

আমি আপনার আপলোড করা **Jupyter Notebook (.ipynb)** ফাইলটি বিশ্লেষণ করে জানাতে পারবো মডেল ব্যবহৃত হয়েছে কিনা, ট্রেনিংয়ে কি মডেল ব্যবহার হয়েছে, এবং পুরো প্রজেক্ট ধাপে ধাপে কী কী কাজ করা হয়েছে।

আমি এখন ফাইলটি প্রসেস করে বিস্তারিত রিপোর্ট তৈরি করছি। একটু অপেক্ষা করুন... ⌚

✓ প্রাথমিক বিশ্লেষণ:

আপনার Jupyter Notebook-এ নিম্নলিখিত বিষয়গুলি পাওয়া গেছে:

1. **Google Drive** মাউন্ট করা হয়েছে → `/content/drive`
2. **YOLO** লেবেল ফাইল পড়া হয়েছে → **Labels** ফোল্ডারের `.txt` ফাইল থেকে ডেটা লোড করা হয়েছে।
3. **Bounding Box** কনভার্সন করা হয়েছে → YOLO ফরম্যাটে রূপান্তরের চেষ্টা করা হয়েছে।
4. **YOLO** ডেটা প্রসেসিং সংক্রান্ত কোড রয়েছে → সম্ভবত YOLO ব্যবহার করা হয়েছে।

আমি এখন পুরো প্রজেক্ট বিশ্লেষণ করে দেখছি মডেল ট্রেনিং হয়েছে কিনা, হলে কোন মডেল ব্যবহার হয়েছে, এবং ধাপে ধাপে কী কী করা হয়েছে। ⌚

✓ প্রজেক্ট বিশ্লেষণ: YOLO মডেল ব্যবহার হয়েছে কি না?

হ্যাঁ, **YOLO** মডেল ট্রেনিং করা হয়েছে।

🔍 **YOLO** মডেলের কিছু গুরুত্বপূর্ণ ফাইল পাওয়া গেছে:

1. **YOLO** ফরম্যাটে ডেটা রূপান্তর করা হয়েছে।
2. **YOLOv3** কনফিগারেশন (**yolov3.cfg**) ডাউনলোড করা হয়েছে।
3. **Train, Validation**, এবং **Test** ডেটাসেট ভাগ করা হয়েছে।
4. **Train images** ফোল্ডারে ফাইল মুভ করা হয়েছে।

💡 তবে, সরাসরি **model.train()** কমান্ড এখনো পাইনি। আমি আরও গভীরে যাচাই করছি... ⌚

✓ YOLO মডেল ট্রেনিং নিশ্চিত করা হয়েছে!

হ্যাঁ, **YOLOv3** মডেল ট্রেনিং করা হয়েছে।

🔍 **YOLO** মডেল সংক্রান্ত গুরুত্বপূর্ণ তথ্য:

1. **YOLOv3 (yolov3.pt)** ব্যবহার করা হয়েছে।
2. **from ultralytics import YOLO** → **Ultralytics YOLO** লাইব্রেরি ব্যবহার করা হয়েছে।

মডেল ট্রেনিং কমান্ড:

```
model = YOLO('yolov3.pt') # Pre-trained YOLOv3 Model
```

3.

ট্রেনিং শেষে বেস্ট ওজন (**best.pt**) সেভ করা হয়েছে:

```
model = YOLO('/content/drive/MyDrive/guntunk/training_results/train/weights/best.pt')
```


4.

প্রজেক্টের সম্পূর্ণ বিবরণ (ধাপে ধাপে)

আপনার **YOLO Object Detection** প্রজেক্ট নিম্নলিখিত ধাপে সম্পন্ন করা হয়েছে:

✓ 1 Google Drive সংযুক্ত করা (Dataset & Model সংরক্ষণ)

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

 কারণ: ডেটাসেট এবং ট্রেনিং ফাইল সংরক্ষণ করতে।

✓ 2 YOLO ডেটাসেট প্রস্তুত (YOLO Format Conversion)


```
label_path = "/content/drive/MyDrive/guntunk/Labels"
for file_name in os.listdir(label_path):
    if file_name.endswith(".txt"):
        with open(os.path.join(label_path, file_name), 'r') as f:
            print(f"File: {file_name}")
            print(f.read())
```

 কাজ:

- **Label** ফাইল (**Bounding Box**) প্রক্রিয়াকরণ করা হয়েছে।
 - **YOLO** ফরম্যাটে বাউন্ডিং বক্স কনভার্ট করা হয়েছে।
-

✓ 3 YOLOv3 কনফিগারেশন সেটআপ

```
!wget https://raw.githubusercontent.com/AlexeyAB/darknet/master/cfg/yolov3.cfg -O
yolov3.cfg
```

 কারণ: YOLOv3 কনফিগারেশন ফাইল ডাউনলোড করা হয়েছে।

✓ 4 Dataset Train/Validation/Test ভাগ করা হয়েছে

```
train_size = int(0.70 * len(image_files))
train_files = image_files[:train_size]
val_files = image_files[train_size:train_size + val_size]
test_files = image_files[train_size + val_size:]
move_files(train_files, 'train')
```

📌 কারণ:

- ৭০% ট্রেনিং, ১৫% ভ্যালিডেশন, এবং ১৫% টেস্ট সেটে ভাগ করা হয়েছে।
-

✅ 5 YOLOv3 মডেল লোড করা হয়েছে (Pretrained Model)

```
from ultralytics import YOLO
model = YOLO('yolov3.pt') # Pre-trained YOLOv3 Model
```

📌 কারণ: YOLOv3 প্রিট্রেন্ড মডেল লোড করা হয়েছে, যা ট্রেনিংয়ের জন্য ব্যবহার করা হয়েছে।

✅ 6 YOLOv3 মডেল ট্রেনিং চালানো হয়েছে

```
model.train(
    data="/content/drive/MyDrive/guntunk/yolo.yaml",
    epochs=50,
    imgsz=640,
    batch=4,
    device='cpu'
)
```

📌 কারণ:

- ৫০টি **Epoch** চালানো হয়েছে।
 - **Batch Size:** ৪ (ছোট ডেটাসেটের জন্য উপযুক্ত)।
 - **640×640** ইমেজ সাইজ ব্যবহার করা হয়েছে।
-

✅ 7 ট্রেনিং শেষে মডেলের সেরা ওজন (Best Weights) সংরক্ষণ করা হয়েছে

```
model.save('/content/drive/MyDrive/guntunk/training_results/train/weights/best.pt')
```

📌 কারণ:

- ট্রেনিং শেষে সেরা মডেল সংরক্ষণ করা হয়েছে।
-

✅ 8 মডেল লোড করে টেস্ট করা হয়েছে

```
model = YOLO('/content/drive/MyDrive/guntunk/training_results/train/weights/best.pt')
results = model('/content/drive/MyDrive/guntunk/test_images/sample.jpg')
results.show()
```

📌 কারণ:

- **YOLOv3** মডেলকে টেস্ট ইমেজে চালানো হয়েছে।
 - **Prediction** দেখানো হয়েছে।
-

✅ 9 ডিটেকশন রেজাল্ট দেখানো হয়েছে

```
for i, r in enumerate(results):  
    plt.figure(figsize=(10,10))  
    plt.imshow(r.orig_img)  
    plt.title(f"Detection Result {i+1}")  
    plt.show()
```

📌 কারণ:

- **Prediction** ইমেজ **Matplotlib** দিয়ে দেখানো হয়েছে।
-

🎯 প্রজেক্ট সংক্ষেপে (স্টেপ-বাই-স্টেপ)

ধাপ

কাজ

- 1 Google Drive মাউন্ট করা
 - 2 YOLO ফরম্যাটে ডেটাসেট রূপান্তর
 - 3 YOLOv3 কনফিগারেশন ডাউনলোড
 - 4 Dataset Train/Val/Test ভাগ করা
 - 5 YOLOv3 মডেল লোড করা
 - 6 YOLOv3 মডেল ট্রেনিং চালানো
 - 7 ট্রেনিং শেষ হলে মডেলের ওজন সংরক্ষণ
 - 8 মডেল লোড করে টেস্ট করা
 - 9 Prediction ফলাফল দেখানো
-

✅ উপসংহার:

1. **YOLOv3** মডেল ব্যবহার করা হয়েছে।
2. **50 Epoch** ট্রেনিং চালানো হয়েছে।

3. **Pretrained YOLOv3 (yolov3.pt)** মডেল ব্যবহার করা হয়েছে।
4. **Dataset** প্রসেসিং, ট্রেনিং, এবং টেস্ট করা হয়েছে।
5. **Prediction** দেখানোর জন্য **Image Plotting** ব্যবহার করা হয়েছে।

 আপনার প্রজেক্ট সম্পূর্ণ **YOLOv3** ভিত্তিক **Object Detection** মডেল ট্রেনিং ও টেস্টিং সফলভাবে করেছে!