

دفتر التدريب العملي يوماً بيوم — الأيدي على لوحة المفاتيح

عشر أوراق عمل: المهارة، والأوامر الجاهزة، والتحدي، وقائمة الإنجاز
كل الأجهزة: ويندوز 11 • كل الأوامر تنفذ حرفياً

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



كيف تستعمل هذا الدفتر؟

هذا الدفتر هو المختبر العملي للدورة: لكل يوم ورقة عمل فيها المهارة المستهدفة، والأوامر الجاهزة للتنفيذ الحرفي، وتحديّ ختامي، وقائمة إنجاز يوقعها المتدرب. النظرية في الشرائح المرافقة — وهنا الأيدي على لوحة المفاتيح.

✓ قواعد المختبر الثلاث

1. اكتب الأمر بيدك أول مرة (لا نسخ)، ثم انسخ في المرات التالية؛ اليد تتعلم قبل العين.
2. اقرأ سطر الموجه قبل كل أمر: هل أنت في البيئة الصحيحة (medai)؟ وفي المجلد الصحيح؟
3. كل ناتج مهم يحفظ فوراً — وفي Colab: ينسخ إلى Drive قبل نهاية الجلسة.

سلم المهارات. كل يوم يضيف مهارة واحدة تبني على ما قبلها:

اليوم	المهارة الجديدة	تبني على
1	أشغل دفتر Colab وأنفذ خلايا	—
2	أنشئ بيئة معزولة وأثبت مكتبة بسطر أوامر	مفهوم البرنامج (يوم 1)
3	أكتب متغيرات ودوال تمثل سجل مريض	تشغيل الخلايا (يوم 1)
4	أفعل GPU وأنقل ملفات بين حاسوبي والسحابة	الطرفية (يوم 2)
5	أقرأ وأنظف جدول مرضى Pandas	الاستيراد والمكتبات (يوم 2-3)
6	أرسم البيانات وأصنع ميزات طبية جديدة	Pandas (يوم 5)
7	أدرب أول شبكة عصبية على جدول	التنظيف والميزات (يوم 5-6)
8	أثبت PyTorch بدعم GPU وأتحقق منه	البيئات (يوم 2) + GPU (يوم 4)
9	أدرب YOLO على صور مجهرية وأقيمه	كل ما سبق — يوم الذروة
10	أسلم مشروعاً كاملاً من البيانات إلى القرار	كل شيء

جدول 0.1: سلم المهارات اليومي.

المراجع المرافقة. شرائح الأساسيات (المفاهيم) وشرائح الأدوات العملية (الأوامر)، والبطاقة الرسمية conda-cheatsheet (تطبع لكل متدرب)، ومجلد البيانات datasets بملفاته السبعة، والتقارير البحثي المحكم لمنصة Roboflow.

أول تشغيل: دفتر Colab يعمل بين يديك

✓ مهارة اليوم

أشغل دفتر ملاحظات سحابياً وأنفذ خلايا برمجية وأقرأ نتائجها. المراجع: شرائح الأساسيات — القسمان 1 و2.

البرنامج وصفه، والحاسوب مطبخ، والنتائج طبق. مستشفياتنا تولد طوفان بيانات (LIS وEHR وRT-PCR) لا يحلله أحد — وبحلول اليوم العاشر ستحولون جدولاً خاماً إلى أداة فرز تساعد الطبيب، والقرار الأخير دائماً للإنسان.

1.1 خطوات العمل

1. افتح colab.research.google.com وسجل بحساب Google، ثم New notebook. وسم الدفتر `day01_first_run`.
2. نفذ الخلايا الثلاث الآتية بيدك (Shift+Enter للتنفيذ):

```
1 # cell 1: the computer as a calculator
2 7 * 24
3
4 # cell 2: variables = labeled boxes (basics deck, section 2)
5 patient_name = "Ali"      # text (str)
6 patient_age = 54          # integer (int)
7 glucose_mgdl = 132.5     # decimal (float)
8 is_smoker = True         # boolean (bool)
9 print(patient_name, patient_age, glucose_mgdl, is_smoker)
10
11 # cell 3: the first programmed medical decision
12 print("High glucose:", glucose_mgdl > 126)
```

3. أضف خلية نصية (Text) واكتب فيها بالعربية: اسمك وقسمك وما تتوقعه من الدورة، ثم احفظ الدفتر.

🔗 التحدي السريري — ثلاثون دقيقة

غير قيمة `glucose_mgdl` ثلاث مرات (95 ثم 130 ثم 210) ولاحظ ناتج المقارنة؛ وأنشئ متغيرات لمريض ثانٍ من واقع عملك (بيانات وهمية) واطبعها؛ وناقش زميلك: أي أعمدة يحتاجها جدول مريض في قسمكما؟

1.2 قائمة الإنجاز

- ☐ فتحت Colab وأنشأت دفترًا باسمي
- ☐ نفذت الخلايا الثلاث بنجاح ورأيت النواتج
- ☐ أعرف الأنواع الأربعة: نص، صحيح، عشري، منطقي
- ☐ حفظت الدفتر وأعرف أين أجده غدًا

سطر الأوامر وMiniconda: التحكم الكامل بالبيئات

✓ مهارة اليوم

أنشئ بيئة بايثون معزولة على حاسوبي وأتحكم بها بالأوامر: إنشاء وتفعيل وتثبيت واستنساخ وتصدير وحذف. المراجع: شرائح الأدوات — القسمان 2 و3، والبطاقة الرسمية `conda-cheatsheet`.

مشروعان يتنازعان بايثون واحداً = جحيم الاعتماديات (Dependency Hell). الحل بنوي: غرفة مختبر معزولة لكل مشروع (Environment) تديرها أداة واحدة اسمها `conda`.

2.1 التثبيت والتحقق (ويندوز 11)

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

:: verify - in a NEW "Anaconda Prompt (Miniconda3)" window:
conda info
conda --version
```

2.2 بيئة الدورة

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

اقرأ سطر الموجه: `C:\Users\...> (medai)` — القوسان يؤكدان أنك داخل البيئة.

2.3 التحكم الكامل — من البطاقة الرسمية

```
:: --- query ---
conda info --envs           :: all my envs (* = active)
conda list -n medai         :: what is inside one env
conda search pandas --info   :: package info before install
conda COMMAND --help        :: help for any command

:: --- install / update / remove ---
conda install -n medai openpyxl -y :: install into a named env
conda update --all -n medai         :: update everything in it
conda uninstall -n medai openpyxl   :: remove a package
conda update -n base conda          :: update conda itself
```

```

:: --- manage environments ---
conda create --clone medai -n medai_backup
conda rename -n medai_backup medai_v2
conda remove -n medai_v2 --all -y
conda list -n medai --revisions          :: change history
conda install -n medai --revision 1      :: roll back!

:: --- share (reproducibility) ---
conda export --from-history > environment.yml
conda env create -n medai --file environment.yml

:: --- housekeeping ---
conda clean --all
conda deactivate

```

شيفرة 2.1: بطاقة التحكم الكامل بكوندا على ويندوز 11.

ملاحظتان: أضف -y لأي أمر كي لا يسألك التأكيد؛ و pip install يعمل داخل البيئة أيضاً للحزم غير المتوفرة في conda (مثل ultralytics).

التحدي السريري — ثلاثون دقيقة

أنشئ بيئة test بنسخة python=3.10 وفعلها وتحقق من النسخة؛ ثم استنسخها باسم test2، وأعد تسمية test2 إلى demo، ثم احذف الاثنيتين؛ وصدّر بيئة medai إلى environment.yml وافتح الملف بالمفكرة واقراه مع زميلك؛ وجرب conda list --revisions واشرح ما تراه.

2.4 قائمة الإنجاز

□ Miniconda مثبت وconda info يعمل

□ بيئة medai منشأة ومفعلة والقوسان ظاهران في الموجه

□ أستطيع: إنشاء وتفعيل واستنساخ وإعادة تسمية وحذف بيئة دون الرجوع للورقة

□ صدرت environment.yml وفهمت محتواه

□ أحتفظ بنسخة مطبوعة من البطاقة الرسمية قرب الشاشة

أساسيات بايثون للبيانات الطبية

✓ مهارة اليوم

أمثل سجل مريض بقاموس (Dictionary) وأكتب دالة (Function) تحسب مؤشراً حيوياً وأعيد استخدامها. المراجع: شرائح الأساسيات — القسمان 2 و3.

سجل المريض في بايثون قاموس: مفتاح فقيمة، كاستمارة المختبر تماماً. والدالة أمر جاهز نكتبه مرة ونستدعيه ألف مرة.

3.1 خطوات العمل

```

1 # 1) the patient record = a dictionary
2 patient = {
3     "id": "P-1042",
4     "name": "Ali Hussein",
5     "age": 54,
6     "weight_kg": 88.0,
7     "height_m": 1.72,
8     "glucose_mgdl": 132.5,
9     "smoker": True,
10 }
11 print(patient["name"], "-", patient["age"], "years")
12
13 # 2) a function computing BMI
14 def bmi(weight_kg, height_m):
15     return weight_kg / height_m ** 2
16 print("BMI =", round(bmi(patient["weight_kg"], patient["height_m"]), 1))
17
18 # 3) a clinical classification function (fasting glucose, mg/dL)
19 def glucose_class(value):
20     if value < 100:
21         return "Normal"
22     elif value < 126:
23         return "Pre-diabetic"
24     else:
25         return "Diabetic"
26 print(patient["name"], "->", glucose_class(patient["glucose_mgdl"]))
27
28 # 4) a list of patients = a list of dictionaries (bridge to Pandas)
29 patients = [
30     {"id": "P-1042", "glucose_mgdl": 132.5},
31     {"id": "P-1043", "glucose_mgdl": 96.0},
32     {"id": "P-1044", "glucose_mgdl": 210.0},
33 ]
34 for p in patients:
35     print(p["id"], glucose_class(p["glucose_mgdl"]))

```

ملاحظة منهجية: هذه آخر حلقة for تكتبها في الدورة — من اليوم الخامس نستبدلها بالعمليات الموجهة (Vectorization): عمود كامل بسطر واحد.

التحدي السريري — ثلاثون دقيقة

أضف للدالة bmi تقريباً لرقم عشري واحد داخلها؛ واكتب دالة hba1c_class(pct): أقل من 5.7 طبيعي، حتى 6.4 ما قبل السكري، فما فوق سكري؛ ثم أضف المفتاح hba1c_pct لكل مريض وصنف الثلاثة.

3.2 قائمة الإنجاز

- ☐ أنشأت قاموس مريض وقرأت قيمه بالمفاتيح
- ☐ كتبت دالتين واستدعيتهما على أكثر من مريض
- ☐ أفهم لماذا سنترك حلقات for بعد اليوم

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

Google Colab بعمق: GPU مجاني وجسر الملفات

✓ مهارة اليوم

أفعل GPU في Colab، وأستعمل علامة ! كطرفية، وأنقل ملفات بين حاسوبي (ويندوز 11) والسحابة عبر Drive. المراجع: شرائح الأدوات — القسمان 5 و6.

Colab حاسوب افتراضي عند غوغل تكتب له من متصفحك — فيه GPU مجاني. والقاعدة المصيرية: قرص الجلسة مؤقت وDrive دائم — كل ناتج مهم ينسخ إلى Drive فوراً.

4.1 خطوات العمل

فعل المعالج أولاً: قائمة Runtime فـ Change runtime type فاختر T4 GPU فـ Save، ثم:

```
!nvidia-smi                # Tesla T4 appears? the GPU is on
!python --version          # ! = the cell becomes a terminal
!pip install ultralytics -q
!ls /content
```

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

الجسر الكامل من ويندوز 11 إلى الجلسة والعودة:

```
:: on your PC (PowerShell, in the folder containing sample_folder):
tar -a -c -f sample.zip sample_folder
:: then drag sample.zip into drive.google.com (or the Drive app folder)
```

```
# in Colab:
!unzip -q /content/drive/MyDrive/sample.zip -d /content/
!ls /content/sample_folder
# and the way back - rescue any output into Drive:
!cp /content/result.txt /content/drive/MyDrive/
```

📁 الطرق الثلاث لنقل الملفات

لوحة الملفات بالسحب والإفلات: سريعة لكنها تمنح مع الجلسة. files.upload() و files.download(): من داخل الشيفرة للملفات الصغيرة. Google Drive: الطريقة المعتمدة للبيانات والنماذج — دائمة ومنظمة.

التحدي السريري — ثلاثون دقيقة

أنشئ على حاسوبك مجلدًا فيه ملفان نصيان، اضغطه، ارفعه، فكّه في الجلسة، واعرض محتواه بأمر `ls`! ثم أنشئ في الجلسة ملفًا وانسخه إلى Drive؛ ثم اقطع الاتصال وأعد الاتصال: ماذا بقي في قرص الجلسة؟ وماذا بقي في Drive؟ فسر.

4.2 قائمة الإنجاز

- ☐ فعلت GPU T4 ورأيت ناتج `!nvidia-smi`
- ☐ أفرق بين خلية بايثون وخلية ! طرفية
- ☐ ربطت Drive ونقلت ملفًا ذهابًا وإيابًا
- ☐ أحفظ عن ظهر قلب: قرص الجلسة مؤقت وDrive دائم

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

Pandas و NumPy: قراءة سجلات المرضى وتنظيفها

✓ مهارة اليوم

أستورد ملف CSV مختبرياً، وألتقط ملامحه بأمرين، وأعزل الحالات بشرط منطقي، وأعالج القيم المفقودة. المراجع: شرائح الأساسيات — القسمان 3 و4؛ البيانات: مجلد datasets.

المكتبة صندوق عدة جاهز نستورده بسطر واحد. والجدول الملوّث يعطي نموذجاً ملوّثاً — لذا خط التنظيف يسبق أي تحليل.

5.1 خطوات العمل

```

1 # 1) the course import card (basics deck, section 3)
2 import numpy as np                # fast math
3 import pandas as pd               # patient tables
4 import matplotlib.pyplot as plt   # charts
5
6 # 2) import and first look - never skip these two
7 df = pd.read_csv("patients_demographics.csv")
8 df.head()
9 df.info()
10 df.describe()
11
12 # 3) boolean masks instead of loops (Vectorization)
13 elderly = df[df["age"] >= 65]
14 long_stay = df[df["length_of_stay_days"] > 7]
15 print(len(elderly), len(long_stay))
16
17 # 4) grouping: mean stay per department, one line
18 df.groupby("department")["length_of_stay_days"].mean().round(1)
19
20 # 5) cleaning the deliberately-messy file
21 messy = pd.read_csv("messy_lab_data.csv")
22 messy.isna().sum()
23 messy = messy.drop_duplicates()
24 messy["glucose_mgdl"] = messy["glucose_mgdl"].fillna(
25     messy["glucose_mgdl"].median())

```

🔥 التحدي السريري — ثلاثون دقيقة

كم مريضاً مدخناً فوق الستين؟ (قناع بشرطين مع &): وأي محافظة لديها أعلى متوسط رقود؟ (groupby ثم sort_values)؛ وفي الملف الملوّث: اعثر على قيمة عمر مستحيلة (فوق 120) وقرر: حذف أم تصحيح؟ ووثق قراراتك في خلية نصية — فالقرار سريري لا برمجي.

5.2 قائمة الإنجاز

- ❑ أكتب بطاقة الاستيراد من الذاكرة
- ❑ `info()` و `describe()` أول ما أنفذه على أي ملف جديد
- ❑ أعزل الحالات بقناع منطقي دون حلقة `for`
- ❑ عالجت قيماً مفقودة ومكررات ووثقت القرار

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

الرسوم البيانية وهندسة الميزات الطبية

✓ مهارة اليوم

أرسم توزيع قيمة مختبرية وخريطة ارتباط، وأشتق أعمدة سريرية جديدة (BMI وفئات HbA1c) بسطر واحد لكل عمود. البيانات: `heart_risk.csv` و `chemistry_panel.csv`

الرسم يكشف ما لا يظهره المتوسط: التوزيع والذبول والشواذ. والميزة الجيدة (Feature) تختصر خبرة سريرية في عمود — نصف النموذج صناعة ميزات.

6.1 خطوات العمل

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4
5 chem = pd.read_csv("chemistry_panel.csv")
6
7 # 1) histogram of glucose with the diagnostic cut-off
8 plt.hist(chem["glucose_mgdl"], bins=40)
9 plt.axvline(126, color="red", linestyle="--", label="Diabetes cut-off")
10 plt.xlabel("Glucose (mg/dL)"); plt.ylabel("Patients"); plt.legend()
11 plt.show()
12
13 # 2) correlation heatmap
14 sns.heatmap(chem.corr(numeric_only=True), cmap="coolwarm", center=0)
15 plt.show()
16
17 # 3) feature engineering: new columns, one line each (no loops!)
18 chem["chol_hdl_ratio"] = chem["total_chol_mgdl"] / chem["hdl_mgdl"]
19 chem["hba1c_class"] = pd.cut(chem["hba1c_pct"],
20                             bins=[0, 5.7, 6.5, 100],
21                             labels=["Normal", "Pre-diabetic", "Diabetic"])
22 chem["hba1c_class"].value_counts()
23
24 # 4) categorical encoding - models do not read text
25 hr = pd.read_csv("heart_risk.csv")
26 hr_encoded = pd.get_dummies(hr, columns=["chest_pain_type"], drop_first=True)
```

🔗 التحدي السريري — ثلاثون دقيقة

ارسم مدرج creatinine_mgdl وحدد بصرياً أين تقع القيم الشاذة؛ ومن خريطة الارتباط: ما أقوى زوجين مرتبطين؟ وهل الارتباط منطقي سريرياً؟ ثم أنشئ عمود risk_flag يساوي True لمن سكره فوق 126 ونسبته chol_hdl_ratio فوق 5.

6.2 قائمة الإنجاز

- ❑ أرسم مدرجاً بعتبة قرار وأفسره
- ❑ اشتق عموداً جديداً وفئات `pd.cut` دون حلقات
- ❑ أفهم لماذا نرسم الأعمدة النصية قبل النمذجة

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

TensorFlow/Keras: أول شبكة عصبية على جدول طبي

✓ مهارة اليوم

أقسم البيانات درساً وامتحاناً، وأبني نموذج Keras تسلسلياً، وأدربه وأقرأ دقته قراءة ناقدة. البيانات: `diabetes_dataset.csv`: التدريب في Colab.

النموذج يتعلم من أمثلة موسومة: ميزات X فهدف y ، ويمتحن على بيانات لم يرها — وإلا صار الطالب الحافظ (Overfitting). على ويندوز 11 ندرّب TensorFlow في Colab (دعم GPU الأصلي على ويندوز توقف عند النسخة 2.10).

7.1 خطوات العمل (في Colab)

```

1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.preprocessing import StandardScaler
4 import tensorflow as tf
5
6 df = pd.read_csv("diabetes_dataset.csv")
7 X = df.drop(columns=["outcome"])      # features
8 y = df["outcome"]                    # target
9
10 # 1) Lesson vs exam: 80/20, stratified
11 X_train, X_test, y_train, y_test = train_test_split(
12     X, y, test_size=0.2, stratify=y, random_state=42)
13
14 # 2) one ruler for everybody - fitted on the Lesson only (no Leakage)
15 scaler = StandardScaler()
16 X_train_s = scaler.fit_transform(X_train)
17 X_test_s = scaler.transform(X_test)
18
19 # 3) a sequential model: two hidden layers, sigmoid output
20 model = tf.keras.Sequential([
21     tf.keras.layers.Dense(16, activation="relu"),
22     tf.keras.layers.Dense(8, activation="relu"),
23     tf.keras.layers.Dense(1, activation="sigmoid"),
24 ])
25 model.compile(optimizer="adam", loss="binary_crossentropy",
26               metrics=["accuracy"])
27
28 # 4) train - watch both curves together
29 history = model.fit(X_train_s, y_train, epochs=30,
30                    validation_split=0.2, verbose=1)
31
32 # 5) the honest exam
33 loss, acc = model.evaluate(X_test_s, y_test)

```

34 `print("Test accuracy:", round(acc, 3))`

ومحلياً على ويندوز 11 (للتجارب الصغيرة فقط — معالج فقط):

```
conda create -n tfcpu python=3.11 -y
conda activate tfcpu
pip install tensorflow
python -c "import tensorflow as tf; print(tf.config.list_physical_devices('GPU'))"
:: [] means no GPU on native Windows - expected; serious training in Colab/WSL2
```

التحدي السريري — ثلاثون دقيقة

ارفع epochs إلى 200: هل تحسنت دقة الامتحان أم دقة الدرس فقط؟ سم الظاهرة؛ وجرب طبقة واحدة بثمانية عصبونات وقارن؛ ثم أجب: لماذا لا نحكم على النموذج بدقة بيانات التدريب أبداً؟

7.2 قائمة الإنجاز

- أفرق بين X و y وأقسم مع `stratify`
- أعاير بالمسطرة الصحيحة دون تسريب
- دربت نموذج Keras وقرأت دقة الامتحان
- أشرح فرط التخصيص بمثال الطالب الحافظ

PyTorch على ويندوز 11: تثبيت GPU والتحقق منه

✓ مهارة اليوم

أفحص بطاقة جهازي، وأثبت PyTorch بدعم GPU في بيئة نظيفة، وأثبت بالقياس أن GPU أسرع. المراجع: شرائح الأدوات — القسم 4 كاملاً.

سلم GPU: تعريف NVIDIA، CUDA/cuDNN (تصل داخل حزمة pip تلقائياً)، PyTorch، فشيفرتك. كل ما يلزمك: تعريف حديث وأمر تثبيت صحيح.

8.1 خطوات العمل

```
:: 1) ALWAYS check first:
nvidia-smi
:: table with card name + CUDA version -> proceed
:: command fails -> no NVIDIA GPU: use the CPU build or Colab

:: 2) clean environment + the official install path
conda create -n yolo python=3.11 -y
conda activate yolo
pip install torch torchvision --index-url https://download.pytorch.org/whl/cu126
:: (no NVIDIA card:) pip install torch torchvision

:: 3) the triple check
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))"
:: wanted: a version ending in +cu126, then True, then your card name
```

شيفرة 8.1: الفحص والتثبيت والتحقق الثلاثي.

8.2 سباق المعالج ضد البطاقة

```
1 import torch, time
2
3 a = torch.rand(4000, 4000); b = torch.rand(4000, 4000)
4 t0 = time.time(); _ = a @ b; cpu_t = time.time() - t0
5
6 a_gpu, b_gpu = a.cuda(), b.cuda()
7 torch.cuda.synchronize()
8 t0 = time.time(); _ = a_gpu @ b_gpu; torch.cuda.synchronize()
9 gpu_t = time.time() - t0
10 print(f"CPU: {cpu_t:.3f}s | GPU: {gpu_t:.3f}s | x{cpu_t/gpu_t:.0f}")
```

```

11
12 # the bridge from Pandas to Tensors
13 import pandas as pd
14 df = pd.read_csv("diabetes_dataset.csv")
15 X = torch.tensor(df.drop(columns=["outcome"]).values, dtype=torch.float32)
16 print(X.shape, X.device)

```

🔥 التحدي السريري — ثلاثون دقيقة

كرر السباق بمصفوفة 1000x1000 ثم 8000x8000: متى يتسع الفارق ولماذا؟ وانقل X إلى البطاقة بـ `cuda()`. وتحقق من `X.device`: ثم شخص لزميل ظهر عنده `False`: السببان الأرجح وعلاجهما.

8.3 قائمة الإنجاز

- ❑ قرأت ناتج `nvidia-smi` وفهمت خاناته
- ❑ بيئة `yolo` فيها PyTorch بنسخة `cu126` (أو CPU عن وعي)
- ❑ التحقق الثلاثي كله سليم
- ❑ قست تسريع GPU بنفسني ودونت الرقم

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

يوم الذروة: تدريب YOLO على خلايا الدم

✓ مهارة اليوم

أحصل على مجموعة بيانات موسومة من Roboflow، وأدرب YOLO11n عليها بسطر أوامر واحد على GPU مجاني، وأقيم النموذج وأشغله محلياً على ويندوز 11. المراجع: شرائح الأدوات — القسمان 6 و7؛ والتقرير البحثي المحكم المرافق.

الكشف (Detection) يجيب سؤال المختبري الحقيقي: أين كل خلية وكم عددها من كل نوع. مجلد train هو الدرس و valid هو الامتحان النزيه — فكرة اليوم السابغ نفسها لكن على الصور.

9.1 الحصول على البيانات (خطوات متحققة بحثياً)

1. حساب مجاني على app.roboflow.com (خطة Public — بلا بطاقة ائتمانية).
2. مفتاحك الخاص في app.roboflow.com/settings/api — سري ككلمة المرور، لا يلصق في دفتر مشترك.
3. افتح صفحة BCCD: <https://public.roboflow.com/object-detection/bccd> (364 صورة، 4,888 تسمية، الأصناف WBC/RBC/Platelets، الترخيص MIT).
4. Dataset Download Dataset Download dataset فالتصيغة YOLOv11 PyTorch TXT (أو YOLOv8 — البنية واحدة) فاختر Show download code.
5. الصق المقتطف المولد كما هو في Colab — يمتلئ بمفتاحك وأسماء المشروع الصحيحة تلقائياً، وينزل ويجهز data.yaml وحده.

✗ خطأ شائع — تحذير وزاري

كل ما يرفع إلى حساب Roboflow مجاني ينشر علناً على Universe — لا ترفعوا أي صور مرضى أو بيانات للوزارة إلى الحسابات المجانية أبداً. التنزيل آمن — الخطر في الرفع.

بدل بلا إنترنت (جهزه المدرب سلفاً — لا يحتاج أي مفتاح): ملف blood_cells.zip يرفع إلى Drive ثم يفك في الجلسة؛ وعندئذ يضبط المفتاح path في data.yaml إلى جذر المجموعة (مطب المسارات النسبية في المسار اليدوي).

9.2 الوصفة الكاملة في Colab

```
# cell 1 - the tools
!pip install ultralytics roboflow -q
!yolo checks

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

# cell 3 - smoke test (2 min), then the real training on the T4
!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 - THE most important line of the day
!cp /content/runs/detect/train/weights/best.pt /content/drive/MyDrive/blood_cells_best.pt
```

شيفرة 9.1: الخلايا الأربع لليوم التاسع في Colab.

مع المقتطف البرمجي يكون مسار البيانات data={dataset.location}/data.yaml — النمط المثبت في دفتر Roboflow الرسمي لتدريب YOLO11.

9.3 التقييم ثم التشغيل المحلي

```
!yolo detect val model=/content/runs/detect/train/weights/best.pt data=/content/blood_cells/
data.yaml
# -> mAP50 above 0.85 = an excellent teaching start
```

افتح results.png (خسارة تهبط و mAP يصعد = تدريب سليم) و confusion_matrix.png (أي الأصناف تختلط). ثم نزل blood_cells_best.pt من Drive إلى حاسوبك:

```
conda activate yolo
pip install ultralytics
yolo detect predict model=blood_cells_best.pt source=smear_test.jpg conf=0.4 save=True
:: annotated image lands in runs/detect/predict - CPU is enough here
```

التحدي السريري — ثلاثون دقيقة

جرب conf=0.25 ثم conf=0.6 على الصورة نفسها: ماذا يحدث للإنذارات الكاذبة وللخلايا الفاتئة؟ اربط ذلك بعتبة القرار والحساسية والنوعية؛ وعد الخلايا البيضاء التي اكتشفها النموذج وقارن بعدك اليدوي؛ ومن مصفوفة الالتباس: ما أكثر خلط يقع فيه النموذج ولماذا برأيك؟

9.4 قائمة الإنجاز

- ❑ حصلت على مجموعة موسومة وفهمت بنيتها و data.yaml
- ❑ نجح فحص الدخان coco8 قبل البيانات الحقيقية
- ❑ دربت 50 حقبة على T4 ونسخت best.pt إلى Drive فوراً
- ❑ قرأت mAP50 و results.png ومصفوفة الالتباس
- ❑ شغلت النموذج محلياً على ويندوز 11 ورأيت النتيجة المرسومة

المشروع الختامي: من البيانات الخام إلى قرار مدعوم

✓ مهارة اليوم

أنجز مساراً كاملاً بفريق: بيانات فتنظيف فنموذج فتنظيم فعرض قرار — وألزم حوكمة بيانات المرضى.

10.1 اختر مسار فريقك (أربعون دقيقة عمل)

المسار أ — رؤية حاسوبية: أعد تدريب نموذج الأمس بمعاملين مختلفين (مثلاً `epochs=80` أو `imgsz=800` أو `model=yolo11s.pt`)، وقارن mAP50 قبل وبعد في جدول من سطرين، وأعرض صورة تنبؤ قبل وبعد.
المسار ب — بيانات جدولية: على سجل الخطورة القلبية: نظف، واهندس ميزة جديدة واحدة على الأقل، ودرب نموذج تصنيف، واحسب الحساسية والنوعية من مصفوفة الالتباس — لا تكتف بالدقة — واقترح عتبة قرار وسوغها سريراً.

10.2 عرض النتائج والحوكمة

لكل فريق خمس دقائق وشرحية واحدة: المشكلة، فاليبيانات، فالنموذج، فالرقم الذي تفخرون به، فماذا بعد. وإلزامي ذكر حدود النموذج: متى لا نثق به؟

✗ خطأ شائع — قواعد الحوكمة الثلاث

لا بيانات مرضى حقيقية إلى السحابة إلا مموهة بالكامل وبموافقة رسمية؛ والنموذج يرتب ويقترح والإنسان يقرر (Human-in-the-loop) وكل تجربة توثق — ملف `args.yaml` يحفظ إعداداتكم تلقائياً فأرفقوه بأي تقرير.

10.3 خارطة طريقك بعد الدورة

1. الأسبوعان الأولان: أعد تمارين الأيام 5-9 على بيانات قسمك (مموهة) — المهارة تثبت بالترار.
2. الشهر الأول: وسم مئة صورة من مجالك بأداة توسيم ودرب YOLO عليها.
3. باستمرار: بطاقتنا الأوامر قرب الشاشة، ومجتمعنا Ultralytics وconda للأسئلة.

10.4 قائمة إنجاز الدورة كاملة

- ☐ أنجزت مشروع الفريق وعرضته مع ذكر الحدود
- ☐ أحفظ قواعد الحوكمة الثلاث وأستطيع شرحها لزميل
- ☐ لدي بيتا medai وyolo تعملان على جهازي
- ☐ لدي نموذج best.pt دربته بيدي ويعمل محلياً
- ☐ أعرف خطواتي القادمة بعد الدورة

السكربتات الجاهزة في مجلد train/scripts

توزع هذه السكربتات مع الدفتر وتشغل من Anaconda Prompt على ويندوز 11.

A.1 01_make_env_medai.bat — بيئة اليوم الثاني

```
conda create -n medai python=3.11 -y
conda install -n medai numpy pandas matplotlib seaborn scikit-learn openpyxl -y
```

A.2 02_make_env_yolo.bat — بيئة اليوم الثامن

يتحسس البطاقة تلقائياً: إن نجح nvidia-smi ثبت نسخة CUDA 12.6 وإلا ثبت نسخة المعالج:

```
conda create -n yolo python=3.11 -y
:: GPU found:
pip install torch torchvision --index-url https://download.pytorch.org/whl/cu126
:: else:
pip install torch torchvision
pip install ultralytics
```

A.3 check_gpu.py — هل يرى PyTorch البطاقة؟

```
1 import torch
2 print("PyTorch version :", torch.__version__)
3 print("CUDA available :", torch.cuda.is_available())
4 if torch.cuda.is_available():
5     print("GPU name      :", torch.cuda.get_device_name(0))
6 else:
7     print("1) CPU-only torch installed -> reinstall with cu126 index")
8     print("2) old/missing NVIDIA driver -> update, or use Colab")
```

A.4 03_smoke_test.bat — فحص الدخان لليوم التاسع

```
yolo checks
yolo detect train model=yolo11n.pt data=coco8.yaml epochs=3 imgsz=640
```

قالب data.yaml لمسار ZIP اليدوي

يستعمل هذا القالب فقط عند فك ZIP يدوياً (المسار البرمجي يصحح الملف وحده). يعدل سطر path حسب مكان التدريب ولا يلمس الباقي — وبالشرطات الأمامية حتى على ويندوز:

```
1 # Colab:      path: /content/blood_cells
2 # Windows 11: path: C:/datasets/blood_cells
3 path: /content/blood_cells
4 train: train/images
5 val: valid/images
6 test: test/images
7 names:
8   0: Platelets
9   1: RBC
10  2: WBC
```

الملف الأصلي القادم مع تصدير Roboflow هو المرجع للأصناف وترتيبها — هذا القالب لإصلاح المسارات لا لتغيير الأصناف.

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