

```
#Mount to google drive
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

Install required package and libraries

```
!pip install -q git+https://github.com/THU-MIG/yolov10.git
import os
import numpy as np
import cv2
from ultralytics import YOLOv10
import xml.etree.ElementTree as ET
import shutil
import random
import albumentations as A
from albumentations.pytorch import ToTensorV2
from IPython.display import Image
import yaml
HOME = os.getcwd()
print(HOME)
```

Installing build dependencies ... done
 Getting requirements to build wheel ... done
 Preparing metadata (pyproject.toml) ... done
 Building wheel for ultralytics (pyproject.toml) ... done
 /content

Convert COCO annotation format to YOLO

```
def convert_voc_to_yolo(xml_file, txt_file, image_width, image_height):
    tree = ET.parse(xml_file)
    root = tree.getroot()

    with open(txt_file, 'w') as f:
        for obj in root.findall('object'):
            class_name = obj.find('name').text
            class_id = class_name_to_id[class_name] # Convert class name to class ID

            bndbox = obj.find('bndbox')
            xmin = int(bndbox.find('xmin').text)
            ymin = int(bndbox.find('ymin').text)
            xmax = int(bndbox.find('xmax').text)
            ymax = int(bndbox.find('ymax').text)

            # Convert from Pascal VOC format to YOLO format
            x_center = (xmin + xmax) / 2.0 / image_width
            y_center = (ymin + ymax) / 2.0 / image_height
            width = (xmax - xmin) / image_width
            height = (ymax - ymin) / image_height

            f.write(f"{class_id} {x_center:.6f} {y_center:.6f} {width:.6f} {height:.6f}\n")

#Funtion to create YOLO annotation file
def create_yolo_annotations(xml_folder, output_folder, image_folder):
    if not os.path.exists(output_folder):
        os.makedirs(output_folder)

    for xml_file in os.listdir(xml_folder):
        if xml_file.endswith('.xml'):
            xml_path = os.path.join(xml_folder, xml_file)
            image_name = os.path.splitext(xml_file)[0] + '.jpg'
            image_path = os.path.join(image_folder, image_name)

            # Implement this function to get image size
            image_width, image_height = get_image_size(image_path)

            txt_file = os.path.join(output_folder, os.path.splitext(xml_file)[0] + '.txt')
            convert_voc_to_yolo(xml_path, txt_file, image_width, image_height)

def get_image_size(image_path):
    image = cv2.imread(image_path)
    return image.shape[1], image.shape[0]

# Define folder path for images, COCO annotation and YOLO annotation
xml_folder = '/content/drive/MyDrive/NEU-DET/annotation'
output_folder = '/content/drive/MyDrive/NEU-DET/YOLO_annotation'
image_folder = '/content/drive/MyDrive/NEU-DET/images'
```

```
# Mapping actual class names to corresponding IDs
class_name_to_id = {
    'crazing': 0,
    'inclusion': 1,
    'patches': 2,
    'pitted_surface': 3,
    'rolled-in_scale': 4,
    'scratches': 5
}

#Execute function
create_yolo_annotations(xml_folder, output_folder, image_folder)
```

Split dataset into Train, test, and validation set.

```

#Define data splitting function
def split_data_with_annotations(source_dir, annotation_dir, train_dir, val_dir, test_dir,
                               train_anno_dir, val_anno_dir, test_anno_dir,
                               train_split=0.7, val_split=0.15, test_split=0.15):
    """
    Splits the image data and their corresponding annotations into training, validation, and testing sets.

    Parameters:
    source_dir (str): The directory containing the original images.
    annotation_dir (str): The directory containing the annotations for the images.
    train_dir (str): The directory where the training images will be saved.
    val_dir (str): The directory where the validation images will be saved.
    test_dir (str): The directory where the test images will be saved.
    train_anno_dir (str): The directory where the training annotations will be saved.
    val_anno_dir (str): The directory where the validation annotations will be saved.
    test_anno_dir (str): The directory where the test annotations will be saved.
    train_split (float): The proportion of images to include in the training set.
    val_split (float): The proportion of images to include in the validation set.
    test_split (float): The proportion of images to include in the test set.
    """

    # Ensure the destination directories exist
    os.makedirs(train_dir, exist_ok=True)
    os.makedirs(val_dir, exist_ok=True)
    os.makedirs(test_dir, exist_ok=True)
    os.makedirs(train_anno_dir, exist_ok=True)
    os.makedirs(val_anno_dir, exist_ok=True)
    os.makedirs(test_anno_dir, exist_ok=True)

    # Get all image file names from the source directory
    image_files = [f for f in os.listdir(source_dir) if os.path.isfile(os.path.join(source_dir, f)) and f.endswith('.jpg')]

    # Shuffle the image files
    random.shuffle(image_files)

    # Calculate the split indices
    train_idx = int(len(image_files) * train_split)
    val_idx = train_idx + int(len(image_files) * val_split)

    # Split the images
    train_files = image_files[:train_idx]
    val_files = image_files[train_idx:val_idx]
    test_files = image_files[val_idx:]

    # Copy the files and their corresponding annotations to their respective directories
    for file_name in train_files:
        shutil.copy(os.path.join(source_dir, file_name), os.path.join(train_dir, file_name))
        shutil.copy(os.path.join(annotation_dir, file_name.replace('.jpg', '.txt')),
                    os.path.join(train_anno_dir, file_name.replace('.jpg', '.txt')))

    for file_name in val_files:
        shutil.copy(os.path.join(source_dir, file_name), os.path.join(val_dir, file_name))
        shutil.copy(os.path.join(annotation_dir, file_name.replace('.jpg', '.txt')),
                    os.path.join(val_anno_dir, file_name.replace('.jpg', '.txt')))

    for file_name in test_files:
        shutil.copy(os.path.join(source_dir, file_name), os.path.join(test_dir, file_name))
        shutil.copy(os.path.join(annotation_dir, file_name.replace('.jpg', '.txt')),
                    os.path.join(test_anno_dir, file_name.replace('.jpg', '.txt')))

    print(f"Training set: {len(train_files)} images")
    print(f"Validation set: {len(val_files)} images")
    print(f"Test set: {len(test_files)} images")

#Define folder path
source_directory = '/content/drive/MyDrive/NEU-DET/images'
annotation_directory = '/content/drive/MyDrive/NEU-DET/YOLO_annotation'
train_directory = '/content/drive/MyDrive/NEU-DET/train/images'
train_annotation_directory = '/content/drive/MyDrive/NEU-DET/train/labels'
validation_directory = '/content/drive/MyDrive/NEU-DET/val/images'
validation_annotation_directory = '/content/drive/MyDrive/NEU-DET/val/labels'
test_directory = '/content/drive/MyDrive/NEU-DET/test/images'
test_annotation_directory = '/content/drive/MyDrive/NEU-DET/test/labels'

#Execute data splitting function
split_data_with_annotations(source_directory, annotation_directory,
                           train_directory, validation_directory, test_directory,
                           train_annotation_directory, validation_annotation_directory, test_annotation_directory)

```

```

Training set: 1260 images
Validation set: 270 images
Test set: 270 images

```

Perform data augmentation on training set

```

# Define the augmentation pipeline
augmentation_pipeline = A.Compose([
    A.HorizontalFlip(p=0.5),
    A.VerticalFlip(p=0.5),
    A.RandomRotate90(p=0.5),
    A.RandomBrightnessContrast(p=0.2),
    A.GaussianBlur(p=0.2),
    ToTensorV2()
]), bbox_params=A.BboxParams(format='yolo', label_fields=['labels'], min_visibility=0.2))

# Function to augment image and bounding boxes in YOLO format
def augment_image_and_bboxes_yolo(image_path, bboxes, labels, images_output_dir, annotations_output_dir, aug_pipeline, num_augmentation):
    image = cv2.imread(image_path)
    image_name = os.path.basename(image_path).split('.')[0]

    for i in range(num_augmentations):
        augmented = aug_pipeline(image=image, bboxes=bboxes, labels=labels)
        augmented_image = augmented["image"]
        augmented_bboxes = augmented["bboxes"]
        augmented_labels = augmented["labels"]

        # Convert the image to the correct format and save it
        augmented_image = augmented_image.permute(1, 2, 0).cpu().numpy()
        augmented_image_path = os.path.join(images_output_dir, f"{image_name}_aug_{i}.jpg")
        cv2.imwrite(augmented_image_path, augmented_image)

        # Save augmented bounding boxes in YOLO format
        augmented_bboxes_path = os.path.join(annotations_output_dir, f"{image_name}_aug_{i}.txt")
        with open(augmented_bboxes_path, 'w') as f:
            for bbox, label in zip(augmented_bboxes, augmented_labels):
                x_center, y_center, bbox_width, bbox_height = bbox
                f.write(f"{label} {x_center} {y_center} {bbox_width} {bbox_height}\n")

        print(f"Saved augmented image and bboxes: {augmented_image_path}, {augmented_bboxes_path}")

# Path where images and annotations are stored
images_dir = '/content/drive/MyDrive/NEU-DET/train/images'
annotations_dir = '/content/drive/MyDrive/NEU-DET/train/labels'
images_output_dir = '/content/drive/MyDrive/NEU-DET/train/images'
annotations_output_dir = '/content/drive/MyDrive/NEU-DET/train/labels'

# Create output directories if they don't exist
os.makedirs(images_output_dir, exist_ok=True)
os.makedirs(annotations_output_dir, exist_ok=True)

# List all images
image_files = [f for f in os.listdir(images_dir) if f.endswith('.jpg')]

# Augment each image
for image_file in image_files:
    image_path = os.path.join(images_dir, image_file)
    annotation_path = os.path.join(annotations_dir, image_file.replace('.jpg', '.txt'))

    # Read annotations in YOLO format
    bboxes = []
    labels = []
    with open(annotation_path, 'r') as f:
        for line in f.readlines():
            label, x_center, y_center, width, height = map(float, line.strip().split())
            labels.append(int(label))
            bboxes.append([x_center, y_center, width, height]) # Keep in YOLO format

    # Execute data augmentation function
    augment_image_and_bboxes_yolo(image_path, bboxes, labels, images_output_dir, annotations_output_dir, augmentation_pipeline)

```

Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_72_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/labels/pitted_surface_72_aug_0.txt
 Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_72_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/labels/pitted_surface_72_aug_1.txt
 Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/patches_63_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/labels/patches_63_aug_0.txt
 Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/patches_63_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/labels/patches_63_aug_1.txt
 Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/scratches_33_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/labels/scratches_33_aug_0.txt
 Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/scratches_33_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/labels/scratches_33_aug_1.txt
 Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/crazing_186_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/labels/crazing_186_aug_0.txt
 Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/crazing_186_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/labels/crazing_186_aug_1.txt
 Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/crazing_289_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/labels/crazing_289_aug_0.txt
 Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/crazing_289_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/labels/crazing_289_aug_1.txt
 Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/scratches_268_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/labels/scratches_268_aug_0.txt
 Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/scratches_268_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/labels/scratches_268_aug_1.txt

```

Saved augmented image and bboxes: /content/drive/MyDrive/NEU-DET/train/images/rolled-in_scale_37_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/rolled-in_scale_37_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_10_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_10_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_269_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_269_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_136_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_136_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/crazing_184_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/crazing_184_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_9_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_9_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_140_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_140_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/inclusion_221_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/inclusion_221_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_197_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_197_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/scratches_29_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/scratches_29_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/inclusion_23_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/inclusion_23_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_262_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_262_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_226_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_226_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_56_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_56_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/crazing_277_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/crazing_277_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/scratches_5_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/scratches_5_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/rolled-in_scale_12_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/rolled-in_scale_12_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/crazing_18_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/crazing_18_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_166_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_166_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_249_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/patches_249_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/crazing_192_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/crazing_192_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_77_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_77_aug_1.jpg, /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_65_aug_0.jpg, /content/drive/MyDrive/NEU-DET/train/images/pitted_surface_65_aug_1.jpg

```

Model training

#Download the YOLOv10 pre-trained weight

```

!mkdir -p {HOME}/weights
!wget -P {HOME}/weights -q https://github.com/jameslahm/yolov10/releases/download/v1.0/yolov10n.pt
!wget -P {HOME}/weights -q https://github.com/jameslahm/yolov10/releases/download/v1.0/yolov10s.pt
!wget -P {HOME}/weights -q https://github.com/jameslahm/yolov10/releases/download/v1.0/yolov10m.pt
!wget -P {HOME}/weights -q https://github.com/jameslahm/yolov10/releases/download/v1.0/yolov10b.pt
!wget -P {HOME}/weights -q https://github.com/jameslahm/yolov10/releases/download/v1.0/yolov10x.pt
!wget -P {HOME}/weights -q https://github.com/jameslahm/yolov10/releases/download/v1.0/yolov10l.pt
!ls -lh {HOME}/weights

```



```

total 408M
-rw-r--r-- 1 root root 80M May 23 09:38 yolov10b.pt
-rw-r--r-- 1 root root 100M May 23 09:38 yolov10l.pt
-rw-r--r-- 1 root root 64M May 23 09:38 yolov10m.pt
-rw-r--r-- 1 root root 11M May 23 09:38 yolov10n.pt
-rw-r--r-- 1 root root 32M May 23 09:38 yolov10s.pt
-rw-r--r-- 1 root root 123M May 23 09:38 yolov10x.pt

```

Write data.yaml file for training

```
#Define the data.yaml file for training
data = {
  'names': [
    'crazing',
    'inclusion',
    'patches',
    'pitted_surface',
    'rolled-in_scale',
    'scratches'
  ],
  'nc': 6, #Number of defect class
  'path': '/content/drive/MyDrive/NEU-DET',#Dataset path
  'test': 'test/images',#Testing data path
  'train': 'train/images',#Training data path
  'val': 'val/images'#Validation data path
}
```

```
# Path to the output data.yaml file
yaml_file_path = '/content/drive/MyDrive/NEU-DET/data.yaml'
```

```
# Write the content to the YAML file
with open(yaml_file_path, 'w') as file:
  yaml.safe_dump(data, file)
```

```
print(f"data.yaml file has been successfully written to {yaml_file_path}")
```

data.yaml file has been successfully written to /content/drive/MyDrive/NEU-DET/data.yaml

```
#Train the YOLOv10-X model for defect detection
```

```
%cd {HOME}
```

```
!yolo task=detect mode=train epochs=300 plots=True verbose=True batch=8 optimizer='SGD' lr=0.001 imgsz=640 patience=100 \
model={HOME}/weights/yolov10x.pt \
data={HOME}/drive/MyDrive/NEU-DET/data.yaml
```

/content
/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py:733: FutureWarning: You are using `torch.load` with `weights_only`
ckpt = torch.load(file, map_location="cpu")
New <https://pypi.org/project/ultralytics/8.2.89> available 😊 Update with 'pip install -U ultralytics'
Ultralytics YOLOv8.1.34 Python-3.10.12 torch-2.4.0+cu121 CUDA:0 (NVIDIA L4, 22700MiB)
engine/trainer: task=detect, mode=train, model=/content/weights/yolov10x.pt, data=/content/drive/MyDrive/NEU-DET/data.yaml, epoch
Downloading <https://ultralytics.com/assets/Arial.ttf> to '/root/.config/yolov10/Arial.ttf'...
100% 755k/755k [00:00<00:00, 72.7MB/s]
2024-09-07 00:20:50.237783: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:485] Unable to register cuFFT factory: Atte
2024-09-07 00:20:50.259129: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:8454] Unable to register cuDNN factory: Att
2024-09-07 00:20:50.265676: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1452] Unable to register cuBLAS factory: A
Overriding model.yaml nc=80 with nc=6

	from	n	params	module	arguments
0	-1	1	2320	ultralytics.nn.modules.conv.Conv	[3, 80, 3, 2]
1	-1	1	115520	ultralytics.nn.modules.conv.Conv	[80, 160, 3, 2]
2	-1	3	436800	ultralytics.nn.modules.block.C2f	[160, 160, 3, True]
3	-1	1	461440	ultralytics.nn.modules.conv.Conv	[160, 320, 3, 2]
4	-1	6	3281920	ultralytics.nn.modules.block.C2f	[320, 320, 6, True]
5	-1	1	213120	ultralytics.nn.modules.block.SCDOWN	[320, 640, 3, 2]
6	-1	6	4604160	ultralytics.nn.modules.block.C2fCIB	[640, 640, 6, True]
7	-1	1	417920	ultralytics.nn.modules.block.SCDOWN	[640, 640, 3, 2]
8	-1	3	2712960	ultralytics.nn.modules.block.C2fCIB	[640, 640, 3, True]
9	-1	1	1025920	ultralytics.nn.modules.block.SPPF	[640, 640, 5]
10	-1	1	1545920	ultralytics.nn.modules.block.PSA	[640, 640]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
13	-1	3	3122560	ultralytics.nn.modules.block.C2fCIB	[1280, 640, 3, True]
14	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
15	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
16	-1	3	1948800	ultralytics.nn.modules.block.C2f	[960, 320, 3]
17	-1	1	922240	ultralytics.nn.modules.conv.Conv	[320, 320, 3, 2]
18	[-1, 13]	1	0	ultralytics.nn.modules.conv.Concat	[1]
19	-1	3	2917760	ultralytics.nn.modules.block.C2fCIB	[960, 640, 3, True]
20	-1	1	417920	ultralytics.nn.modules.block.SCDOWN	[640, 640, 3, 2]
21	[-1, 10]	1	0	ultralytics.nn.modules.conv.Concat	[1]
22	-1	3	3122560	ultralytics.nn.modules.block.C2fCIB	[1280, 640, 3, True]
23	[16, 19, 22]	1	4396596	ultralytics.nn.modules.head.v10Detect	[6, [320, 640, 640]]

YOLOv10x summary: 688 layers, 31666436 parameters, 31666420 gradients, 171.1 GFLOPs

Transferred 1123/1135 items from pretrained weights

TensorBoard: Start with 'tensorboard --logdir runs/detect/train', view at <http://localhost:6006/>

Freezing layer 'model.23.dfl.conv.weight'

AMP: running Automatic Mixed Precision (AMP) checks with YOLOv8n...

Downloading <https://github.com/ultralytics/assets/releases/download/v8.1.0/yolov8n.pt> to 'yolov8n.pt'...

100% 6.23M/6.23M [00:00<00:00, 499MB/s]

/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py:733: FutureWarning: You are using `torch.load` with `weights_only`
ckpt = torch.load(file, map_location="cpu")

/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py:641: FutureWarning: `torch.cuda.amp.autocast(args...)` is dep
with torch.cuda.amp.autocast(True):

AMP: checks passed ✓

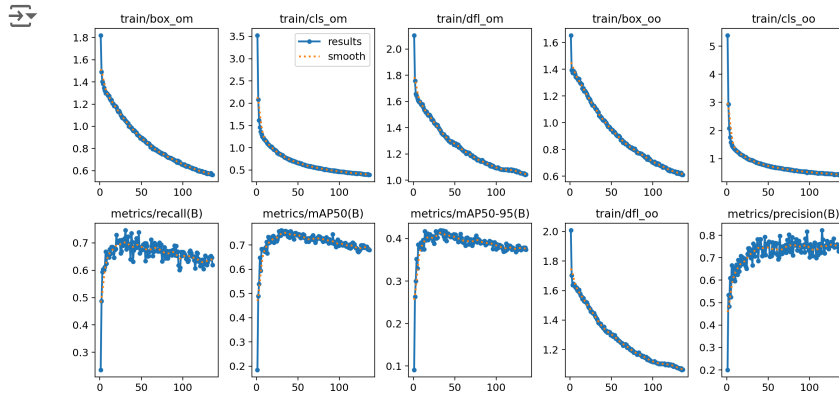
```

/usr/local/lib/python3.10/dist-packages/ultralytics/engine/trainer.py:276: FutureWarning: `torch.cuda.amp.GradScaler(args...)` is
self.scaler = torch.cuda.amp.GradScaler(enabled=self.amp)
train: Scanning /content/drive/MyDrive/NEU-DET/train/labels.cache... 3780 images, 0 backgrounds, 0 corrupt: 100% 3780/3780 [00:00
train: WARNING ⚠ /content/drive/MyDrive/NEU-DET/train/images/crazing_120.jpg: 1 duplicate labels removed
train: WARNING ⚠ /content/drive/MyDrive/NEU-DET/train/images/crazing_120.jpg: 1 duplicate labels removed
train: WARNING ⚠ /content/drive/MyDrive/NEU-DET/train/images/crazing_120.jpg: 1 duplicate labels removed

```

#Visualise the training process

Image(filename=f'{HOME}/runs/detect/train/results.png', width=600)



Model Testing

#Perform prediction on testing set

```

model=YOLOv10(f'{HOME}/runs/detect/train/weights/best.pt')
model.predict(source='/content/drive/MyDrive/NEU-DET/test/images', save=True)

```

```

/usr/local/lib/python3.10/dist-packages/ultralytics/nn/tasks.py:733: FutureWarning: You are using `torch.load` with `weights_only`
ckpt = torch.load(file, map_location="cpu")

```

```

image 1/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_101.jpg: 640x640 2 crazings, 18.5ms
image 2/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_122.jpg: 640x640 1 crazing, 18.6ms
image 3/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_126.jpg: 640x640 2 crazings, 24.2ms
image 4/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_135.jpg: 640x640 2 crazings, 18.5ms
image 5/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_153.jpg: 640x640 2 crazings, 18.4ms
image 6/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_154.jpg: 640x640 2 crazings, 17.5ms
image 7/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_155.jpg: 640x640 4 crazings, 17.2ms
image 8/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_164.jpg: 640x640 2 crazings, 17.5ms
image 9/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_176.jpg: 640x640 2 crazings, 17.5ms
image 10/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_188.jpg: 640x640 3 crazings, 17.3ms
image 11/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_191.jpg: 640x640 3 crazings, 17.3ms
image 12/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_195.jpg: 640x640 (no detections), 18.4ms
image 13/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_196.jpg: 640x640 4 crazings, 17.9ms
image 14/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_2.jpg: 640x640 2 crazings, 17.5ms
image 15/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_200.jpg: 640x640 3 crazings, 17.7ms
image 16/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_203.jpg: 640x640 2 crazings, 19.5ms
image 17/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_212.jpg: 640x640 3 crazings, 18.1ms
image 18/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_217.jpg: 640x640 2 crazings, 18.5ms
image 19/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_229.jpg: 640x640 3 crazings, 17.4ms
image 20/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_233.jpg: 640x640 4 crazings, 17.5ms
image 21/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_241.jpg: 640x640 2 crazings, 17.9ms
image 22/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_253.jpg: 640x640 (no detections), 17.5ms
image 23/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_261.jpg: 640x640 4 crazings, 17.1ms
image 24/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_262.jpg: 640x640 2 crazings, 16.7ms
image 25/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_267.jpg: 640x640 1 crazing, 17.1ms
image 26/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_268.jpg: 640x640 3 crazings, 17.2ms
image 27/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_275.jpg: 640x640 2 crazings, 17.5ms
image 28/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_284.jpg: 640x640 2 crazings, 18.6ms
image 29/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_287.jpg: 640x640 1 crazing, 18.0ms
image 30/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_299.jpg: 640x640 2 crazings, 19.9ms
image 31/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_30.jpg: 640x640 2 crazings, 16.9ms
image 32/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_40.jpg: 640x640 2 crazings, 17.9ms
image 33/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_52.jpg: 640x640 1 crazing, 17.3ms
image 34/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_56.jpg: 640x640 2 crazings, 16.9ms
image 35/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_6.jpg: 640x640 3 crazings, 17.3ms
image 36/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_76.jpg: 640x640 2 crazings, 19.1ms
image 37/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_81.jpg: 640x640 1 crazing, 18.9ms
image 38/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_83.jpg: 640x640 2 crazings, 18.4ms
image 39/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_87.jpg: 640x640 3 crazings, 17.9ms
image 40/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_92.jpg: 640x640 3 crazings, 17.8ms
image 41/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_95.jpg: 640x640 3 crazings, 18.2ms
image 42/270 /content/drive/MyDrive/NEU-DET/test/images/crazing_97.jpg: 640x640 5 crazings, 18.2ms
image 43/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_110.jpg: 640x640 2 inclusions, 18.9ms
image 44/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_115.jpg: 640x640 6 inclusions, 17.6ms
image 45/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_125.jpg: 640x640 7 inclusions, 17.2ms
image 46/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_127.jpg: 640x640 2 inclusions, 23.0ms
image 47/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_140.jpg: 640x640 3 inclusions, 18.5ms
image 48/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_141.jpg: 640x640 3 inclusions, 17.5ms

```

```
image 49/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_143.jpg: 640x640 4 inclusions, 16.7ms
image 50/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_147.jpg: 640x640 7 inclusions, 17.9ms
image 51/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_153.jpg: 640x640 2 inclusions, 17.2ms
image 52/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_164.jpg: 640x640 5 inclusions, 16.8ms
image 53/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_168.jpg: 640x640 3 inclusions, 17.4ms
image 54/270 /content/drive/MyDrive/NEU-DET/test/images/inclusion_180.jpg: 640x640 3 inclusions, 16.8ms
```

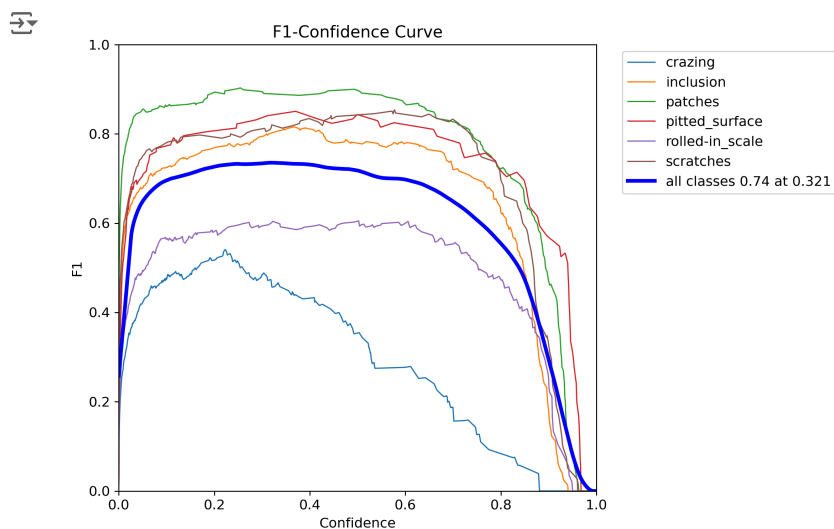
```
#Evaluate trained yolov10 on testing set
test_results = model.val(split='test',plots=True)
```

```
Ultralytics YOLOv8.1.34 Python-3.10.12 torch-2.4.0+cu121 CUDA:0 (NVIDIA L4, 22700MiB)
val: Scanning /content/drive/MyDrive/NEU-DET/test/labels.cache... 270 images, 0 backgrounds, 0 corrupt: 100%|██████████| 270/270 [06:17<00:00, 1.67it/s]
Class      Images  Instances  Box(P  R      mAP50  mAP50-95): 100%|██████████| 17/17 [00:10<00:00, 1.67it/s]
all         270      621        0.756  0.72   0.777   0.469
crazing     270      101        0.583  0.374  0.457   0.189
inclusion    270      153        0.801  0.817  0.854   0.502
patches     270      136        0.914  0.864  0.928   0.615
pitted_surface 270      44        0.851  0.841  0.875   0.633
rolled-in_scale 270      111       0.589  0.593  0.653   0.316
scratches   270      76         0.797  0.829  0.897   0.562

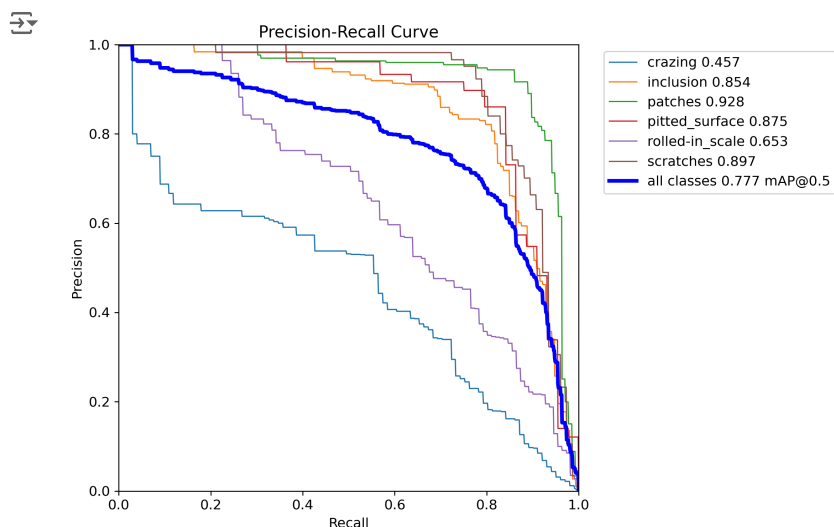
Speed: 0.9ms preprocess, 28.0ms inference, 0.0ms loss, 1.5ms postprocess per image
Results saved to runs/detect/val
```

Visualise testing result

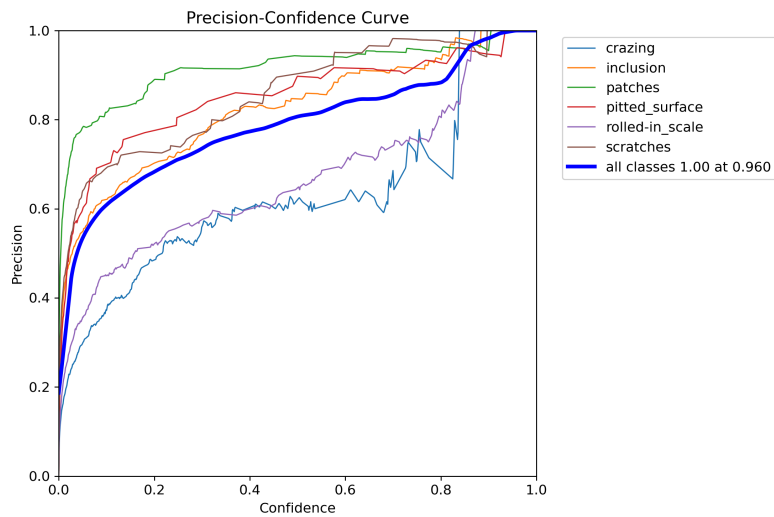
```
#F1 curve
Image(filename=f'{HOME}/runs/detect/val/F1_curve.png', width=600)
```



```
#Precision-recall curve
Image(filename=f'{HOME}/runs/detect/val/PR_curve.png', width=600)
```

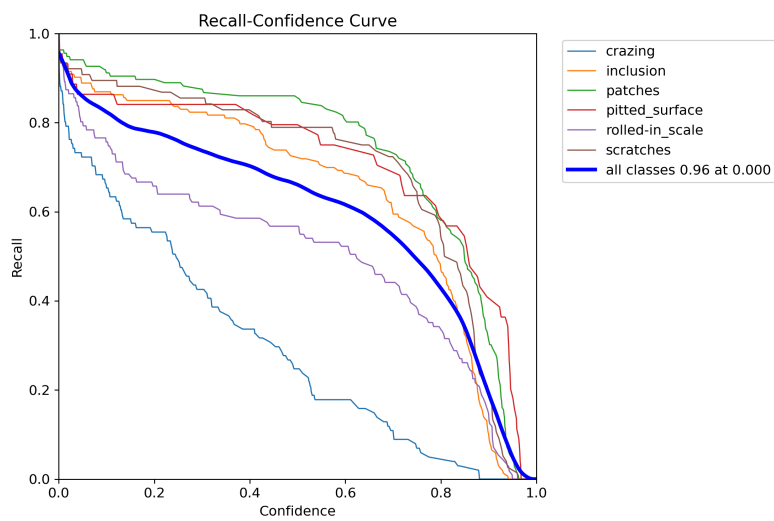


```
#precision-confidence curve
Image(filename=f'{HOME}/runs/detect/val/P_curve.png', width=600)
```

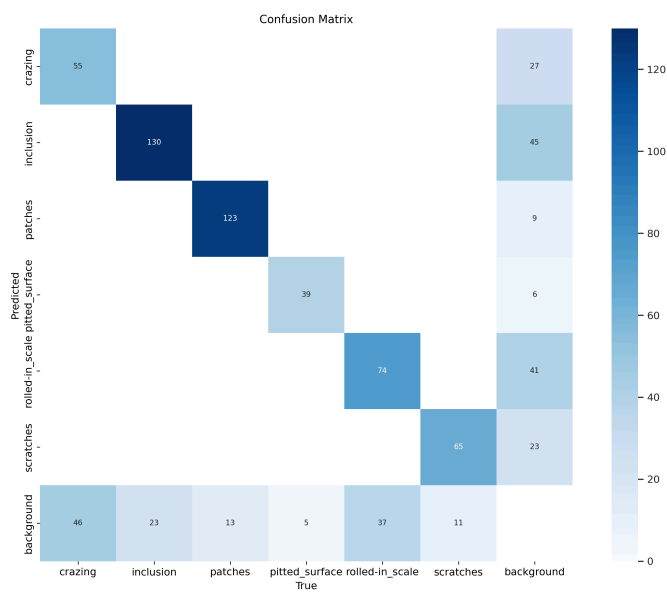
```
#recall-confidence curve
```

```
Image(filename=f'{HOME}/runs/detect/val/R_curve.png', width=600)
```



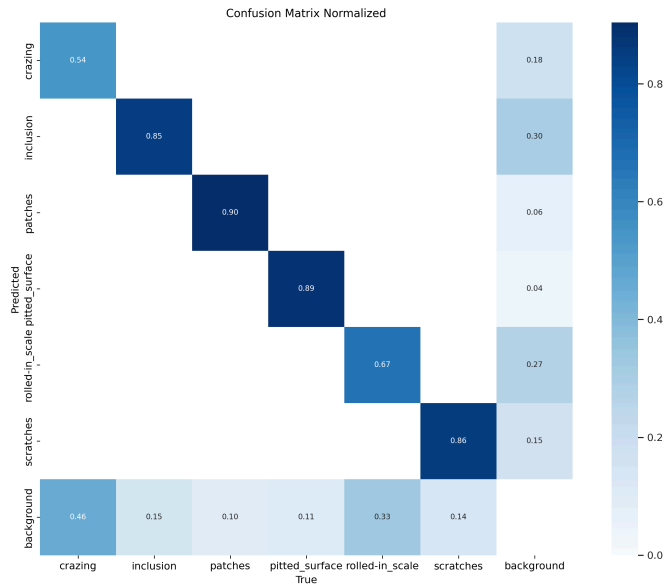
```
#Confusion matrix
```

```
Image(filename=f'{HOME}/runs/detect/val/confusion_matrix.png', width=600)
```



```
#Normalised confusion matrix
```

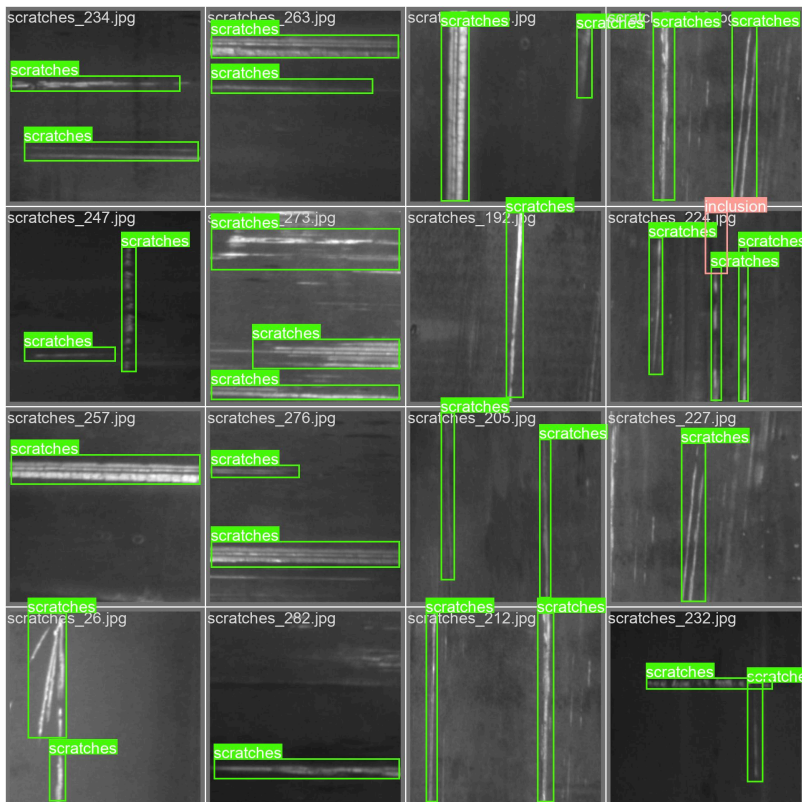
```
Image(filename=f'{HOME}/runs/detect/val/confusion_matrix_normalized.png', width=600)
```



Compare actual label with model prediction

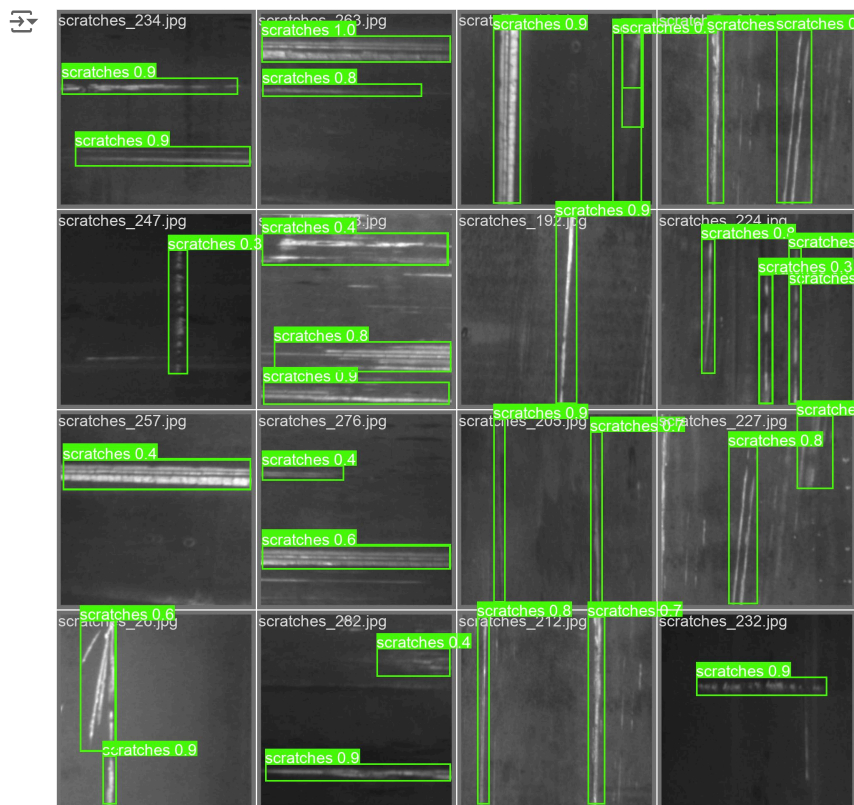
```
#Defect labels for testing batch 0
```

```
Image(filename=f'{HOME}/runs/detect/val/val_batch0_labels.jpg', width=600)
```



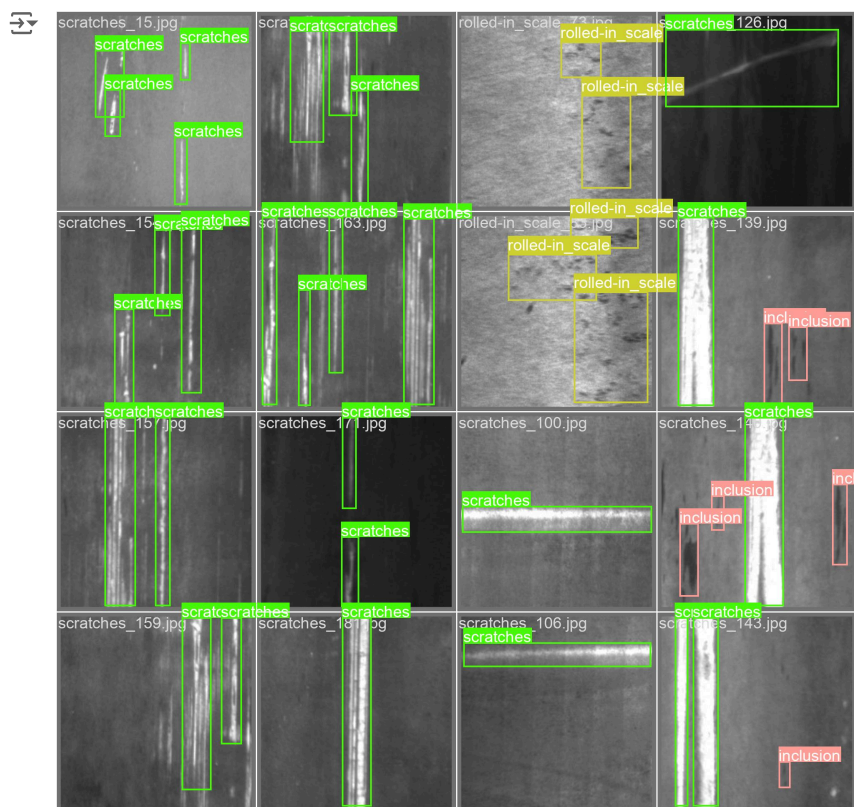
```
#Model prediction on testing batch 0
```

```
Image(filename=f'{HOME}/runs/detect/val/val_batch0_pred.jpg', width=600)
```



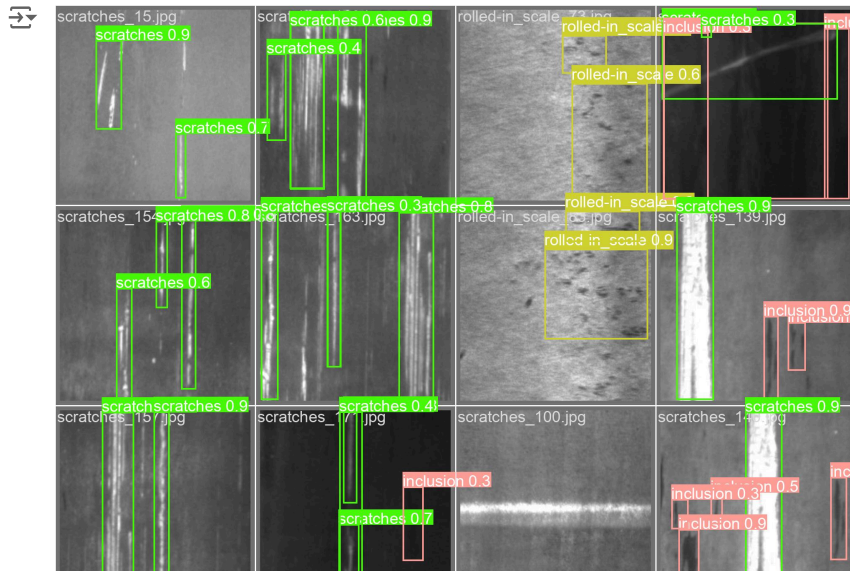
#Defect labels on testing batch 1

Image(filename=f'{HOME}/runs/detect/val/val_batch1_labels.jpg', width=600)



#Model prediction on testing batch 1

Image(filename=f'{HOME}/runs/detect/val/val_batch1_pred.jpg', width=600)



```
#Defect labels on testing batch 2
```

```
Image(filename=f'{HOME}/runs/detect/val/val_batch2_labels.jpg', width=600)
```

