03/11/26 LAB: 15

Implement a Yolo model to detect objects

*~~X~~ Aim :=

To implement a Yolo-based object decodor to locate and classify object in images

* objective :-

1. Use a yolo pretrained model as a starting point
2. Fine-tune(train) on your dataset in Yolo format
3. Run inference to detect objects and visualize bounding boxes
4. Evaluate performance (mAP, precision/recall)

* pseudocode :=

1. prepare dataset in Yolo format (images + labels + tx + p
image or loc
2. Install ultralytics (yolov8) or clone Yolov5 repo
3. Load pretrained Yolo model weights
4. Replace/adjust head or hyperparams for custom classe
5. Train model on dataset for N epochs
6. Run inference on test images, draw boxes & class
7. Evaluate using mAp / visualize results.

* result :-

successfully implemented a yolo model to detect objects

output

448x640 1 car, 322.2ms

Speed: 34.0ms preprocess, 322.2ms

inference 132.6ms postprocess per

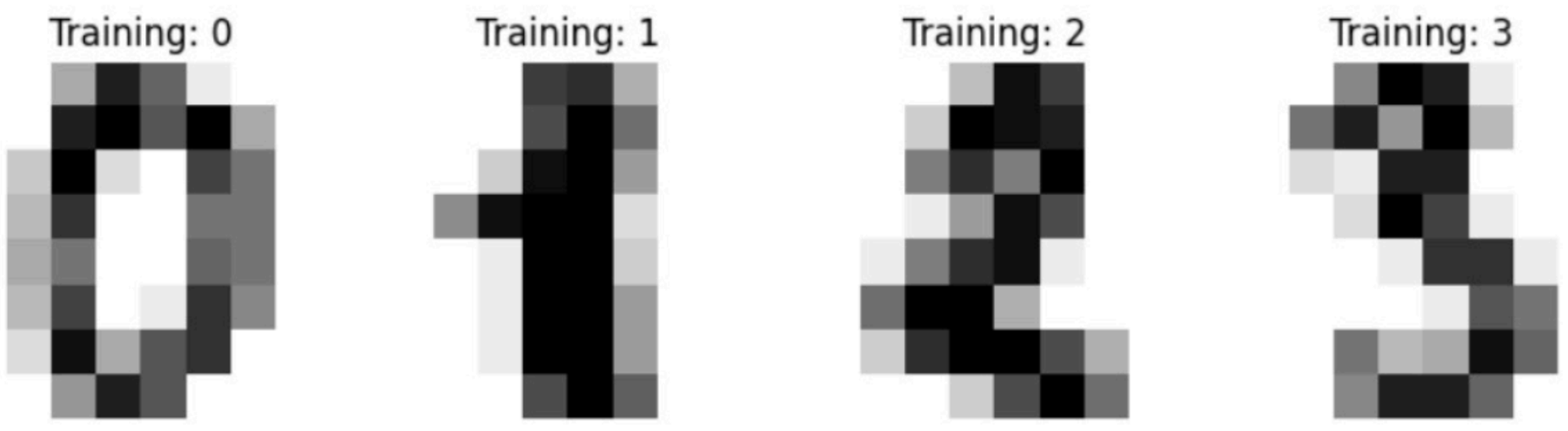
image at shape (1, 3, 448, 640)

In [101…

```
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

In [103…

```
_, axes = plt.subplots(nrows=1, ncols=4, figsize=(10, 3))
for ax, image, label in zip(axes, digits.images, digits.target):
    ax.set_axis_off()
    ax.imshow(image, cmap=plt.cm.gray_r, interpolation="nearest")
    ax.set_title("Training: %i" % label)
```



In [102…

```
digits = load_digits()
data = digits['data']
```

In [104…

```
digits.images[0].shape
```

In [104…

```
digits.images[0].shape
```

Out[104… (8, 8)

In [105…

```
n_samples = len(digits.images)
data = digits.images.reshape((n_samples, -1))
```

In [106…

```
from sklearn import svm,metrics
clf = svm.SVC(gamma=0.001)
```

In [107…

```
X_train, X_test, y_train, y_test = train_test_split(
    data, targets, test_size=0.3, shuffle=True
)

clf.fit(X_train, y_train)

predicted = clf.predict(X_test)
```

In [108…

```
print(
    f"Classification report for classifier {clf}:\n"
    f"{metrics.classification_report(y_test, predicted)}\n"
)
```

```
Classification report for classifier SVC(gamma=0.001):
              precision    recall  f1-score   support
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	49
1	0.97	1.00	0.98	60
2	1.00	1.00	1.00	59
3	1.00	1.00	1.00	44
4	1.00	1.00	1.00	52
5	1.00	0.96	0.98	57
6	0.98	1.00	0.99	55
7	0.98	1.00	0.99	54
8	1.00	0.96	0.98	57
9	0.98	0.98	0.98	53
accuracy			0.99	540
macro avg	0.99	0.99	0.99	540
weighted avg	0.99	0.99	0.99	540

In [109…

```
disp = metrics.ConfusionMatrixDisplay.from_predictions(y_test, predicted)
disp.figure_.suptitle("Confusion Matrix")

plt.show()
```