbox = bbox * 1024

Eg: we place the location 0 padded to 4 digit
bbox = [300, 400, 500, 600]
<loc 0300> <loc 0400> <loc 0500> <loc 0600>

Finally we append class name after loc tag.

<loc0300> <loc 0400> <loc 0500> <loc 0600> dog

>> Dataset From Roboflow has following format:

dataset
>> _annotations.test.jsonL
>> _annotations.valid.jsonL
>> _annotations.train.jsonL
images

Eg: test.jsonL

{ image: path to image
prefix: "detect 0; 1; 2; 3 ....
suffix: <loc 0048> <loc 0083> <loc 0683>
<loc 0944> 9"
}