Помощь компьютерного зрения в поиске людей

Выполнила:
Сукнева Виолетта,
Ученица 11 «А» класса
Руководитель:
Киреева Елена Николаевна,
Учитель информатики

Актуальность

- Несмотря на развитие технологий, проблема пропажи родных и близких все еще актуальна. Плакаты о поиске исчезнувших без вести, к сожалению, не ушли в прошлое. Поэтому я решила создать приложение, которое поможет федеральным службам в поисках.

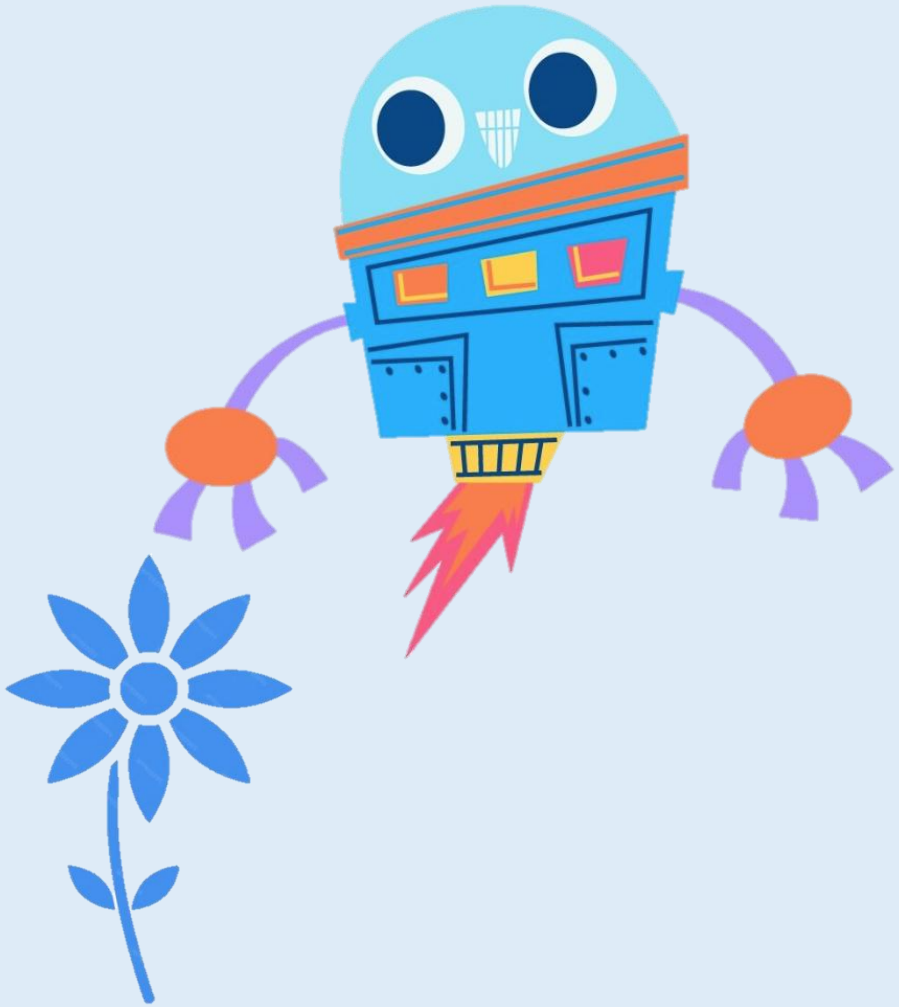
Цель проекта

Создание мобильного приложения, способного определить человека и выдать ответ с помощью внедрённой нейронной сети

Задачи проекта

1) Изучить литературу о компьютерном, машинном зрении, нейронных сетях, инструментах, используемых для создания мобильного приложения;
2) Создать нейронную сеть, тренировать её на разном количестве эпох, сравнить метрики и выбрать лучший вариант;
3) Создание простого мобильного приложения, привязка к нейронной сети.

Инструментарий



Android Studio

roboflow

Jetpack Compose

Yolo

PC

Flask
web development,
one drop at a time

Google colab

Обучение нейросети

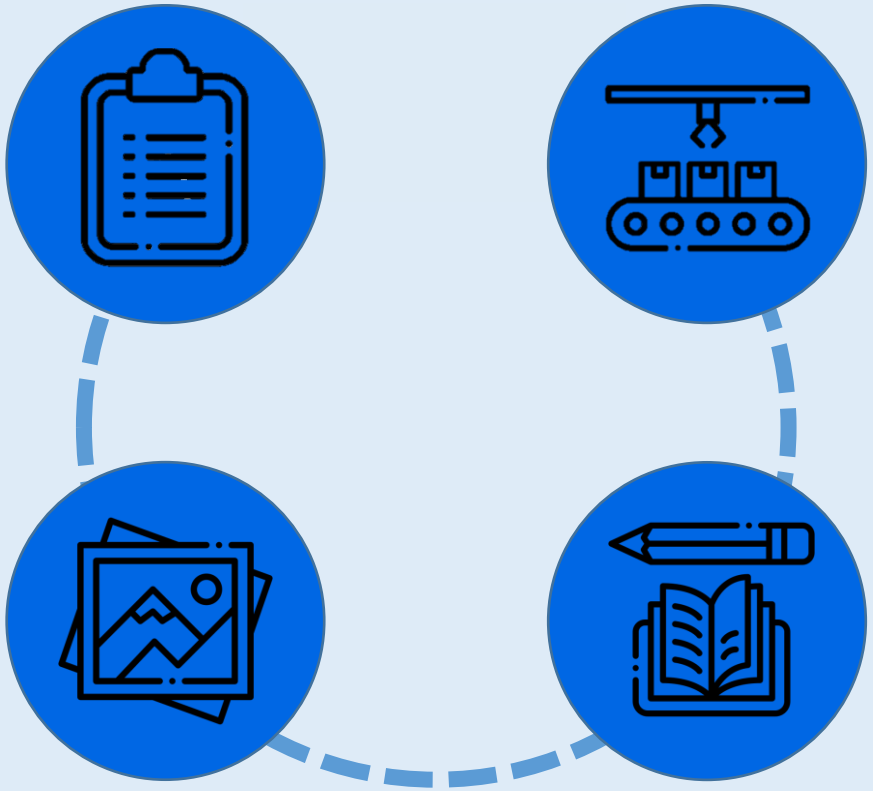
Процесс работы с нейронной сетью

1. Выбор метода локализации

Выбираем детекцию, так как нам достаточно знать лишь местонахождение объекта: форма, размер не важны.

2. Подготовка данных

Ищем и размечиваем нужные фотографии для датасета.

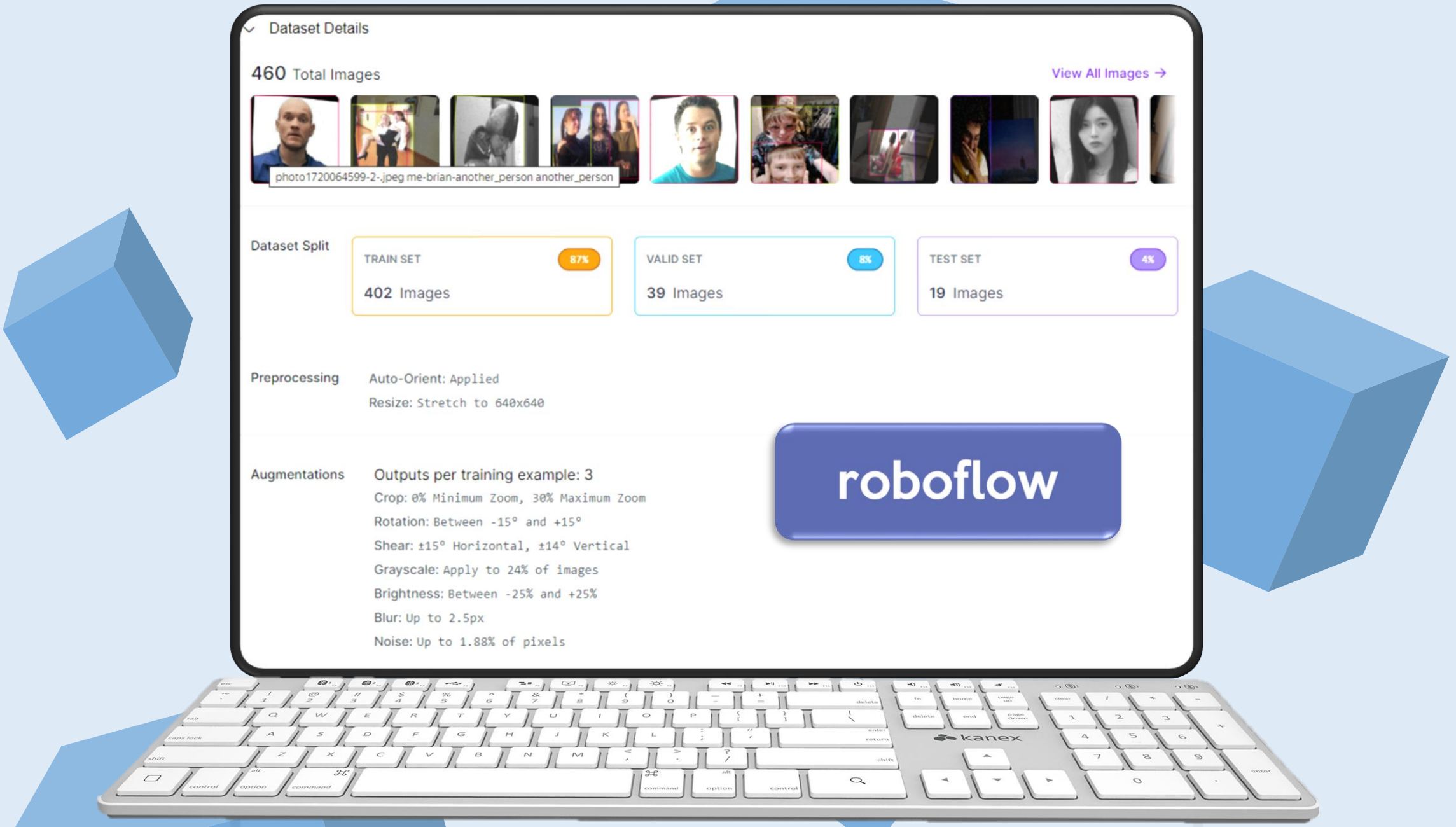


3. Аугментация

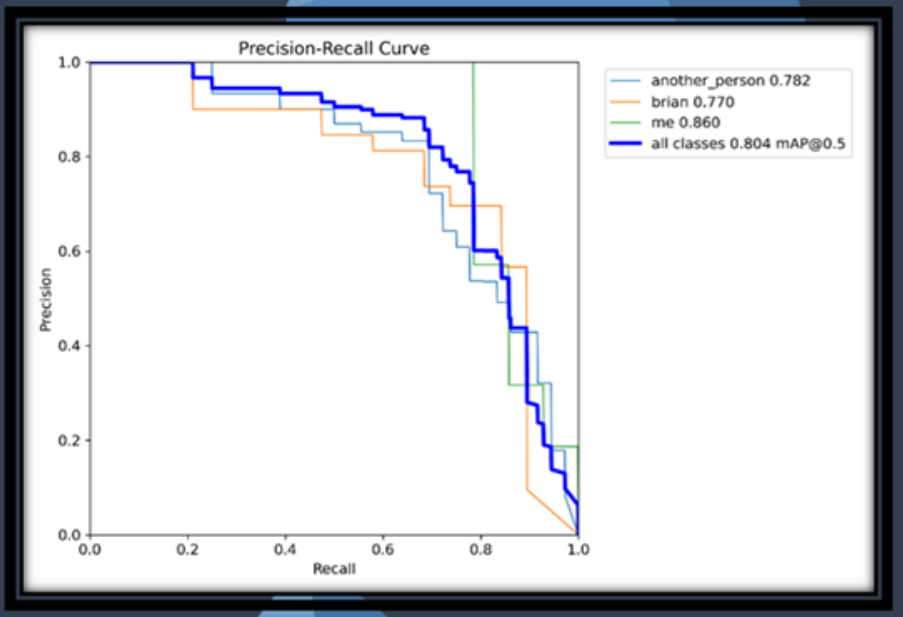
Создаем дополнительные обучающие данные.

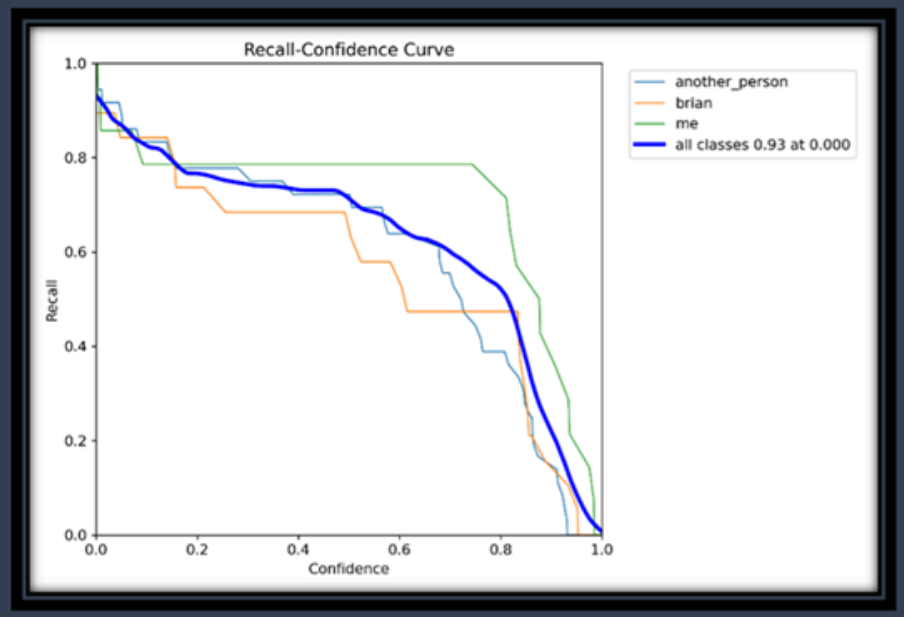
4. Обучение

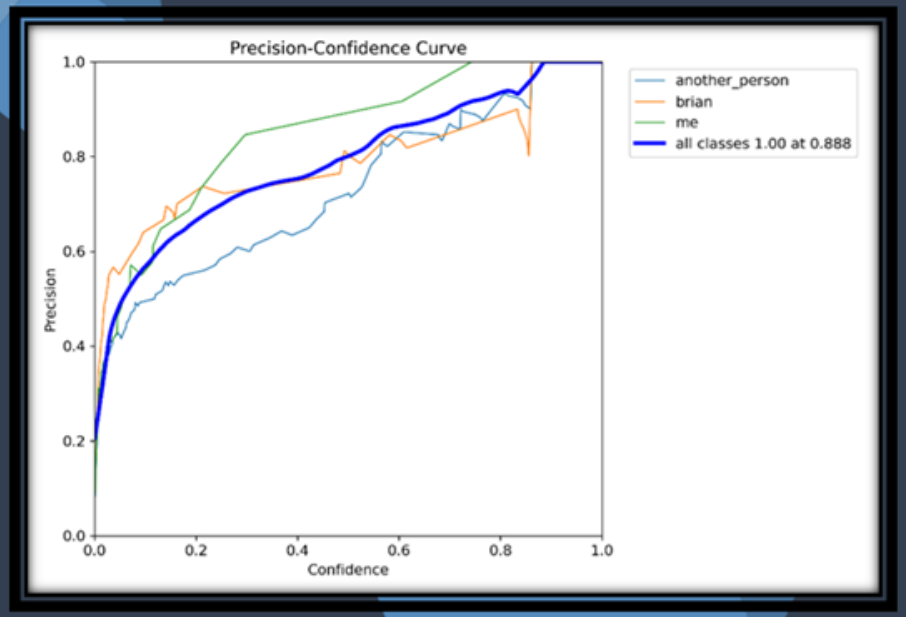
Загрузка датасета и вычислительные эксперименты.
Тренировка, валидация и тестирование модели архитектуры YOLOv8.

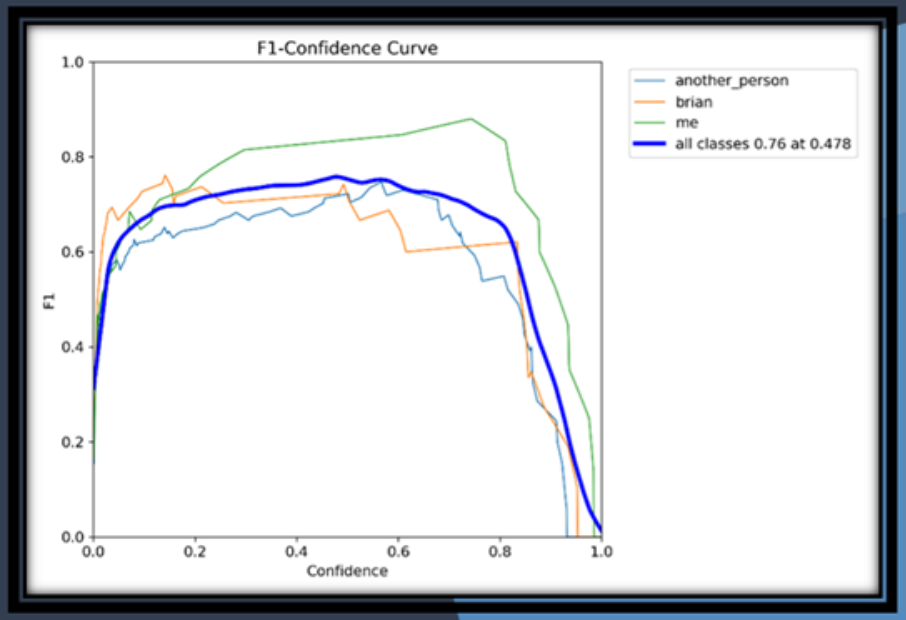

Dataset Details
460 Total Images
View All Images →
photo1720064599-2-.jpeg me-brian-another_person another_person
Dataset Split
TRAIN SET
87%
402 Images
VALID SET
8%
39 Images
TEST SET
4%
19 Images
Preprocessing
Auto-Orient: Applied
Resize: Stretch to 640x640
Augmentations
Outputs per training example: 3
Crop: 0% Minimum Zoom, 30% Maximum Zoom
Rotation: Between -15° and +15°
Shear: ±15° Horizontal, ±14° Vertical
Grayscale: Apply to 24% of images
Brightness: Between -25% and +25%
Blur: Up to 2.5px
Noise: Up to 1.88% of pixels
roboflow
kanex

50 эпох

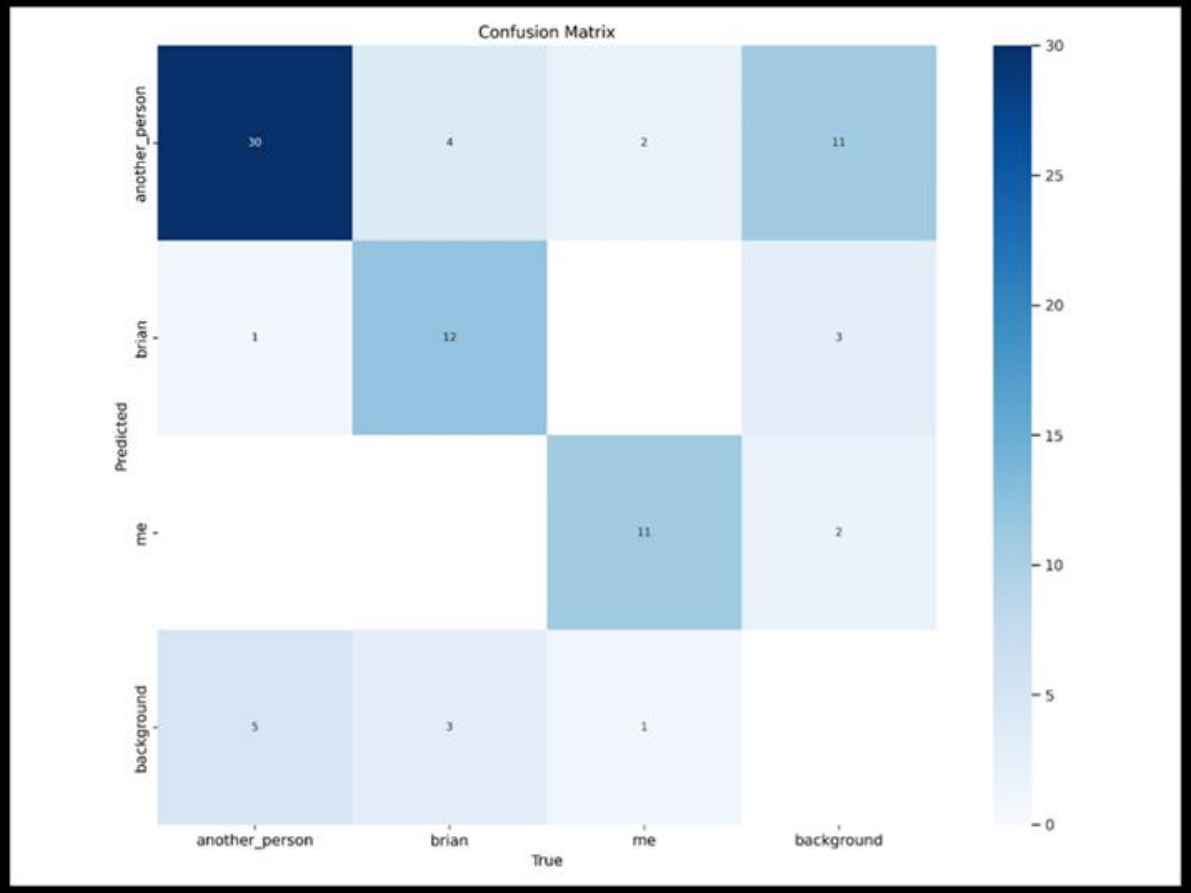

Precision-Recall Curve
another_person 0.782
brian 0.770
me 0.860
all classes 0.804 mAP@0.5
Precision
Recall


Recall-Confidence Curve
another_person
brian
me
all classes 0.93 at 0.000
Recall
Confidence

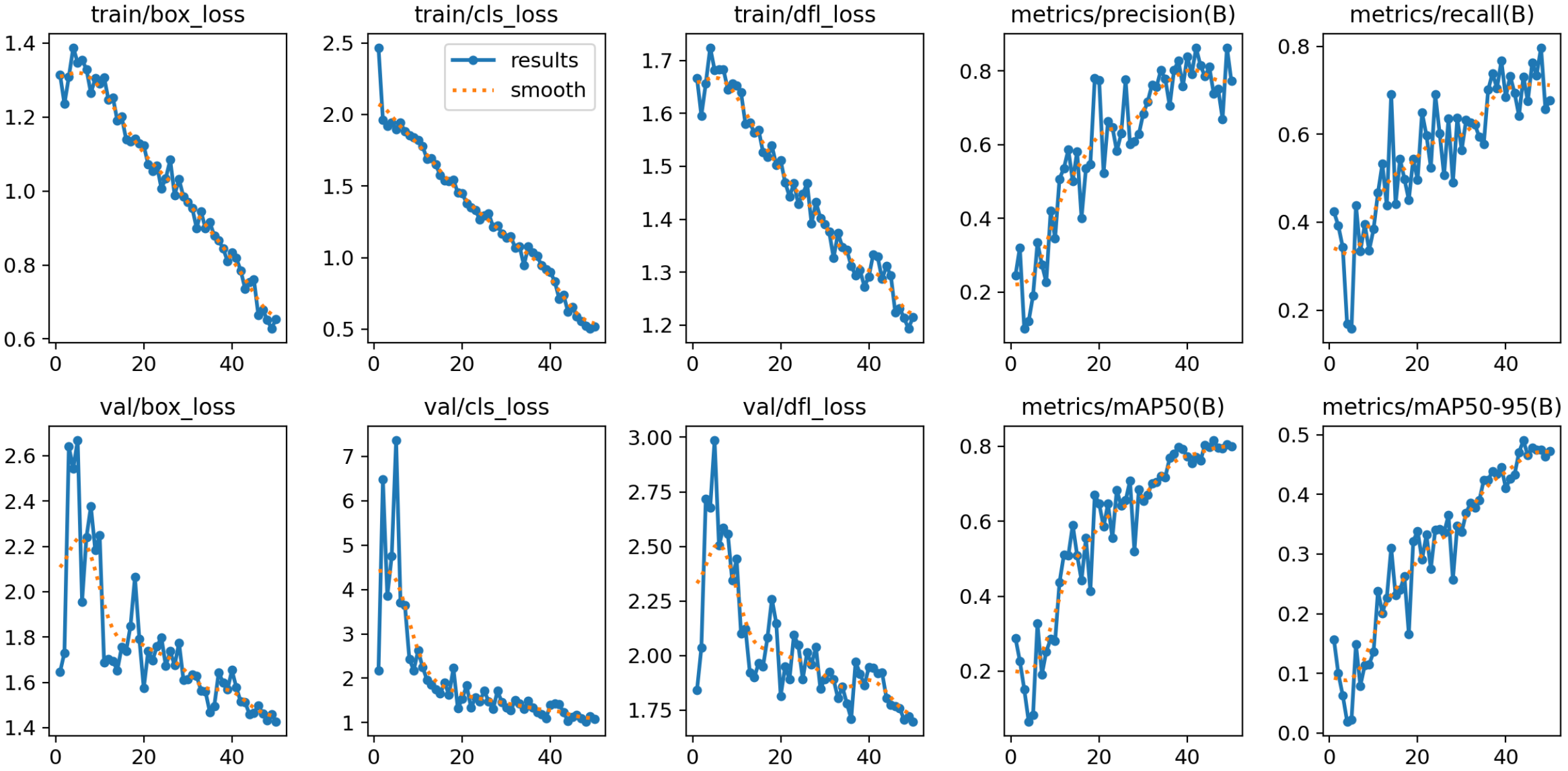

Precision-Confidence Curve
another_person
brian
me
all classes 1.00 at 0.888
Precision
Confidence

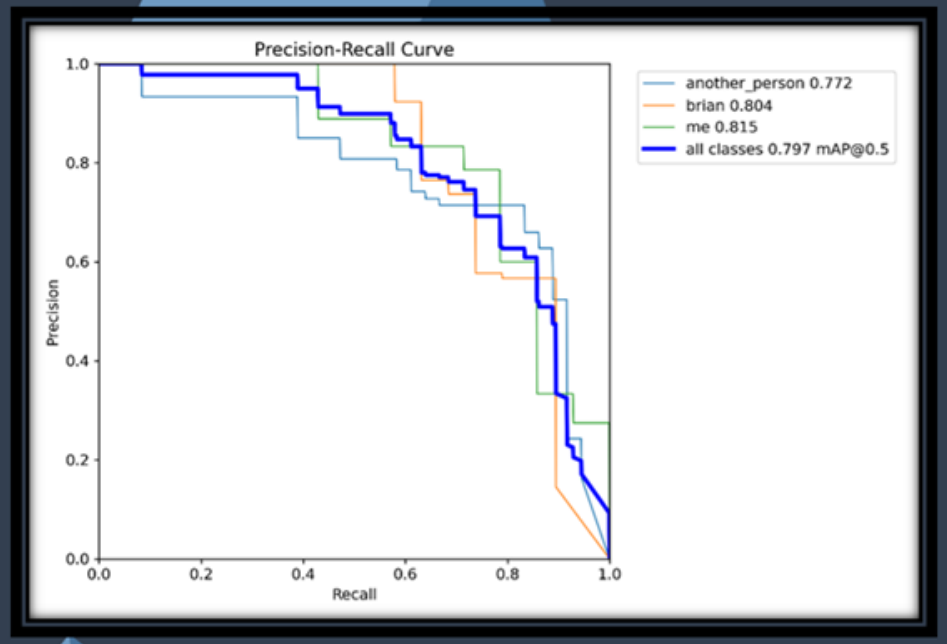

F1-Confidence Curve
another_person
brian
me
all classes 0.76 at 0.478
F1
Confidence

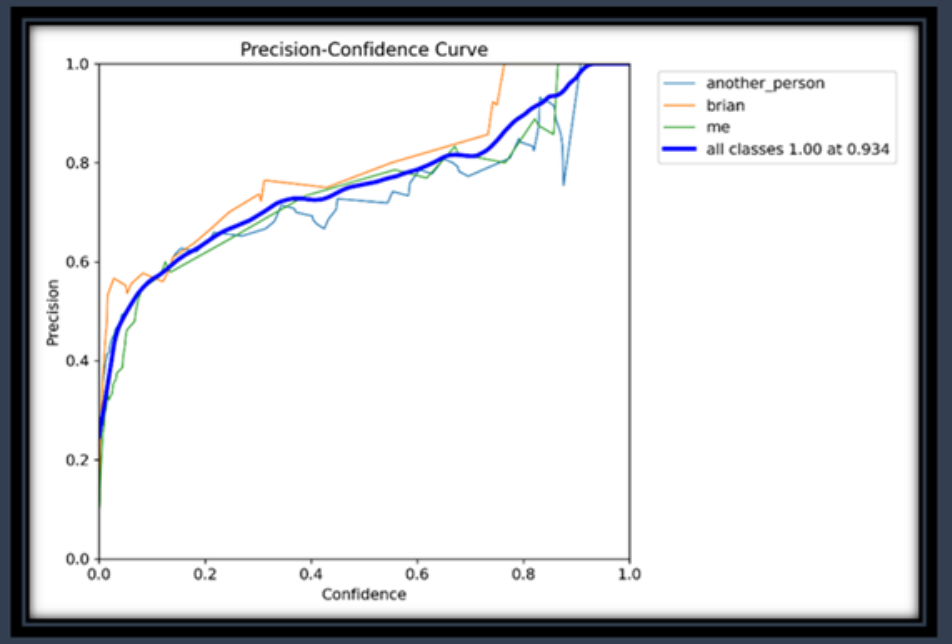
50 эпох

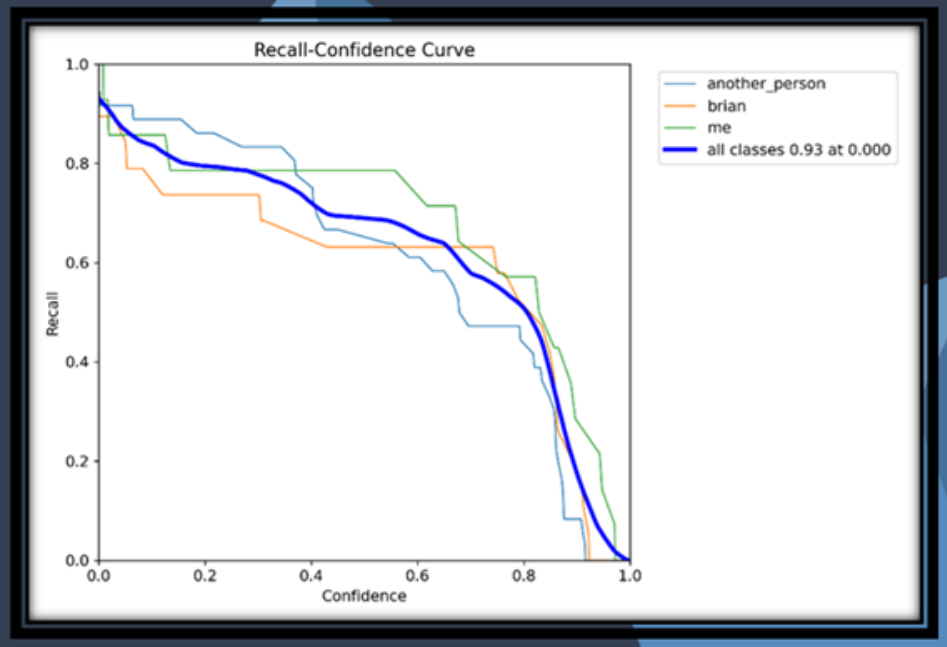

Confusion Matrix
another_person
brian
me
background
Predicted
True
30
4
2
11
1
12
3
11
2
5
3
1
30
25
20
15
10
5
0

another_person 0.5
another_person 0.8
another_person 0.7
another_person 0.6
another_person 0.9

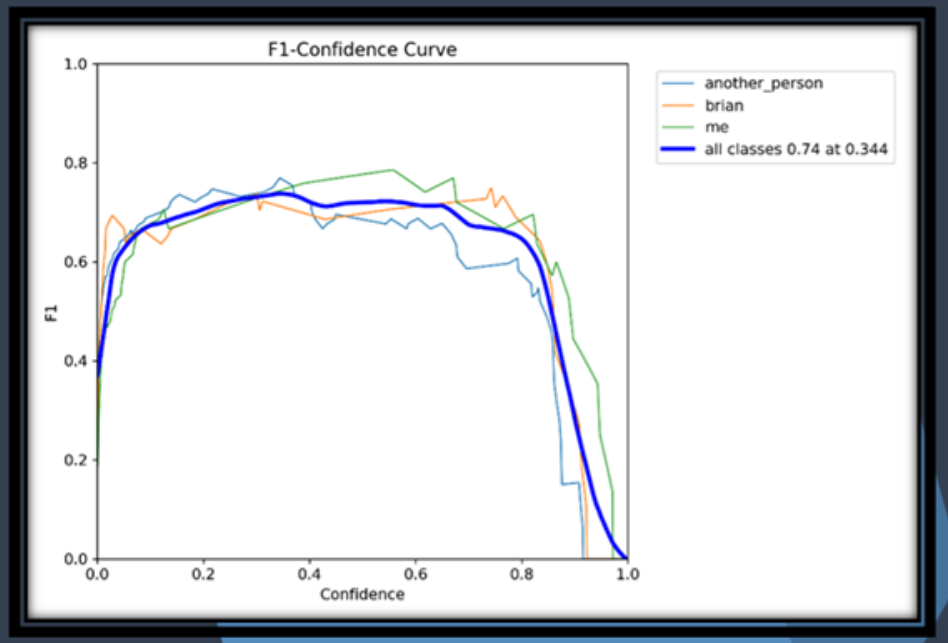

train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)


Precision-Recall Curve
Precision
Recall
another_person 0.772
brian 0.804
me 0.815
all classes 0.797 mAP@0.5

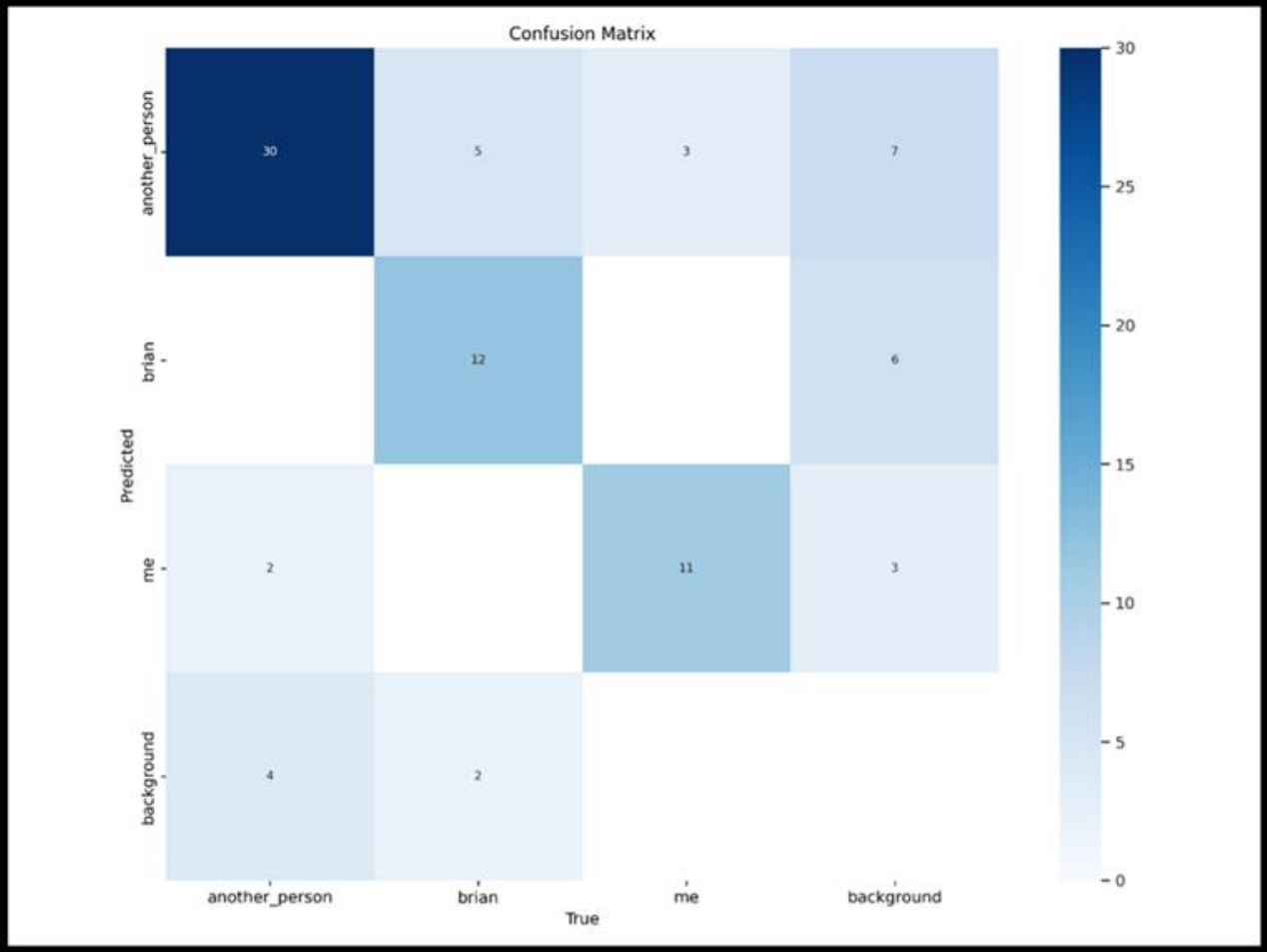

Precision-Confidence Curve
Precision
Confidence
another_person
brian
me
all classes 1.00 at 0.934


Recall-Confidence Curve
Recall
Confidence
another_person
brian
me
all classes 0.93 at 0.000

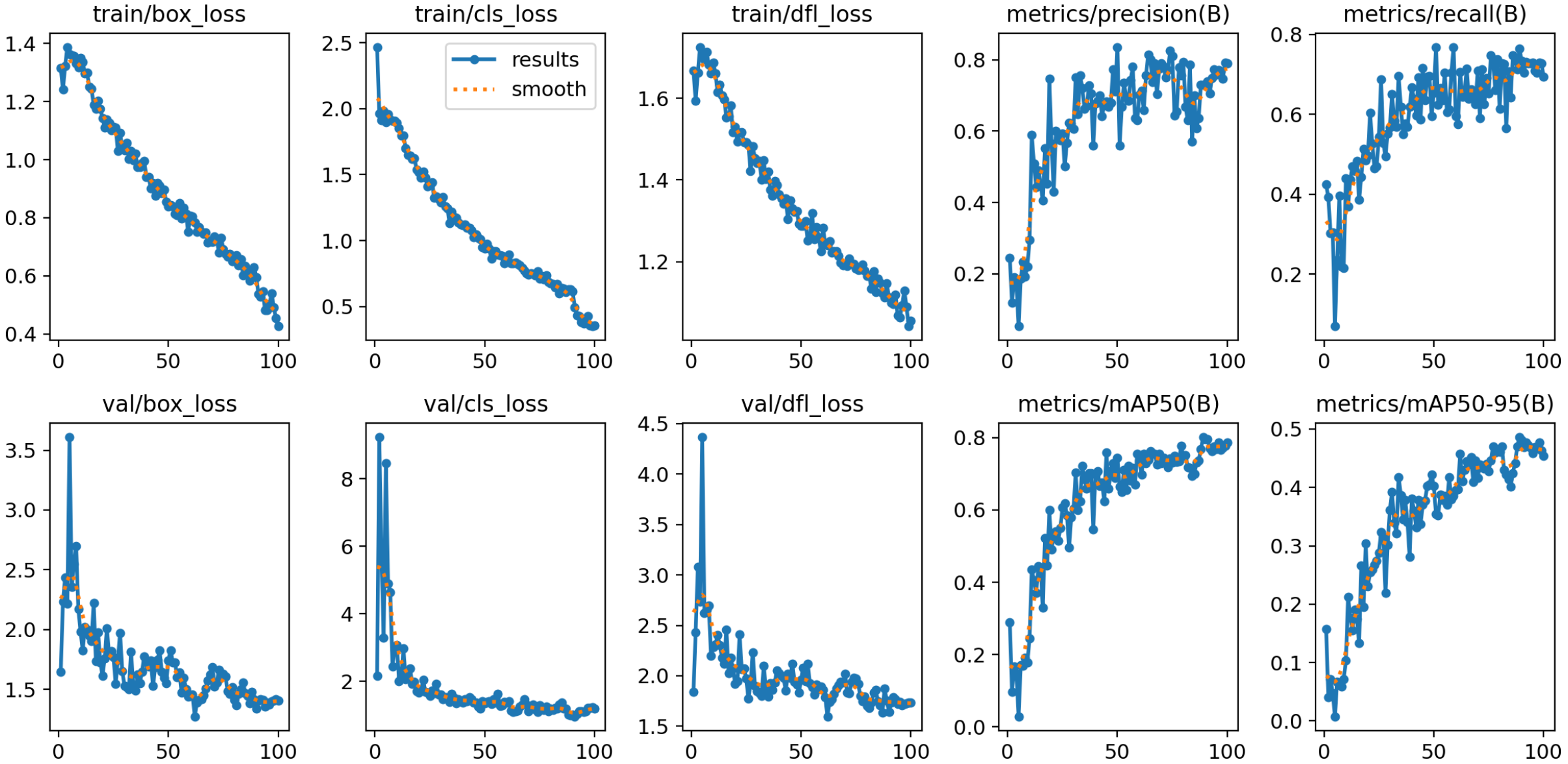
100 эпох

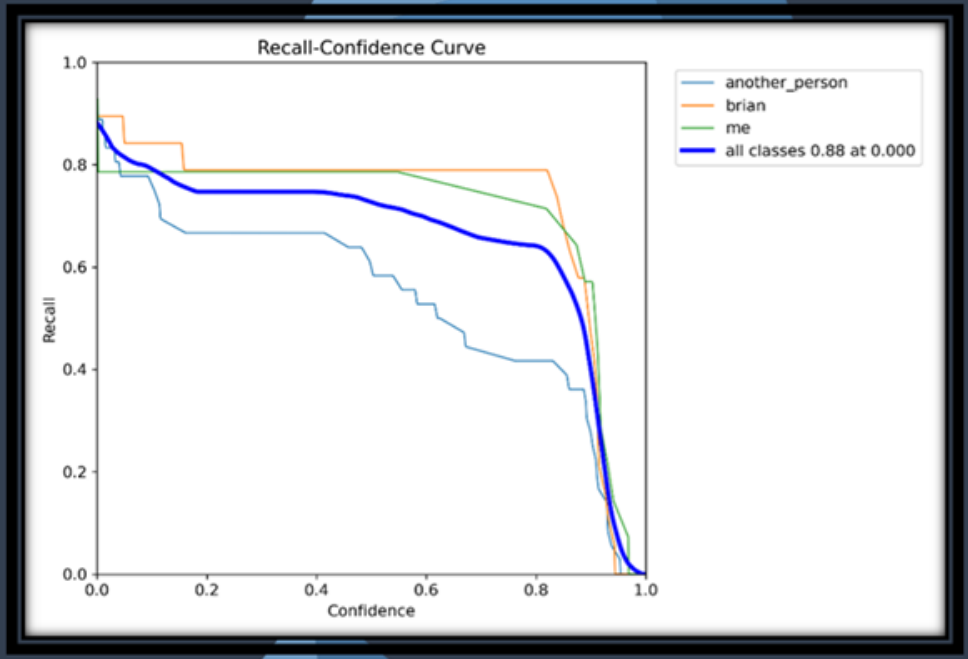

F1-Confidence Curve
F1
Confidence
another_person
brian
me
all classes 0.74 at 0.344

another_person 0.7
another_person 0.3
another_person 0.9
another_person 0.8
another_person 0.4
another_person 0.7
another_person 0.3
another_person 0.8
another_person 0.9

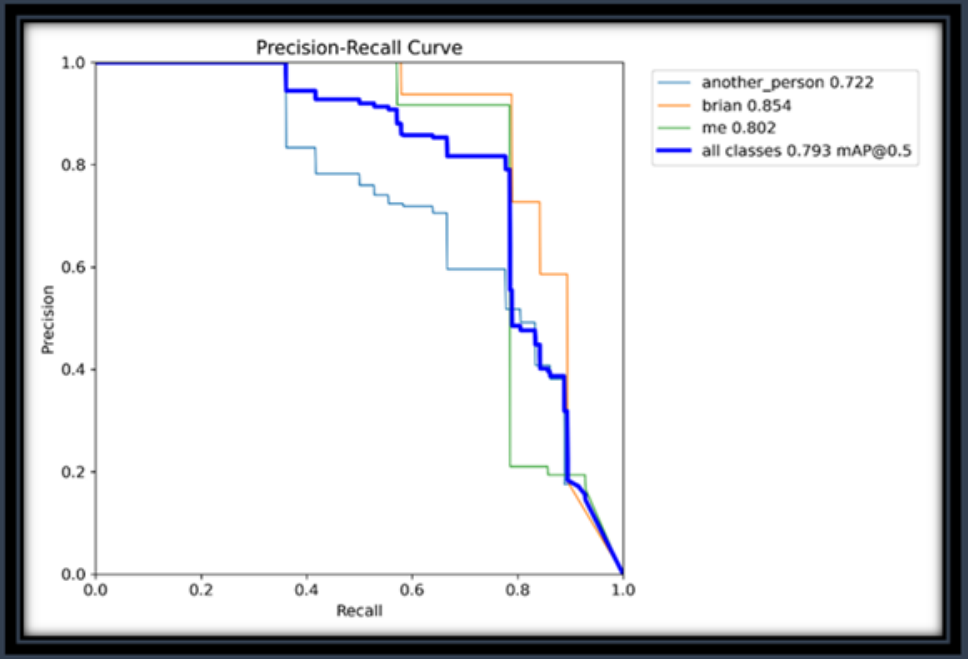

Confusion Matrix
another_person
brian
me
background
Predicted
True
30
5
3
7
12
6
2
11
3
4
2
0
5
10
15
20
25
30

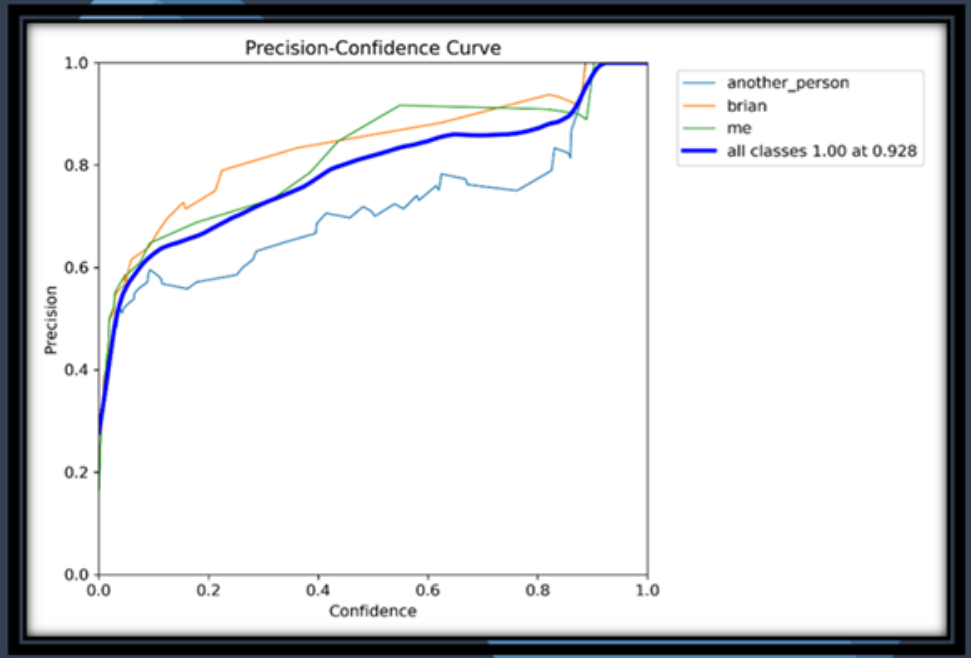
100 эпох

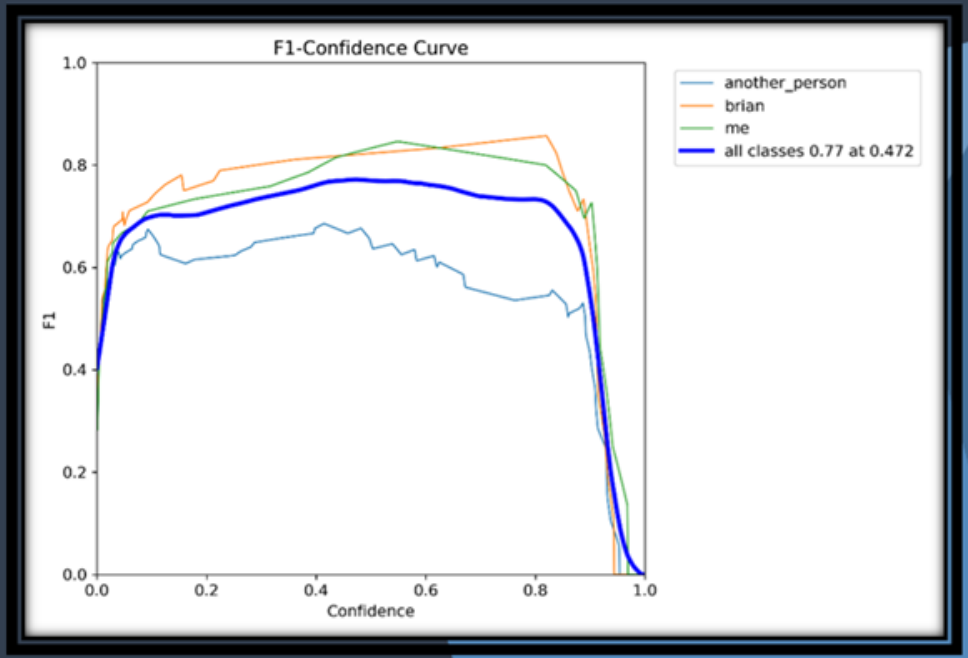

train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)

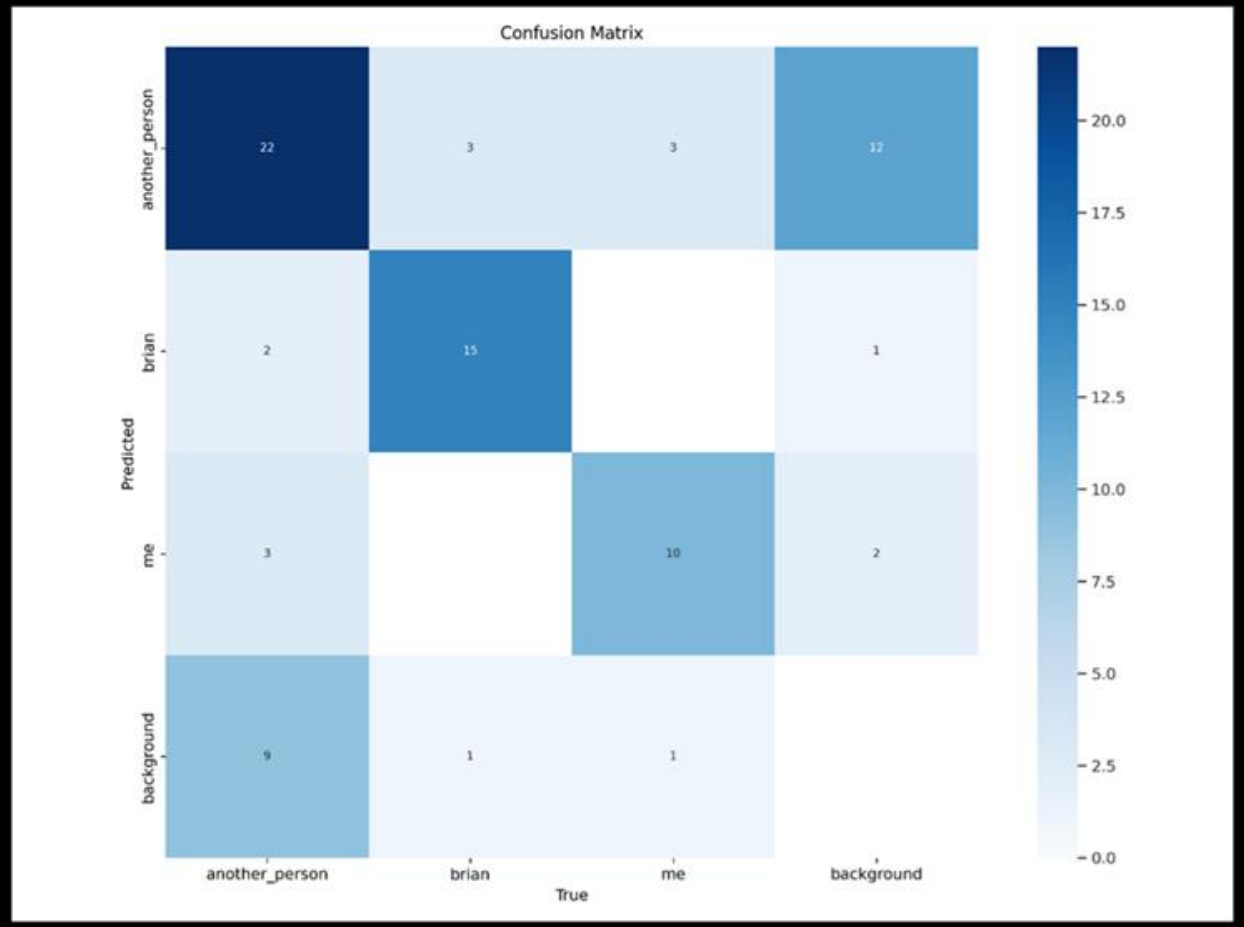

Recall-Confidence Curve
another_person
brian
me
all classes 0.88 at 0.000
Recall
Confidence
0.0
0.2
0.4
0.6
0.8
1.0

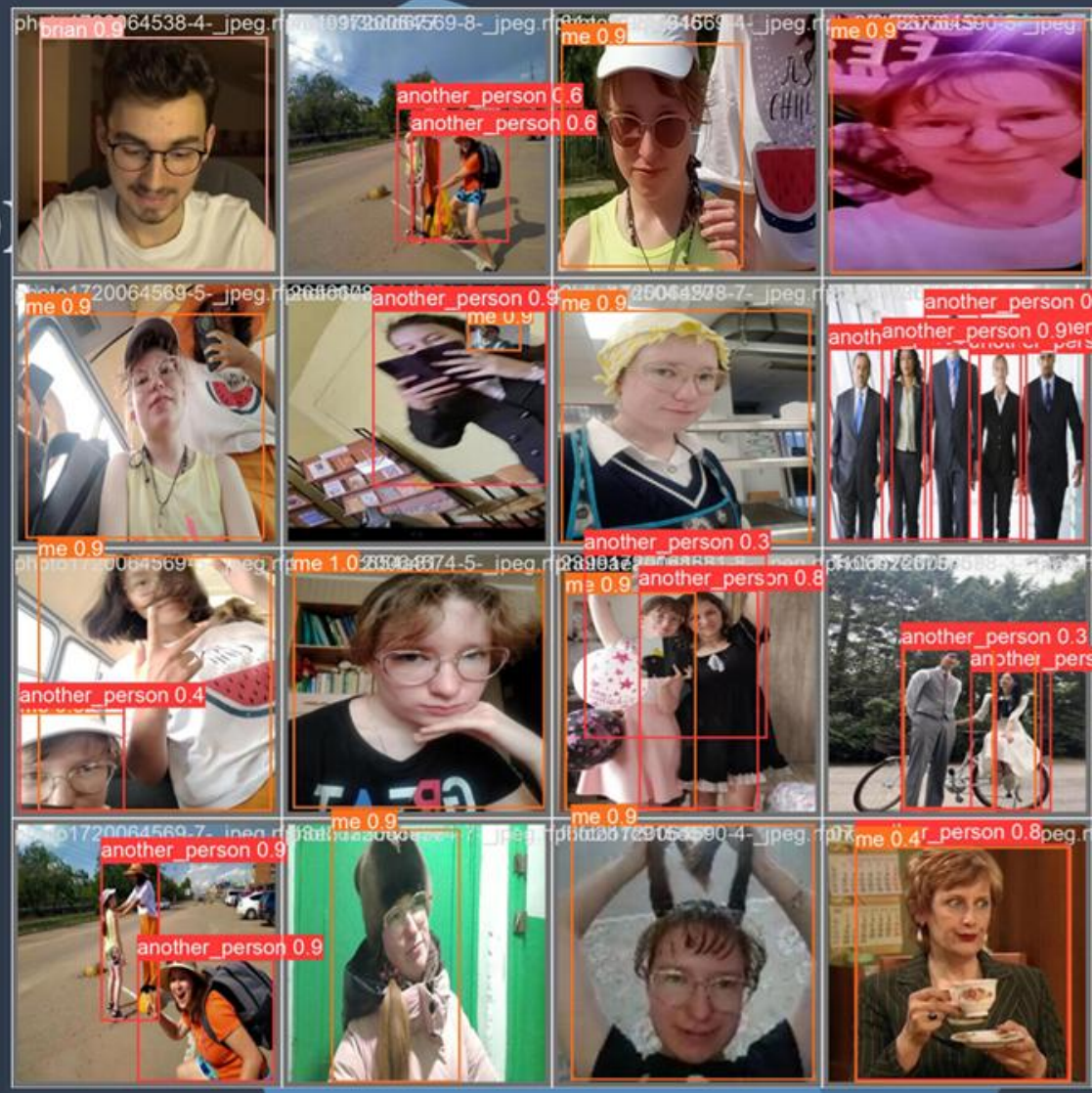
200 эпох

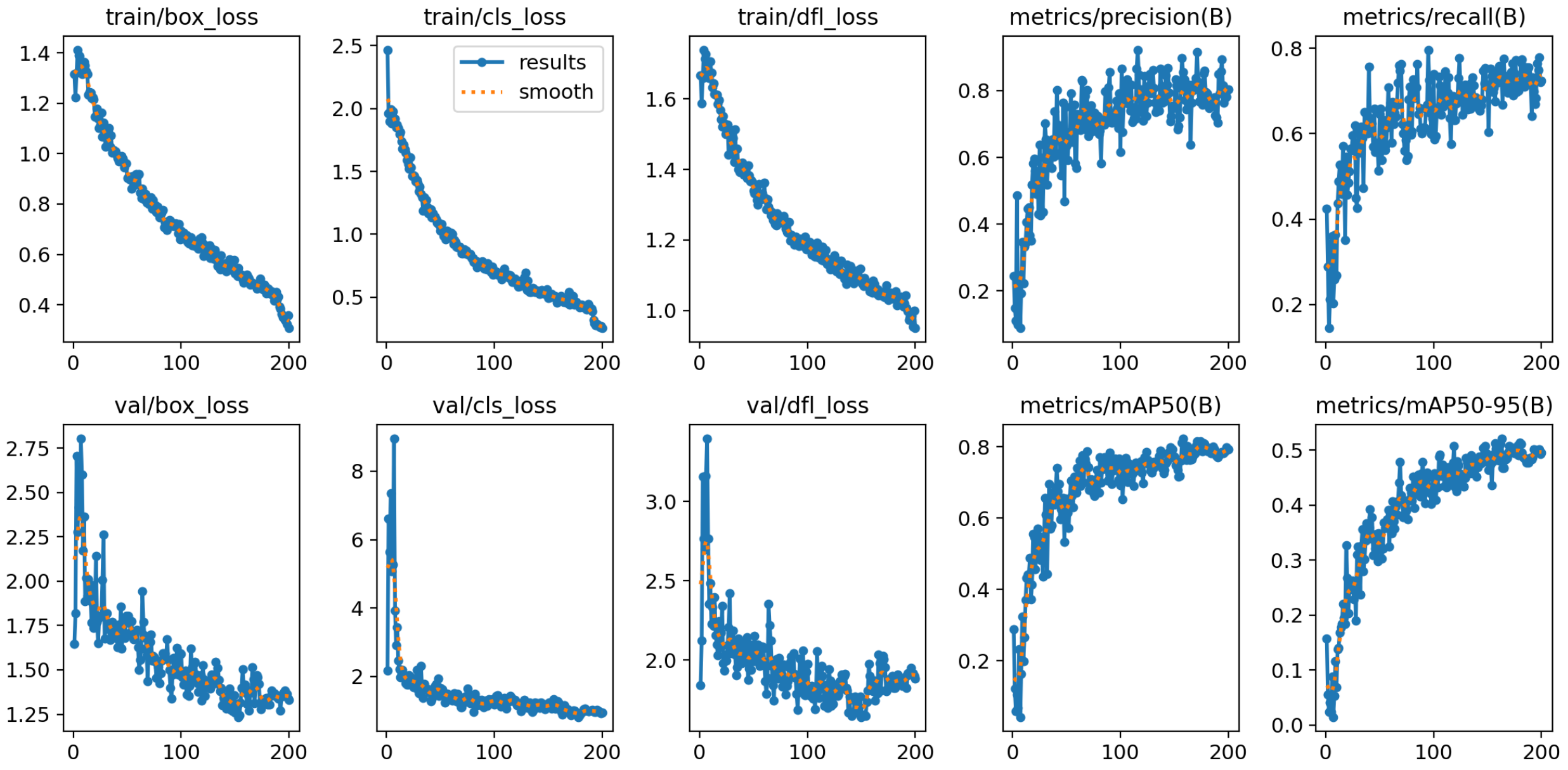

Precision-Recall Curve
another_person 0.722
brian 0.854
me 0.802
all classes 0.793 mAP@0.5
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0


Precision-Confidence Curve
another_person
brian
me
all classes 1.00 at 0.928
Precision
Confidence
0.0
0.2
0.4
0.6
0.8
1.0


F1-Confidence Curve
another_person
brian
me
all classes 0.77 at 0.472
F1
Confidence
0.0
0.2
0.4
0.6
0.8
1.0


Confusion Matrix
another_person
brian
me
background
Predicted
True
22
3
3
12
2
15
1
3
10
2
9
1
1
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0


brian 0.9
another_person 0.6
another_person 0.6
me 0.9
me 0.9
me 0.9
another_person 0.9
me 0.9
me 0.9
another_person 0.9
another_person 0.3
me 0.9
another_person 0.4
another_person 0.8
another_person 0.3
another_person 0.9
another_person 0.9
me 0.9
me 0.9
me 0.4
another_person 0.8


train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)

Разработка мобильного приложения

Как работает приложение?

Демонстрация!

https://drive.google.com/drive/folders/1U_qJPnNm5mNoxzbYdEuT2BenbXqRykJ0?usp=drive_link

Вывод

Было разработано мобильное приложение с нейросетевой моделью распознавания людей

Разработанное приложение будет ускорять и облегчать работу по поиску людей

Я использовала новообретенные знания на практике

СПАСИБО ЗА ВНИМАНИЕ!