

المنهج المدمج المحدث: بايثون والذكاء الاصطناعي للتشخيص الطبي

عشرة أيام عملية — من سطر الأوامر إلى تدريب نموذج YOLO
سطر الأوامر • Miniconda • GPU TensorFlow/PyTorch • Colab • YOLO

د. حسام صلاح مهدي
رئيس قسم الذكاء الاصطناعي — جامعة المصطفى
Dr. Husam Salah Mahdi — Head of AI Department, Al-Mustafa University



باب 1

الأهداف والمنهجية

الجهة المستفيدة: مديرية التشخيص — وزارة الصحة العراقية. مدة البرنامج: عشرة أيام بمعدل 90 دقيقة يومياً بإيقاع ثابت: 20 دقيقة نظرية، ثم 40 دقيقة تطبيقاً موجهاً، ثم 30 دقيقة تحدياً عملياً. المستوى: مبتدئ متأسس وصولاً إلى تدريب نموذج رؤية حاسوبية حقيقي. الشرائح والوثائق المرافقة: شرائح الأساسيات (المفاهيم، 63 شريحة)، وشرائح الأدوات العملية (الأوامر، 64 شريحة)، ودفتر التدريب العملي اليومي، والتقرير البحثي المحكم لمنصة Roboflow.

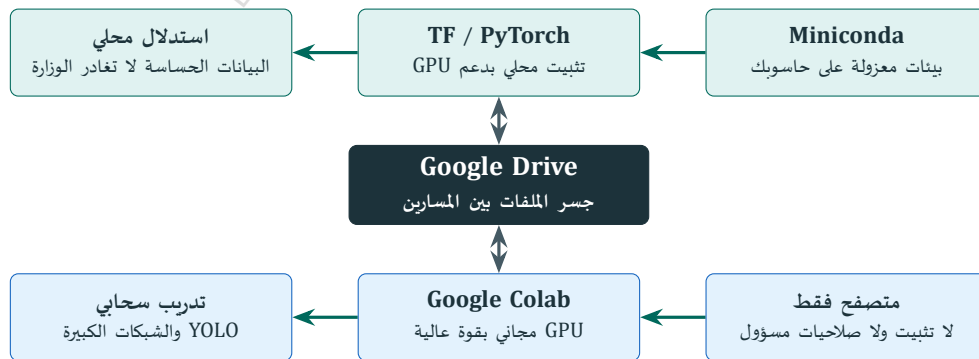
1.1 الأهداف الاستراتيجية

✓ الأهداف الستة للدورة

1. بناء أساس برمجي رصين بلغة Python للكودر الصحية والتقنية.
2. إتقان سطر الأوامر (CLI) بوصفه أداة المحترفين لإدارة أدوات الذكاء الاصطناعي.
3. تهيئة بيئات عمل معزولة محلياً عبر Miniconda وسحابياً عبر Google Colab.
4. تثبيت TensorFlow وPyTorch بدعم المعالج الرسومي (GPU) تثبيتاً صحيحاً موثقاً.
5. تدريب نموذج كشف الأجسام YOLO على صور مجهرية بسطر أوامر واحد — سحابياً ومحلياً.
6. إتقان دورة التكامل: تجهيز البيانات محلياً، فتدريب سحابي مجاني، فتشغيل النموذج محلياً داخل الوزارة.

1.2 مساران متوازيان وجسر بينهما

تسير الدورة على مسارين متوازيين لا نختار أحدهما بل نتقهما معاً: مسار محلي على حواسيب الوزارة ببيئات Miniconda المعزولة، ومسار سحابي عبر Google Colab بمعالجه الرسومي المجاني؛ ويجسر بينهما Google Drive لنقل البيانات والنماذج.



شكل 1.1: المساران المتوازيان طوال الدورة والجسر الدائم بينهما.

✖ خطأ شائع — القاعدة الذهبية للحكومة

البيانات الحساسة الحقيقية لا تغادر أجهزة الوزارة أبداً؛ التدريب السحابي يجري حصراً على بيانات اصطناعية أو مموهة بالكامل (Anonymized). ويضاف إليها تحذير منصة Roboflow: كل ما يرفع إلى حساب مجاني ينشر علناً — فلا ترفع بيانات مرضى إلى الحسابات المجانية إطلاقاً.

1.3 إيقاع الجلسة الثابت

تتبع كل جلسة الإيقاع نفسه، فيختفي الشكل بعد اليوم الثالث وينصرف الانتباه كله إلى المحتوى:

الدقائق	الجزء	المحتوى والأنشطة
0-20	نظرية	المفهوم الطبي والتقني لليوم، مرسخاً في سيناريو تشخيصي قبل ظهور أي شيفرة.
20-60	تطبيق موجه	برمجة وأوامر حية يقودها المدرب؛ كل متدرب ينفذ بيده ويرى نتيجته — وفي أيام الأدوات (2، 4، 7، 8، 9) تكون الكتلة كلها أوامر تنفذ حرفياً.
60-90	تحليل عملي	مسألة مفتوحة تحل في أزواج وتناقش نتيجتها بوصفها معطى تشخيصياً.

باب 2

جدول الأيام العشرة

يلخص الجدول الآتي قوس الدورة كاملاً: الأسبوع الأول للتأسيس والبنية التحتية، والثاني للنماذج والتدريب الحقيقي، وذروته اليوم التاسع بتدريب YOLO.

اليوم	العنوان	النتائج العملية للموسم
1	خارطة الطريق وجولة الأدوات	أول دفتر Colab يعمل + فهم المنظومة كاملة
2	سطر الأوامر و Miniconda	بيئة medai معزولة منشأة ومفعلة على حاسوب كل متدرب
3	أساسيات بايثون للبيانات الطبية	دوال تحسب مؤشرات حيوية على سجل مريض (قاموس)
4	Google Colab والجسر مع الحاسوب	جلسة GPU مفعلة + Drive مربوط + ملف مضغوط رفع وفك
5	NumPy وPandas: معالجة السجلات	تنظيف ملف مختبري ملوث واستخراج إحصاءات المرضى
6	الرسوم البيانية وهندسة الميزات	مخططات المجالات المرجعية + عمود BMI وتصنيف HbA1c
7	TensorFlow: أول شبكة عصبية	نموذج Keras مدرب على سجل السكري في Colab
8	PyTorch: المصفوفات وتسريع GPU	PyTorch مثبت محلياً + مقارنة سرعة CPU/GPU
9	تدريب YOLO لكشف خلايا الدم	نموذج best.pt مدرب سحابياً ويعمل محلياً
10	المشروع الختامي وخارطة الطريق	مشروع فريق كامل + تقييم + شهادات

جدول 2.1: الدورة في لمحة واحدة.

باب 3

الخطة اليومية المفصلة

يعرض هذا الفصل كل يوم بمكوناته الثلاثة، مع الأوامر الجاهزة للتنفيذ الحرفي. النسخ الورقية للمتدربين في دفتر التدريب العملي المرافق.

3.1 اليوم 1 — خارطة الطريق: الذكاء الاصطناعي في التشخيص وجولة الأدوات

- نظرية: التحول من الحدس إلى القرار المستند إلى البيانات؛ حالات واقعية (فرز الأورام، عد الخلايا، إنذار وبائي مبكر)؛ ولماذا بايثون هي اللغة المهيمنة طيباً.
- تطبيق: فتح أول دفتر في colab.research.google.com، تنفيذ خلايا حسابية، وجولة مصورة على المنظومة كاملة: CLI Conda، GPU Colab، YOLO.
- تحدي: كل متدرب يعدل خلية ويشغلها ويصف بجملته واحدة أين يخدم الذكاء الاصطناعي قسمه.

3.2 اليوم 2 — سطر الأوامر Miniconda: بناء البيئة المحلية

مفهوم اليوم: تعارض الاعتماديات (Dependency Hell) وحله البنيوي بالبيئات المعزولة. التثبيت على ويندوز 11 بأحد خيارين ثم بناء بيئة الدورة:

```
:: Option A - one command on Windows 10/11:
winget install -e --id Anaconda.Miniconda3
:: Option B - official installer (Just Me):
:: https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe

conda --version
conda create -n medai python=3.11 -y
conda activate medai
conda install numpy pandas matplotlib -y
```

شيفرة 3.1: تثبيت Miniconda وإنشاء بيئة الدورة.

بطاقة التحكم الكامل (من البطاقة الرسمية [conda-cheatsheet](#)):

```
conda info --envs           :: all environments (* = active)
conda list -n medai         :: what is inside an env
conda install -n medai openpyxl -y  :: install into a named env
conda update --all -n medai      :: update everything in it
conda create --clone medai -n backup  :: clone an environment
conda rename -n backup medai_v2    :: rename
conda remove -n medai_v2 --all -y   :: delete completely
conda list -n medai --revisions    :: change history
conda install -n medai --revision 1  :: roll back!
conda export --from-history > environment.yml
```

```
conda env create -n medai --file environment.yml
conda clean --all :: housekeeping
```

شيفرة 3.2: التحكم الكامل بالبيئات: استعمال وتحديث واستنساخ وتراجع ومشاركة.

3.3 اليوم 3 — أساسيات بايثون للبيانات الطبية

- المتغيرات وأنواع البيانات الأربعة؛ سجل المريض بوصفه قاموساً (Dictionary)؛ فلسفة الدورة: لا حلقات تكرارية — عمليات موجهة لاحقاً.
- دوال (Functions) تحسب مؤشر كتلة الجسم وتصنف سكر الصائم بالعتبات السريرية، وتعاد على أكثر من مريض.

3.4 اليوم 4 — Google Colab بعمق والجسر مع الحاسوب

مفهوم اليوم المصيري: قرص الجلسة مؤقت وDrive دائم. الخطوات: تفعيل GPU T4 من قائمة Runtime، ثم:

```
!nvidia-smi # is the GPU on?
!pip install ultralytics -q # ! turns the cell into a terminal
```

```
1 from google.colab import drive
2 drive.mount('/content/drive') # files under /content/drive/MyDrive/
```

والجسر الكامل من ويندوز 11: ضغط المجلد ثم رفعه إلى Drive ففكه في الجلسة:

```
tar -a -c -f sample.zip sample_folder :: on your PC
!unzip -q /content/drive/MyDrive/sample.zip -d /content/ # in Colab
!cp /content/result.txt /content/drive/MyDrive/ # rescue outputs
```

3.5 اليوم 5 — Pandas و NumPy: استيراد السجلات وتنظيفها

على ملفات الدورة (patients_demographics.csv و messy_lab_data.csv): الاستيراد، ثم النظرة الأولى بأمرين لا يتقدم قبلهما أحد، ثم الأقنعة المنطقية بدل الحلقات:

```
1 import pandas as pd
2 df = pd.read_csv("patients_demographics.csv")
3 df.info(); df.describe() # first look, always
4 elderly = df[df["age"] >= 65] # boolean mask, no loops
5 df.groupby("department")["length_of_stay_days"].mean()
6 messy = pd.read_csv("messy_lab_data.csv")
7 messy = messy.drop_duplicates()
8 messy["glucose_mgdl"] = messy["glucose_mgdl"].fillna(
9     messy["glucose_mgdl"].median())
```

3.6 اليوم 6 — الرسوم البيانية وهندسة الميزات

مدرج تكراري لسكر الدم مع خط عتبة التشخيص عند 126، وخريطة ارتباط حرارية، ثم اشتقاق أعمدة سريرية: نسبة الكوليسترول، وفتات HbA1c عبر pd.cut بعتبات 5.7 و 6.5، والترميز الفئوي بـ get_dummies.

3.7 اليوم 7 — TensorFlow: أول شبكة عصبية على الجداول

الخريطة الصادرة لدعم GPU في TensorFlow

في Google Colab: جاهز فوراً بلا تثبيت — خيار الدورة الأول. في WSL2 (لينكس داخل ويندوز): دعم كامل حديث عبر `pip install tensorflow[and-cuda]`. على ويندوز الأصلي: النسخ الحديثة تعمل بالمعالج فقط — دعم GPU الأصلي توقف عند النسخة 2.10.

التطبيق في Colab على سجل السكري: تقسيم طبقي 80/20، معيار StandardScaler من بيانات الدرس فقط، نموذج Keras تسلسلي بطبقتين خفيتين ومخرج احتمالي sigmoid، ثم قراءة دقة الامتحان لا دقة الدرس.

3.8 اليوم 8 — PyTorch: المصفوفات وتسريع GPU محلياً

الفحص الأول دائماً `nvidia-smi`، ثم التثبيت في بيئة نظيفة بالمسار الرسمي الحديث (`pip` داخل بيئة `conda` — القناة الرسمية القديمة في `conda` أوقفت):

```
conda create -n yolo python=3.11 -y
conda activate yolo
:: NVIDIA GPU build (CUDA 12.6 wheels):
pip install torch torchvision --index-url https://download.pytorch.org/whl/cu126
:: CPU-only fallback:
pip install torch torchvision

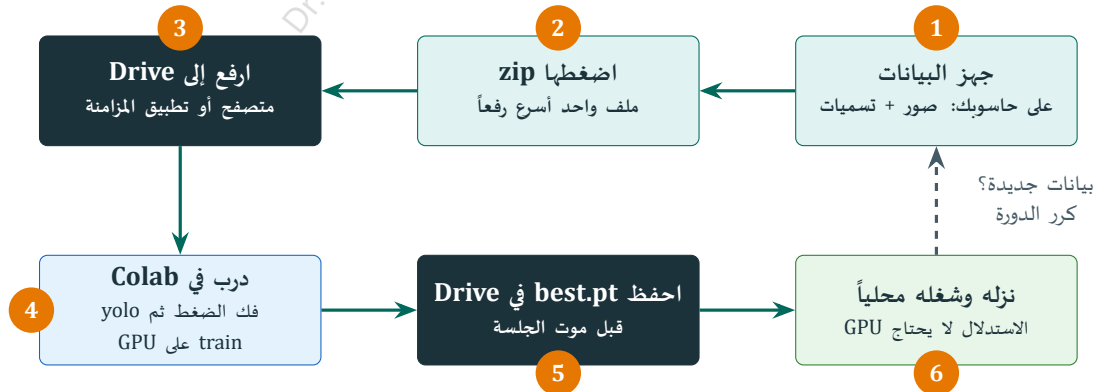
python -c "import torch; print(torch.__version__)"
python -c "import torch; print(torch.cuda.is_available())"
python -c "import torch; print(torch.cuda.get_device_name(0))"
```

شيفرة 3.3: تثبيت PyTorch بدعم GPU والتحقق الثلاثي.

المطلوب: رقم نسخة فيه `cu126`، ثم `True`، ثم اسم البطاقة. ويختم اليوم بسباق قياس: ضرب مصفوفتين كبيرتين على المعالج ثم على البطاقة وتدون نسبة التسريع.

3.9 اليوم 9 — تدريب YOLO لكشف خلايا الدم (ذروة الدورة)

الكشف (Detection) يجيب سؤال المختبري الحقيقي: أين كل خلية وكم عددها من كل نوع. مجموعة الدورة: BCCD (364 صورة، 4,888 تسمية، الأصناف WBC/RBC/Platelets، الترخيص MIT) من منصة Roboflow؛ وتفاصيل الحصول عليها الموثقة في التقرير البحثي المرافق ودفتر التدريب.



شكل 3.1: رحلة التكامل الكاملة: تجهيز محلي، فتدريب سحابي، فتشغيل محلي.

بنية تصدير Roboflow القياسية وبطاقة التعريف:

```
blood_cells/
|-- data.yaml
|-- train/ (images/ + labels/)      205 x .jpg + .txt
|-- valid/ (images/ + labels/)      87 x .jpg + .txt
`-- test/ (images/ + labels/)       72 x .jpg + .txt
```

كل سطر تسمية جسم واحد بخمسة أرقام (h w c) class منسوبة من 0 إلى 1)، وملف data.yaml يكتب المفتاح val تلقائياً فيقبله ultralytics كما هو. الوصفة الكاملة في Colab:

```
# cell 1 - install
!pip install ultralytics roboflow -q

# cell 2 - dataset (paste YOUR generated snippet: Show download code)
# ... or the classroom ZIP:
!unzip -q /content/drive/MyDrive/blood_cells.zip -d /content/blood_cells

# cell 3 - smoke test (2 min) then the real training
!yolo detect train model=yolo11n.pt data=coco8.yaml epochs=3 imgsz=640
!yolo detect train model=yolo11n.pt data=/content/blood_cells/data.yaml epochs=50 imgsz=640
device=0

# cell 4 - rescue the weights BEFORE the session ends
!cp /content/runs/detect/train/weights/best.pt /content/drive/MyDrive/blood_cells_best.pt
```

شيفرة 3.4: وصفة Colab الكاملة لليوم التاسع.

وعلى حاسوب ويندوز 11 الأمر نفسه (يتغير المسار وتغيّب علامة !):

```
conda activate yolo
pip install ultralytics
yolo detect train model=yolo11n.pt data=C:/datasets/blood_cells/data.yaml epochs=50 imgsz=640
device=0
:: no NVIDIA card? add device=cpu (overnight run)

yolo detect val model=best.pt data=C:/datasets/blood_cells/data.yaml
yolo detect predict model=best.pt source=smear_042.jpg conf=0.4 save=True
```

★ لؤلؤة سريرية — قراءة النتائج قراءة مختبري

mAP50 فوق 0.85 بداية ممتازة تعليمياً؛ منحى خسارة يهبط ومنحنى mAP يصعد في results.png يعنيان تدريباً سليماً؛ وعتبة الثقة conf مؤشر يحرك كما تحرك عتبة أي فحص: رفعها يقلل الإنذارات الكاذبة وخفضها يقلل الفائت — والقرار الأخير دائماً لعين المختبري.

3.10 اليوم 10 — المشروع الختامي وخارطة الطريق

- مشروع جماعي: إما إعادة تدريب YOLO بمعاملات محسنة أو على بيانات موسومة جديدة، وإما نموذج جدولي كامل على سجل الخطورة القلبية من التنظيف إلى التقييم بالحساسية والنوعية.
- الحوكمة والأخلاقيات: التحيز والخصوصية وسياسة الوزارة والمعايير الدولية، ومبدأ الإنسان في الحلقة (Human-in-the-loop): النموذج يرتب والطبيب يقرر.
- الختام: عرض النتائج، والتقييم، وتسليم الشهادات، ومناقشة مشاريع مقترحة للوزارة.

باب 4

الملاحق التقنية

4.1 بطاقة الأوامر النهائية

تعلق قرب الشاشة وتغني عن الرجوع إلى الملائم في العمل اليومي:

```
:: === setup (once per machine) ===
conda create -n yolo python=3.11 -y
conda activate yolo
pip install torch torchvision --index-url https://download.pytorch.org/whl/cu126
pip install ultralytics

:: === every session ===
conda activate yolo
yolo checks

:: === train / evaluate / use ===
yolo detect train model=yolo11n.pt data=data.yaml epochs=50 imgsz=640 device=0
yolo detect val model=runs/detect/train/weights/best.pt data=data.yaml
yolo detect predict model=best.pt source=image.jpg conf=0.4 save=True
```

في Colab: الأسطر نفسها مسبقة بعلامة ! وبمسارات لينكس، ولا حاجة لسطري conda ولا لتثبيت torch.

4.2 الأخطاء الخمسة الشائعة وعلاجها

رصدت هذه الأخطاء في أفواج حقيقية؛ من يحفظها يوفر على نفسه أسبوع حيرة:

الخطأ	العلاج الجاهز
ثبت الحزم خارج البيئة	تحقق من وجود (yolo) في أول سطر الموجه قبل أي <code>pip install</code>
أغلق Colab قبل النسخ إلى Drive	انسخ <code>best.pt</code> فور انتهاء التدريب لا في نهاية اليوم
مسار ويندوز في <code>data.yaml</code> داخل كولا (أو العكس)	عدل سطر <code>path</code> وحده عند تبديل البيئة
<code>device=0</code> يفشل رغم وجود البطاقة	ثبت نسخة CPU من <code>torch</code> ؛ أعد التثبيت بفهرس <code>cu126</code> وتحقق <code>cuda.is_available</code>
نفدت ذاكرة GPU أثناء التدريب	أنقص <code>batch</code> إلى 8 ثم 4، أو صغر <code>imgsz</code> إلى 480

جدول 4.1: الأخطاء الخمسة وعلاجها.

4.3 مصادر البيانات الموسومة

1. التوسيم اليدوي: أدوات LabelImg و CVAT و Roboflow: ترسم الصناديق بالفأرة وتصدر بصيغة YOLO مباشرة — جودة التوسيم سقف جودة النموذج.
2. مجموعات طبية جاهزة: BCCD لخلايا الدم — الرابط: <https://public.roboflow.com/object-detection/bccd> (ترخيص MIT).
3. للتجربة الفورية: data=coco8.yaml تنزل تلقائياً وتثبت أن السلسلة كلها تعمل.

4.4 مراجع رسمية سريعة

- Miniconda: <https://www.anaconda.com/docs/getting-started/miniconda/install>
- محدد أوامر تثبيت PyTorch: <https://pytorch.org/get-started/locally/>
- TensorFlow (بما فيه مسار WSL2): <https://www.tensorflow.org/install/pip>
- توثيق YOLO Ultralytics: <https://docs.ultralytics.com/modes/train/>
- Google Colab: <https://colab.research.google.com>

أعد هذا المنهج المدمج بدمج منهج Colab التدريبي مع المنهج العشري المفصل لمديرية التشخيص، وأضيفت إليه وحدات سطر الأوامر و Miniconda وتثبيت GPU وتدريب YOLO والتكامل المحلي السحابي.
د. حسام صلاح مهدي — رئيس قسم الذكاء الاصطناعي، جامعة المصطفى — آب 2026