Training/Customizing YOLOv11 on face mask detection (from Kaggle: custom dataset with 3-class labels)

7 Steps:

#1
pip install ultralytics

#2
Create (and cd to) directory/folder YOLO (or some other name):
create (and cd to) yolov11
download yolo11n.pt from https://github.com/ultralytics/ultralytics

#3
(while still under yolov11 above) Create train_custom/masks.yaml with:

train: <rest of full path>/train_custom/train
val: <rest of full path>/train_custom/val
test: <rest of full path>/train_custom/test

# Classes
nc: 3 # number of classes
names: ['with_mask', 'without_mask', 'mask_weared_incorrect']

#4
Get raw images and labels from Kaggle - https://www.kaggle.com/datasets/andrewmvd/face-mask-detection
.. and save both annotations and images dirs under the dir YOLO mentioned above.

#5
Data prep (by running yolo_data_prep.py under YOLO dir above)
- Convert Pascal VOC labels to Yolo format
- Split data into subfolders: the annotations and images dirs above will be copied to (and re-organized under) yolov11:

    yolov11\train_custom\train\images
    yolov11\train_custom\train\labels
    yolov11\train_custom\val\images
    yolov11\train_custom\val\labels
    yolov11\train_custom\test\images
    yolov11\train_custom\test\labels

For the *.txt files under the dir labels above, the 1st column there would be with values 0, 1, 2 respectively for:
['with_mask', 'without_mask', 'mask_weared_incorrect']

.. and the remaining columns would be for the bounding box (in yolo format): x center, y center, width, height
Note that Pascal VOC would have been this format: top-left point (xmin, ymin), bottom-right (xmax, ymax)

#7
Perform training (python perform_train.py) under yolov11 (as needed, increase #epochs to get better quality):

```
#model = YOLO("yolov11n.yaml")  # build a new model from scratch
model = YOLO("yolo11n.pt")  # load a pretrained model (recommended for training)
```

```
results = model.train(data="train_custom/masks.yaml", epochs=50, imgsz=512,
batch=4, verbose=True, device='cpu')
# device='cuda' to use GPU
```

In case of this error (`ModuleNotFoundError: No module named 'cv2'`), then:
pip install opencv-python

#8
Prediction, see: perform_pred.py

.. would need to update the model path to use the *.pt from last step above:

```
!yolo detect predict model="runs/detect/train6/weights/best.pt"
source="test/DJT.jpg" conf=0.3
!yolo detect predict model="runs/detect/train6/weights/best.pt"
source="test/maksssksksss10.png" conf=0.3
!yolo detect predict model="runs/detect/train6/weights/best.pt"
source="test/Mask wearing.mp4" conf=0.3
```

.. and put the inputs (*.jpg, *.png, *.mp4) above under: <rest of full path>/yolov11/runs/test

.. the results(images or videos) will be under: <rest of full path>/yolov11/runs/detect/predict*/