Yolov7-based Graph Arrow Detection Model

This project features a YOLOv7-based model designed to detect arrows in graphs and predict their direction, whether upward or downward. The model underwent training on a custom dataset comprising 504 images of arrows, divided into 80% for training and 20% for validation. The training spanned 100 epochs, resulting in a well-performing model on test images.

Model Performance and Recommendations

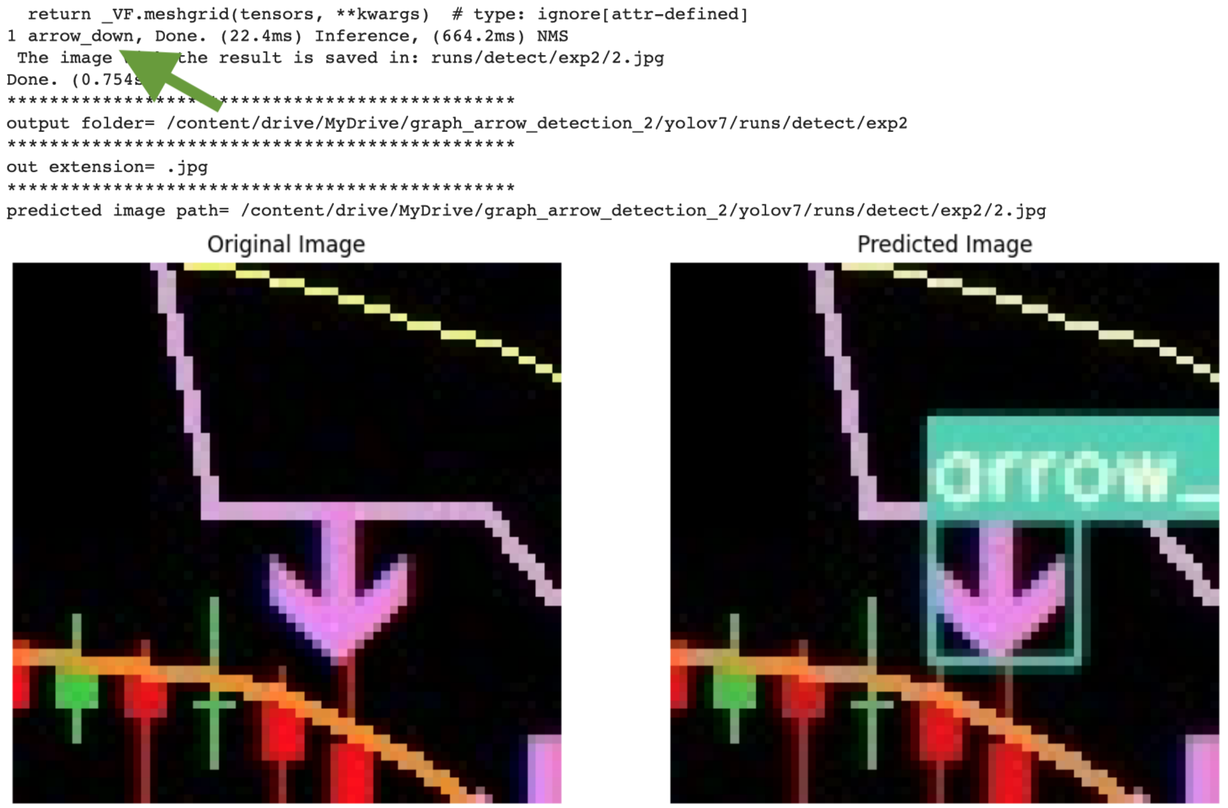
The model demonstrates proficiency in detecting arrows in various graph scenarios. To enhance its performance further, consider expanding the dataset and increasing the number of training epochs. This could lead to improved generalization and accuracy.

Unpredicted Images Explanation

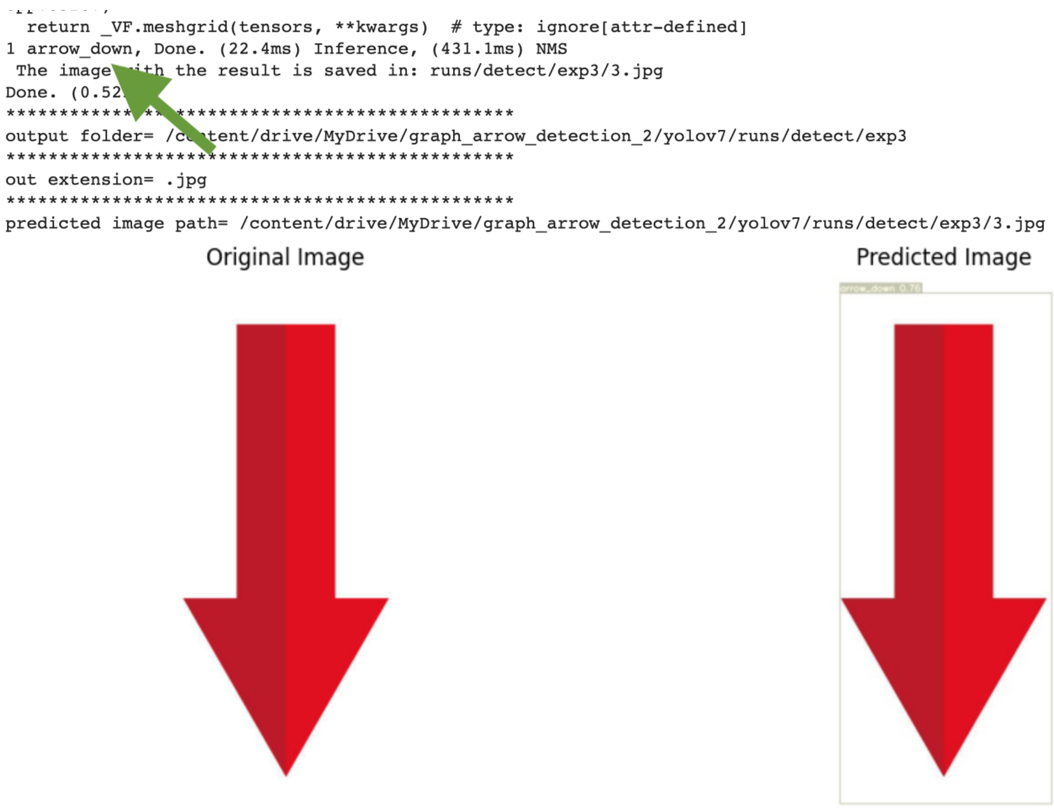
It's noteworthy that the model did not make predictions on the last two specific images. This occurred because there were no instances of such images within the provided dataset. During the entire training process, the model did not encounter examples of this specific type, leading to a lack of prediction capability. To address this, including diverse images in the dataset that represent all possible scenarios would be beneficial.

Model Results

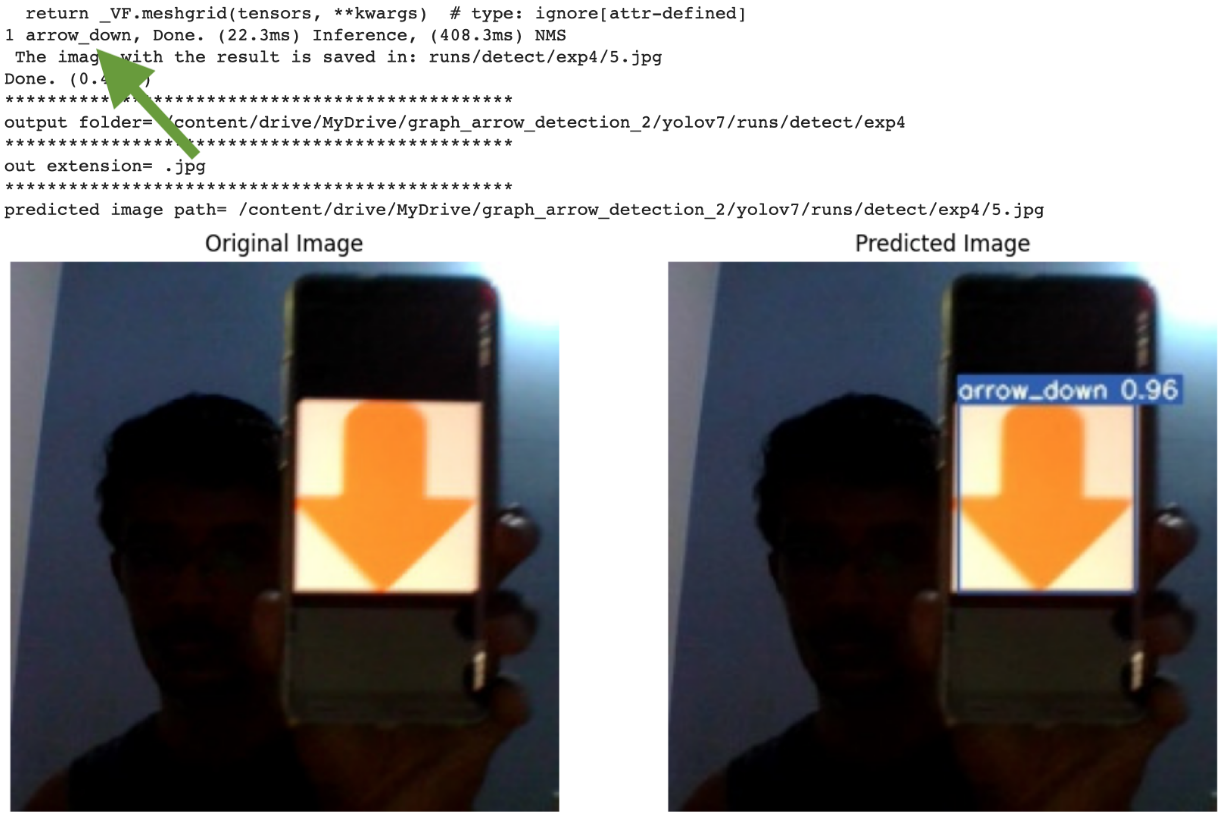
The model's results on the provided test images are detailed below:



```
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
1 arrow_down, Done. (22.4ms) Inference, (664.2ms) NMS
 The image with the result is saved in: runs/detect/exp2/2.jpg
Done. (0.754s
************************************************
output folder= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp2
************************************************
out extension= .jpg
************************************************
predicted image path= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp2/2.jpg
```



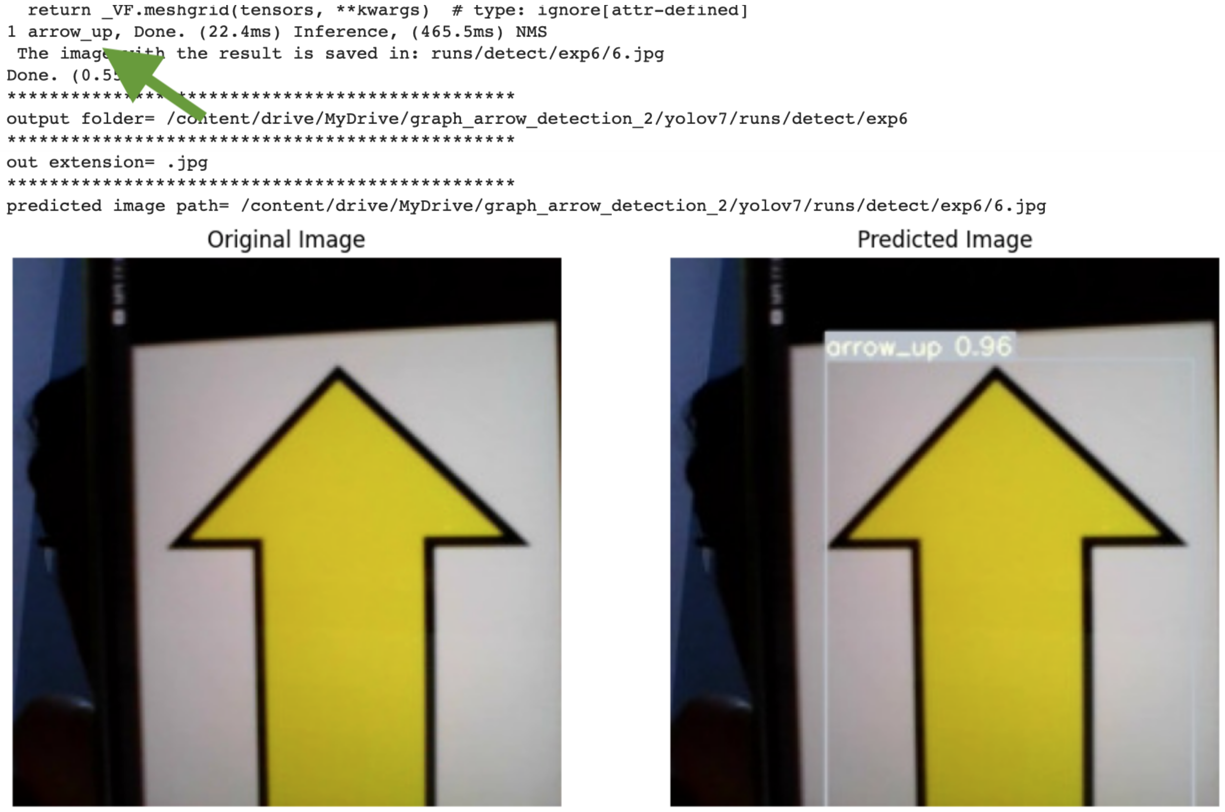
```
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
1 arrow_down, Done. (22.4ms) Inference, (431.1ms) NMS
 The image with the result is saved in: runs/detect/exp3/3.jpg
Done. (0.52
************************************************
output folder= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp3
************************************************
out extension= .jpg
************************************************
predicted image path= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp3/3.jpg
```



```
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
1 arrow_down, Done. (22.3ms) Inference, (408.3ms) NMS
 The image with the result is saved in: runs/detect/exp4/5.jpg
Done. (0.4[illegible])
************************************************
output folder= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp4
************************************************
out extension= .jpg
************************************************
predicted image path= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp4/5.jpg
```

Original Image

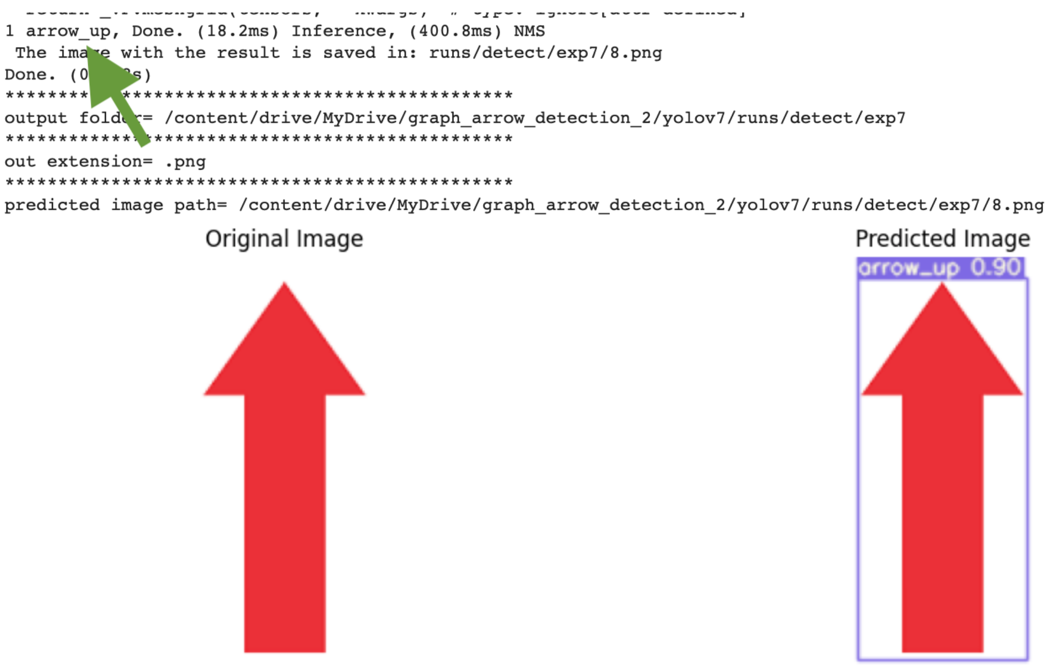
Predicted Image



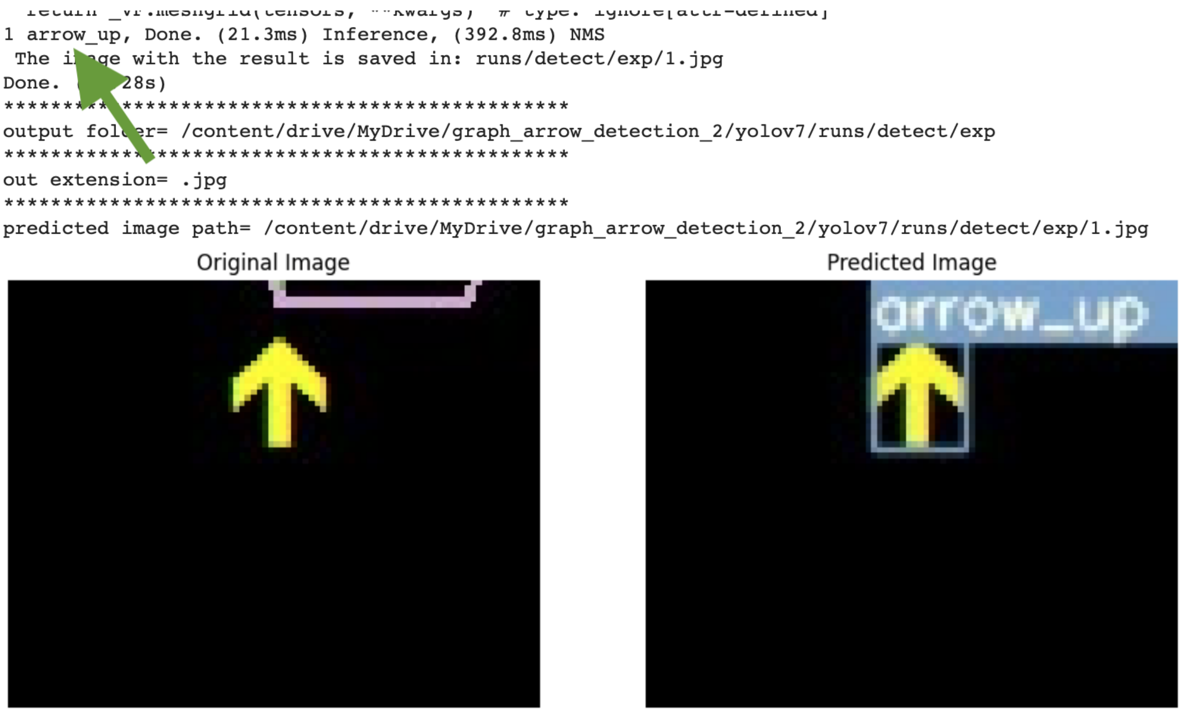
```
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
1 arrow_up, Done. (22.4ms) Inference, (465.5ms) NMS
 The image with the result is saved in: runs/detect/exp6/6.jpg
Done. (0.55[illegible]
************************************************
output folder= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp6
************************************************
out extension= .jpg
************************************************
predicted image path= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp6/6.jpg
```

Original Image

Predicted Image

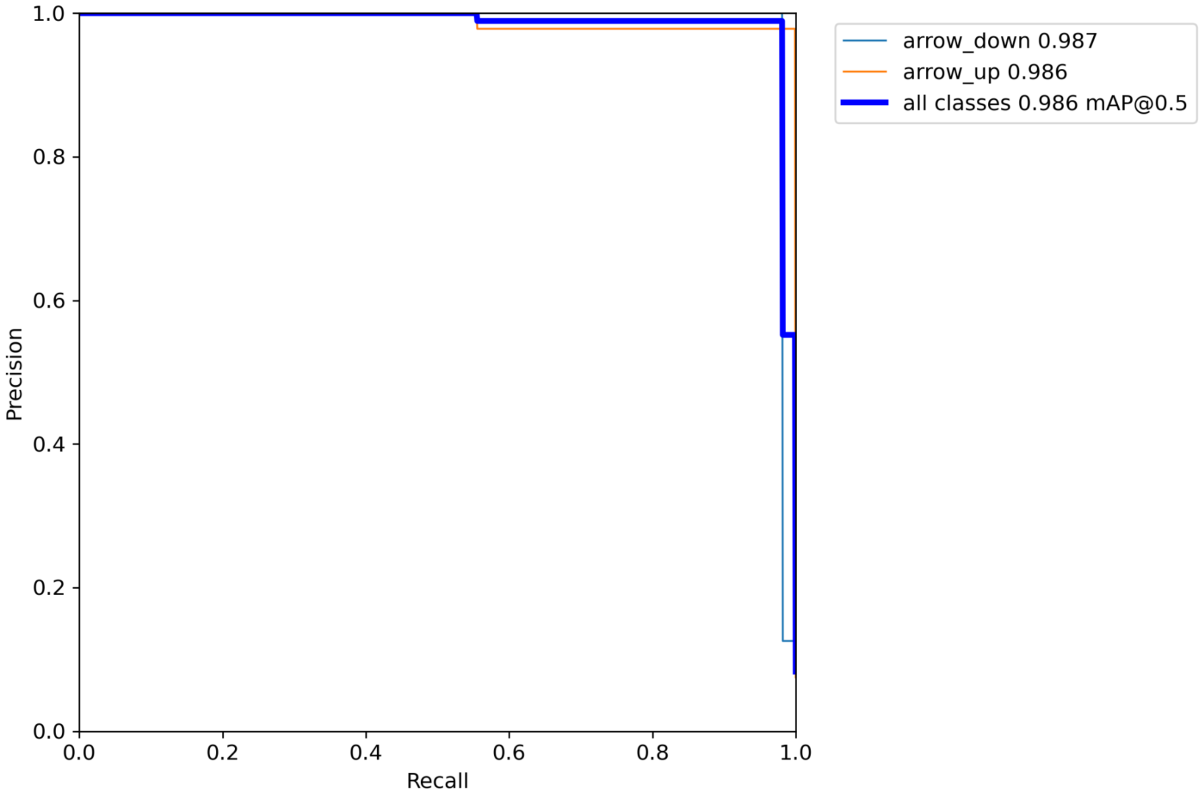


```
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
1 arrow_up, Done. (18.2ms) Inference, (400.8ms) NMS
 The image with the result is saved in: runs/detect/exp7/8.png
Done. (0[illegible]s)
************************************************
output folder= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp7
************************************************
out extension= .png
************************************************
predicted image path= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp7/8.png
```

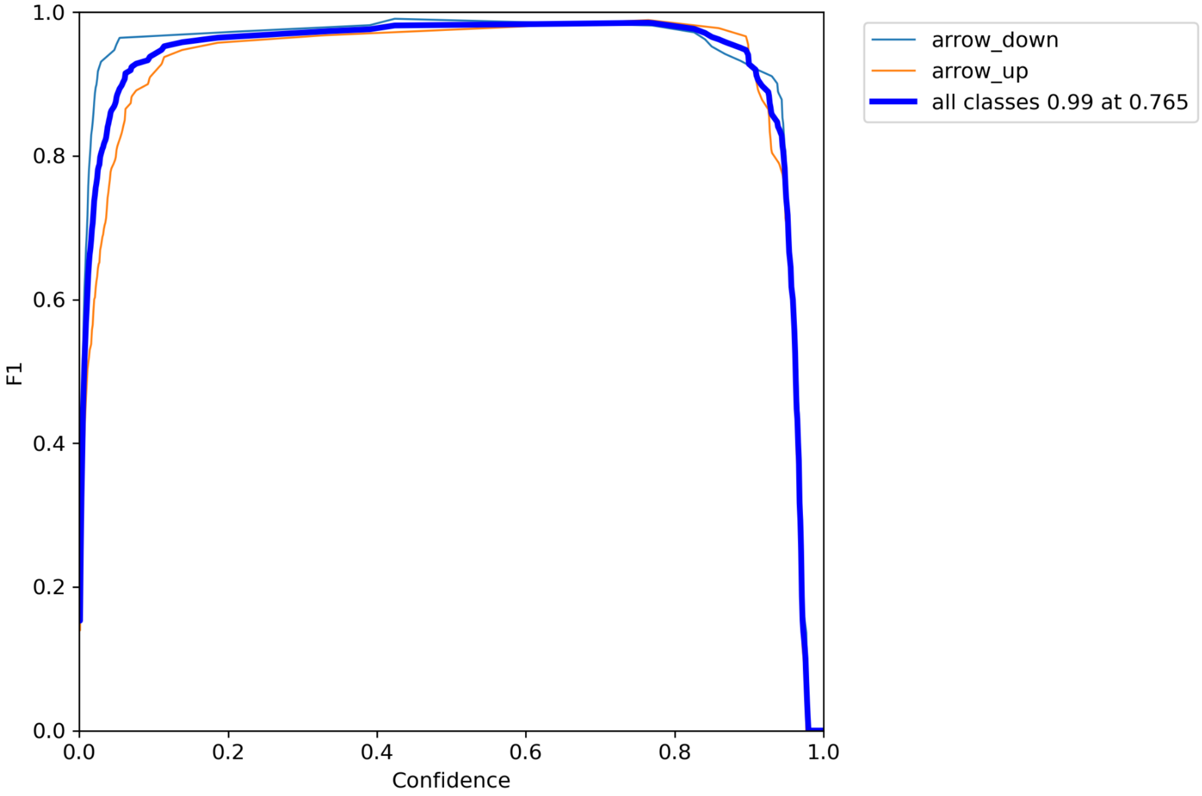


```
  return _VF.meshgrid(tensors, **kwargs)  # type: ignore[attr-defined]
1 arrow_up, Done. (21.3ms) Inference, (392.8ms) NMS
 The image with the result is saved in: runs/detect/exp/1.jpg
Done. ([illegible]28s)
************************************************
output folder= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp
************************************************
out extension= .jpg
************************************************
predicted image path= /content/drive/MyDrive/graph_arrow_detection_2/yolov7/runs/detect/exp/1.jpg
```

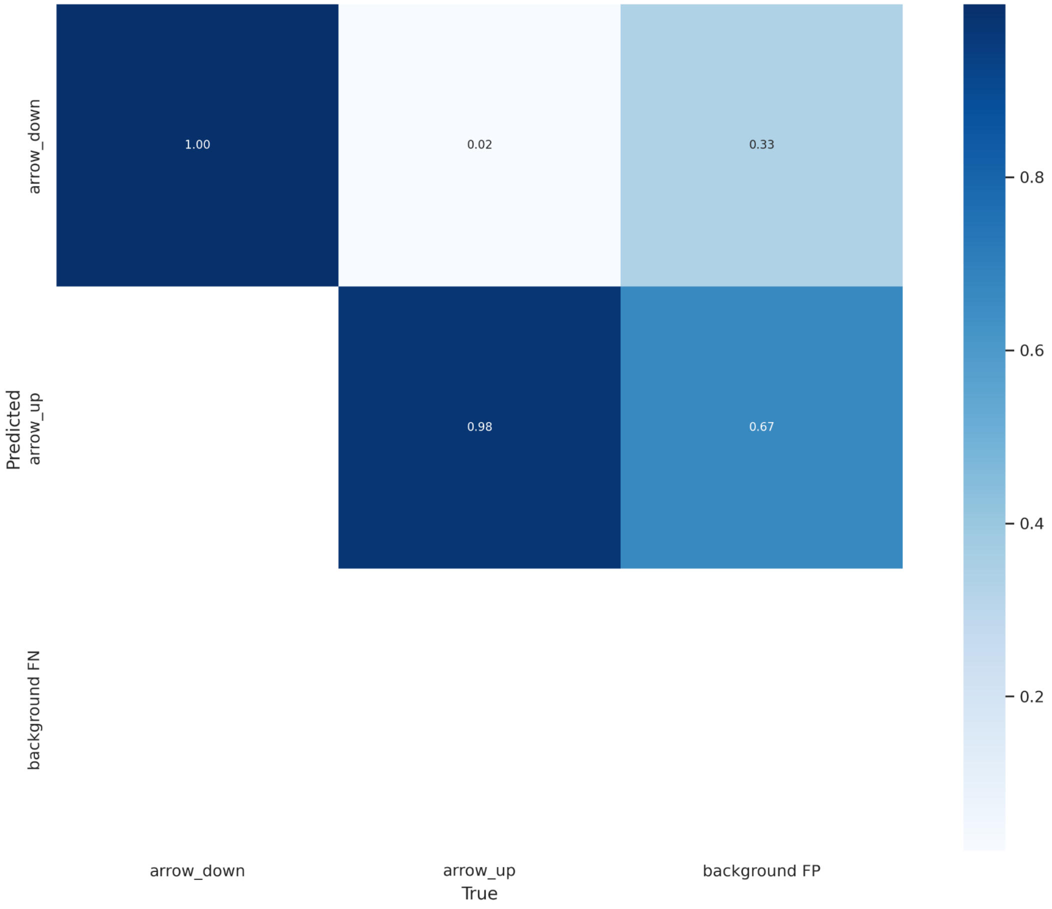
Precision-Recall Curve:



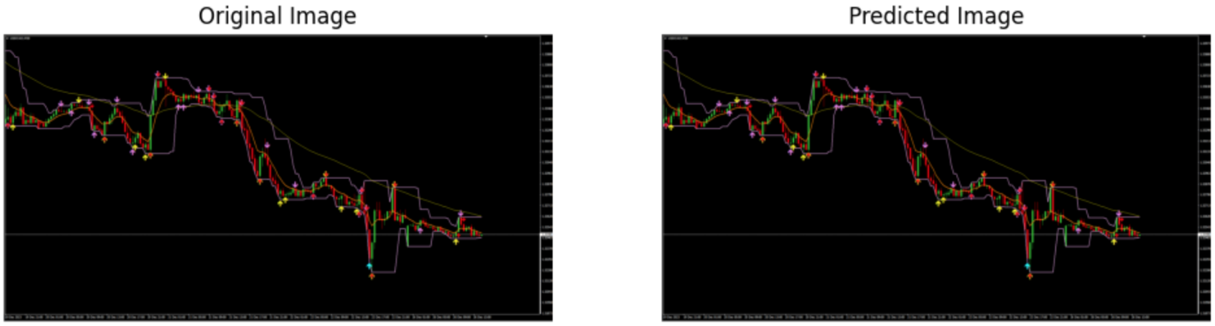
F1 Score Curve:

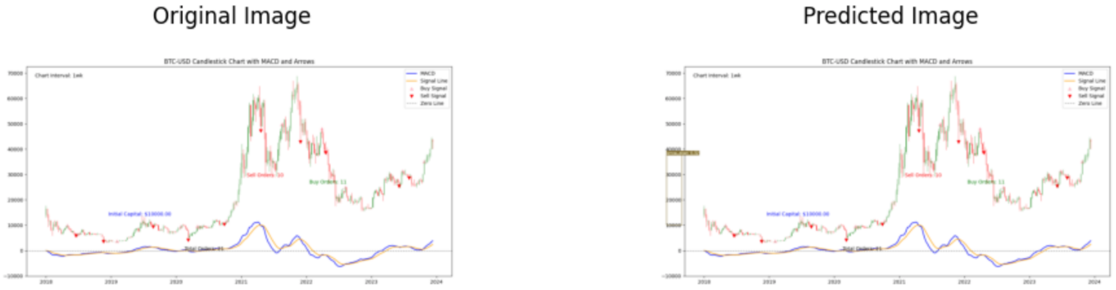


Confusion Matrix:



No prediction images:


Original Image
Predicted Image


Original Image
Predicted Image
BTC-USD Candlestick Chart with MACD and Arrows
Chart Interval: 1wk
MACD
Signal Line
Buy Signal
Sell Signal
Zero Line
Initial Capital: $10000.00
Sell Orders: 10
Buy Orders: 11
Total Orders: 21