

## ✓ Detección de Objetos Custom con YOLO NAS

### ✓ 1. Requerimientos (es necesario utilizar el acelerador de hardware GPU)

```
1
2 %%capture
3 !pip install -q super-gradients==3.1.1
4 !pip install -q roboflow
5 !pip install -q supervision
```

⚠ Una vez instalados los requerimientos, al trabajar en Colab, hay que hacer reset del runtime (este bug 🐛 debería desaparecer próximamente)

# Tiene formato de código

### ✓ 2. Dataset

Se utilizará un dataset de Roboflow, porque el formato es compatible.

```
1 from roboflow import Roboflow
2 rf = Roboflow(api_key="EHbEWh0mXtzTndb0yzQ5")
3 project = rf.workspace("shaafz").project("isl-yphyg")
4 dataset = project.version(2).download("yolov5")
5
```

```
↔ loading Roboflow workspace...
   loading Roboflow project...
   Downloading Dataset Version Zip in ISL-2 to yolov5pytorch: 53% [36470784 / 67]
```

### ✓ 3. Definición de clase `class Config`

```
1 from typing import List, Dict
```

```

2 class config:
3     # Project paths
4     DATA_DIR: str = "ISL-2"
5     CHECKPOINT_DIR: str = "/checkpoints"
6     EXPERIMENT_NAME: str = "ISL"
7
8     # Datasets
9     TRAIN_IMAGES_DIR: str = "train/images"
10    TRAIN_LABELS_DIR: str = "train/labels"
11    VAL_IMAGES_DIR: str = "valid/images"
12    VAL_LABELS_DIR: str = "valid/labels"
13    TEST_IMAGES_DIR: str = "test/images"
14    TEST_LABELS_DIR: str = "test/labels"
15
16    # Classes
17    CLASSES: List[str] = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C',
18    NUM_CLASSES: int = len(CLASSES)
19
20    # Model
21    DATALOADER_PARAMS: Dict = {
22        'batch_size':16,
23        'num_workers':1
24    }
25    MODEL_NAME: str = 'yolo_nas_l'
26    PRETRAINED_WEIGHTS: str = 'coco'

```

## ✓ 4. Inicialización de Dataloaders

```

1 from super_gradients.training import models
2 from super_gradients.training import Trainer
3 from super_gradients.training.dataloaders.dataloaders import coco_detection_yo
4 from super_gradients.training.dataloaders.dataloaders import coco_detection_yo
5 from super_gradients.training.losses import PPYOloELoss
6 from super_gradients.training.metrics import DetectionMetrics_050
7 from super_gradients.training.models.detection_models.pp_yolo_e import PPYOloE
8

```

```

[2023-06-29 20:26:32] INFO - crash_tips_setup.py - Crash tips is enabled. You
The console stream is logged into /root/sg_logs/console.log
Setuptools is replacing distutils.
[2023-06-29 20:26:37] WARNING - __init__.py - Failed to import pytorch_quanti:
[2023-06-29 20:26:37] WARNING - calibrator.py - Failed to import pytorch_quan
[2023-06-29 20:26:37] WARNING - export.py - Failed to import pytorch_quantiza
[2023-06-29 20:26:37] WARNING - selective_quantization_utils.py - Failed to in

```

```

1 train_data = coco_detection_yolo_format_train(
2     dataset_params={
3         'data_dir': config.DATA_DIR

```

```

3         'data_dir': config.DATA_DIR,
4         'images_dir': config.TRAIN_IMAGES_DIR,
5         'labels_dir': config.TRAIN_LABELS_DIR,
6         'classes': config.CLASSES
7     },
8     dataloader_params=config.DATALOADER_PARAMS
9 )
10
11 test_data = coco_detection_yolo_format_val(
12     dataset_params={
13         'data_dir': config.DATA_DIR,
14         'images_dir': config.TEST_IMAGES_DIR,
15         'labels_dir': config.TEST_LABELS_DIR,
16         'classes': config.CLASSES
17     },
18     dataloader_params=config.DATALOADER_PARAMS
19 )
20
21 val_data = coco_detection_yolo_format_val(
22     dataset_params={
23         'data_dir': config.DATA_DIR,
24         'images_dir': config.VAL_IMAGES_DIR,
25         'labels_dir': config.VAL_LABELS_DIR,
26         'classes': config.CLASSES
27     },
28     dataloader_params=config.DATALOADER_PARAMS
29 )

```

```

Caching annotations: 100%|██████████| 2244/2244 [00:00<00:00, 5332.17it/s]
Caching annotations: 100%|██████████| 111/111 [00:00<00:00, 5472.82it/s]
Caching annotations: 100%|██████████| 213/213 [00:00<00:00, 5346.32it/s]

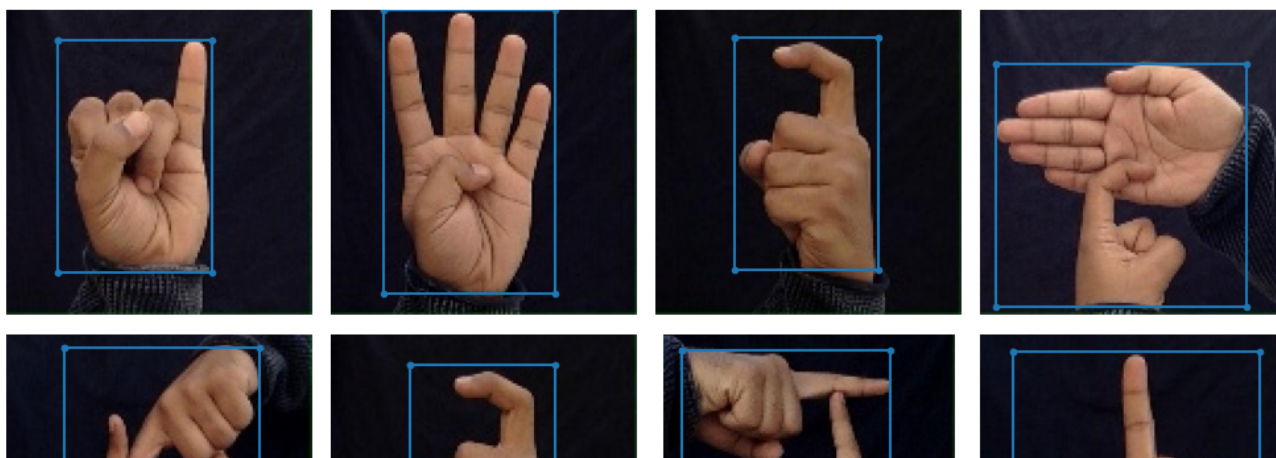
```

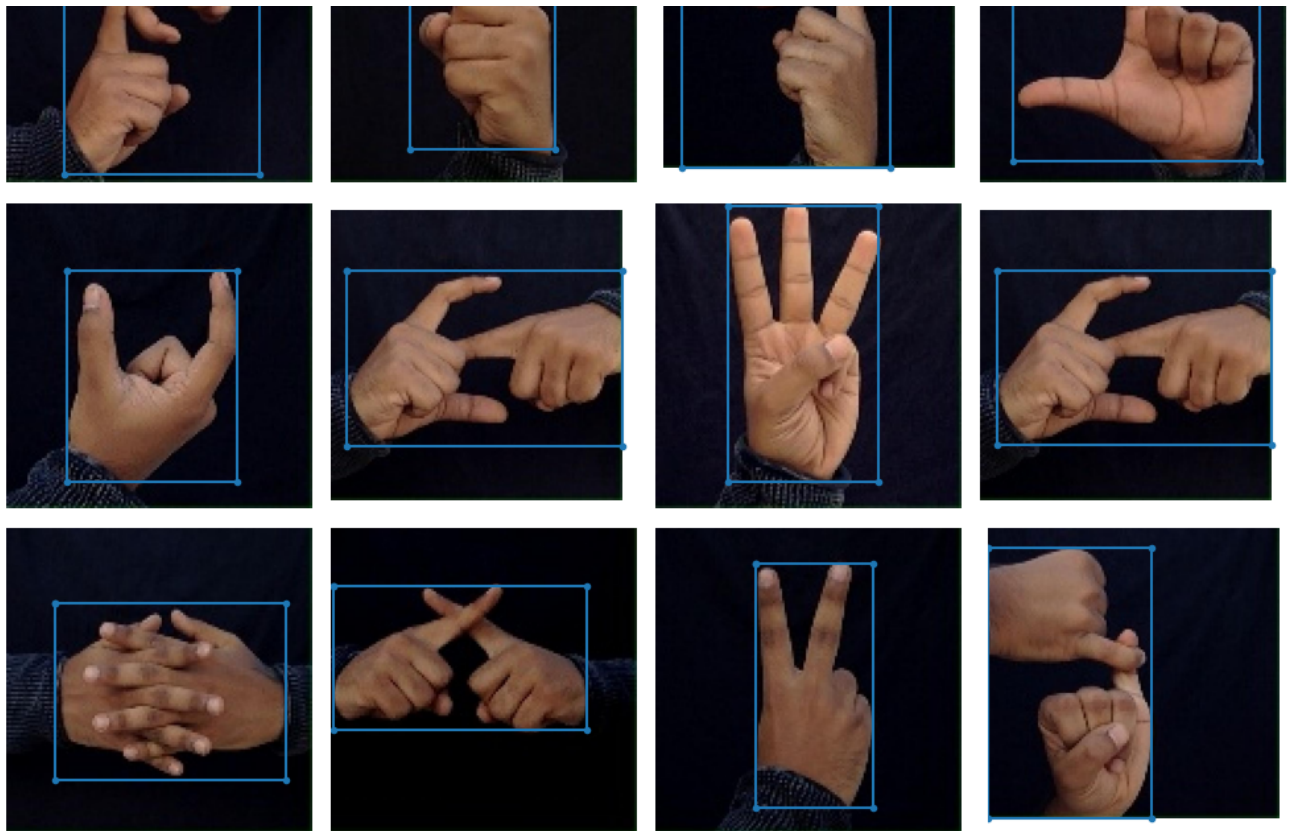
## ✓ 4.1 Visualización

```

1
2 val_data.dataset.plot()

```





## 5. Hiperparámetros de entrenamiento

```
1
2 train_params = {
3     "average_best_models": True,
4     "warmup_mode": "linear_epoch_step",
5     "warmup_initial_lr": 1e-6,
```

```

6     "lr_warmup_epochs": 1,
7     "initial_lr": 5e-4,
8     "lr_mode": "cosine",
9     "cosine_final_lr_ratio": 0.1,
10    "optimizer": "Adam",
11    "optimizer_params": {"weight_decay": 0.001},
12    "zero_weight_decay_on_bias_and_bn": True,
13    "ema": True,
14    "ema_params": {"decay": 0.9, "decay_type": "threshold"},
15    "max_epochs": 5,
16    "mixed_precision": True,
17    "loss": PPYOloELoss(
18        use_static_assigner=False,
19        num_classes=config.NUM_CLASSES,
20        reg_max=16
21    ),
22    "valid_metrics_list": [
23        DetectionMetrics_050(
24            score_thres=0.1,
25            top_k_predictions=300,
26            num_cls=config.NUM_CLASSES,
27            normalize_targets=True,
28            post_prediction_callback=PPYOloEPostPredictionCallback(
29                score_threshold=0.01,
30                nms_top_k=1000,
31                max_predictions=300,
32                nms_threshold=0.8
33            )
34        )
35    ],
36    "metric_to_watch": 'mAP@0.50'
37 }
38

```

## ✓ 6. Entrenamiento

### ✓ 6.1 Descarga el modelo

```
1 model = models.get(config.MODEL_NAME, num_classes=config.NUM_CLASSES, pretrain
```

[2023-06-29 20:27:13] INFO - checkpoint\_utils.py - License Notification: YOLO-  
<https://github.com/Deci-AI/super-gradients/blob/master/LICENSE.YOLONAS.md>

By downloading the pre-trained weight files you agree to comply with these te  
 Downloading: "[https://sghub.deci.ai/models/yolo\\_nas\\_l\\_coco.pth](https://sghub.deci.ai/models/yolo_nas_l_coco.pth)" to /root/.cach

100%

256M/256M [00:04<00:00, 67.3MB/s]

## ▼ 6.2 Inicializa el trainier

```
1 trainer = Trainer(experiment_name=config.EXPERIMENT_NAME, ckpt_root_dir=config
```

## ▼ 6.3 Entrenar! 🚀🚀🚀🚀🚀

```
1 trainer.train(model=model, training_params=train_params, train_loader=train_da
```

```
[2023-06-29 20:27:37] INFO - sg_trainer.py - Using EMA with params {'decay': (
[2023-06-29 20:27:42] INFO - utils.py - NumExpr defaulting to 2 threads.
The console stream is now moved to /checkpoints/ISL/console_Jun29_20_27_42.tx
[2023-06-29 20:27:48] INFO - sg_trainer_utils.py - TRAINING PARAMETERS:
  - Mode:                               Single GPU
  - Number of GPUs:                     1           (1 available on the machine)
  - Dataset size:                       2244         (len(train_set))
  - Batch size per GPU:                 16           (batch_size)
  - Batch Accumulate:                   1           (batch_accumulate)
  - Total batch size:                   16           (num_gpus * batch_size)
  - Effective Batch size:               16           (num_gpus * batch_size * batch
  - Iterations per epoch:               140          (len(train_loader))
  - Gradient updates per epoch:         140          (len(train_loader) / batch_acci
```

```
[2023-06-29 20:27:48] INFO - sg_trainer.py - Started training for 5 epochs (0,
```

```
Train epoch 0: 100%|██████████| 140/140 [03:23<00:00, 1.45s/it, PPYoloELoss/
Validation epoch 0: 100%|██████████| 13/13 [00:08<00:00, 1.48it/s]
```

### =====

#### SUMMARY OF EPOCH 0

```
├── Training
│   ├── Ppyoloeloss/loss = 3.6906
│   ├── Ppyoloeloss/loss_cls = 2.5202
│   ├── Ppyoloeloss/loss_dfl = 1.3999
│   └── Ppyoloeloss/loss_iou = 0.1882
├── Validation
│   ├── F1@0.50 = 0.0
│   ├── Map@0.50 = 0.0
│   ├── Ppyoloeloss/loss = 3.7645
│   ├── Ppyoloeloss/loss_cls = 2.84
│   ├── Ppyoloeloss/loss_dfl = 1.2265
│   ├── Ppyoloeloss/loss_iou = 0.1245
│   ├── Precision@0.50 = 0.0
│   └── Recall@0.50 = 0.0
```

```
=====
[2023-06-29 20:31:31] INFO - base_sg_logger.py - Checkpoint saved in /checkpo
[2023-06-29 20:31:31] INFO - sg_trainer.py - Best checkpoint overridden: valid
Train epoch 1: 100%|██████████| 140/140 [03:21<00:00, 1.44s/it, PPYoloELoss/
Validation epoch 1: 100%|██████████| 13/13 [00:09<00:00, 1.42it/s]
```

```
=====
SUMMARY OF EPOCH 1
├── Training
│   ├── Ppyoloeloss/loss = 2.967
│   │   ├── Best until now = 3.6906 (↘ -0.7236)
│   │   └── Epoch N-1      = 3.6906 (↘ -0.7236)
│   ├── Ppyoloeloss/loss_cls = 1.9128
│   │   ├── Best until now = 2.5202 (↘ -0.6074)
│   │   └── Epoch N-1      = 2.5202 (↘ -0.6074)
│   ├── Ppyoloeloss/loss_dfl = 1.3099
│   │   ├── Best until now = 1.3999 (↘ -0.09)
│   │   └── Epoch N-1      = 1.3999 (↘ -0.09)
│   └── Ppyoloeloss/loss_iou = 0.1597
│       ├── Best until now = 0.1882 (↘ -0.0285)
│       └── Epoch N-1      = 0.1882 (↘ -0.0285)
└── Validation
    ├── F1@0.50 = 0.0135
    └── Best until now = 0.0 (↗ 0.0135)
```

## ✓ 7. Cargar el mejor modelo

```
1 import os
2 avg_model = models.get(config.MODEL_NAME,
3                          num_classes=config.NUM_CLASSES,
4                          checkpoint_path=os.path.join(config.CHECKPOINT_DIR,
5
[2023-06-29 20:49:09] INFO - checkpoint_utils.py - Successfully loaded model \
```

## ✓ 8. Evaluación con test\_data\_loader

```
1
2 trainer.test(model=avg_model,
3               test_loader=test_data,
4               test_metrics_list=DetectionMetrics_050(score_thres=0.1,
5               top_k_predictions=300,
6               num_cls=config.NUM_CLASSES,
7               normalize_targets=True,
8               post_prediction_callback=PP
9               ))
```

```
Test: 100%|██████████| 6/6 [00:05<00:00, 1.20it/s]
{'PPYoloELoss/loss_cls': 1.1896276,
 'PPYoloELoss/loss_iou': 0.06636592,
 'PPYoloELoss/loss_dfl': 0.80759615,
 'PPYoloELoss/loss': 1.7593406,
 'Precision@0.50': tensor(0.0524),
 'Recall@0.50': tensor(0.9937),
 'mAP@0.50': tensor(0.7030),
 '-----'}
```

```
'F1@0.50': tensor(0.0981)}
```

```
1 import supervision
2
3 # Utiliza dir() para obtener los atributos y métodos de supervision
4 print(dir(supervision))
```

```
['BaseDataset', 'BoxAnnotator', 'ClassificationDataset', 'Classifications', 'C'
```

```
1 import supervision
2
3 # Accede al atributo deseado
4 attribute_value = supervision.dataset
5
6 # Imprime el valor del atributo
7 print(attribute_value)
8
```

```
<module 'supervision.dataset' from '/usr/local/lib/python3.10/dist-packages/si
```

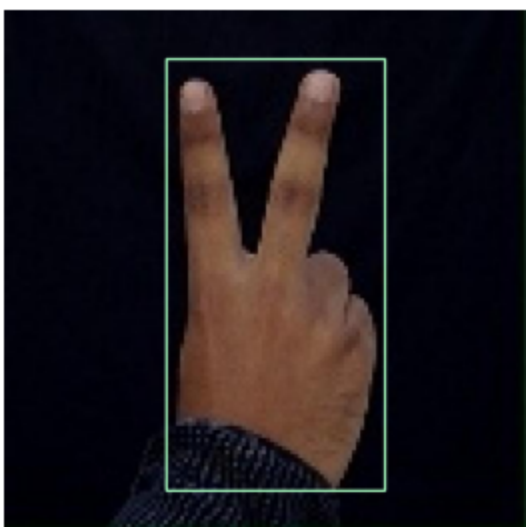
## ✓ 9. Visualización

```
1 import random
2 import supervision as sv
3 random.seed(10)
4 ds = sv.DetectionDataset.from_yolo(
5     images_directory_path=f"{config.DATA_DIR}/test/images",
6     annotations_directory_path=f"{config.DATA_DIR}/test/labels",
7     data_yaml_path=f"{config.DATA_DIR}/data.yaml",
8     force_masks=False
9 )
10
11
12
13 predictions = {}
14
15 CONFIDENCE_THRESHOLD = 0.3
16
17 #Inferencia
18 for image_name, image in ds.images.items():
19     result = list(avg_model.predict(image, conf=CONFIDENCE_THRESHOLD))[0]
20     detections = sv.Detections(
21         xyxy=result.prediction.bboxes_xyxy,
22         confidence=result.prediction.confidence,
23         class_id=result.prediction.labels.astype(int)
24     )
25     predictions[image_name] = detections
```



```
26
27
28
29 #Plot anotaciones vs predicciones
30 MAX_IMAGE_COUNT = 10
31
32 n = min(MAX_IMAGE_COUNT, len(ds.images))
33
34 keys = list(ds.images.keys())
35 keys = random.sample(keys, n)
36
37 box_annotator = sv.BoxAnnotator()
38
39 images = []
40 titles = []
41
42 for key in keys:
43     frame_with_annotations = box_annotator.annotate(
44         scene=ds.images[key].copy(),
45         detections=ds.annotations[key],
46         skip_label=True
47     )
48     images.append(frame_with_annotations)
49     titles.append('annotations')
50     frame_with_predictions = box_annotator.annotate(
51         scene=ds.images[key].copy(),
52         detections=predictions[key],
53         skip_label=True
54     )
55     images.append(frame_with_predictions)
56     titles.append('predictions')
57
58 %matplotlib inline
59 sv.plot_images_grid(images=images, titles=titles, grid_size=(n, 2), size=(2 * .
```

annotations



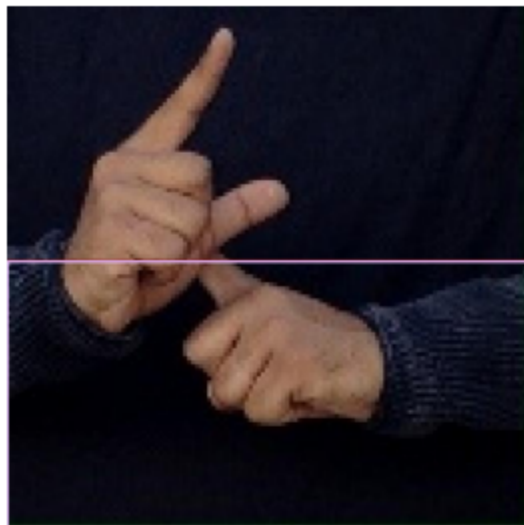
predictions



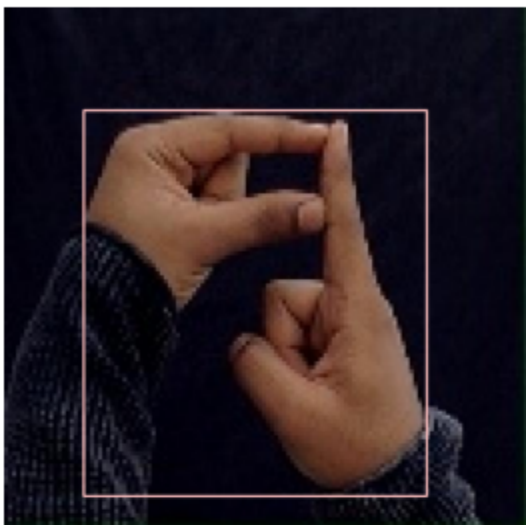
annotations



predictions



annotations



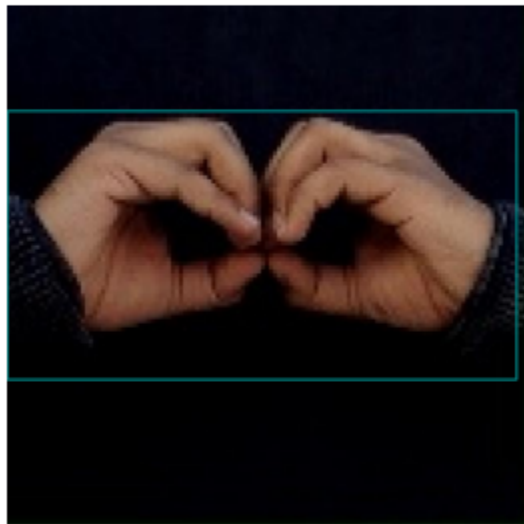
predictions



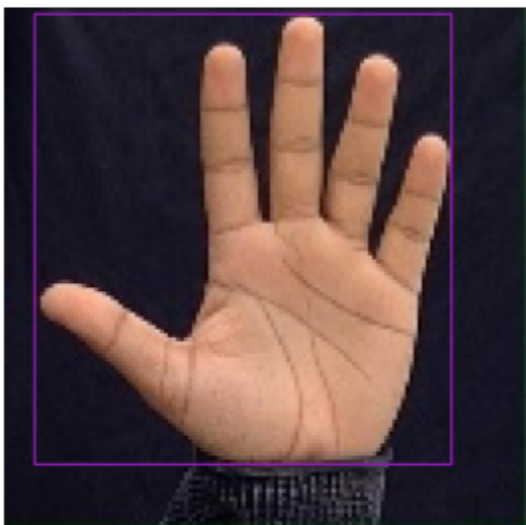
annotations



predictions



annotations



predictions



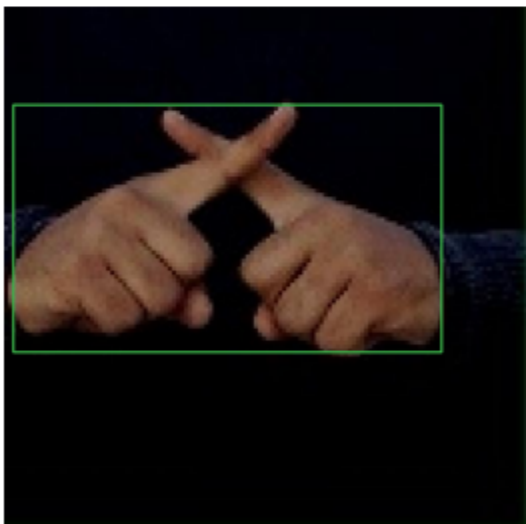
annotations



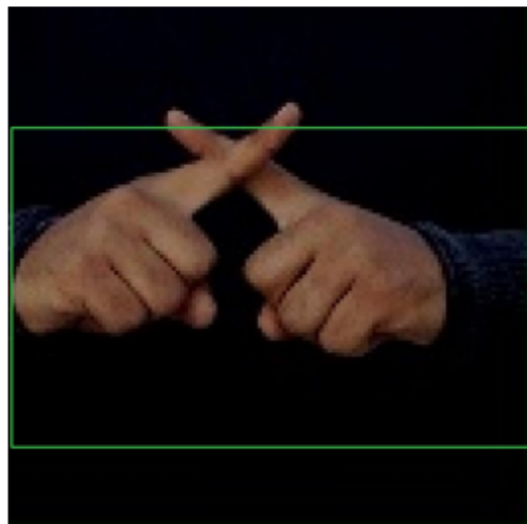
predictions



annotations

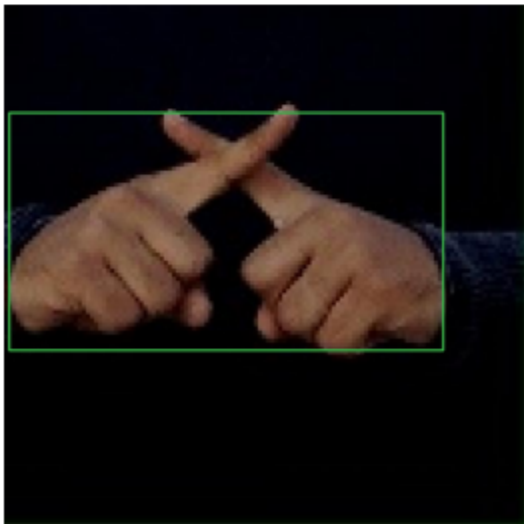


predictions

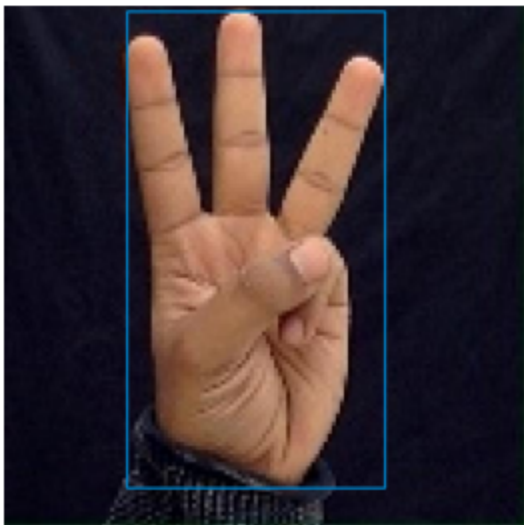


annotations

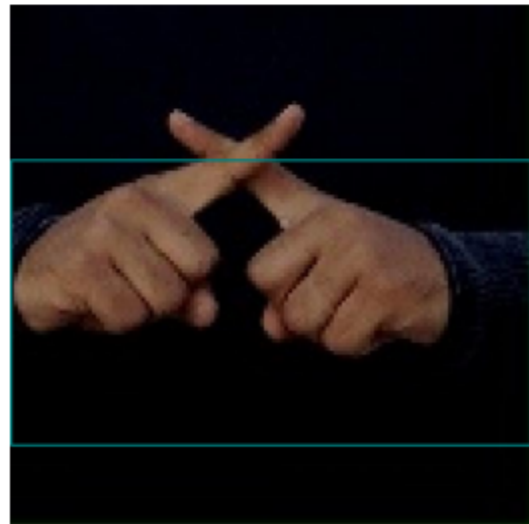
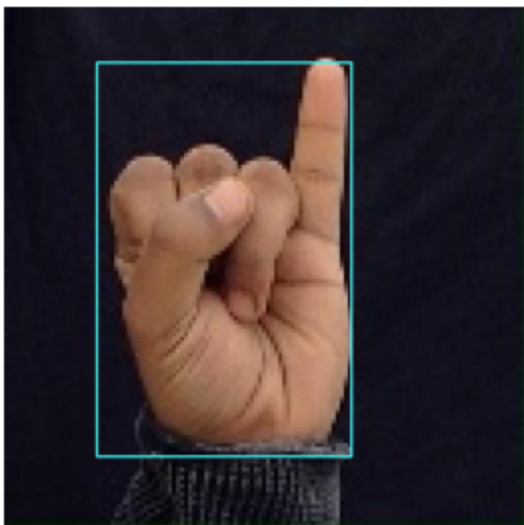
predictions



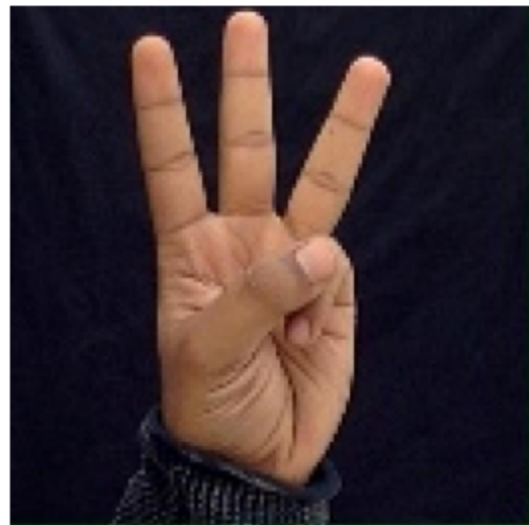
annotations



annotations



predictions



predictions



