قارئ الأرقام

تطبيق ويب للتعرف على الأرقام المكتوبة بخط اليد

مشروع مادة الذكاء الاصطناعي العملي

إشراف: أ/ شيماء عبية

إعداد: أسامه راشد · سامي الخضر · محمد المنتصر

تاريخ إعداد التقرير: 2026-09-06

الموقع: ai.tgchnls.com

من رسم بسيط إلى قرار يمكن تفسيره

يبني المشروع جسرًا بين مفهوم الأوزان في الشبكات العصبية وتجربة عملية يراها المتعلم: يرسم رقمًا أو يرفع صورته، فيشاهد تجهيز الصورة والتنبؤ واحتمالات الأرقام، ثم يدرب شبكة صغيرة أمامه ويرى دقتها تتغير.

نتيجة موثقة	القيمة
دقة النموذج الأساسي على الاختبار	99.24%
حجم الاختبار المستقل	10,000 صورة
عدد الأوزان والمعاملات	197,482
مدة التدريب الأساسي	89.55 ثانية
دقة التدريب الحي قبل التعلم وبعده	من 8.1% إلى 94.2%

القيم مأخوذة من ملفات التدريب الفعلية؛ دقة الاختبار على MNIST لا تعني ضمان النتيجة لكل خط يدوي أو صورة هاتف.

1. المشكلة والهدف التعليمي

يتعامل الحاسوب مع صورة الرقم باعتبارها قيمًا عددية، ولا يفهم اسم الرقم مباشرة. يهدف التطبيق إلى تصنيف رقم واحد من 0 إلى 9، وإظهار كيف يؤدي تجهيز المدخلات وتعلم الأوزان إلى نتيجة قابلة للملاحظة. الواجهة عربية ومن اليمين إلى اليسار، لتخدم عرض المشروع أمام الطلاب والمشرفة.

العصبون والأوزان ودالة التنشيط

يجمع العصبون المدخلات بعد ضربها في أوزان، ويضيف انحيازًا، ثم يمرر الناتج إلى دالة تنشيط. في مثال «هل أذهب إلى الكلية؟»: محاضرة مهمة بقيمة 1 ووزن 0.6، وجو جيد بقيمة 1 ووزن 0.3، وصديق بقيمة 0 ووزن 0.2؛ المجموع 0.9 يتجاوز العتبة 0.5، فيكون القرار «اذهب». هذا مثال توضيحي وليس خوارزمية تصنيف الصور المستخدمة في الموقع.

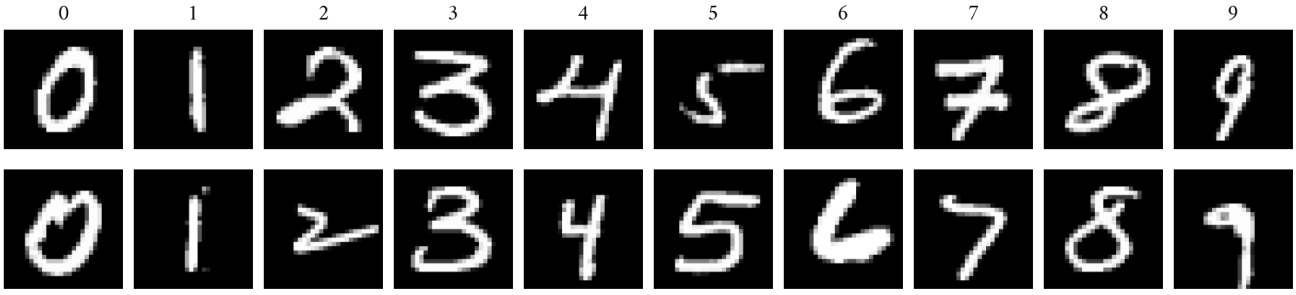
في المشروع نستخدم ReLU داخل الطبقات، وSoftmax في المخرج لإنتاج عشرة احتمالات مجموعها واحد. ينتج التدريب من حساب الخطأ والانتشار العكسي Backpropagation وتحديث الأوزان بمحسّن Adam. أما Dropout فيخفض الاعتماد المفرط على بعض الوحدات أثناء التدريب للحد من فرط التعلّم.

الدورة الخمسية: تجرّب ← تقارن ← تحسب الخطأ ← تعدّل الأوزان ← تعيد المحاولة.

بيانات MNIST وتقسيمها

توفر MNIST صورًا رمادية بحجم 28×28 للأرقام المكتوبة يدويًا: 60,000 صورة في قسم التدريب و10,000 في قسم الاختبار. توثيق مجموعة البيانات في Keras.

في هذا التنفيذ قُسم قسم التدريب إلى **54,000 صورة لتحديث الأوزان** و**6,000 للتحقق**. استُخدم التحقق لاختيار أفضل أوزان التدريب، ثم جرى التقييم النهائي على **10,000 صورة مستقلة**. لم تدخل صور الاختبار في تحديث أوزان النموذج الأساسي.



حُفظت البيانات محليًا في `data/mnist.npz`؛ لا يحتاج تشغيل الصفحات لتنزيلها مرة أخرى. المعرض يعرض عينات متنوعة، ولا يدّعي تساوي عدد أمثلة جميع الفئات.

2. تصميم النظام والنموذج

تتكون الواجهة الجديدة من HTML وCSS وJavaScript مستقلة عن خدمة Python المبنية على FastAPI. استُبدلت واجهة Streamlit السابقة بطلب المستخدم لتثبيت تفاعل الصفحة؛ تغيير أدوات الرسم لا يعيد تنفيذ الصفحة ولا يبدأ التعرف تلقائيًا. تصل طلبات العمليات الصريحة عبر Nginx إلى الواجهة البرمجية، وتُحدّث الواجهة الجزء المعني فقط.

يُحمّل نموذج CNN عند بدء الخدمة، قبل قبول الزيارات، ويُشارك بكاش آمن للتزامن. تعمل الخدمة بصورة دائمة حتى عندما لا يوجد متصفح متصل. التدريب التعليمي مهمة خلفية مستقلة عن التنقل داخل الموقع، ولا يبدأ إلا بطلب المستخدم.

المستخدم ← الرسم أو الصورة ← القص والتحجيم والتوسيط ← الشبكة العصبية ← النتيجة والتصحيح

تُحوّل الصورة إلى تدرج رمادي، وتُعكس الخلفية عند الحاجة ليصبح الحبر فاتحًا. تُنظف صورة الرفع بعتبة أوتسو، ثم يُقص المحتوى، ويُحجّم مع الحفاظ على تناسبه بحيث يكون الضلع الأطول 20 بكسل، ويوضع داخل 28×28. يُنقل مركز كتلة الحبر إلى المركز، وتُطبّع القيم إلى المجال من صفر إلى واحد. تكيف شدة التنعيم مع حجم الحبر حتى لا تمحو تفاصيل الأرقام الصغيرة.

بنية الشبكة الأساسية

شبكة التفاف CNN بأربع طبقات التفاف: 32 ثم 32 ثم 64 ثم 64 مرشحًا، كل مرشح 3×3. تفصل مجموعتي الالتفاف طبقات تجميع وإسقاط، ثم تسطيح وطبقة كاملة الاتصال تضم 128 وحدة، ثم عشرة مخارج. تشترك طبقات الالتفاف في الأوزان عند البحث عن الأنماط، بعكس نموذج Dense البسيط المستخدم للتدريب الحي. شرح طبقات الالتفاف في TensorFlow.

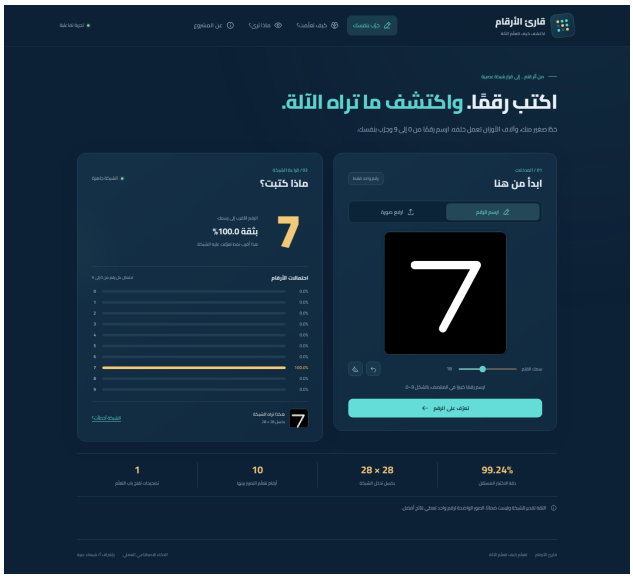
الطبقة	الوظيفة	عدد المعاملات
الطبقة 1	طبقة التفاف (تكتشف الأشكال)	320
الطبقة 2	طبقة التفاف (تكتشف الأشكال)	9,248
الطبقة 3	طبقة تجميع (تصغير)	0
الطبقة 4	طبقة إسقاط (تحد من الحفظ الأعمى)	0
الطبقة 5	طبقة التفاف (تكتشف الأشكال)	18,496
الطبقة 6	طبقة التفاف (تكتشف الأشكال)	36,928
الطبقة 7	طبقة تجميع (تصغير)	0
الطبقة 8	طبقة إسقاط (تحد من الحفظ الأعمى)	0
الطبقة 9	طبقة تسطيح (تحويل إلى قائمة)	0
الطبقة 10	طبقة كاملة الاتصال	131,200
الطبقة 11	طبقة إسقاط (تحد من الحفظ الأعمى)	0
الطبقة 12	طبقة كاملة الاتصال	1,290
المجموع	من النموذج المحفوظ	**197,482**

أُضيف أثناء التدريب فقط دوران حتى نحو 10 درجات، وإزاحة وتكبير طفيفان. حُفظت الأوزان بصيغة keras.، مع سجل التدريب ومصفوفة الالتباس في ملفات مستقلة.

3. صفحات التطبيق: التجربة والتعريف

جرّب بنفسك

لوحة رسم بحبر فاتح على خلفية داكنة تدعم الماوس واللمس والتراجع، وتبويب لرفع صورة رقم مكتوب على ورقة. يُرسل الرسم عند ضغط زر التعرف فقط، وتبقى أدوات الإدخال مستقلة عن عرض النتيجة. تظهر النتيجة كبيرة، ومعها نسب الفئات العشر ومعاينة المدخل 28×28. يؤدي المسح إلى إزالة الرسم والنتيجة. وتُعرض الأخطاء بالعربية دون كشف التتبّع التقني للمستخدم.



عند اختيار «الشبكة أخطأت» يحدد المستخدم التصحيح، فيُحفظ مثال مع رقمه الصحيح وتوقع الشبكة والثقة ومصدر الصورة. لا يتغير النموذج تلقائيًا بمجرد إضافة التصحيح؛ إعادة التدريب على التصحيحات تطوير مستقبلي يحتاج مراجعة البيانات.

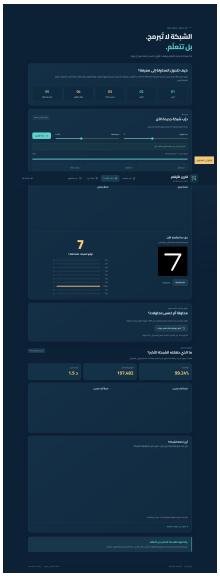
عن المشروع

تعرض الصفحة المادة وأسماء أعضاء المجموعة والمشرفة، والأدوات ومسار العمل والدورة الخمسية. توفر سياقًا مختصرًا يشرح فكرة المشروع قبل التجربة.

4. صفحات التطبيق: التعلم ورؤية البكسلات

كيف تعلّمت؟

يبدأ زر التدريب مهمة خلفية تنشئ شبكة Dense مستقلة، وتعرض دقتها قبل أي تعلم ثم التقدم والدقة والخطأ بعد كل دورة. يمكن إيقاف التدريب بأمان والتنقل بين الصفحات دون إنهاء المهمة، ومقارنة دورة واحدة بخمس دورات بالبذرة والعينة نفسيهما. تُحفظ الحالة والقياسات في SQLite وترتبط بملف ارتباط جلسة موقّع. تحتفظ الخدمة بآخر ثلاثة نماذج تعليمية في الذاكرة؛ يمكن تجربة النموذج المتاح فقط، بينما تبقى قياساته بعد زواله. عند إعادة تشغيل الخدمة تُعلّم المهمة الجارية بأنها منقطعة ولا يستأنف التدريب تلقائيًا. ويعرض قسم منفصل نتائج CNN الفعلية وطبقاتها ومصفوفة الالتباس.

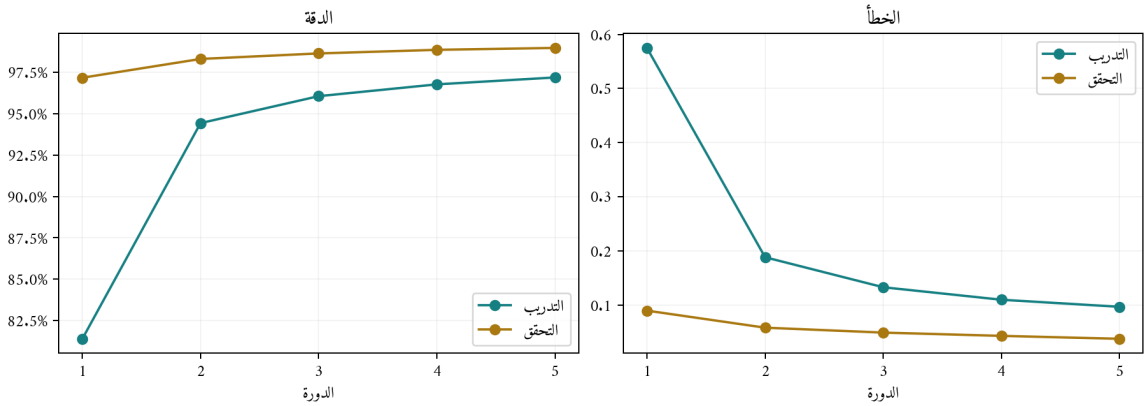


ماذا ترى الشبكة؟

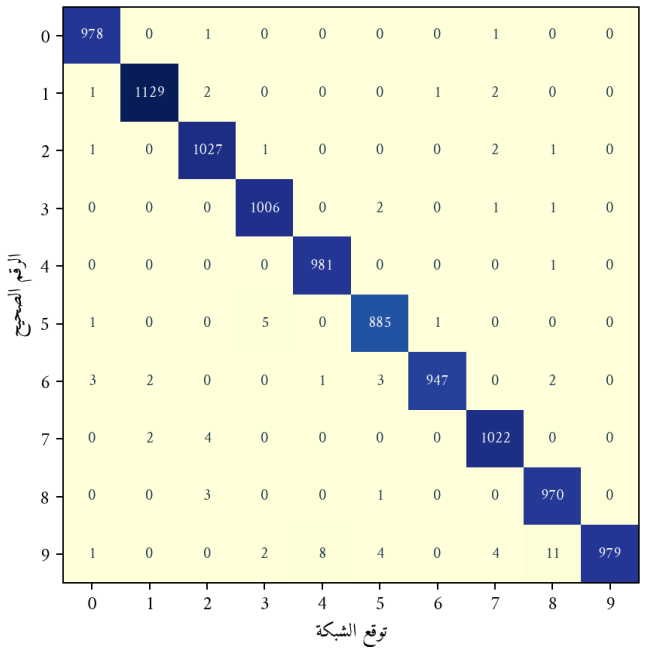
تعرض رحلة الصورة حتى مدخل 28×28، وخريطة تفاعلية لقيم البكسلات، وتوقع النموذج. وتعرض أمثلة التدريب مرتبة بالأرقام، وأوزان المرشحات الـ32 للطبقة الأولى. تمثل ألوان المرشح أوزانه بعد تطبيعها للعرض، وليست صورًا لأرقام محفوظة.

5. النتائج المقيسة

بلغت دقة الاختبار النهائي **99.24%**، أي **9,924 إجابة صحيحة** و**76 خطأً من 10,000 صورة**. قيمة خسارة الاختبار **0.02201**. استغرق التدريب **89.55 ثانية** عبر **5 دورات** على CPU. هذه مدة التنفيذ المقيسة هنا وليست وعدًا بزمن ثابت على كل جهاز أو حمل تشغيل.



المنحنى الذهبي يمثل التحقق أثناء التدريب، وليس مجموعة الاختبار النهائية. قد تتجاوز دقة التحقق دقة التدريب لأن التعزيز والإسقاط يعملان أثناء التدريب. ويُعرض التقييم النهائي المستقل في المقياس أعلى الصفحة.



أكثر زوجين اختلطا بالجمع في الاتجاهين: **8 و9 بعدد 11**، ثم **4 و9 بعدد 8**. قد يفسر تشابه الحلقات أو الذيول بعض الالتباس؛ هذا تفسير شكلي محتمل، وليس سببًا أثبته النموذج لكل مثال.

في اختبار التدريب الخلفي من متصفح الموقع المنشور على 6,000 مثال و1,000 مثال تقييم، ارتفعت الدقة من **8.1%** إلى **94.2%** خلال **1.90 ثانية** وبعد **5 دورات**، واستمر التدريب أثناء التنقل داخل الموقع. يستخدم هذا العرض شبكة مختلفة أصغر؛ لذلك لا تُقارن دقته مباشرة بدقة CNN على الاختبار الكامل.

6. التحديات والاختبارات وحدود الاستخدام

التحدي	الحل المنفذ
اختلاف الرسم عن صور التدريب	قص المحتوى، الحفاظ على التناسب، التوسيط بمركز الكتلة، وتعزيز التدريب
رقم صغير في زاوية اللوحة	تحجيم بعد القص وتنعيم يتكيف مع حجم الحبر
صورة ورقة أو خلفية معكوسة	تقدير لون الخلفية من الحواف وعتبة أوتسو عند الرفع
ملف تالف أو لوحة فارغة	تحقق من المدخل ورسائل عربية وسجلات تقنية منفصلة
تكلفة التدريب المتزامن	قفل مشترك للتدريب وحدود لحجم العينة والدورات والخيوط
إعادة تحميل النموذج	تحميل عند إقلاع الخدمة وكاش مشترك آمن للتزامن؛ التدريب الحي مستقل
إعادة تنفيذ الصفحة أثناء التفاعل	واجهة HTML/CSS/JavaScript مستقلة وتحديث جزئي للنتائج
انقطاع المتصفح أثناء التدريب	مهمة خلفية مع SQLite واستعادة الحالة وإيقاف آمن
الانقطاع عن خدمات خارجية	بيانات MNIST وخطوط الواجهة محلية

تشمل الاختبارات شكل المدخل والقيم ومركز الكتلة، والخلفيات والشفافية، والتصغير والعكس، والكتابة الآمنة للتصحيحات، وتحميل النموذج والتنبؤ، وتحديثات التدريب الحي، وعزل ملكية المهام، والمقارنة العادلة، والإلغاء وحفظ النتائج بعد إعادة تشغيل مدير المهام. اجتازت الواجهة الجديدة 12 فحصًا وظيفيًا من متصفح الدومين العام، شملت الرسم بالماوس واللمس ورفع الرقم 7، وثبات اللوحة أثناء التنقل واستعادة الرسم بعد التحديث، والتراجع والمسح، والتدريب الخلفي والإلغاء والمقارنة، والصفحات الأربع على الهاتف. لم يرسل الرسم طلبات تعرف قبل ضغط الزر، ولم يُحمّل Plotly على الصفحة الرئيسية. وعولج تداخل رأس الصفحة على الهاتف ثم أعيدت الفحوص. لا تمثل الاختبارات الآلية كل ظروف جهاز المستخدم أو الشبكة.

اجتازت المجموعة الكاملة **76 اختبارًا خلال 9.77 ثانية**، مع أربعة تحذيرات إهمال من الاعتماديات ودون اختبارات فاشلة. وفي اختبار الخدمة الحية، أُرسلت 20 عملية تعرف بتزامن أربعة طلبات؛ بلغ متوسط الاستجابة **59 ملي ثانية** وأقصاها **102 ملي ثانية**. أعاد معرض البيانات 60 مثالًا، وعرض فحص الصورة مراحل المعالجة الست. هذه قياسات ظروف التنفيذ الفعلية وليست ضمان أداء مطلقًا.

دقة MNIST لا تكفي لضمان دقة صور الهاتف، وخاصة عند وجود ظلال أو أرقام متعددة أو قص غير مناسب. المشروع يصنف رقمًا واحدًا، ولا يقرأ أعدادًا متعددة الخانات أو نصوصًا أو الحروف العربية. كما أن نسبة Softmax تمثل ترجيح النموذج وليست ضمانًا لصحة التوقع.

المقصود بالعمل دون تنزيل خارجي أن السيرفر لا يحتاج خدمات بيانات أو خطوط أثناء الاستخدام؛ الوصول إلى الموقع البعيد نفسه يحتاج اتصالًا بالإنترنت. للحضور احتياط عملي: لقطات شاشة وفيديو للعرض عند انقطاع اتصال القاعة.

7. النشر والتشغيل

يستمع FastAPI عبر عامل Uvicorn واحد على `127.0.0.1:8502`، ويستقبل Nginx طلبات الدومين عبر HTTPS. تُقدّم أصول HTML وCSS وJavaScript محليًا، وتصل تحديثات التدريب دون إعادة تحميل الصفحة. يمنع قفل خاص تشغيل مديرَي مهام على قاعدة SQLite نفسها. تشغّل خدمة `digit-reader` التطبيق بمستخدم مخصص، وتعيده تلقائيًا عند التعطل وبعد إقلاع السيرفر. تُحفظ الإعدادات الحساسة في `env.`، ولا تُضمّن في الأرشيف المصدري.

في اختبار استعادة نسخة التجهيز، أوقفت العملية عمدًا بإشارة SIGKILL؛ أعادها systemd خلال **7.53 ثانية**. بقيت الجلسة الموقعة صالحة، واستُعيدت المهمة السابقة باعتبارها منقطعة، وتطابقت بصمات جميع ملفات النموذج والتصحيحات قبل الاختبار وبعده. لا يُعاد التدريب تلقائيًا عند إعادة التشغيل.

```
sudo systemctl status digit-reader
sudo systemctl restart digit-reader
sudo journalctl -u digit-reader -n 50
```

لتشغيل نسخة مصدرية جديدة: أنشئ بيئة افتراضية، وثبت requirements.txt، ثم جهز السر والأصول المحلية بالأمر python scripts/prepare_web.py، ودرب النموذج بالأمر python -m core.train وشغّل make run. يُحفظ WEB_SECRET ثابتًا في env. لتوقيع الجلسات؛ تغييره يبطل الوصول من الجلسات القديمة. يُستخرج أصل Plotly من المكتبة المثبتة لتبقى الحزمة المصدرية صغيرة. الأرشيف المصدري لا يضم النموذج أو البيانات أو الأسرار، لذلك لا يعمل التعرف قبل إنشاء النموذج. على السيرفر المنشور تُدار الخدمة بالأوامر السابقة بدل تشغيل نسخة أخرى على المنفذ نفسه.

8. الخاتمة والتطوير المستقبلي

قدم المشروع تطبيقًا عمليًا للتعرف على الرقم، وتجربة تعليمية توضح تعديل الأوزان ورحلة الصورة إلى أرقام. يجمع التنفيذ نموذجًا أساسيًا مدربًا، ومعالجة قابلة للفحص، وواجهة عربية، وتشغيلًا دائمًا على الخادم.

التطوير المقترح: مراجعة التصحيحات وتجهيز مجموعة إضافية تمثل رسوم المستخدمين، ثم قياس تحسنها على اختبار مستقل قبل اعتماد نموذج محسّن. أما الحروف العربية أو قراءة عدة أرقام فتحتاج بيانات وأهداف تدريب مختلفة، ولا تُعد قدرات موجودة في الإصدار الحالي.

خريطة الملفات

المجلد أو الملف	الغرض
`web/api.py` و`web/jobs.py`	الواجهة البرمجية والتدريب الخلفي وحفظ حالة المهام
`web/static/`	صفحات HTML وأنماط CSS ومنطق JavaScript والأصول المحلية
`app.py` و`pages/`	واجهة Streamlit السابقة محفوظة للرجوع
`core/`	الإعدادات والبيانات والمعالجة والنماذج والتدريب والتصحيحات
`ui/`	مكونات الواجهة السابقة ومصدر الخط العربي المحلي
`models/`	النموذج والسجل والمقاييس ومصفوفة الالتباس
`data/`	بيانات MNIST والأمثلة والتصحيحات وقاعدة مهام SQLite
`deploy/` و`scripts/`	الخدمة والوكيل العكسي والتشغيل والفحص والتغليف
`tests/` و`docs/`	التحقق الآلي والتقرير واللقطات

مصادر النتائج وإعادة توليد التقرير

يعتمد هذا التقرير على `models/metrics.json` و`models/history.json` و`models/confusion.json`، وعلى جدول الطبقات المستخرج من النموذج المحفوظ بواسطة `model_summary_ar`. تعتمد تجربة التدريب الخلفي المنشور على جولة المتصفح الموثقة في `docs/audit/live_metrics.json`. أما `docs/التحقق.md` فيحفظ قياسات التجربة الأصلية. حُدثت اللقطات من الواجهة الجديدة في `docs/screenshots/` بعد اجتياز الفحوص الوظيفية. يتحقق مولد التقرير من أن مجموع مصفوفة الالتباس يطابق عدد الاختبار ودقته قبل بناء المستند.

ينشئ `scripts/build_report.py` المنحنيات وملف PDF من النص المراجع باستخدام بيئة التوثيق المنفصلة التي تضم Matplotlib وMarkdown وWeasyPrint ودعم تشكيل العربية. لا تحتاج بيئة تشغيل الموقع إلى مكتبات التوثيق هذه. استُخدمت WeasyPrint 65.1 مع pydyf 0.11.0 لإنتاج نص عربي قابل للقراءة والبحث، وخط Amiri للتقرير والمخططات، ومكتبة النظام Pango لعرض العربية. ويتحقق PyMuPDF من إمكانية استخراج النص العربي بعد التوليد.

المراجع

1. مجموعة MNIST، أعمال Yann LeCun وCorinna Cortes وChristopher J. C. Burges، صفحة MNIST الأصلية وواجهة التحميل في Keras.
2. دليل الشبكات العصبية الالتفافية في TensorFlow.

3. دورة حياة الخدمة في FastAPI. استُخدمت لتهيئة نموذج الاستدلال قبل قبول الطلبات وإغلاق التدريب بأمان.
4. الشفرة المصدرية للمشروع وملفات قياساته الفعلية بتاريخ إعداد التقرير.