# تقرير تدريبي لنموذج كشف الهواتف باستخدام YOLOv8

## 📌 معلومات عامة:

اسم التجربة: yolov8\_exp12

عدد التكرارات (Epochs): 60

مدة التدريب: 2.839 ساعات

نوع النموذج: YOLOv8n (النسخة الخفيفة)

لغة البرمجة: Python 3.12.2

المكتبات المستخدمة: Ultralytics 8.3.105، Torch 2.6.0 + CUDA 11.8

المعالج الرسومي: NVIDIA GeForce GTX 1050 (2GB VRAM)

## 📈 نتائج التحقق (Validation):

عدد الصور: 306

عدد الكائنات: 298

الدقة (Precision): 0.966

الاسترجاع (Recall): 0.956

mAP@0.5: 0.979

mAP@0.5:0.95: 0.759

## ⚙️ مواصفات النموذج النهائي:

عدد الطبقات (Layers): 72

عدد المعاملات (Parameters): 3,006,038

عدد عمليات الفلوبس (GFLOPs): 8.1

حجم النموذج: 6.3MB (تم تجريد Optimizer)

## 🧪 سرعة المعالجة (Per Image):

Preprocess: 0.9ms

Inference: 12.3ms

Loss: 0.0ms

Postprocess: 1.3ms

## 🧰 إعدادات التدريب:

حجم الصور: 640×640

حجم الدفعة: 2

عدد المعالجات: 2

الجهاز المستخدم: GPU (CUDA)

ملف البيانات: data.yaml

مجلد النتائج: themodel/yolov8\_exp12

## 🧪 تحسين البيانات (Augmentation):

تم تحسين الصور باستخدام مكتبة Albumentations:

- LongestMaxSize + PadIfNeeded لضمان أبعاد ثابتة.

- HorizontalFlip: الانعكاس الأفقي.

- RandomBrightnessContrast: تغيير الإضاءة والتباين.

- Blur و MotionBlur: محاكاة الاهتزاز.

- CLAHE: تحسين التباين المحلي.

- RandomGamma: تعديل جاما لتحسين التوزيع اللوني.

تم إنشاء مجلد aug\_test يحوي الصور المعدّلة.

ملف JSON جديد annotations.json يحتوي على الصور المشروحة الجديدة بطريقة COCO.

## ✅ الاستنتاجات:

جودة النتائج ممتازة، ومؤشرات الأداء عالية.

التحسينات ساعدت على تقليل احتمالية التحيّز وزادت تنوع البيانات.

مناسب تمامًا للتطبيقات العملية مثل المراقبة الذكية أو كشف الغش في الامتحانات.