الجمهورية العربية السورية

وزارة التعليم العالي والبحث العلمي

الجامعة الافتراضية السورية

برنامج ماجستير علوم الحاسوب

بناء نظام ذكي لتقييم جودة الخيوط القطنية

بحث مقدّم لنيل درجة ماجستير تأهيل وتخصص في علوم الحاسوب

إعداد الطالب:

غاندي غازي أحمد

ghandi_275004

إشراف الدكتور المهندس:

رائوف حمدان

2025

كلمة شكر

" مَنْ لَمْ يَشْكُرِ المَخْلُوق ... لَمْ يَشكُرِ الخَالِق"

أتقدَّمُ بجزيلِ الشُّكرِ لكلِّ مَنْ ساعدني في إنجازِ هذا البحث:

- **الدكتورة المهندسة: سيرا أستور** *مديرة برنامج علوم الحاسوب (MCS)*
- **الدكتور المهندس: رائوف حمدان** *المشـرف العلمي*
- **الدكتور المهندس: محمد مازن المحايري** *رئيس لجنة التحكيم*
- **الدكتور المهندس: باسل الخطيب** *عضو لجنة التحكيم*
- **المهندس: معتز جنوب** *زميل في برنامج علوم الحاسوب (MCS)*
- **ربا الشبقي** *زميلة في برنامج علوم الحاسوب (MCS)*

الإهداء

إِلَى

مَنْ

أُحِبُّ ...

فهرس المحتويات

Contents

1. الفصل الأول: الإطار العام للأطروحة 11

1.1. المقدّمة INTRODUCTION 11

1.2. مشكلة البحث وأسئلته 12

1.3. أهداف البحث 13

1.4. أهمية البحث ومبرراته 14

1.5. الدراسات السابقة ذات الصلة 14

1.6. الفجوة البحثية ومساهمة الدراسة الحالية 15

1.7. الخلاصة CONCLUSION 16

2. الفصل الثاني: المواصفات التكنولوجية للغزول القطنية 17

2.1. تمهيد: 17

2.2. طريقة إجراء الاختبار 18

2.3. تصنيف وتقييم مظهرية الخيوط 18

2.4. الخلاصة CONCLUSION: 20

3. الفصل الثالث: الصورة الرقميّة ومعالجتها 21

3.1. تمهيد 21

3.2. أساسيات اللون وآلية الرؤية: 21

3.3. رقمنة الألوان (DIGITIZATION): 24

3.4. الصورة الرقمية: 25

3.5. معالجة الصورة الرقمية 27

3.5.1 المراحل الأساسية لمعالجة الصور الرقمية 27

3.5.2. مراحل معالجة الصور الرقمية 28

3.6. الخلاصةCONCLUSION 31

4. الفصل الرابع: الذكاء الصناعيّ وتعلّم الآلة 32

4.1. تمهيد في الذكاء الصناعي: 32

4.2. تعلم الآلة والمكتبات البرمجية المستخدمة في بناء منصة تصنيف الخيوط القطنية 35

4.2.1. تمهيد في تعلّم الآلة 35

4.2.2. فروع تعلم الآلة 36

4.2.2.1. التعلم المراقب (Supervised Learning) 36

4.2.2.2. التعلم غير المراقب (Unsupervised Learning) 38

4.2.2.3. التعلم المعزز (Reinforcement Learning) 40

4.3 التعلم العميق والشبكات العصبية التلافيفية 41

4.3.1. الشبكات العصبية التلافيفية (Convolutional Neural Networks - CNNs) 41

4.3.2. أمثلة على معماريات CNNs 44

4.4 المكتبات البرمجية المستخدمة في بناء المنصة 45
4.4.1. مكتبات معالجة وتوليد الصور 45
4.4.2. مكتبات تدريب وتعليم النماذج المخصصة لتقييم صورة الخيط 47
4.4.3. مكتبات Django المستخدمة في بناء المنصة 51
4.4.5. المكاتب العامة المساعدة المستخدمة في بناء المنصة 53
4.5. توليد نماذج التصنيف 54
4.5.1. تقسيم مجموعات البيانات لتدريب نماذج التعلم العميق وتحسين تقييم الأداء 54
4.5.2. تدريب وتقييم نماذج التعلم العميق لتصنيف الصور باستخدام نقل التعلم 54
4.5.2.1. تجهيز البيانات وزيادتها 55
4.5.2.2. نقل التعلم (Transfer Learning) 56
4.5.2.3. حلقة التدريب والتقييم 56
4.5.3. استدلال نماذج التعلم العميق لتصنيف الصور: معالجة البيانات وتقنيات التنبؤ 57
5. القسم العملي (الدراسة التجريبيّة) 59
5.1. تمهيد 59
5.2. توليد الصور اللازمة لعملية تدريب النماذج 59
5.3. توليد نماذج التصنيف 62
5.3.1. تقسيم مجموعات البيانات لتدريب نماذج التعلم العميق وتحسين تقييم الأداء 62
5.3.2. تدريب وتقييم نماذج التعلم العميق لتصنيف الصور 62
5.4. تقييم أداء نماذج التصنيف باستخدام مصفوفة الارتباك CONFUSION MATRIX 70
5.4.1. تمهيد 70
5.4.2. مكونات مصفوفة الارتباك 70
5.4.3. مقاييس الأداء المستخلصة من مصفوفة الارتباك 71
5.4.4. نتائج تطبيق مصفوفة الارتباك على نماذج التصنيف الناتجة 72
5.5. الدراسة التحليلية والتصميمية للمنصة المطلوبة 76
5.5.1. تحليل النظام Analysis 76
5.5.1.1. المتطلبات الوظيفية (Functional Requirements) 76
5.5.1.2. متطلبات الأداء (Performance Requirements) 77
5.5.1.3. قيود التصميم (Design Constraints) 77
5.5.1.4. متطلبات الواجهة (Interface Requirements) 77
5.5.2. حالات الاستخدام Use Cases 78
5.5.2.1 الممثلون (Actors) 78
5.5.2.2. قائمة حالات الاستخدام الرئيسية 78
5.5.2.3. صلاحيات المستخدمين 78
5.5.2.4 عمليات التصنيف (Workflow) 79
5.5.2.5 مواصفات حالات الاستخدام (Use Case Specifications) 79
5.5.2.6 رسم مخطط حالات الاستخدام (Use Cases Diagram) 80
5.5.2.7 مخطط التسلسل (sequence) لأنشطة المستخدم 81
5.6. إنشاء قاعدة بيانات المنصة 82

5.6.1. توصيف قاعدة البيانات (Database Specification) 82
5.6.2. تحديد العلاقات بين الكيانات: 84
5.6.3. مخطط كيان-علاقة (ERD): 84
5.7. إنشاء المنصة على الخادم المحلي 84
5.7.1. شجرة الموقع 84
5.7.2 توصيف ملفات الموقع 85
5.7.3 آلية الربط المتزامن بين الواجهة الأمامية والخلفية في منصة Django الذكية 97
5.7.3.1 الواجهة الأمامية القوالب (Django Templates) ودورها في إرسال الطلبات 98
5.7.3.2 الواجهة الخلفية (Django - urls.py, views.py, models.py) 98
5.7.3.3 مسار البيانات خلال عملية التصنيف (دورة الطلب-الاستجابة) 99
6. مناقشة النتائج، التوصيات، والآفاق المستقبلية 101
6.1. مناقشة النتائج 101
6.2. التوصيات والمقترحات 103
6.3. الآفاق المستقبلية للمشروع 104
6.4. الخاتمة 105
7. المراجع العلمية 106
8. الملاحق 111
8.1. الملحق (1): توليد 250 صورة جديدة من صورة معيارية واحدة 111
8.2. الملحق (2): توليد 250 صورة جديدة من صورة معيارية واحدة 113
8.3. الملحق (3): دمج الصور الناتجة عن الكودين السابقين 114
8.4. الملحق (4): تقسيم الصور للقيام بعملية التدريب 115
8.5. الملحق (5): تدريب نماذج التصنيف الخمسة 116
8.6. الملحق (6): خوارزمية تقييم نماذج التدريب والمستخدمة في البحث 119
8.7. الملحق (7): الخوارزمية المستخدمة في رسم مصفوفة الارتباك لكل نموذج 121

فهرس الأشكال

الشكل (2-1): جهاز مظهرية الغزول القطنية 18

الشكل (2-2): المقارنة بين عينة الخيط القطني وصور التقييم المعيارية 19

الشكل (2-3): الصور المعيارية للسلسلة الخامسة حسب المواصفة القياسية المعتمدة 20

الشكل (3-1): مجال أطوال الأمواج المرئية ضمن مجال الأشعة الضوئية 21

الشكل (3-2): آلية رؤية اللون الأحمر 22

الشكل (3-3): امتصاص الضوء من مخاريط اللون الأحمر، الأخضر والأزرق 23

الشكل (3-4): مزج الألوان بالإضافة (A) وبالطرح (B) 24

الشكل (3-5): مكعب الألوان RGB 25

الشكل (3-6) عملية التقاط الصورة الرقمية، (A) مصدر الطاقة (الإضاءة)، (B) عنصر المشهد، (C) نظام التصوير، (D) إسقاط المشهد على مستوي الصورة، (E) الصورة الرقمية 26

الشكل (3-7): عناصر مصفوفة الصورة الرقمية (البكسلات) 26

الشكل (3-8): مستويات معالجة الصورة الرقمية 27

الشكل (3-9): مراحل معالجة الصور الرقمية 29

الشكل (4-1): نظرة شاملة لفروع الذكاء الصناعي 34

الشكل (4-2): نظرة شاملة لفروع تعلّم الآلة 35

الشكل (4-3): بنية عامة لشبكة عصبية تلافيفية (CNN) 41

الشكل (4-4): عملية الالتفاف (Convolution Operation) 43

الشكل (4-5): عملية التجميع الأقصى (Max Pooling Operation) 43

الشكل (5-1): بعض الأمثلة على التحويلات الناتجة عن توسيع البيانات المتقدمة باستخدام Albumentations 61

الشكل (5-2): البدء بعملية تدريب النموذج الخاص بتصنيف السلسلة الثانية 63

الشكل (5-3): الانتهاء من عملية تدريب النموذج الخاص بتصنيف السلسلة الثانية 64

الشكل (5-4): منحني تدريب السلسلة الأولى 64

الشكل (5-5): منحني تدريب السلسلة الثانية 65

الشكل (5-6): منحني تدريب السلسلة الثالثة 65

الشكل (5-7): منحني تدريب السلسلة الرابعة 66

الشكل (5-8): منحني تدريب السلسلة الخامسة 66

الشكل (5-9): منحني دقة النموذج الأول 67

الشكل (5-10): منحني دقة النموذج الثاني 67

28الشكل (5-11): منحني دقة النموذج الثالث 68
29الشكل (5-12): منحني دقة النموذج الرابع 68
30الشكل (5-13): منحني دقة النموذج الخامس 69
31الشكل (5-14): مصفوفة الارتباك الخاص بنموذج التصنيف للسلسلة الأولى..... 74
32الشكل (5-15): مصفوفة الارتباك الخاص بنموذج التصنيف للسلسلة الثانية 74
33الشكل (5-16): مصفوفة الارتباك الخاص بنموذج التصنيف للسلسلة الثالثة 75
34الشكل (5-17): مصفوفة الارتباك الخاص بنموذج التصنيف للسلسلة الرابعة 75
35الشكل (5-18): مصفوفة الارتباك الخاص بنموذج التصنيف للسلسلة الخامسة 76
36الشكل (5-19): مخطط حالات الاستخدام 80
37الشكل (5-20): مخطط التسلسل لعمليات المستخدم 81
38الشكل (5-21): مخطط كيان-علاقة للمنصة المطلوبة 84
39الشكل (5-22): الهيكل العام للمشروع (PROJECT STRUCTURE) 85
40الشكل (5-23): الصفحة الرئيسية للمنصة 88
41الشكل (5-24): صفحة إنشاء حساب جديد 89
42الشكل (5-25): صفحة تسجيل الدخول 89
43الشكل (5-26): صفحة DASHBOARD المستخدم 90
44الشكل (5-27): صفحة PROFILE المستخدم..... 90
45الشكل (5-28): صفحة رفع الصورة وإدخال النمرة 91
46الشكل (5-29): صفحة عرض النتيجة 91
47الشكل (5-30): صفحة تظهر إضافة النتيجة الجديدة إلى DASHBOARD 92
48الشكل (5-31): صفحة تظهر النتائج بشكل ملف PDF 92
49الشكل (5-32): صفحة تسجيل الخروج..... 93
50الشكل (5-33): صفحة مشرف الموقع..... 93
51الشكل (5-34): صفحة إدارة الموقع 94
52الشكل (5-35): صفحة تُظهر الصور المُحمّلة 94
53الشكل (5-36): صفحة تظهر PROFILE المستخدمين المسجلين 95
54الشكل (5-37): صفحة لتعديل بيانات المستخدمين 95
55الشكل (5-38): صفحة تظهر تأكيد عملية تعديل بيانات المستخدم 96
56الشكل (5-39): صفحة تظهر بيانات المستخدم بعد تعديلها من قبل مشرف الموقع 96

57الشكل (5-40): صفحة تظهر المستخدمين المسجلين في الموقع 97
58الشكل (5-41): آلية الربط بين واجهة وخلفية الموقع 100

فهرس الجداول

1الجدول (2-1): مجالات نمر السلاسل المعيارية لمظهرية الغزول 17
2الجدول (5-1): البنية الأساسية لمصفوفة الارتباك 71
3الجدول (5-2): نتائج مقاييس مصفوفة الارتباك لنموذج التصنيف الخاص بالسلسلة الأولى..... 72
4الجدول (5-3): نتائج مقاييس مصفوفة الارتباك لنموذج التصنيف الخاص بالسلسلة الثانية 73
5الجدول (5-4): نتائج مقاييس مصفوفة الارتباك لنموذج التصنيف الخاص بالسلسلة الثالثة 73
6الجدول (5-5): نتائج مقاييس مصفوفة الارتباك لنموذج التصنيف الخاص بالسلسلة الرابعة 73
7الجدول (5-6): نتائج مقاييس مصفوفة الارتباك لنموذج التصنيف الخاص بالسلسلة الخامسة 73
8الجدول (5-7): المتطلبات الوظيفية للنظام 76
9الجدول (5-8): المتطلبات الأداء الخاصة بالنظام 77
10الجدول (5-9): متطلبات واجهة النظام 77
11الجدول (5-10): المتطلبات الوظيفية للنظام 78
12الجدول (5-11): المتطلبات الوظيفية للنظام 78
13الجدول (5-12): مواصفات قاعدة بيانات كيان المستخدم USER 82
14الجدول (5-13): مواصفات قاعدة بيانات الكيان USERPROFILE 83
15الجدول (5-14): مواصفات قاعدة بيانات الكيان UPLOADEDIMAGE 83
16الجدول (5-15): العلاقة بين الكيانات 84
17الجدول (5-16): توصيف وظائف ملفات HTML المستخدمة في المنصة 88

الملـخّـص

تتناول هذه الأطروحة المعضلة الجوهرية للتقييم الذاتي وغير الموضوعي لمظهرية الخيوط القطنية في صناعة الغزل والنسيج، وذلك وفقاً للمواصفة القياسية ASTM D2255-09. يعتمد التقييم التقليدي على الخبرة البشرية والمقارنة البصرية، مما يؤدي إلى عدم الاتساق، ويُسبب أخطاء في تحديد الجودة والتسعير، وينتج عنه هدر كبير في الوقت والتكاليف.

يهدف هذا البحث إلى تطوير نظام ذكي وموضوعي بالكامل لتقييم مظهرية الخيوط، مستثمراً القدرات التحليلية للرؤية الحاسوبية والذكاء الاصطناعي. لتحقيق استقلالية تامة عن التدخل البشري وتحسين دقة التقييم، جرى تصميم وتدريب خمسة نماذج بالاعتماد على الشبكات العصبونية التلافيفية (CNNs)، وهي من أبرز تقنيات التعلم العميق في مجال تحليل الصور، وذلك لإجراء تصنيف دقيق لمظهرية الخيوط. بعد التقييم الشامل للنماذج، تم بناء منصة ويب تفاعلية لتمكين المصانع من تحقيق تقييم مهني وموضوعي للجودة. سيسهم هذا النظام في توفير ملموس في الموارد والوقت والكلفة، بالإضافة إلى تحسين مستمر في جودة المنتج النهائي. يشكل هذا البحث تطبيقاً عملياً للتقنيات الحديثة في تقييم خواص الخيوط، ويقدم قاعدة انطلاق قوية للأبحاث المستقبلية في قطاع الصناعات النسيجية.

الكلمات المفتاحية: مظهرية الخيوط القطنية، معالجة الصور الرقمية، تعلّم الآلة، التعلّم العميق، تصنيف آلي، منصة ويب.

1. الفصل الأول: الإطار العام للأطروحة

An Overview of The Research

1.1. المقدّمة Introduction

تتميّز الغزول القطنية بالعديد من المواصفات التكنولوجية التي تُحدّد جودة الغزل. تُعدّ مظهرية الغزل من المواصفات الهامة في تحديد جودته. تعتمد عملية تصنيف الخيوط حسب المواصفات القياسية المعتمدة (ASTM- D2255-09) [1] على الأسلوب اليدوي، حيث تتمّ عملية تقييم مظهرية الغزل بالطريقة التقليدية حيث يقوم العامل بإعطاء درجة لمظهرية الغزول بالمقارنة البصرية بين العينة المُختبرة والصورة المعيارية وبالتالي تزداد احتمالية الوقوع في الخطأ بحسب قدرات العامل القائم على عملية التقييم. من الجدير بالذكر أن أسعار الخيوط القطنية تختلف تبعاً لجودتها، وطالما أن عملية التقييم تتم بالاعتماد على العامل البشري، فإن ذلك يجعل عملية تحديد أسعار الخيوط مرتبطة بقدرة العامل على تقييم جودة الخيوط بشكل الموضوعي.

من جهة أخرى، تُعدّ معالجة الصور الرقمية من العلوم الحديثة التي تتضمّن العديد من التقنيات والخوارزميات المتطوّرة باستمرار والقادرة على استخلاص المعلومات من الصور الرقمية. فكما يُقال: *"صورة واحدة تُغنِي عن أكثر من عشرة آلاف كلمة"* [2]. من جهة أخرى يعدّ الذكاء الصناعي من علوم الحاسب الآلي الحديثة والمتطوّرة، والذي يقوم بمحاكاة عمليات الذكاء التي تتم داخل الدماغ البشري. للذكاء الصناعي تطبيقات عديدة كالإدراك، التعليل، التنبّؤ، التصنيف، التقييم، التعرّف على الأنماط، واتخاذ القرارات... الخ.

إنّ التوجّه العام للدراسات والأبحاث الحديثة يعتمد على ربط العديد من العلوم بآن واحد لإتمام العملية البحثية وتجاوز المشاكل التي تواجه الباحثين. وبالمحصّلة ضمان زيادة جودة المنتج والحفاظ عليها بأقل كلفة وزمن ممكن. لذلك تسعى الشركات المنتجة إلى الابتعاد عن تدخل العامل البشري في عملية تقييم جودة المنتج، بغية الحفاظ على الموضوعية والمهنية في اتخاذ القرارات. لذلك لا بدّ من الاستعانة بتقنيات وأدوات جديدة وغير تقليدية للحصول على تقييم دقيق وصحيح بعيداً عن الأخطاء المرافقة لطرق التقييم التقليديّة المُعتمدة على العامل البشري. من خلال هذا البحث ستتم أتمتة عملية تصنيف جودة الخيوط القطنية باستخدام تقنيات معالجة الصور الرقميّة وباستخدام الذكاء الصناعي.

تتضمّن هذه الأطروحة خمسة فصول، فيما يلي لمحة موجزة عن كل فصل. يتضمّن *الفصل الأوّل* أدبيات البحث والذي يتكوّن من الإطار العام للأطروحة (المشكلة التي يُعالجها البحث، أهداف البحث، وأهمية ومبررات البحث) ومن الدراسات والأبحاث المرجعية المتعلّقة بموضوع البحث. يتناول *الفصل الثاني* الدراسة التكنولوجية لبنية الغزل القطني والمواصفات والاختبارات على مظهرية الغزول الناتجة. يُقدِّم *الفصل الثالث* آلية الرؤية لدى الإنسان وأساسيات الصورة الرقمية وتقنيات معالجة الصورة الرقمية المستخدمة بالبحث. يُعرّف *الفصل الرابع* أدوات الذكاء الصناعي المُستخدمة بالبحث. يُبيِّن *الفصل الخامس* القسم العملي والدراسة التجريبية للبحث والنتائج التي تمّ التوصّل إليها عن طريق تطبيق تقنيات الذكاء الصناعي ومعالجة الصورة الرقمية للوحة الغزل. فيما يتناول *الفصل السادس* الأخير مناقشةً للنتائج التي تمّ الحصول عليها والتوصيات الناتجة عن هذا البحث بالإضافة إلى التطلعات والآفاق المستقبلية التي يمكن بناؤها من هذا البحث.

1.2. مشكلة البحث وأسئلته

تتمّ عملية تقييم مظهرية الغزل وفقاً للمواصفة القياسية (ASTM- D2255-09) بالطريقة التقليدية. حيث يقوم العامل بلفّ عينة غزل على لوحة سوداء باستخدام جهاز مظهرية الغزول. ثمّ يقوم بإعطاء درجة لمظهرية الغزول بالمقارنة البصرية بين العينة المُختبرة والصور المعيارية المرافقة للمواصفة القياسية. حيث يوجد سلاسل معيارية لصور مظهرية الغزول، كل سلسلة تُمثّل مجال معيّن لنمر الغزول. تحتوي السلسلة الواحدة أربع صور معيارية (A (best)، B, C, D (worst) . باستخدام الطريقة التقليدية في تقييم مظهرية الغزول تزداد احتمالية الوقوع في الخطأ حيث تتم عملية التقييم بالاعتماد على المَلكات الشخصية للعامل مما يجعلها عرضة لتقييم خاطئ وإعطاء درجة لمظهرية الغزل قد لا تُمثِّل الحقيقة. وهنا تكمن المشكلة التي يسعى البحث لمعالجتها فبسبب محدودية عدد الخبراء وما يحملونه من خبرات، وصعوبة إيجاد النظرة المنظمة للوصول إلى التقييم الموضوعي الأمثل؛ كل ذلك يؤدّي إلى زيادة الوقت والتكاليف في تقييم جودة الغزل. كل ذلك يؤدي إلى خفض جودة المنتج وهدر في الوقت والجهد وزيادة في كلفة الإنتاج مقابل منتج قد يتم تنخيبه بمستوى متدنٍ.

يمكن طرح الأسئلة التي يسعى البحث للإجابة عليها على الشكل التالي:

- هل يمكن تقييم درجة مظهرية الغزل باستخدام تقنيات معالجة الصورة الرقمية والذكاء الصناعي دون تدخّل العامل البشري؟
- هل يمكن تطوير عملية تقييم مظهرية الغزل لتوافق الواقع أكثر من الطريقة التقليدية؟
- هل يمكن بناء منصة على الويب لاستثمار النظام الذكي المقترح بطريقة فعالة؟

1.3. أهداف البحث

يهدف البحث إلى الحصول على خوارزميات تقوم بتقييم موضوعي ودقيق لمظهرية الغزل بعيداً عن التدخل البشري في اتخاذ القرار. مما يؤدي إلى تجنّب الأخطاء المرافقة لعملية التقييم بالطريقة التقليدية. ستُبنى الخوارزميات باستخدام أدوات معالجة الصورة الرقمية والذكاء الصناعي. وطالما أن الصورة الرقمية تحتوي الكثير من البيانات، لذا سيتم استخلاص المعلومات من الصورة الرقمية لتدريب نظام ذكي يقوم بتقييم مهرية الغزل. كل ذلك سيؤدي إلى توفير معلومات هامة عن جودة الغزل المُختبر. كل ذلك يؤدي إلى توفير الوقت والجهد والكلفة وتحسين جودة المنتج عن طريق إجراء تغذية خلفية دائمة لخطوط الإنتاج وتزويد القائمين عليها بالمعلومات المتعلقة بجودة المنتج مما يدفعهم إلى تحسين مستمر في عمليات الإنتاج للحصول على الجودة المطلوبة.

من الأهداف غير المباشرة للبحث إجراء تطبيق عملي للاستفادة من قدرات الذكاء الصناعي في قطاع الصناعات النسيجية. حيث تتميّز العلاقة بين مواصفات الألياف ومتغيرات العملية الإنتاجية بأنها غير خطية. وبالتالي فإن إنشاء نماذج لإيجاد هذه العلاقة يستلزم حل شبكة مترابطة من التعقيدات. وليس من السهولة فهم كيفية سلوك الألياف بعد غزلها. لذا فإنّ التطبيق العملي لهذا البحث يفتح أفقاً جديداً في تطوير تكنولوجيا الغزل عن طريق تسليط الضوء على إمكانية إدخال الذكاء الصناعي في هذه التكنولوجيا. فإذا تمكّن هذا البحث من معالجة بعض الحالات (وهي: تقييم مظهرية الغزل)، يمكن بعد ذلك تعميم تطبيقات الذكاء الصناعي لحل المشاكل التي تواجه قطاع الصناعات النسيجية.

1.4. أهمية البحث ومبرراته

هناك حاجة متزايدة عالمياً لخفض زمن انتظار المنتج إذ يجب أن تُؤدّى النشاطات في المعمل بالتوازي (التكامل في تطوير المنتج). من هذه النشاطات: عملية الإنتاج وعملية تقييم جودة المنتج باستمرار وعملية التغذية الخلفية لتحسين جودة المنتج. كل ذلك يؤدّي إلى تلبية حاجات الزبون، وفتح أسواق جديدة تهتم بجودة المنتج، وابتكار تقنيات حديثة خلال مرحلة التصنيع لتطوير المنتج. لذا يبرز دور تقنيات الحاسوب وتطبيقاته لتحقيق هذه الغايات. فالنجاحات الواضحة التي حققتها البرمجيات الحاسوبية في مختلف التطبيقات العملية تشير إلى أهمية وتعاظم دورها في السنوات القادمة، حيث إنّ توظيف تقنيات البرمجيات الحاسوبية سيقود حتماً إلى أنظمة مكوّنة من آلات ذات ذكاء عالٍ.

يشكّل هذا البحث قاعدة يمكن الانطلاق منها في أبحاث متقدمة لمواجهة المشاكل التي تعترض قطاع الصناعات النسيجية. حيث يعتمد هذا البحث على التقنيات الحديثة كأدوات الذكاء الصناعي ومعالجة الصورة الرقمية للتعرّف على بعض خواص الغزول القطنيّة وتقييم جودة المنتج. فبعد بناء خوارزميات البحث (خوارزميات تصنيف مظهرية الغزول القطنية) وإجراء الاختبارات اللازمة للتحقّق من دقة النتائج، يمكن لأي معمل غزل استثمار هذه الخوارزميات لتقييم جودة المنتج بطريقة مهنية وموضوعية دون تدخّل العامل في عملية التقييم. كل ذلك يؤدي إلى توفير بالجهد وكلفة الإنتاج والزمن والعمل على تحسين جودة المنتج باستمرار.

1.5. الدراسات السابقة ذات الصلة

لقد تناولت العديد من الدراسات السابقة جوانب مختلفة من فحص جودة المنسوجات باستخدام الرؤية الحاسوبية والذكاء الاصطناعي. يمكن تصنيف هذه الدراسات بناءً على تركيزها:

1) **الكشف عن عيوب القماش باستخدام التعلم العميق:** ركزت العديد من الأبحاث على استخدام CNNs للكشف عن عيوب الأقمشة. على سبيل المثال، استخدم (Adibpour et al) [3] شبكات CNN لتصنيف عيوب الأقمشة المنسوجة بدقة عالية. كما استعرضت (Liu et al) [4] أحدث التطورات في

فحص عيوب النسيج باستخدام التعلم العميق، مؤكدة على فعالية هذه التقنيات مقارنة بالطرق التقليدية.

2) **تصنيف عيوب الخيوط باستخدام الرؤية الحاسوبية**: بعض الدراسات تناولت تصنيف عيوب الخيوط. على سبيل المثال، اقترح (Zhang et al) [5] نظاماً يعتمد على معالجة الصور لاستخراج ميزات من صور الخيوط وتصنيف العيوب. ومع ذلك، غالباً ما تركز هذه الدراسات على عيوب محددة (مثل العقد أو التكتلات) بدلاً من المظهرية العامة للخيط وفقاً لمعيار قياسي.
3) **تطبيقات الذكاء الاصطناعي في صناعة النسيج**: دراسات أوسع نطاقاً ناقشت تطبيقات الذكاء الاصطناعي في جوانب مختلفة من صناعة النسيج، من التصميم إلى الإنتاج ومراقبة الجودة [6] .(Chen et al)هذه الدراسات تؤكد على الإمكانات التحويلية للذكاء الاصطناعي في هذا القطاع.
4) **نشر نماذج الذكاء الاصطناعي على الويب**: على الرغم من وجود العديد من الأمثلة على نشر نماذج التعلم العميق لتصنيف الصور عبر الويب في مجالات عامة (مثل تصنيف الحيوانات أو الأشياء)، إلا أن الدراسات التي تركز على نشر نماذج مخصصة لتقييم جودة المنسوجات أو الخيوط وفقاً لمعايير صناعية محددة (مثل ASTM) عبر منصات ويب تفاعلية لا تزال محدودة.

1.6. الفجوة البحثية ومساهمة الدراسة الحالية

بناءً على الاستعراض الشامل للأدبيات، يمكن تحديد الفجوات البحثية التالية:

1. التركيز على مظهرية الخيوط القطنية وفقاً لمعيار ASTM D2255–09: على الرغم من وجود أبحاث حول الكشف عن عيوب الخيوط، إلا أن هناك نقصاً في الدراسات التي تركز بشكل مباشر على أتمتة تصنيف مظهرية الخيوط القطنية باستخدام نماذج التعلم العميق بناءً على المعيار البصري القياسي ASTM D2255–09. معظم الدراسات تركز على عيوب محددة أو انتظام الخيط بدلاً من التقييم الشامل للمظهرية.
2. دمج التعلم العميق مع النشر على الويب: قلة الدراسات التي تجمع بين تطوير نماذج التعلم العميق لتصنيف مظهرية الخيوط ونشرها كمنصة ويب تفاعلية وسهلة الاستخدام للمستخدمين النهائيين في الصناعة. هذا الدمج ضروري لتمكين التطبيق العملي للحلول الذكية.

3. التحقق من صحة النموذج على بيانات حقيقية وفقاً لمعيار: تفتقر العديد من الدراسات إلى التحقق الشامل من أداء النماذج على بيانات حقيقية تم جمعها وتصنيفها بدقة وفقاً لمعيار صناعي موثوق به.

تُسهم الدراسة الحالية في سد هذه الفجوات من خلال:

1. **تطوير نموذج تصنيف ذكي:** بناء وتدريب نموذج تعلم عميق (باستخدام نقل التعلم على ResNet18) قادر على تصنيف مظهرية الخيوط القطنية بدقة بناءً على صور لوحات ASTM D2255–09.
2. **تصميم وتطوير منصة ويب متكاملة:** إنشاء منصة ويب تفاعلية تسمح للمستخدمين بتحميل صور لوحات مظهرية الخيوط والحصول على تصنيف فوري وموضوعي.
3. **التحقق العملي:** اختبار وتقييم أداء المنصة والنموذج على بيانات حقيقية لضمان فعاليتها في بيئة صناعية.

1.7. الخلاصة Conclusion

لقد استعرض هذا الفصل الأدبيات العلمية المتعلقة بتقييم جودة الخيوط، وتطبيقات الرؤية الحاسوبية والتعلم العميق في صناعة النسيج، بالإضافة إلى أهمية معيار ASTM D2255–09 ودور المنصات الويب في نشر نماذج الذكاء الاصطناعي. أظهر الاستعراض أن هناك حاجة واضحة لتطوير حلول آلية وموضوعية لتقييم مظهرية الخيوط القطنية وفقاً للمعايير الصناعية، ونشر هذه الحلول في صيغة قابلة للاستخدام. تُساهم الدراسة الحالية في معالجة هذه الفجوات من خلال تقديم منهجية متكاملة لتصميم وتطوير منصة ذكية لتصنيف مظهرية الخيوط، مما يعزز من كفاءة ودقة عمليات مراقبة الجودة في صناعة النسيج.

2. الفصل الثاني: المواصفات التكنولوجية للغزول القطنية

2.1. تمهيد:

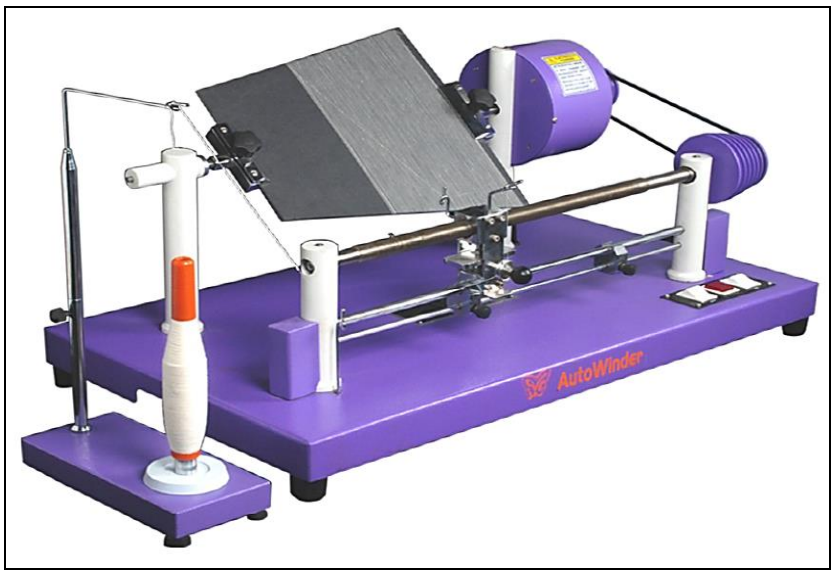
اعتمدت المواصفات القياسية لتقييم مظهرية الغزول القطنية لأول مرة في عام 1938 ونُقّحت في عام 1964 وكانت تتكون من خمس سلاسل من الصور. كل سلسلة تُمثّل مجال نمر غزول محدد وتتكون كل سلسلة من أربع صور، كل صورة تمثل درجة مظهرية الغزل (A, B, C, D) حيث الدرجة (A) هي الأفضل و(D) هي الأسوأ. تمّ تنقيح السلسلة الثالثة في عام 1975 مع الحفاظ على نفس نمر الغزول لكل سلسلة. وبعد تقييم التحسينات، تقرر مراجعة كل السلاسل وتضييق النطاق في السلاسل المتوسطة الأكثر نشاطاً بإضافة سلسلة جديدة وبذلك أصبحت المواصفة القياسية لتقييم المظهرية تتكون من ست سلاسل من الصور المعيارية. وبعد إنتاج هذه السلاسل الممثلة للمظهرية تقرر استخدام طباعة صورة الأوفست (offset photo printing) ونشرها في عام 1979. وقد تبيّن أن درجات المظهرية (A,B) من السلسلة الثانية لعام 1979 لم تكن واضحة لذا تم تصحيح السلسلة الثانية في عام 1987. وفي نفس العام 1987 أجريت دراسة استقصائية عالمية حول كيفية استخدام لوحات مظهرية الغزل. وفي ضوء نتائج الدراسة تقرر اعتماد لوحات المظهرية لتشمل تصنيف جميع أنواع الغزل المفردة. بالمحصلة، تتكون المواصفة القياسية لتقييم مظهرية الغزول القطنية من خمس سلاسل من الصور. كل سلسلة تُمثّل مجال نمر غزول محدد حسب الجدول (2–1). تتكوّن كل سلسلة من أربع صور معيارية، كل صورة تمثل درجة مظهرية الغزل (A, B, C, D) حيث الدرجة (A) هي الأفضل و(D) هي الأسوأ، بالمحصلة يوجد 20 صورة معيارية لتصنيف الخيوط القطنية.

/الجدول (2–1): مجالات نمر السلاسل المعيارية لمظهرية الغزول

السلسلة	مجالات نمر الغزول القطنية	
	نمرة الغزل المفرد (Ne)	tex
1	3's to 7's	200 to 84+
2	7+'s to 16's	84 to 37+
3	16+'s to 36's	37 to 16+
4	36+'s to 70's	16 to 8.4+
5	70+'s to 135's	8.4 to 4.4+

2.2. طريقة إجراء الاختبار

يتم اختبار مظهرية الغزول باستخدام جهاز مظهرية الغزول المُوضّح بالشكل (2–1) ووفقاً للمواصفة القياسية (ASTM D-2255) [1]. حيث يتم اختيار توضع السير الناقل بين المخروط القائد والمُقاد (A, B, C, D, E, F) بالعلاقة مع مجالات نمر الغزول في كل سلسلة معيارية. نتيجة اختيار المكان المناسب للسير الناقل، يتم تغيير عدد لفات الغزول المطلوبة في واحدة الطول على اللوحة السوداء، وذلك بسبب تغير نسبة نقل الحركة بين المخروط القائد والمخروط المُقاد.



الشكل (2–1): جهاز مظهرية الغزول القطنية

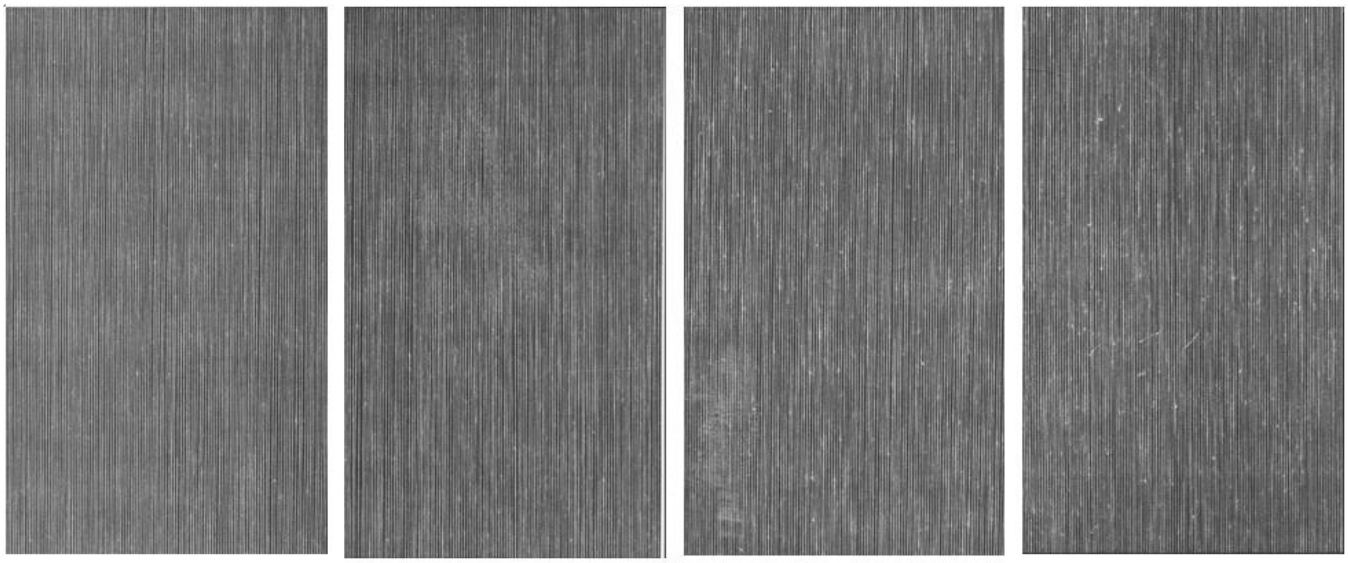
2.3. تصنيف وتقييم مظهرية الخيوط

يتم إزالة لوحة الاختبار من جهاز المظهرية ومقارنتها مع صور اللوحات المعيارية لنفس السلسلة التي تنتمي إليها نمرة الغزل المُختبر حسب المواصفة (ASTM-D2255-09) كما يوضح الشكل (2–2)، حيث يظهر المخبري وهو يقوم بالمقارنة بين عينة الغزل الملفوفة على اللوحة السوداء مع صور السلسلة المعيارية تحت درجة إضاءة مناسبة على أن يكون الضوء موزعاً توزيعاً منتظماً على جميع أجزاء اللوحة. ثم يقوم العامل بإعطاء درجة تقييم لمظهرية الغزول حسب درجة المظهرية الأقرب لعينة الاختبار (A, B, C, D). تُعرّف كل درجة من درجات تقييم المظهرية وتبعاً للمواصفة القياسية [1] كما يلي:

- الدرجة A: لا تحتوي على عقد طويلة، أي لا يزيد حجم العقد عن ثلاثة أضعاف قطر الخيط. تتمتع الغزول في هذه الدرجة بانتظامية جيدة. لا تحتوي على زغب زائد ولا على مواد غريبة.
- الدرجة B: لا تحتوي على عقد طويلة ولكن تحتوي على عقد صغيرة. لا تظهر أكثر من ثلاثة أجزاء من المواد الغريبة على كامل اللوح الأسود. تحتوي على زغب وعدم انتظامية أكثر بقليل من الدرجة A.
- الدرجة C: تحتوي على عقد أكثر وأطول وكمية أكبر من الزغب والمواد الغريبة بالمقارنة مع الدرجة B. والاختلاف بين الأماكن الثخينة والرفيعة وقطر الغزل تكون أكبر بالمقارنة مع الدرجة A.
- الدرجة D: تحتوي على العقد الكبيرة والتي يزيد حجمها عن ثلاثة أضعاف قطر الغزل. وتحتوي على زغب ومواد غريبة أكثر وعلى مناطق ثخينة ورفيعة متعددة.

الشكل (2-2): المقارنة بين عينة الخيط القطني وصور التقييم المعيارية2

يوضح الشكل (2-3) مثالاً للصور المعيارية الممثّلة للسلسلة الخامسة حسب المواصفة القياسية المعتمدة (ASTM-D2255-09).



3الشكل (2–3): الصور المعيارية للسلسلة الخامسة حسب المواصفة القياسية المعتمدة

2.4. الخلاصة Conclusion:

في هذا الفصل، يُلاحظ أن أسلوب تقييم مظهرية الغزل يعتمد على الأسلوب اليدوي مما يجعله عرضة لارتكاب الخطأ أثناء التقييم كونه يتعلق بمهارة ومَلكات المخبري والتي تقيدها العديد من الظروف المتغيرة. وهنا تظهر المشكلة التي يُحاول البحث حلها عن طريق تحييد العامل البشري أثناء القيام بعملية التقييم والاستعاضة عنه بأسلوب مؤتمت يعتمد على معالجة الصور الرقمية وخوارزميات الذكاء الصناعي من أجل الحصول على نتائج موضوعية لا تتعلق بالمتغيرات المحيطة بعملية الاختبار والتقييم.

يمكن اعتبار هذا الفصل قاعدة أساسية في هندسة الغزل لفهم مظهرية الغزل. مما يساعد بتطوير النموذج المنشود من هذا البحث للحصول على خوارزمية تصنيف مظهرية الغزول والتي ستعتمد على تقنيات معالجة الصور الرقمية والذكاء الصناعي بشكل أساسي.

3. الفصل الثالث: الصورة الرقميّة ومعالجتها

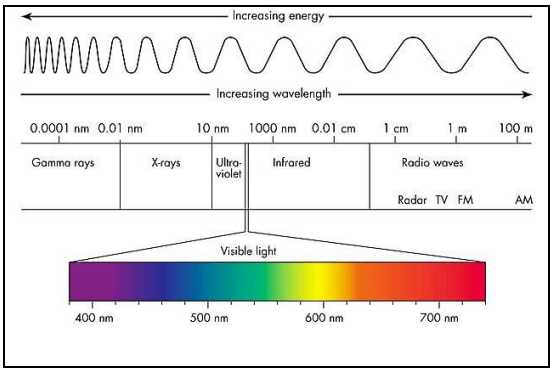
Digital Image & Processing

3.1. تمهيد

في هذا الفصل تم تعريف عدداً من مفاهيم الصورة وبعض التعابير الرياضية المستخدمة، مما يمنح فهماً واضحاً لطريقة تمثيل آلية رؤية الصورة ورقمنتها ومعالجتها. لذا في البداية تم التعرّف بشكل موجز على آلية جهاز الإبصار عند الإنسان بما في ذلك تشكّل الصورة في العين. ثم الانتقال إلى طريقة تحويل الصورة الطبيعية إلى صورة رقمية. ثم تم التعرّف على معالجة الصورة الرقمية. بالتالي سيساعد هذا الفصل على فهم كافٍ لسلوك الصورة مما يسهل التعامل معها، على اعتبار أن صور السلاسل المعيارية الخاصة بتصنيف مظهرية الخيوط بحاجة لمعالجة رقمية قبل تطبيق تقنيات الذكاء الصناعي عليها من أجل القيام بعملية مظهرية الخيوط القطنية.

3.2. أساسيات اللون وآلية الرؤية:

يتألف الضوء سواء أكان من مصادر طبيعية (كالشمس) أو مصادر صناعية، من أمواج كهرطيسية (electromagnetic waves) مختلفة بطول الموجة تنتشر على شكل كمّات من الطاقة تُدعى فوتونات (photons). يقع مجال الضوء المرئي ضمن أطوال موجية تتراوح بين (380-740 nm)، كما يظهر في الشكل (3-1) الحاوي على الطيف الكهرطيسي (The electromagnetic spectrum). يمكن تحليل الأشعة المرئية بتمريرها على موشور زجاجي إلى ألوانها الأساسية السبعة.



4الشكل (3-1): مجال أطوال الأمواج المرئية ضمن مجال الأشعة الضوئية

وبالتالي تختلف أطوال الموجات الضوئية الواردة على الجسم الملون باختلاف المصدر الضوئي، مما يسبب اختلافاً باللون المنعكس على العين. تُسمّى هذه الظاهرة (Metamerism) والتي تُعرّف على أنها اختلاف اللون باختلاف المصدر الضوئي [7]. يُعبّر عن الطيف الكهرطيسي باستخدام المعادلات (3–1) و(3–2) والمصطلحات التالية: طول موجة wavelength (λ)، الترددfrequency (ν) والطاقة energy (E) [8].

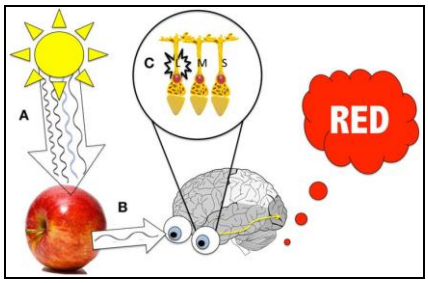
$$\lambda = \frac{c}{\nu} \qquad (1-3)$$

حيث C: سرعة الضوء، $c=2.998\times108$ m/s $\approx 3\times108$ m/s

$$E = h.\nu = \frac{h.c}{\lambda} \qquad (2-3)$$

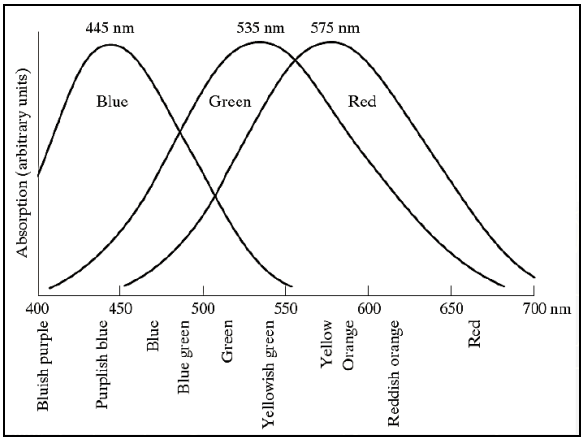
حيث h: ثابت بلانك، $h= 6.62606891\times10^{-34}$ J.s $\approx 6.63\times10^{-34}$ J.s. وتُقاس الطاقة (E) بواحدة الإلكترون–فولط (electron-volt).

يُوضّح الشكل (3–2) مثالاً لرؤية اللون الأحمر ممثلاً بتفاحة حمراء. حيث يظهر أنّ أشعة المصدر الضوئي (A) تصل إلى الجسم بعدة أطوال موجية مما يؤدي إلى تهيج الإلكترونات الحرة من الجسم وانطلاقها إلى سويات طاقية أعلى وعند عودة بعض هذه الإلكترونات إلى مستواها الأصلي تطلق من جديد أشعة بطول موجة جديد (B). أي في هذه الحالة، لا تصدر التفاحة الحمراء ضوءاً أحمراً وإنما تمتص التفاحة جميع ترددات الضوء المرئي الساقط عليها ما عدا مجموعة ترددات اللون الأحمر التي تنعكس عن التفاحة إلى أن تصل إلى الخلايا الحسية (C) في العين فتتحسّس اللون الأحمر وتدركها العين ويقرأ الدماغ أن لون الجسم هو اللون الأحمر. ترى العين البشرية التفاحة حمراء لأنها تستطيع التمييز بين أطوال الموجات المختلفة. *فاللون هو شيء في الدماغ وليس خاصية للأجسام.*



الشكل (3–2): آلية رؤية اللون الأحمر5

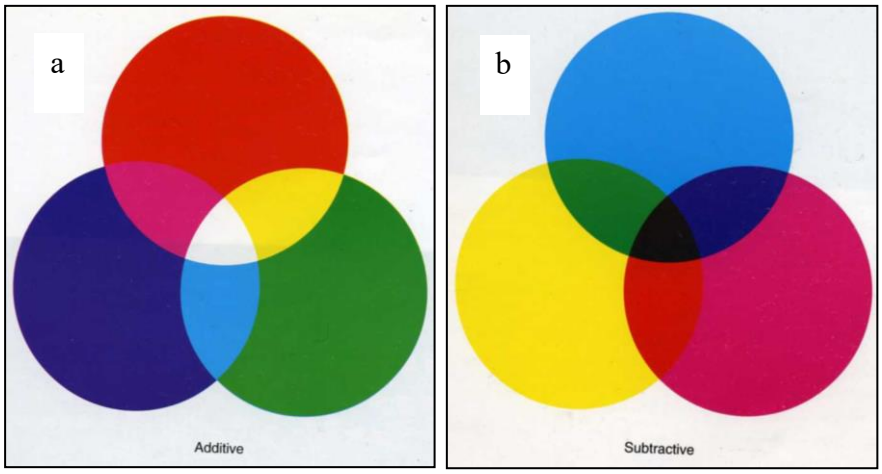
يُظهر الشكل (3–3) منحنيات امتصاص المخاريط للون الأحمر، الأخضر والأزرق تبعاً لطول موجة الضوء الساقط على العين [9].



6الشكل (3–3): امتصاص الضوء من مخاريط اللون الأحمر، الأخضر والأزرق

يُلاحظ من هذه المنحنيات أنه لا يوجد لون بمفرده يُدعى أحمر، أخضر، أو أزرق. وبالتالي لا يمكن الحصول على كامل ألوان الطيف من مزج الألوان الأولية (Primary Colors- RGB) فقط إلا إذا تم مراعاة طول الموجة الناتجة، ومن المبادئ المستخدمة لمزج الألوان [10]:

1) مزج الألوان بالإضافة Additive Color mixing: وهي الظاهرة الأساسية في إدراك اللون. وتتضمن إضافة الأطوال الموجية للأشعة الضوئية لتوليد قيمة أعلى للون. عندما تتداخل الألوان الأولية، تُمزج أطوالها الموجية مع بعضها وتولّد الضوء الأبيض. كما يُظهر بالشكل (3-4-a). وعندما يتداخل لونين فقط من الألوان الأولية تتجمع الأطوال الموجية لهذين اللونين لتشكيل الألوان الثانوية (Secondary colors) وهي الأرجوان (Magenta)، التركواز (cyan) والأصفر (yellow).
2) مزج الألوان بالطرح Subtractive Color Mixing: والتي تنتج باستخدام المواد الملونة (الصبغات، الحبر، الطلاء) وتقوم هذه المواد بإزالة الأطوال الموجية الساقطة من الأشعة الضوئية على المواد وتعكس أشعة ضوئية بطول موجة جديد مما يسمح برؤية لون جديد. عند مزج ألوان الملونات الأولية وهي الأرجوان (Magenta)، التركواز (cyan) والأصفر (yellow) مع بعضها ينتج اللون الأسود كما يوضح الشكل (3-4-b). يُلاحظ أن الألوان الثانوية في المبدأ الأول هي ألوان أولية في المبدأ الثاني. والألوان الأولية في المبدأ الأول هي ألوان ثانوية في المبدأ الثاني.



الشكل (3-4): مزج الألوان بالإضافة (*a*) وبالطرح (*b*)7

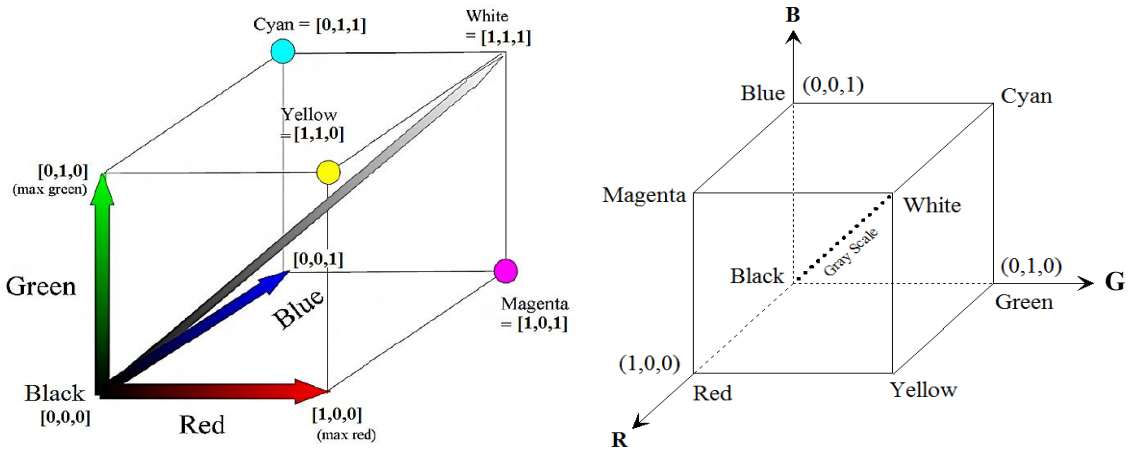
3.3. رقمنة الألوان (Digitization):

يمكن تمثيل الألوان بطريقة رقمية اعتماداً على آلية تمثيل الألوان بالصورة الطبيعية. فالألوان بالصورة الطبيعية تُدرك بالعين باستخدام المخاريط التي تتحسس اللون الأحمر(R)، الأخضر (G) والأزرق (B). وبناءً عليه يمكن تمثيل الألوان بالصورة الرقمية باستخدام نفس الألوان (RGB). يوجد عدة طرق لنمذجة الألوان من أجل استخدامها في العتاد الصلب (hardware) مثل الكاميرات وشاشات المراقبة والعرض، وتُستخدم أيضاً في التطبيقات التي تعالج الألوان مثل المخططات والرسومات. من أكثر طرق النمذجة شيوعاً نموذج RGB (Red, Green, Blue) المُستخدم في شاشات العرض والكاميرات، نموذج CMY (Cyan, Magenta, Yellow) و CMYK (Cyan, Magenta, Yellow, Black) التي تُستخدم في طباعة الألوان، ونموذج HSI (Hue, Saturation, Intensity) الذي يصف ويفسّر الألوان بشكل قريب من أسلوب الإنسان [2].

- **نموذج الألوان RGB:**

في هذا البحث سيتم استخدام نموذج RGB، لذا سيتم شرحه بشيء من التفصيل. في هذا النموذج، يظهر كل لون باستخدام المركبات الطيفية الأولية اللون الأحمر(R)، الأخضر (G) والأزرق (B). يعتمد هذا النموذج على نظام الإحداثيات الديكارتي كما يظهر في الشكل (3-5). حيث تتوضّع الألوان الأولية (The primary colors) في ثلاث زوايا وتكون إحداثياتها red=(1,0,0), green=(0,1,0), and blue=(0,0,1)، والألوان الثانوية (The secondary colors) بالزوايا الثلاث الأخرى وتكون إحداثياتها cyan=(0,1,1), magenta=(1,0,1) and

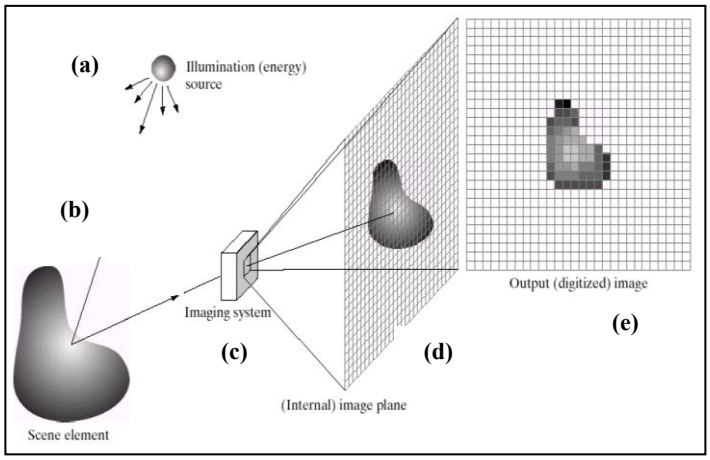
yellow=(1,1,0)، والأسود في مركز الإحداثيات black=(0,0,0)، والأبيض في الزاوية البعيدة عن مركز الإحداثيات white=(1,1,1) حيث تكون: R=G=B=1. ويمتد مقياس التدرج الرمادي (والذي يُمثِّل النقاط التي تكون فيها قيم R,G,B متساوية) من اللون الأسود إلى اللون الأبيض. وباقي الألوان تتوضّع إما على أو داخل مكعب الألوان وتُعرف بشعاع يمتد من مركز الإحداثيات. يظهر من مكعب الألوان أيضاً أن قيم كل من R, G, B تقع ضمن المجال [0،1]، بالتالي تُمثّل باقي الألوان بقيم تتراوح ضمن أبعاد المكعب (1،1،1). تجدر الإشارة إلى أن طبيعة نظام الألوان المتبع في RGB هو نظام الإضافة (additive) للحصول على ألوان الصورة.



الشكل (3–5): مكعب الألوان *RGB*8

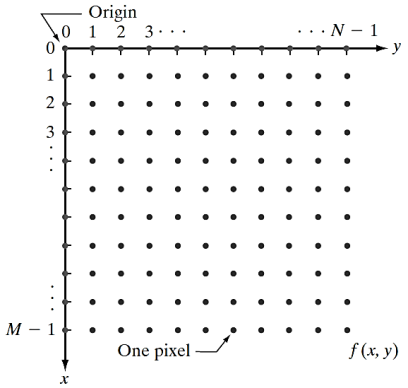
3.4. الصورة الرقمية:

يبين الشكل (3–6) الطريقة الرئيسية التي تُستخدم بها أجهزة مستشعرات المصفوفة لالتقاط الصورة. حيث يظهر طاقة مصدر الإضاءة (a) التي تنعكس من عنصر المشهد (b). تتمثل الوظيفة الأولى لنظام التصوير (c) في تجميع الطاقة الواردة من عنصر المشهد وتركيزها على مستوي الصورة. إذا كانت الإضاءة خفيفة ، فإن الواجهة الأمامية لنظام التصوير عبارة عن عدسة لتساعد في تركيز الطاقة وعرض المشهد المنظور على المستوي المحرقي للعدسة (d). تنتج مجموعة المستشعرات إشارات تتناسب مع الضوء المُستقبل من كل مستشعر. تقوم الدارات الرقمية والتناظرية في نظام التصوير بمعالجة مخرجات المستشعرات بعدة مراحل وتحويلها إلى صورة رقمية (e) [2]. عندما تُستخدم مصفوفة المستشعرات لالتقاط الصورة، فإنّ عدد المستشعرات في المصفوفة يحدد حدود الصورة في كلا الاتجاهين.



9الشكل (3–6) عملية التقاط الصورة الرقمية، (*a*) مصدر الطاقة (الإضاءة)، (*b*) عنصر المشهد، (*c*) نظام التصوير، (*d*) إسقاط المشهد على مستوي الصورة، (*e*) الصورة الرقمية

ممّا سبق يمكن الإشارة إلى الصور بواسطة مصفوفة ثنائية الأبعاد f(x,y). حيث قيمة f عند الإحداثيات المكانية (x, y) هي قيمة موجبة يُحدد معناها الفيزيائي بمصدر الصورة. فعندما يتم إنشاء صورة من عملية فيزيائية، تكون قيم عناصر الصورة متناسبة مع الطاقة المشعة بواسطة مصدر فيزيائي (مثل الأمواج الكهرطيسية). ونتيجة لذلك تكون قيمة f(x,y) غير صفرية ومنتهية. ويُلاحظ أنّ الصورة الرقمية تتكوّن من مصفوفة تحتوي عدداً محدداً من العناصر، لكل عنصر موقع معين (x,y) وقيمة معينة (f). يُشار إلى هذه العناصر على أنّها عناصر الصورة (picture elements)، (image elements)، (pels) و(pixels) كما يوضّح الشكل (3–7). يُعدّ البكسل (Pixel) المصطلح الأكثر استخداماً للدلالة على عناصر الصورة الرقميّة، ويُعرّف بأنّه أصغر عناصر الصورة وهو على شكل نقطة أو مستطيل، تتشكّل الصورة من مجموعة بكسلات في سطح ثنائي الأبعاد. كلّما كانت البكسلات صغيرة وبكثافة كبيرة زادت جودة الصورة.



10الشكل (3–7): عناصر مصفوفة الصورة الرقمية (البكسلات)

ممّا تقدّم يمكن كتابة مصفوفة صورة رقمية ذات أبعاد M×N وفق المعادلة (3−3).

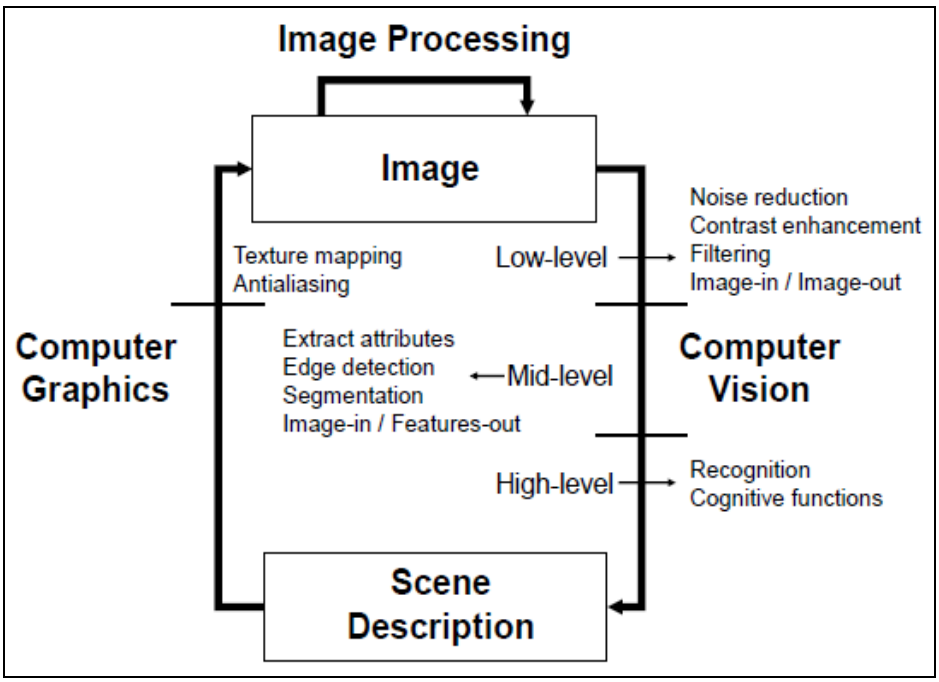
$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(\ 0,N-1\ ) \\ f(1,0) & f(1,1) & \cdots & f(\ 1,N-1\ ) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \quad (3-3)$$

3.5. معالجة الصورة الرقمية

تُعالج الصور الرقمية لعدّة أسباب منها ما يتعلق بتصحيح توازن الألوان، إعادة بناء وترميم الصور، إمكانية ضبط حجم الصورة من أجل العرض أو الطباعة، سهولة تخزين الصور الرقمية وإرسالها عبر الإنترنت، تحسين رؤية بعض الأمراض من الصور الشعاعية، استعادة الصورة عن طريق إزالة الخدوش من الصور القديمة، واستخلاص المعلومات المهمة من الصور.

3.5.1 المراحل الأساسية لمعالجة الصور الرقمية

يمكن تصنيف العمليات الحاسوبية التي تمر بها الصورة الرقمية أثناء المعالجة إلى ثلاث مستويات، كما يوضح الشكل (3−8) [11].

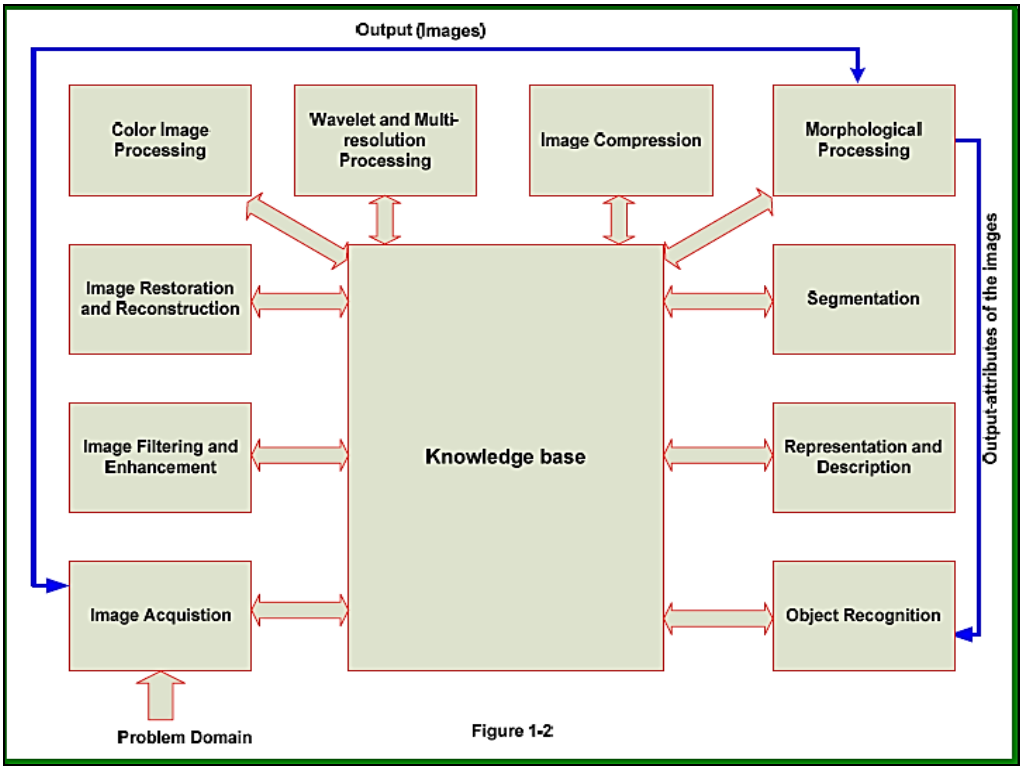


11الشكل (3−8): مستويات معالجة الصورة الرقمية

- *عمليات المستوى المنخفض (Low-level processes):* تشمل معالجة أولية للصور الرقمية (image preprocessing) لتقليل الضجيج (reduce noise)، تحسين التباين (contrast enhancement)، زيادة حدة الصورة (image sharpening)، الترشيح (filtering). يتميّز هذا المستوى أن المدخلات والمخرجات عبارة عن صورة.
- *المستوى المتوسط لمعالجة الصور (Mid-level processing):* وهو مستوى تحليل الصورة (image analysis). يتضمن هذا المستوى مهاماً مثل تجزئة الصورة (segmentation)، توصيف العناصر (description)، تصنيف عناصر الصورة بشكل إفرادي (classification). يتميّز هذا المستوى أن المدخلات تكون صورة والمخرجات تكون سمات ومميزات الصورة (كالحواف، والإطار المحيط..).
- *المستوى العالي لمعالجة الصور (high-level processing):* يتضمن هذا المستوى صناعة المشهد (making sense) من مجموعة العناصر المُدركة من مرحلة تحليل الصورة (image analysis). ويشمل هذا المستوى على تصنيف المشهد (classification)، الاكتشاف (detection)، الإدراك (recognition)، الملاحقة والتعقب (tracking). وفي النهاية تُنفّذ الوظائف المُدركة من جميع المستويات لتكوين الرؤية الصناعية.

3.5.2. مراحل معالجة الصور الرقمية

لإجراء معالجة للصور الرقمية لا بدّ من القيام بالمراحل التالية والموضحة بالشكل (3-9) [2].



12الشكل (3–9): مراحل معالجة الصور الرقمية

1) *الحصول على الصورة (Image Acquisition):* هي الخطوة الأولى من الخطوات الأساسية لمعالجة الصور الرقمية. وتشتمل دراسة طرق الحصول على الصورة من أجهزة خاصة (مثل الكاميرا الرقمية أو الماسح الضوئي...)، وأبسط أشكال هذه العملية هو الحصول على صورة رقمية من الحاسب.

2) *تحسين الصورة (Image Enhancement):* يُقصد بها العمليات التي يتم بها إيضاح بعض التفاصيل المهمة بالصورة أو التركيز على بعض السمات المهمة في الصورة. من الأمثلة على هذه المرحلة هو زيادة التباين في الصورة وذلك لأن البشر يرون في ذلك وضع أفضل لفهم محتوى الصورة.

3) *استعادة الصورة (Image Restoration):* تهتم هذه العملية بتحسين مظهر الصورة وتعتمد عملية التحسين هذه على بعض النماذج الرياضية والإحصائية لمعالجة الصورة بينما تعتمد عملية التحسين السابقة (Enhancement) على متطلبات بشرية مثل مقدار جودة الصورة بالنسبة لمُشاهِدها.

4) *معالجة الصور الملونة (Color Image Processing):* ترجع أهمية هذه المرحلة إلى كثرة استخدام الصور الملونة على الإنترنت، والحاجة دوماً إلى معالجتها بالإضافة إلى أنه يمكن استخلاص بعض السمات والخصائص من الصورة بناءً على اللون.

5) *المعالجة ذات الدقة المتعددة باستخدام الموجيات (Wavelets and Multi-Resolution Processing):* وهو حجر الأساس في تمثيل الصورة بدرجات متفاوتة من الدقة (resolution) وتُستخدم عادةً في ضغط الصور والبيانات حيث يتم تقسيم الصورة على التوالي إلى مناطق أصغر لضغط البيانات والتمثيل الهرمي.

6) *الضغط (Compression):* يُستخدم في اختزال حجم الذاكرة المطلوبة لتخزين الصورة أو امتداد الصورة (bandwidth) لتسهيل إرسالها عبر الإنترنت وتقليل حجم البيانات. مثل امتداد ملف jpg المستخدم في ضغط الصور (Joint Photographic Experts Group) JPEG .

7) *معالجة التشكيل (Morphological Processing):* تهتم هذه المرحلة باستخلاص مكونات الصورة والتي تكون مفيدة في تمثيل وتوصيف شكل معين بالصورة وتُعدّ هذه العملية أول العمليات التي يكون خرجها هو خصائص الصورة.

8) *التجزئة أو التقسيم (Segmentation):* تهتم هذه المرحلة بتقسيم الصورة إلى مكوناتها وعناصرها الجزئية. بشكل عام، تعد التجزئة واحدة من أصعب المهام في معالجة الصور الرقمية. حيث تؤدي التجزئة الخاطئة إلى جعل عملية المعالجة بعيدة عن الحل الناجح لمشاكل التصوير، كالعمليات التي تتطلب التعرف على الكائنات بشكل مستقل. يؤدي تطبيق خوارزميات التجزئة الضعيفة أو غير المنتظمة إلى الفشل في العملية النهائية لمعالجة الصورة. بشكل عام، كلما كانت التجزئة أكثر دقة، كلما كان من المرجح أن يتحقق النجاح في عملية المعالجة.

9) *التمثيل والتوصيف (Representation and Description):* وهي العملية التالية لعملية التقسيم (Segmentation). يتم فيها تمثيل مكونات الصورة بإحدى الطريقتين: التمثيل الإطاري (boundary of a region) الذي يهتم بالشكل الخارجي لعناصر الصورة، أو تمثيل مساحي (all the points in the region) الذي يهتم بالخصائص الداخلية لعناصر الصورة. بينما تهتم عملية

التوصيف (Description) باستخلاص مميزات وصفات الصورة والتي تكون ذات أهمية عالية في عملية التصنيف (classification).

10) *التعرف على العناصر (Object recognition):* تهتم هذه العملية بتسمية عنصر ما في الصورة بناءً على صفاته وميزاته.

11) *قاعدة المعرفة (Knowledge Base):* قد تكون المعرفة بسيطة مثل تفصيل مناطق الصورة التي تحتوي على المعلومات المفيدة، مما يحدّ من البحث الذي يجب إجراؤه عند البحث عن تلك المعلومات. كما يمكن أن تكون قاعدة المعارف معقدة للغاية، مثل قائمة مترابطة من العيوب الرئيسية المحتملة في مشكلة فحص المواد أو احتواء قاعدة البيانات على صور أقمار صناعية (satellite images) عالية الدقة لمنطقة مرتبطة بتطبيقات متغيرة.

3.6. الخلاصة Conclusion

في هذا الفصل، تم التعرّف على آلية الرؤية لدى الإنسان وكيفية تأثير طول الموجة الكهرطيسية للألوان على مخاريط العين (الحمراء، الخضراء، والزرقاء) مما يؤدي إلى إثارة هذه المخاريط لتوليد رؤية ملونة للصور والأشكال. ثم تم التعرّف على كيفية تحويل الألوان الطبيعية إلى ألوان رقمية، عن طريق أتمتة عملية الرؤية الطبيعية للصور لدى الإنسان للحصول على الصور الرقمية ومن أشهر هذه الطرق هو تمثيل الألوان بثلاث مستويات (R, G, B)، وهذا المبدأ مُستوحى من مخاريط العين البشرية. حيث يتم تحويل الصورة الطبيعية عن طريق أداة التقاط الصور الرقمية إلى مصفوفة ثلاثية الأبعاد. ثم تم الانتقال إلى معالجة الصور الرقمية، حيث هنالك الكثير من المعالجات الرقمية للصور ولكن تم التعرّف على التقنيات المفيدة للبحث والتي سيتم استخدامها في القسم العملي من هذا البحث.

إن إدراج هذا الفصل سيوفّر فهماً كافياً للتقنيّات التي ستستخدم في معالجة الصور الرقمية للغزول القطنية. بالتالي ستُهيّأ الصور الرقمية للوحات مظهرية الغزول لتكون مُدخلات لأدوات الذكاء الصناعي التي ستقوم بعملية التصنيف وفقاً للمواصفات المعتمدة. لذا لا بدّ من الحديث في الفصل القادم عن الذكاء الصناعي وبعض التقنيات المستخدمة لتحقيق الهدف من هذا البحث.

4. الفصل الرابع: الذكاء الصناعيّ وتعلّم الآلة

The Artificial Intelligence (AI) & Machine Learning (ML)

4.1. تمهيد في الذكاء الصناعي:

يُعرّف الذكاء الاصطناعي (Artificial Intelligence - AI) بشكل عام على أنه قدرة الآلة على محاكاة القدرات المعرفية البشرية، مثل التفكير، التعلم من الخبرة، حل المشكلات، والإدراك [14]. منذ منتصف القرن العشرين، ومع التطورات المتسارعة في قدرات الحوسبة، أظهرت الآلات إمكانات غير متوقعة في أداء مهام معقدة، مثل إثبات النظريات الرياضية المعقدة واللعب بمستوى عالٍ في ألعاب مثل الشطرنج [15]. على الرغم من التقدم الهائل في سرعة المعالجة وسعة التخزين، ظلت مرونة العقل البشري وقدرته على الاستنتاجات اليومية التلقائية تشكل تحدياً كبيراً للأنظمة الاصطناعية [16]. ومع ذلك، فقد نجحت بعض تطبيقات الذكاء الاصطناعي في مضاهاة مستوى أداء الخبراء البشريين في مهام محددة، مثل التشخيص الطبي، تطوير محركات البحث، والتعرف على الصوت والكتابة اليدوية [17].

يتكون الذكاء الاصطناعي من مجموعة واسعة من المعارف والعلوم الفرعية التي تتداخل وتتكامل لتشكل هذا المجال المعقد [18].

- **أنظمة التعلم (Learning Systems):** وتشمل تعلم الآلة (Machine Learning) والتعلم العميق (Deep Learning)، حيث تتعلم الأنظمة من البيانات لتحسين أدائها.
- **تمثيل المعرفة والاستدلال (Knowledge Representation and Reasoning):** يركز على كيفية تمثيل المعلومات بشكل يسمح للآلات بمعالجتها واستخلاص استنتاجات منطقية.
- **التخطيط (Planning):** يتعلق بقدرة الأنظمة على تحديد تسلسل الإجراءات اللازمة لتحقيق أهداف معينة.
- **اكتساب المعرفة (Knowledge Acquisition):** يُعنى بأساليب جمع وتنظيم المعرفة اللازمة للأنظمة الذكية.
- **الحوسبة اللينة (Soft Computing):** وهي مجموعة من المنهجيات التي تتعامل مع عدم اليقين والتقريب، وتتضمن:

- **المنطق الضبابي (Fuzzy Logic):** للتعامل مع المفاهيم غير الدقيقة والغامضة.
- **الشبكات العصبية الاصطناعية (Artificial Neural Networks - ANNs):** نماذج مستوحاة من الدماغ البشري للتعلم من البيانات.
- **الخوارزميات الجينية (Genetic Algorithms):** أساليب تحسين مستوحاة من التطور البيولوجي.

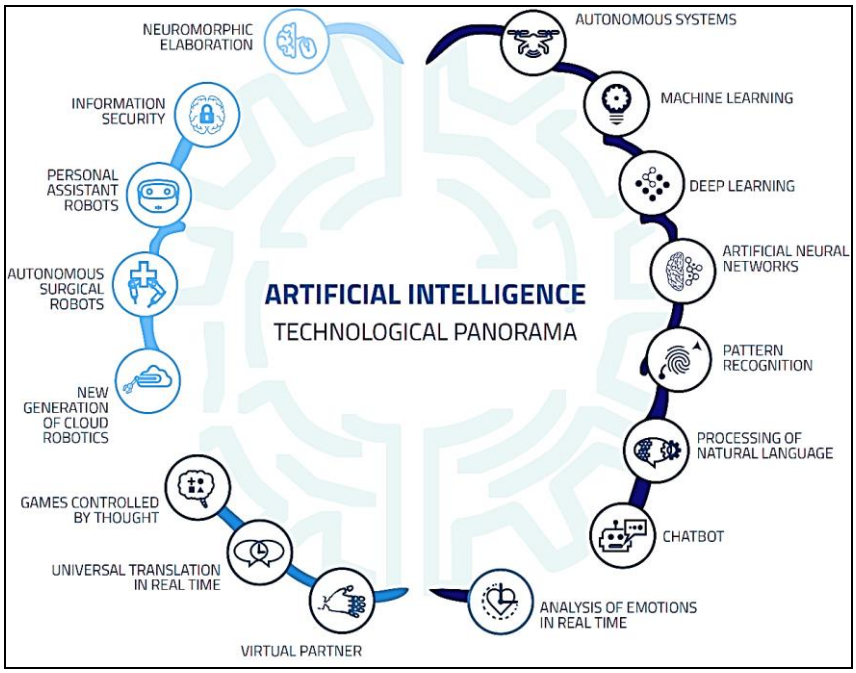
- **الذكاء الاصطناعي المنطقي (Logical AI):** يركز على استخدام المنطق الرسمي لتمثيل المعرفة والاستدلال.
- **البحث الذكي (Intelligent Search):** يتناول خوارزميات البحث الفعالة في فضاءات الحلول المعقدة.
- **البرمجة المنطقية (Logic Programming):** لغة برمجة تعتمد على المنطق الرسمي.
- **إدارة مسائل عدم اليقين (Management of Imprecision and Uncertainty):** يختص بالتعامل مع البيانات غير الكاملة أو غير المؤكدة.

تاريخياً، شهد مجال الذكاء الاصطناعي تحولات مهمة. ففي الفترة من 1949 وحتى أواخر الستينيات، عمل المهندس الكهربائي الأمريكي آرثر صموئيل (Arthur Samuel) على تطوير الذكاء الاصطناعي من مجرد التفاعل مع الاحتمالات إلى مرحلة التعلم من التجربة [19]. هذا التطور فتح الباب أمام تطبيقات متقدمة في مجالات معقدة، مثل تحليل مجموعات الجينوم الكبيرة لمنع الأمراض، وتشخيص الاكتئاب بناءً على أنماط الكلام، وتحديد الأفراد ذوي الميول الانتحارية [20].

في الثمانينيات، برزت الشبكات العصبية الاصطناعية (Neural Networks) كمنهجية قوية للتعلم، خاصة لتعلم القواعد من البيانات الأولية دون برمجة صريحة [21]. تعتمد هذه الشبكات على بنية شبيهة بآلية عمل الدماغ البشري، حيث تتكون من طبقات من العصبونات المترابطة. وقد أدت قدرتها على التعلم من الأنماط المعقدة إلى استخدامها في مجالات واسعة مثل الاتصالات، الخدمات المصرفية، الطب الحيوي، الكشف عن المخدرات، تحليل البصمة الوراثية، اكتشاف العقاقير الجديدة في الصيدلة، خدمات البحث عن الصور، فهم اللغة البشرية للبحث الصوتي، وتطبيقات السيارات ذاتية القيادة [22]. هذا التوسع في التطبيقات دفع الشركات، خاصة تلك المتخصصة في معالجة الرسوميات (GPUs)، إلى التنافس الشديد في تطوير المزيد

من تطبيقات التعلم العميق والذكاء الاصطناعي نظراً لقدرة وحدات معالجة الرسوميات الفائقة على تسريع عمليات التدريب المعقدة.

يُعد التعلم الآلي (Machine Learning) التجسيد العملي للذكاء الاصطناعي، حيث يقلل من الحاجة إلى البرمجة اليدوية الصريحة للآلة لكل سيناريو محتمل [23]. ومع التقدم إلى مستويات أعلى وأكثر تطوراً من التعلم الآلي، ظهر ما يُسمى **التعلم العميق (Deep Learning)**. يتطلب التعلم العميق بنية معقدة من الشبكات العصبية متعددة الطبقات (شبكات عصبية عميقة) لتقليد الشبكات العصبية في الدماغ البشري، مما يمكنه من فهم الأنماط والسلوكيات المعقدة وأبعادها في الأوضاع الطبيعية، وحتى في وجود الضوضاء أو التفاصيل المفقودة [24]. نظراً لأن التعلم العميق يحتاج إلى كميات هائلة من البيانات) البيانات الضخمة - Big Data)وقوة حاسوبية هائلة، فقد أدى ذلك إلى تطورات مثل مشروع Google Brain في عام 2012، حيث تم بناء شبكات عصبية ضخمة (باستخدام 16000 معالج ومليار وصلة) وتدريبها على 100 مليون صورة ثابتة من موقع YouTube للقيام بعملية تصنيف الصور [24,25]. كان هذا بمثابة بداية عصر جديد لتطبيقات التعلم العميق في التعرف على الوجوه، الصوت، والأنماط التي يتعرف عليها الإنسان بشكل فطري. يبيّن الشكل (4-1) نظرة تكنولوجية شاملة لفروع الذكاء الصناعي.

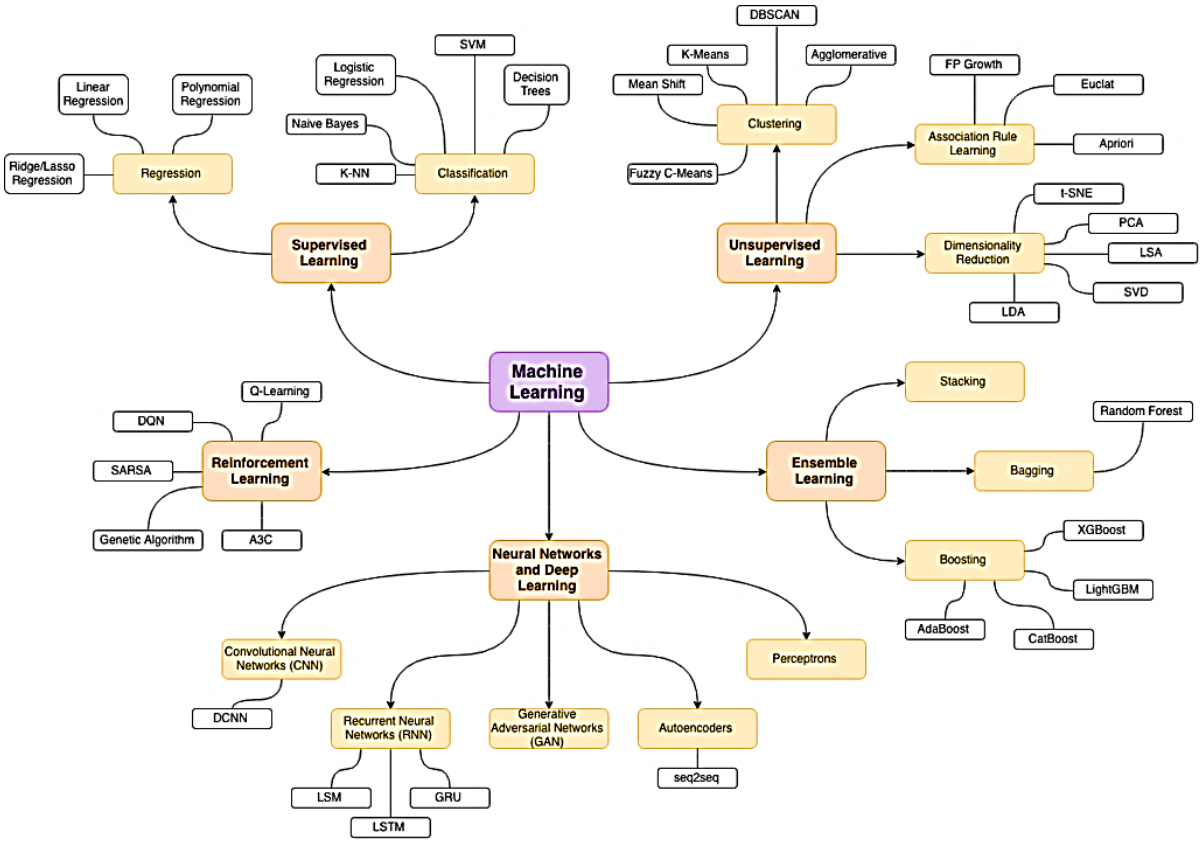


13الشكل (4-1): نظرة شاملة لفروع الذكاء الصناعي

4.2. تعلم الآلة والمكتبات البرمجية المستخدمة في بناء منصة تصنيف الخيوط القطنية

4.2.1. تمهيد في تعلّم الآلة

تُعد معالجة الصور وتصنيفها من الركائز الأساسية في مجال الرؤية الحاسوبية، حيث تتيح للأنظمة الذكية فهم المحتوى البصري وتفسيره بدقة متزايدة. في سياق الصناعة، وخاصة في فحص جودة المواد الخام مثل الخيوط القطنية، يمثل التصنيف الآلي خطوة حاسمة لضمان معايير الجودة، وتحسين كفاءة الإنتاج، وتقليل الأخطاء البشرية المحتملة. لقد أحدث الذكاء الاصطناعي (AI) وتعلم الآلة (ML) ثورة في القدرة على تحليل مجموعات البيانات المعقدة، بما في ذلك البيانات المرئية، مما يوفر حلولاً متقدمة لمهام التصنيف التي كانت تتطلب سابقاً تدخلاً بشرياً مكثفاً [26, 27]. يوضّح الشكل (4–2) فروع تعلّم الآلة.



الشكل (4–2): نظرة شاملة لفروع تعلّم الآلة14

4.2.2. فروع تعلم الآلة

ينقسم تعلم الآلة إلى عدة فروع رئيسية، كل منها مصمم للتعامل مع أنواع مختلفة من البيانات ومهام التعلم. في سياق مشروع تصنيف الخيوط القطنية، تُعد بعض هذه الفروع ذات أهمية محورية، بينما يقدم البعض الآخر سياقاً أوسع لفهم المشهد العام لتعلم الآلة.

4.2.2.1. التعلم المراقب (Supervised Learning)

التعلم المراقب هو فئة من تعلم الآلة تستخدم مجموعات بيانات مُعلّمة لتدريب الخوارزميات على التنبؤ بالنتائج والتعرف على الأنماط. على عكس التعلم غير المراقب، تُزوّد خوارزميات التعلم المراقب ببيانات تدريب تحتوي على أمثلة لكل من المدخلات (الميزات) والمخرجات الصحيحة (التسميات). تتعلم الخوارزمية من هذه الأزواج المُعلّمة لاستنتاج العلاقة بين المدخلات والمخرجات. بمجرد تدريب النموذج واختباره، يمكن استخدامه لإجراء تنبؤات على بيانات جديدة غير معروفة بناءً على المعرفة التي اكتسبها مسبقاً [28].

يُعد التعلم المراقب هو النهج الأساسي لمنصة تصنيف الخيوط القطنية. يتم تدريب النموذج على صور خيوط مُعلّمة مسبقاً، مثل "خيط سليم" أو "خيط معيب" أو أنواع محددة من العيوب. من خلال هذه البيانات المُعلّمة، يتعلم النموذج كيفية استخلاص الميزات المرئية ذات الصلة وتحديد الفئة الصحيحة لصور الخيوط الجديدة. يُقسم التعلم المراقب إلى عدة أنواع منها:

a) خوارزميات التصنيف (Classification Algorithms)

تُستخدم خوارزميات التصنيف لتصنيف البيانات عن طريق التنبؤ بتسمية فئوية أو متغير إخراج بناءً على بيانات الإدخال [28, 29]. تُستخدم هذه الخوارزميات عندما تكون متغيرات الإخراج فئوية، أي توجد فئتان أو أكثر. في سياق تصنيف الخيوط، الهدف هو تعيين صورة الخيط إلى فئة جودة محددة.

من الأمثلة الشائعة على خوارزميات التصنيف:

1) الانحدار اللوجستي **(Logistic Regression):** على الرغم من اسمه، يُستخدم الانحدار اللوجستي بشكل أساسي للتصنيف الثنائي (فئتين). يقوم بتحويل أي رقم حقيقي إلى نطاق باستخدام دالة

سيجمويد، مما يسمح بتفسير المخرجات كاحتمالات. يمكن تطبيق عتبات مختلفة بناءً على هذه الاحتمالات لتصنيف البيانات [30].

2) آلات المتجهات الداعمة **(Support Vector Machines - SVM):** تهدف آلات المتجهات الداعمة إلى إيجاد المستوى الفائق (hyperplane) الأمثل الذي يفصل الفئات في مساحة الميزات بفعالية، مع تعظيم الهامش بين أقرب نقاط البيانات من الفئات المختلفة (متجهات الدعم). يمكنها تصنيف البيانات غير القابلة للفصل خطياً في مساحة الإدخال الأصلية باستخدام "خدعة النواة kernel)" (trick التي تنقل البيانات إلى مساحة ذات أبعاد أعلى حيث يمكن فصلها خطياً [30].

3) أشجار القرار **(Decision Trees):** توفر أشجار القرار نهجاً مباشراً للتصنيف، حيث تقوم بفرز مجموعة بيانات إلى مجموعات فرعية أصغر وأكثر تفصيلاً وفقاً لقيم الميزات. يتم اتخاذ القرارات في كل عقدة بناءً على معايير مثل عدم نقاء جيني (Gini impurity) أو الإنتروبي (entropy) على الرغم من سهولة تفسيرها، إلا أنها عرضة للانحدار الزائد (overfitting)، خاصة عندما تكون عميقة جداً [30].

4) الغابات العشوائية **(Random Forest):** تُعد الغابات العشوائية تحسيناً على أشجار القرار الفردية، حيث تستخدم مجموعات فرعية عشوائية من الميزات والبيانات لزيادة تباين النموذج. يتم تشكيل تنبؤ النموذج من متوسط مخرجات كل شجرة فردية. تُظهر الغابات العشوائية نجاحاً عالياً في تطبيقات مثل تصنيف الصور، وتتفوق على أشجار القرار الفردية في الدقة والمتانة [30].

5) الجيران الأقرب **(k-Nearest Neighbors - k-NN):** هي خوارزمية تعلم قائمة على الأمثلة (instance-based learning) تصنف نقطة بيانات غير مرئية بناءً على تصويت الأغلبية بين أقرب k من الجيران في مساحة الميزات تتسم بالبساطة ولا تتطلب مرحلة تدريب صريحة لبناء نموذج، بل تخزن مجموعة البيانات بأكملها [30].

من الجدير بالذكر أنه في سياق تصنيف صور الخيوط، تُعد الشبكات العصبية التلافيفية (CNNs) هي الخوارزميات الأساسية الأكثر استخداماً نظراً لقدرتها الفائقة على معالجة البيانات المرئية، وسيتم تفصيلها لاحقاً. ومع ذلك، فإن فهم خوارزميات التصنيف التقليدية يضع الأساس النظري للتمييز بين مهام تعلم الآلة المختلفة.

b) خوارزميات الانحدار (Regression Algorithms)

تُستخدم خوارزميات الانحدار للتنبؤ بقيمة حقيقية أو مستمرة، حيث تكتشف الخوارزمية علاقة بين متغيرين أو أكثر [28, 31]. تُعد جزءاً لا يتجزأ من أي نموذج تنبؤي أو تحليلي. على سبيل المثال، يمكن استخدام الانحدار للتنبؤ بسعر منزل بناءً على مساحته وموقعه. من أنواع الانحدار الشائعة:

1) الانحدار الخطي البسيط **(Simple Linear Regression):** يرسم خطاً مستقيماً عبر نقاط البيانات لتقليل الخطأ بين الخط ونقاط البيانات، ويُستخدم عندما يكون هناك متغير مستقل واحد [31].
2) الانحدار الخطي المتعدد **(Multiple Linear Regression):** يُستخدم عندما يكون هناك أكثر من متغير مستقل للتنبؤ بالمتغير التابع [31].

على الرغم من أن مشروع تصنيف الخيوط القطنية يركز بشكل أساسي على التصنيف (تحديد فئة الجودة)، إلا أن فهم خوارزميات الانحدار ضروري للتمييز بين أنواع مهام تعلم الآلة المختلفة والتعرف على متى يكون الانحدار هو النهج المناسب.

4.2.2.2. التعلم غير المراقب (Unsupervised Learning)

في التعلم غير المراقب، تتعلم أنظمة تعلم الآلة من البيانات دون إشراف بشري، أي دون الحاجة إلى تسميات مسبقة للبيانات. على عكس التعلم المراقب، تُعطى نماذج التعلم غير المراقب بيانات غير مُعلّمة وتُسمح لها باكتشاف الأنماط والرؤى والهياكل الكامنة في البيانات دون أي توجيه صريح. هذه الخوارزميات مناسبة لمهام المعالجة الأكثر تعقيداً، مثل تنظيم مجموعات البيانات الكبيرة في مجموعات (clusters) أو تقليل أبعادها [32]. يُقسم التعلم غير المراقب إلى عدة أنواع منها:

a) خوارزميات التجميع (Clustering Algorithms)

التجميع (Cluster Analysis): هو طريقة تعلم آلة يتم فيها تجميع نقاط البيانات معاً وفقاً لمفهوم معين للتشابه. الهدف هو أن تكون نقاط البيانات داخل المجموعة الواحدة أكثر تشابهاً بشكل ملحوظ من نقاط البيانات بين المجموعات. نظراً لعدم توفر معلومات حول عضوية المجموعة مسبقاً، يندرج التجميع ضمن عائلة خوارزميات التعلم غير المراقب [33]. من الأمثلة البارزة على خوارزميات التجميع **K-means**، وهي

الخوارزمية الأكثر شهرة واستخداماً في التجميع، حيث تقوم بتقسيم نقاط البيانات إلى عدد محدد K من المجموعات [32, 33]. تعمل عن طريق تعيين كل نقطة بيانات إلى أقرب مركز مجموعة (centroid) ثم تحديث مراكز المجموعات بشكل متكرر.

يمكن أن يلعب التجميع دوراً حاسماً في مرحلة استكشاف البيانات لمشروع تصنيف الخيوط القطنية. بدلاً من مجرد تصنيف العيوب المعروفة، يمكن لخوارزميات التجميع أن تكتشف أنماطاً جديدة أو أنواعاً غير متوقعة من العيوب في صور الخيوط التي لم يتم تحديدها مسبقاً بواسطة الخبراء البشريين. هذه القدرة على "اكتشاف المجهول" أو "الأنماط الخفية" هي ميزة فريدة للتعلم غير المراقب، ويمكن أن تؤدي إلى تحسين فهمنا لجودة الخيوط، وتطوير فئات عيوب جديدة، وبالتالي تعزيز دقة وشمولية نظام التصنيف المراقب لاحقاً.

b) خوارزميات تقليل الأبعاد (Dimensionality Reduction Algorithms)

تقليل الأبعاد هي مهمة تقليل عدد الميزات في مجموعة بيانات مع الحفاظ على الخصائص الهامة. تُستخدم هذه التقنية لتحويل البيانات من مساحة ميزات عالية الأبعاد إلى مساحة ميزات منخفضة الأبعاد. تُعد هذه العملية حاسمة عند التعامل مع مشكلة "لعنة الأبعاد (Curse of Dimensionality)"، حيث يؤدي العدد الكبير من الميزات إلى زيادة تعقيد النمذجة وزيادة مخاطر الانحدار الزائد [34, 35]. من الأمثلة على خوارزميات تقليل الأبعاد، تحليل المكونات الرئيسية **(Principal Component Analysis - PCA)**، وهو طريقة غير مُراقبة لتقليل الأبعاد تهدف إلى إيجاد مساحة ذات أبعاد أقل عن طريق الحفاظ على أكبر قدر ممكن من التباين الموجود في مساحة الإدخال عالية الأبعاد. تعمل PCA عن طريق تحويل البيانات خطياً إلى مجموعة جديدة من الإحداثيات (المكونات الرئيسية) التي تكون متعامدة وغير مترابطة [32, 34].

في معالجة صور الخيوط، خاصة إذا كانت الصور عالية الدقة، يمكن أن تنتج مجموعات بيانات ذات أبعاد عالية جداً (عدد كبير من البكسلات). استخدام خوارزميات تقليل الأبعاد مثل PCA في مرحلة ما قبل المعالجة يمكن أن يقلل بشكل كبير من عدد الميزات مع الحفاظ على المعلومات الأساسية. هذا لا يؤدي فقط إلى تسريع عملية تدريب نماذج التعلم العميق (CNNs) المستخدمة في تصنيف الخيوط، بل ويساعد أيضاً على تحسين قدرة النموذج على التعميم على بيانات جديدة غير مرئية من خلال إزالة الميزات الزائدة أو

الضوضاء. وبالتالي، فإن تقليل الأبعاد هو خطوة تمهيدية حاسمة لتعزيز أداء النموذج واستقراره من خلال التخفيف من تحديات لعنة الأبعاد [34, 35].

c) خوارزميات قواعد الارتباط (Association Rule Learning Algorithms)

تُعد قواعد الارتباط طريقة تعلم آلة قائمة على القواعد لاكتشاف العلاقات المثيرة للاهتمام بين المتغيرات في قواعد البيانات الكبيرة. تهدف إلى تحديد القواعد القوية باستخدام مقاييس مثل الدعم (Support) الذي يشير إلى مدى تكرار ظهور مجموعة من العناصر معاً، والثقة (Confidence) التي تقيس مدى تكرار ظهور عنصر معين عندما يظهر عنصر آخر. من الأمثلة الشائعة على هذا النوع من الخوارزميات هو، خوارزمية **Apriori:** هي إحدى التقنيات الرئيسية لتعدين قواعد الارتباط، والتي تهدف بشكل أساسي إلى إيجاد مجموعات العناصر الأكثر تكراراً في مجموعة بيانات[36, 37] .

قد لا تكون هذه الخوارزميات ذات صلة مباشرة بتصنيف صور الخيوط بحد ذاتها، ولكن يمكن استخدامها لتحليل بيانات أخرى ذات صلة في سياق المشروع، مثل سجلات الإنتاج أو بيانات العيوب المرتبطة بظروف معينة. على سبيل المثال، يمكن استخدامها لتحديد أنماط مشتركة، مثل ظهور عيوب معينة دائماً مع نوع معين من المادة الخام أو في ظروف بيئية محددة.

4.2.2.3. التعلم المعزز (Reinforcement Learning)

التعلم المعزز هو نوع من تعلم الآلة يركز على اتخاذ القرارات بواسطة العملاء المستقلين. يتعلم العميل (Agent)كيفية التصرف في بيئته من خلال التفاعل، باستخدام إشارات التعزيز (Reinforcers) التي تشير إلى مدى جودة أو سوء إجراء معين [38]. يتم التعلم من خلال التجربة والخطأ، حيث يتلقى العميل مكافآت أو عقوبات بناءً على أفعاله، ويهدف إلى تعظيم مجموع المكافآت على المدى الطويل. تتضمن المكونات الرئيسية للتعلم المعزز العميل (الذي يتخذ القرارات)، والبيئة (التي يتفاعل معها العميل)، والهدف (المكافأة التي يسعى العميل لتعظيمها)، والسياسة (Policy) التي تحدد سلوك العميل، ودالة القيم (Value Function) التي تحدد المنفعة طويلة المدى للحالة أو الإجراء [39].

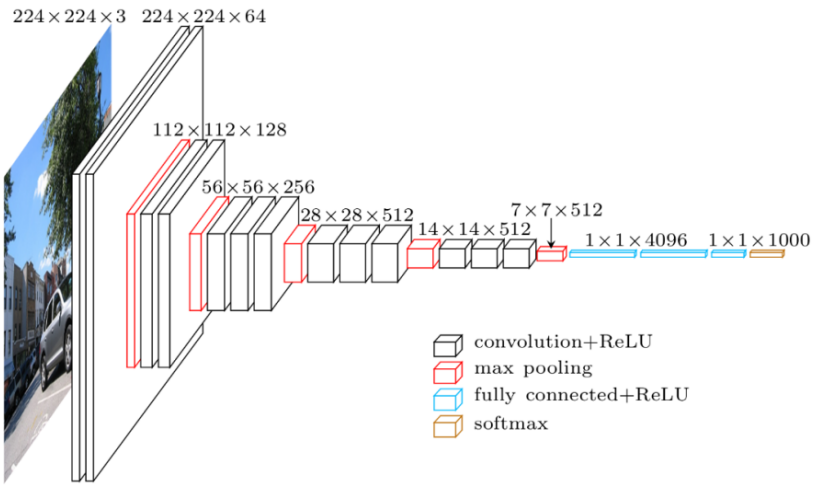
لا يبدو التعلم المعزز مستخدماً بشكل مباشر في مهمة تصنيف صور الخيوط القطنية، حيث أنها مهمة تصنيف ثابتة تعتمد على بيانات مُعلّمة مسبقاً وليست مهمة اتخاذ قرار تسلسلي في بيئة ديناميكية تتطلب استكشافاً واستغلالاً. ومع ذلك، يُعد التعلم المعزز فرعاً مهماً في الذكاء الاصطناعي، خاصة في مجالات الروبوتات والألعاب وأنظمة التحكم[38,39].

4.3. التعلم العميق والشبكات العصبية التلافيفية

يمثل التعلم العميق (Deep Learning) مجالاً فرعياً من تعلم الآلة يستخدم شبكات عصبية متعددة الطبقات، تُعرف بالشبكات العصبية العميقة، لمحاكاة قدرة الدماغ البشري على اتخاذ القرارات المعقدة. تُعد هذه التقنية أساس معظم تطبيقات الذكاء الاصطناعي الحديثة [27]. في هذا البحث تم استخدام الشبكات العصبية التلافيفية (Convolutional Neural Networks - CNNs)، لذا كان لا بدّ من دراستها والتعرّف عليها بشيء من التفصيل.

4.3.1. الشبكات العصبية التلافيفية (Convolutional Neural Networks - CNNs)

أصبحت الشبكات العصبية التلافيفية (CNNs) محورية في مجال التعلم العميق، وقد أحدثت ثورة في تكنولوجيا التعرف على الصور. تحاكي CNNs نظام الرؤية البشري وتؤدي بشكل ممتاز في مهام مثل: تصنيف الصور، الكشف، والتجزئة عن طريق التعلم التلقائي للميزات في الصور. هذا يلغي الحاجة إلى استخراج الميزات يدوياً ويحسن الدقة بشكل كبير مقارنة بالأساليب التقليدية التي تعتمد على الميزات المصممة يدوياً [42, 43]. الشكل (4-3) رسم تخطيطي يوضح البنية العامة لشبكة CNN. يجب أن تبدأ الصورة بـ "صورة المدخلات (Input Image)"، تليها عدة "طبقات التفافية (Convolutional Layers)" و"طبقات تجميع (Pooling Layers)" متتالية (ممثلة بمكعبات أو مستطيلات تتناقص أبعادها وتزداد عمقها). بعد ذلك، تُظهر الطبقات المسطحة (Flatten Layer) والطبقات الخطي (Fully Connected Layers)، وتنتهي بطبقة المخرجات (Output Layer) التي تُظهر احتمالات التصنيف.



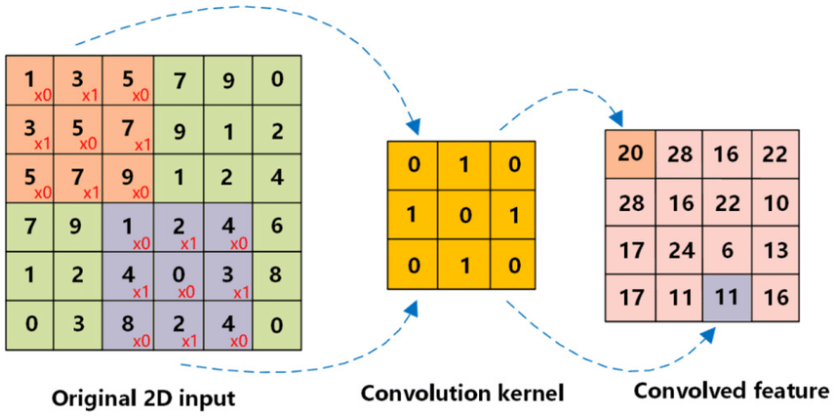
الشكل (4-3): بنية عامة لشبكة عصبية تلافيفية (*CNN*)15

تتكون CNNs من عدة طبقات، كل منها يلعب دوراً حيوياً في استخراج الميزات الهرمية ومعالجة الأنماط المعقدة في البيانات المرئية:

1) **طبقات التلافيف (Convolutional Layers):** هي الطبقة الأكثر أهمية في CNN، حيث تستخدم عملية رياضية تسمى التلافيف (convolution operation) لاستخراج الميزات من الصور. تقوم هذه الطبقات بتطبيق مجموعة من المرشحات (filters) أو النوى (kernels) على الصورة المدخلة لإجراء حساب مرجح بناءً على خصائص البكسلات، مما يولد خرائط ميزات جديدة تبرز أنماطاً معينة مثل الحواف أو الأنسجة.
2) **طبقات التجميع (Pooling Layers):** تُستخدم لتقليل حجم خرائط الميزات (downsampling) مع الاحتفاظ بالميزات الرئيسية، مما يحسن كفاءة الحوسبة ويقلل من مخاطر الانحدار الزائد. تشمل طرق التجميع الشائعة التجميع الأقصى (Max Pooling) الذي يأخذ القيمة القصوى في نافذة معينة، والتجميع المتوسط(Average Pooling) الذي يحسب متوسط القيم.
3) **طبقات التنشيط (Activation Layers):** تُدخل اللاخطية (non-linearity) إلى الشبكة، مما يمكنها من تعلم والتعبير عن علاقات الميزات المعقدة. بدون وظائف التنشيط، ستكون الشبكة العصبية مجرد سلسلة من التحويلات الخطية، مما يحد من قدرتها على نمذجة البيانات المعقدة. دالة ReLU (Rectified Linear Unit)شائعة الاستخدام لفعاليتها في تسريع التدريب.
4) **طبقات تطبيع الدفعة (Batch Normalization Layers - BN):** تسرّع تدريب الشبكة العصبية، وتحسن الاستقرار، وتخفف من مشاكل تضاؤل أو انفجار التدرج (gradient vanishing/exploding) عن طريق توحيد مدخلات كل طبقة إلى متوسط موحد (0) وتباين (1).
5) **الطبقات المتصلة بالكامل (Fully Connected Layers - FC):** تربط جميع العصبونات في الطبقة السابقة بكل عصبون في الطبقة الحالية. تُستخدم عادة في نهاية الشبكة بعد استخلاص الميزات بواسطة الطبقات التلافيفية والتجميع، لإجراء التصنيف النهائي بناءً على الميزات المستخلصة.

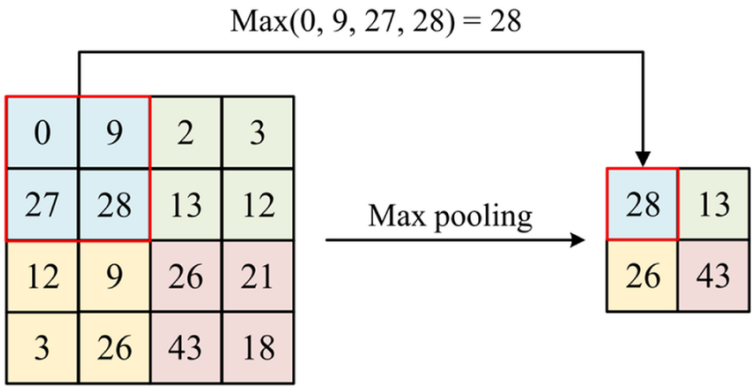
تُعد CNNs الخيار الأمثل لتصنيف صور الخيوط القطنية نظراً لقدرتها الفائقة على استخلاص الميزات الهرمية من البيانات المرئية. يمكنها التعرف على العيوب الدقيقة في نسيج الخيط، مثل الشوائب، التمزقات، أو الاختلافات في السمك، والتي قد تكون صعبة التحديد بالعين المجردة أو باستخدام طرق تعلم آلة تقليدية.

يبيّن الشكل (4-4) رسم توضيحي مبسط لعملية الالتفاف. يجب أن يظهر مربع مدخلات (صورة) ومربع مرشح(filter/kernel) أصغر يتحرك فوق المدخلات خطوة بخطوة (stride). في كل خطوة، يتم حساب المنتج النقطي بين المرشح والجزء المقابل من المدخلات، وتُسجل النتيجة في مربع مخرجات (feature map). يمكن استخدام ألوان مختلفة لتمثيل القيم العددية.



الشكل (4-4): عملية الالتفاف (*Convolution Operation*)16

يبيّن الشكل (4-5) رسم توضيحي مبسط لعملية التجميع الأقصى. يجب أن تظهر خريطة ميزات مدخلة، ومربع تجميع (pooling window) يتحرك فوقها. في كل خطوة، يتم اختيار القيمة القصوى من المنطقة التي يغطيها مربع التجميع، وتُسجل النتيجة في خريطة ميزات مخرجة ذات أبعاد أصغر.



الشكل (4-5): عملية التجميع الأقصى (*Max Pooling Operation*)17

4.3.2. أمثلة على معماريات CNNs

لقد شهدت معماريات CNNs تطوراً كبيراً، مما أدى إلى ظهور نماذج أكثر عمقاً وكفاءة [42]:

1) **LeNet-5:** هو هيكل CNN كلاسيكي اقترحه يان لوكون (Yann LeCun) وزملاؤه في أواخر التسعينيات. تم تحسينه لمشاركة الأوزان واستخراج الميزات الهرمية، مما أدى إلى كفاءة حاسوبية أفضل وقدرة على التعميم. كان أول CNN ناجح تجارياً، واستُخدم بشكل خاص لتصنيف الأرقام المكتوبة بخط اليد في تطبيقات مثل أجهزة الصراف الآلي.
2) **AlexNet:** اقترحه كرزيفسكي وزملاؤه في عام 2012، وحقق نجاحاً كبيراً في تحدي ImageNet لتصنيف الصور، مما أظهر القوة الهائلة للشبكات العميقة واستخدام وحدات معالجة الرسوميات (GPUs)لمعالجة مجموعات البيانات الكبيرة. يتكون من ثماني طبقات (خمس تلافيفية، ثلاث متصلة بالكامل) واستخدم معماريات أعمق، وتنشيط ReLU، وتنظيم Dropout، وتقنيات زيادة البيانات (Data Augmentation).
3) **ResNet (Residual Network):** تمّ اقترحه في عام 2015، وقد حل مشاكل تضاؤل وانفجار التدرج التي كانت تعيق تدريب الشبكات العصبية العميقة جداً. يستخدم ResNet مفهوم *"التعلم المتبقي(residual learning)"* من خلال *"وصلات التخطي (skip connections)"* التي تسمح للمعلومات بالمرور عبر طبقات متعددة دون المرور عبر جميع التحويلات، مما يسهل تدفق التدرج أثناء الانتشار العكسي ويسرع تقارب التدريب. هذا يجعله مناسباً لتعلم الميزات المعقدة للعيوب الدقيقة في الخيوط.
4) **EfficientNet:** اقترحه فريق Google Brain في عام 2019، ومفهومه الأساسي هو القياس المركب (compound scaling)، حيث يقوم بتحسين عمق الشبكة، عرضها، ودقة الإدخال في وقت واحد بطريقة موحدة. يحقق أداء تصنيف أفضل بكفاءة حاسوبية أقل بكثير من CNNs التقليدية، مما يجعله فعالاً للغاية لمختلف مهام الرؤية الحاسوبية، خاصة على الموارد المحدودة.

في هذا البحث المتعلّق بتصنيف جودة الخيوط القطنية، تم اختيار المعمارية ResNet، لما تتميّز به بمقدرة عالية على التعامل مع الشبكات العميقة وتجنب مشكلة تضاؤل التدرج، مما يجعلها مناسبةً لتعلم الميزات

المعقدة للعيوب الدقيقة في الخيوط. في حين يتميّز EfficientNet بقدراته على القياس المركب، يمكن أن يكون مثالياً إذا كانت الموارد الحسابية محدودة أو إذا كان هناك حاجة لنموذج خفيف الوزن للنشر. هذا يعني أن قرار اختيار المعمارية يجب أن يكون مستنيراً بخصائص البيانات (مثل حجم الصور، تنوع العيوب) والقيود التشغيلية (مثل وقت التدريب، متطلبات الأجهزة)، مما يؤدي إلى علاقة سببية بين خصائص المشروع واختيار المعمارية لتحقيق الأداء الأمثل.

4.4. المكتبات البرمجية المستخدمة في بناء المنصة

تعتمد منصة تصنيف الخيوط القطنية على مجموعة من المكتبات البرمجية القوية في لغة بايثون، والتي تغطي مختلف جوانب المشروع من معالجة الصور إلى تدريب النماذج وبناء الواجهة الخلفية للويب.

4.4.1. مكتبات معالجة وتوليد الصور

تُعد هذه المكتبات حجر الزاوية في التعامل مع البيانات المرئية، من التحضير الأولي إلى زيادة البيانات وتعرضها.

1) cv2 (OpenCV):

- الوصف: مكتبة مفتوحة المصدر وشاملة لمعالجة الصور والفيديو، وتدعم مجموعة واسعة من العمليات.
- الميزات: توفر وظائف قوية للتحويل بين مساحات الألوان (مثل RGB إلى تدرج رمادي)، وتصفية الصور(filtering) ، والكشف عن الحواف، وتغيير الحجم، والقص، وتطبيق المرشحات المختلفة. كما تدعم عمليات الرؤية الحاسوبية المتقدمة.
- الدور في المشروع: تُعد OpenCV أداة أساسية في مرحلة ما قبل معالجة صور الخيوط. يمكن استخدامها لتوحيد حجم الصور إلى أبعاد قياسية، تحويلها إلى تدرج رمادي أو قنوات ألوان معينة لتبسيط البيانات، إزالة الضوضاء التي قد تؤثر على دقة التصنيف، أو تطبيق عمليات الكشف عن الحواف لإبراز ميزات نسيج الخيط أو عيوبه الدقيقة.

2) PIL (Pillow):

- الوصف: مكتبة Python لمعالجة الصور، وهي شوكة (fork) محسّنة لمكتبة PIL الأصلية، تُستخدم لتحميل الصور وتحويلها بين الصيغ المختلفة وإجراء تعديلات أساسية عليها.
- الميزات: تدعم مجموعة واسعة من صيغ الصور مثل: (JPEG, PNG, BMP)، وتوفر عمليات أساسية مثل القص، التدوير، تغيير الحجم، وتعديل خصائص الصورة مثل السطوع والتباين.
- الدور في المشروع: تُستخدم Pillow بشكل شائع لتحميل صور الخيوط من نظام الملفات إلى الذاكرة، وإجراء تحويلات بسيطة عليها قبل إدخالها إلى خط أنابيب المعالجة الأعمق بواسطة مكتبات مثل torchvision.transforms أو لحفظ الصور المعالجة أو الصور التي تم توليدها.

3) albumentations:

- الوصف: مكتبة متقدمة لزيادة الصور (Data Augmentation) في رؤية الحاسوب، مصممة خصيصاً لتحسين أداء نماذج التعلم العميق.
- الميزات: تدعم مجموعة واسعة من التحويلات المتطورة مثل التدوير العشوائي، القص العشوائي، إضافة الضوضاء (مثل ضوضاء الملح والفلفل)، التحريف البصري(elastic deformations)، وتغيير الألوان (مثل تعديل التشبع والسطوع) وتتميز بكونها سريعة وفعالة.
- الدور في المشروع: تُعتبر albumentations حيوية لتعزيز قوة نموذج تصنيف الخيوط. نظراً لأن نماذج التعلم العميق تتطلب كميات هائلة من البيانات للتدريب الفعال [26]، فإن زيادة البيانات الاصطناعية باستخدام هذه المكتبة تساعد في توسيع مجموعة بيانات صور الخيوط المتاحة بشكل كبير. هذا يجعل النموذج أكثر قوة للتباينات في ظروف الإضاءة، زوايا التصوير، أو التغيرات الطفيفة في مظهر الخيط التي قد تحدث في بيئة الإنتاج الحقيقية، وبالتالي يقلل من الانحدار الزائد (overfitting)ويحسن قدرة النموذج على التعميم على بيانات جديدة وغير مرئية [42،51].

4) matplotlib.pyplot:

- الوصف: جزء من مكتبة Matplotlib، وهي مكتبة شاملة للرسم البياني في بايثون تُستخدم لتصور البيانات.

- الميزات: تتيح إنشاء مجموعة متنوعة من الرسوم البيانية مثل الرسوم البيانية الخطية، المبعثرة، الأعمدة، وتصور الصور مباشرة. تُستخدم على نطاق واسع في تحليل البيانات وعرض النتائج.
- الدور في المشروع: تُستخدم matplotlib.pyplot لتصور مراحل التدريب والتقييم لنموذج تصنيف الخيوط. على سبيل المثال، رسم منحنيات دقة النموذج (accuracy) وخسارة التدريب (loss) على مجموعتي التدريب والتحقق لمراقبة تقدم التعلم، أو عرض أمثلة لصور الخيوط الأصلية والمعالجة أو الصور التي تم توليدها بواسطة GANs لتقييم جودتها.

5) shutil, glob, os:

- الوصف: هذه المكتبات هي جزء من مكتبة بايثون القياسية وتُستخدم لإدارة الملفات والمجلدات على نظام التشغيل.
- الميزات:
  - shutil: توفر عمليات ذات مستوى أعلى للتعامل مع الملفات والمجلدات، مثل نسخ الملفات والمجلدات، حذفها، أو نقلها.
  - glob: تُستخدم للبحث عن الملفات والمجلدات التي تتطابق مع نمط معين باستخدام أحرف البدل(wildcards) ، مثل البحث عن جميع ملفات الصور ذات الامتداد jpg.
  - os: توفر واجهة للتفاعل مع نظام التشغيل، مما يتيح تنفيذ مهام مثل إنشاء مجلدات جديدة، التعامل مع مسارات الملفات، تغيير الدلائل، وقراءة المتغيرات البيئية.
- الدور في المشروع: تُستخدم هذه المكتبات لتنظيم مجموعة بيانات صور الخيوط بشكل فعال. على سبيل المثال، يمكن استخدامها لنسخ الصور إلى مجلدات مخصصة للتدريب والتحقق والاختبار، البحث عن جميع صور الخيوط بنمط معين داخل دليل، أو إنشاء مجلدات لتخزين النماذج المدربة، أو نتائج التصنيف، أو السجلات.

4.4.2. مكتبات تدريب وتعليم النماذج المخصصة لتقييم صورة الخيط

تُشكل هذه المكتبات جوهر عملية بناء وتدريب نماذج التعلم العميق، وتُعد **PyTorch** بيئة قوية لهذا الغرض.

1) **torch (PyTorch):**

- الوصف: مكتبة **PyTorch** الأساسية لتطوير النماذج العصبية، وهي إطار عمل مفتوح المصدر لتعلم الآلة.
- الميزات: توفر بنية بيانات **Tensor** (مصفوفات متعددة الأبعاد) التي تدعم الحوسبة على وحدات معالجة الرسوميات (GPUs)، مما يسرع بشكل كبير عمليات التدريب والحسابات الموازية. كما توفر نظام التفاضل التلقائي(autograd) الضروري لتطبيق خوارزمية الانتشار العكسي وتحديث أوزان النموذج.
- الدور في المشروع: تُعد torch العمود الفقري لتطوير وتدريب نماذج التعلم العميق (خاصة CNNs) في منصة تصنيف الخيوط. إن قدرتها على الاستفادة من وحدات معالجة الرسوميات (GPUs) حاسمة لتدريب النماذج الكبيرة على مجموعات بيانات الصور الضخمة بكفاءة وفي وقت معقول.

2) **torch.nn:**

- الوصف: وحدة فرعية داخل **PyTorch** تحتوي على الطبقات والخسائر المطلوبة لبناء النماذج العصبية.
- الميزات: توفر وحدات بناء جاهزة للشبكات العصبية مثل الطبقات التلافيفية (nn.Conv2d) طبقات التجميع(nn.MaxPool2d) طبقات التنشيط (nn.ReLU)، والطبقات المتصلة بالكامل (nn.Linear). كما توفر دوال الخسارة (Loss Functions) الشائعة مثل nn.CrossEntropyLoss المستخدمة عادة لمهام التصنيف متعدد الفئات، و nn.MSELoss للانحدار.
- الدور في المشروع: تُستخدم torch.nn لبناء معمارية الشبكة العصبية التلافيفية المخصصة لتصنيف صور الخيوط، وتحديد دالة الخسارة التي سيتم تحسينها أثناء التدريب لتقليل الفرق بين التنبؤات والنتائج الفعلية.

3) **torch.optim:**

- الوصف: وحدة فرعية داخل **PyTorch** تحتوي على خوارزميات تحسين (Optimizers) مثل Adam وSGD لتحديث أوزان النموذج.

- الميزات: توفر تطبيقات محسنة لخوارزميات تحسين التدرج التي تقوم بتعديل أوزان النموذج بشكل تكراري لتقليل دالة الخسارة. تعمل هذه الخوارزميات على حساب التدرجات (gradients) وتطبيق تحديثات على أوزان النموذج بناءً على معدل التعلم (learning rate) المحدد.
- الدور في المشروع: تُستخدم torch.optim لتحديد كيفية تعلم النموذج من الأخطاء. على سبيل المثال، قد يتم استخدام مُحسّن Adam لتحديث أوزان شبكة CNN بناءً على التدرجات المحسوبة من دالة الخسارة، مما يوجه النموذج نحو الأداء الأمثل في تصنيف الخيوط عن طريق تقليل الخطأ تدريجياً.

4) torch.utils.data:

- الوصف: وحدة فرعية داخل **PyTorch** تُستخدم لإنشاء DataLoader لتقسيم البيانات إلى دفعات (batches) وتوزيعها.
- الميزات: توفر فئات مثل Dataset وDataLoader لتجريد عملية تحميل البيانات، التعامل مع الدفعات (مجموعات صغيرة من الأمثلة)، والخلط العشوائي للبيانات، والتحميل المتعدد الخيوط(multi-threading) لتحسين كفاءة إدخال البيانات إلى النموذج.
- الدور في المشروع :تُعد هذه المكتبة ضرورية لإدارة تدفق بيانات صور الخيوط إلى نموذج التدريب بكفاءة. يسمح DataLoader بتحميل دفعات صغيرة من الصور في كل تكرار تدريب، مما يقلل من متطلبات الذاكرة (خاصة لوحدات GPU ذات الذاكرة المحدودة) ويسرع عملية التدريب بشكل كبير.

5) torchvision.models:

- الوصف: وحدة فرعية داخل **PyTorch** توفر نماذج مدربة مسبقاً (Pre-trained Models) لمعماريات رؤية حاسوبية شهيرة مثل ،ResNet, VGG والتي يمكن إعادة استخدامها للتصنيف.
- الميزات :تتضمن تطبيقات لمعماريات CNNs الشهيرة التي تم تدريبها مسبقاً على مجموعات بيانات كبيرة ومتنوعة مثل ImageNet يمكن استخدام هذه النماذج كقاعدة لـ "التعلم النقل (Transfer Learning)" حيث يتم تجميد الطبقات السفلية (التي تعلمت ميزات عامة) وتدريب الطبقات العلوية فقط على البيانات الجديدة.
- الدور في المشروع: تُعد torchvision.models ذات قيمة هائلة في مشروع تصنيف الخيوط. بدلاً من تدريب نموذج CNN من الصفر (الذي يتطلب كميات هائلة من البيانات ووقت حوسبة طويل)، يمكن

تحميل نموذج مدرب مسبقاً (مثل ResNet50) وتعديله قليلاً ليناسب مهمة تصنيف الخيوط. هذا يسرع عملية التطوير بشكل كبير ويحسن الأداء، خاصة إذا كانت مجموعة بيانات صور الخيوط صغيرة نسبياً، حيث يمكن للنموذج الاستفادة من الميزات العامة التي تعلمها من مجموعة بيانات ImageNetالكبيرة [48].

6) torchvision.transforms:

- الوصف: وحدة فرعية داخل **PyTorch** لتطبيق تحويلات مختلفة على الصور لتحويلها إلى تنسيقات قابلة للتغذية للنموذج مثل التحجيم، التحويل إلى **Tensor**.
- الميزات: توفر مجموعة من التحويلات الشائعة التي يمكن تطبيقها على الصور، مثل تغيير الحجم (resizing) القص (cropping) التدوير (rotation) التحويل إلى تنسيق Tensor (تحويل مصفوفات NumPy أو صور PIL إلى Tensors) وتطبيع قيم البكسل (normalization).
- الدور في المشروع: تُستخدم torchvision.transforms كجزء من خط أنابيب إعداد البيانات لضمان أن صور الخيوط يتم تحويلها إلى التنسيق والأبعاد المناسبة التي يتوقعها نموذج CNN المدرب. كما يمكن استخدامها لأغراض زيادة البيانات البسيطة (مثل التدوير العشوائي أو القلب الأفقي) لزيادة تنوع بيانات التدريب وتحسين قدرة النموذج على التعميم.

7) argparse:

- الوصف: مكتبة قياسية في بايثون لإدارة الوسائط في سطر الأوامر، مثل اختيار المعلمات أو المسارات أثناء التدريب.
- الميزات: تسمح للمطورين بتحديد واجهة سطر أوامر سهلة الاستخدام لبرامجهم، مما يتيح تمرير المعلمات (مثل معدل التعلم، حجم الدفعة، مسار مجموعة البيانات، عدد الحقب التدريبية) بسهولة عند تشغيل نصوص التدريب دون الحاجة إلى تعديل الكود مباشرة.
- الدور في المشروع: تُستخدم argparse لجعل نصوص تدريب نموذج تصنيف الخيوط أكثر مرونة وقابلية للتكوين، مما يسهل تجربة المعلمات الفائقة المختلفة (hyperparameters) ومسارات البيانات المتنوعة، وهو أمر حيوي في عملية تحسين أداء النموذج.

4.4.3. مكتبات Django المستخدمة في بناء المنصة

Django هو إطار عمل ويب عالي المستوى في بايثون يشجع على التطوير السريع والتصميم النظيف والعملي. تُستخدم مكوناته لإنشاء الواجهة الخلفية لمنصة تصنيف الخيوط.

1) **django.shortcuts:**

- الوصف: وحدة توفر وظائف مساعدة شائعة لتسهيل تنفيذ العمليات المتكررة في واجهات عرض Django مثل **render** و **redirect**.
- الميزات: تجعل الكود أكثر إيجازاً وقابلية للقراءة من خلال توفير اختصارات للوظائف المستخدمة بشكل متكرر.
- الدور في المشروع: تُستخدم في واجهات عرض **Django (views)** لتقديم قوالب **HTML** للمستخدمين (مثلاً، صفحة تحميل صورة الخيط) أو لإعادة توجيه المستخدمين بعد عملية تصنيف الخيط إلى صفحة النتائج أو صفحة أخرى ذات صلة.

2) **django.urls:**

- الوصف: وحدة لتعريف مسارات التطبيق **(URL routing)** في **Django**.
- الميزات: تسمح بتحديد أنماط عناوين **URL** وكيفية ربطها بوظائف العرض **(views)** في تطبيق **Django**، مما يوجه طلبات الويب الواردة إلى الكود المناسب للمعالجة.
- الدور في المشروع: تُستخدم لتحديد المسارات التي يمكن للمستخدمين الوصول إليها على منصة الويب، مثل **/upload/** لتحميل صورة خيط، أو **/classify/** لتشغيل عملية التصنيف، أو **/results/** لعرض النتائج.

3) **django.db.models:**

- الوصف: وحدة لإنشاء النماذج **(Models)** التي تمثل جداول قاعدة البيانات في **Django**.
- الميزات: يوفر نظام **ORM (Object-Relational Mapper)** الخاص بـ **Django**، والذي يسمح للمطورين بالتعامل مع جداول قاعدة البيانات ككائنات Python مما يبسط التفاعل مع قواعد البيانات ويزيل الحاجة إلى كتابة استعلامات **SQL** يدوياً.

- الدور في المشروع: تُستخدم لتعريف نماذج قاعدة البيانات لتخزين معلومات حول صور الخيوط التي تم تحميلها (مثل مسار الصورة، اسم الملف)، نتائج التصنيف (مثل فئة الجودة، درجة الثقة)، أو بيانات المستخدمين (إذا كانت المنصة تتطلب تسجيل دخول). هذا يربط الواجهة الأمامية بالبيانات الدائمة ويسمح بتتبع العمليات.

4) django.contrib.auth:

- الوصف: وحدة مدمجة في **Django** لإدارة نظام تسجيل الدخول والمستخدمين.
- الميزات: توفر نظام مصادقة وتفويض كامل الميزات، بما في ذلك نماذج المستخدمين، طرق المصادقة (مثل تسجيل الدخول بالاسم وكلمة المرور)، وإدارة الأذونات والمجموعات.
- الدور في المشروع: تُستخدم لتأمين منصة تصنيف الخيوط، مما يسمح فقط للمستخدمين المصرح لهم بتحميل الصور أو الوصول إلى نتائج التصنيف، وهو أمر حيوي لتطبيق صناعي يتطلب التحكم في الوصول وحماية البيانات.

5) django.core.files.storage:

- الوصف: وحدة للتعامل مع تحميل وتخزين الصور المرفوعة من المستخدمين.
- الميزات: توفر واجهة موحدة للتعامل مع تخزين الملفات، سواء كان ذلك على نظام الملفات المحلي (الافتراضي) أو خدمات التخزين السحابي مثل (Amazon S3) من خلال محولات التخزين (storage backends).
- الدور في المشروع: تُستخدم لإدارة عملية تحميل صور الخيوط من قبل المستخدمين عبر واجهة الويب، وتخزينها بشكل آمن على الخادم قبل معالجتها بواسطة نموذج تعلم الآلة.

6) django.http:

- الوصف: وحدة لإنشاء استجابات **HTTP** مخصصة، مثل **JsonResponse** .
- الميزات: توفر فئات لتمثيل طلبات واستجابات **HTTP**، مما يسمح بإنشاء استجابات مخصصة، بما في ذلك واجهات برمجة التطبيقات **(APIs)** التي تُرجع البيانات بتنسيق **JSON.**

- الدور في المشروع :تُستخدم لإنشاء استجابات) **API** على سبيل المثال، إرجاع نتائج تصنيف الخيط بتنسيق**JSON** إلى الواجهة الأمامية لتطبيق الويب أو تطبيق جوال) أو للتعامل مع الأخطاء التي قد تحدث أثناء تفاعل المستخدم مع المنصة عن طريق إرجاع رموز حالة **HTTP** مناسبة ورسائل خطأ.

4.4.5. المكاتب العامة المساعدة المستخدمة في بناء المنصة

تُكمل هذه المكتبات الوظائف الأساسية للمنصة من خلال توفير أدوات لمعالجة البيانات العامة وإدارة الوقت.

1) json:

- الوصف: مكتبة قياسية في بايثون لتخزين وتحميل البيانات بشكل منظم على شكل **JSON (JavaScript Object Notation).**
- الميزات: توفر وظائف لتحويل كائنات Python (مثل القواميس والقوائم) إلى سلاسل **JSON** والعكس، مما يسهل تبادل البيانات بين المكونات المختلفة للمنصة أو مع واجهات برمجة التطبيقات الخارجية **JSON.** هو تنسيق بيانات خفيف الوزن وسهل القراءة والكتابة بواسطة البشر والآلات.
- الدور في المشروع: تُستخدم لتخزين تكوينات النموذج (مثل المعلمات الفائقة)، أو سجلات التدريب، أو نتائج التصنيف في تنسيق يمكن قراءته بسهولة. كما تُستخدم بشكل شائع لإرسال واستقبال البيانات من الواجهة الأمامية للويب (المتصفح) إلى الواجهة الخلفية **(Django)** والعكس.

2) datetime:

- الوصف: مكتبة قياسية في بايثون للتعامل مع كائنات التاريخ والوقت، وتنسيقها في تقارير التصنيف.
- الميزات: توفر فئات للتعامل مع التاريخ والوقت، وإجراء العمليات الحسابية (مثل إضافة أو طرح الأيام/ الساعات)، وتنسيقها في سلاسل نصية بأشكال مختلفة.
- الدور في المشروع: تُستخدم لإضافة طوابع زمنية (timestamps) دقيقة إلى سجلات التصنيف أو تقارير جودة الخيوط. هذا يساعد في تتبع متى تم إجراء كل عملية تصنيف، وتسهيل تحليل البيانات بمرور الوقت، ومراقبة أداء المنصة، وتوفير معلومات حيوية وحيوية لأغراض التدقيق والجودة في البيئة الصناعية.

4.5. توليد نماذج التصنيف

4.5.1. تقسيم مجموعات البيانات لتدريب نماذج التعلم العميق وتحسين تقييم الأداء

تُعد عملية تقسيم مجموعات البيانات (Dataset Splitting) خطوة منهجية أساسية لا غنى عنها في تطوير وتقييم نماذج التعلم الآلي (Machine Learning) والتعلم العميق (Deep Learning)، خاصة في سياق الرؤية الحاسوبية (Computer Vision) [10] [12]. يهدف هذا التقسيم إلى إنشاء مجموعات فرعية متميزة: مجموعة تدريب (training set) لتعليم النموذج، ومجموعة تحقق (validation set) لضبط المعلمات الفائقة (hyperparameters) وتقييم الأداء أثناء التدريب، ومجموعة اختبار (test set) لتقييم الأداء النهائي للنموذج على بيانات غير مرئية تماماً [13]. يُقدّم الرماز المصدري (Code) المستخدم آلية فعالة ومنظمة لتقسيم مجموعة بيانات صور مُعدّة مسبقاً إلى مجموعتي تدريب وتحقّق بنسب محددة وهو أمر حيوي للتحقق من النتائج العلمية [14].

تُساهم هذه المنهجية المنظمة لتقسيم البيانات بشكل مباشر في [15]:

1. **تقييم أداء النموذج بدقة:** حيث تُستخدم مجموعة التحقّق لتقديم مؤشر غير متحيز لأداء النموذج على بيانات جديدة أثناء دورة التدريب.
2. **منع الفرط في الملاءمة:** من خلال مراقبة أداء النموذج على مجموعة التحقق، يمكن اكتشاف علامات الفرط في الملاءمة مبكراً واتخاذ الإجراءات التصحيحية.
3. **تحسين كفاءة التجربة:** توفير بنية بيانات واضحة ومنظمة يُبسط عمليات تحميل البيانات وتغذيتها لنماذج التعلّم العميق، مما يُقلّل من التعقيد البرمجي ويسرّع من دورات التطوير.

تُسهم هذه الخطوات في بناء أساس متين لتدريب وتقييم نماذج التعلم العميق، مما يؤدّي إلى تطوير حلول رؤية حاسوبية أكثر موثوقية وقدرة على التعميم في التطبيقات الواقعية.

4.5.2. تدريب وتقييم نماذج التعلم العميق لتصنيف الصور باستخدام نقل التعلم

يُمثل تدريب نماذج التعلم العميق (Deep Learning) لتصنيف الصور عملية معقدة تتطلب منهجية دقيقة لضمان الأداء الأمثل والتعميم الفعال [8]. يُقدم الرماز المصدري (Code) المستخدم إطاراً شاملاً لتدريب شبكة عصبية تلافيفية(Convolutional Neural Network - CNN) باستخدام تقنية نقل التعلم (Transfer

(Learning، مع التركيز على تحسين المعلمة (Parameter Optimization) بواسطة مُحسن آدم (Adam Optimizer). يهدف هذا النهج إلى تسخير قوة النماذج المدربة مسبقاً على مجموعات بيانات ضخمة، وتكييفها لمهام تصنيف محددة، مما يقلّل بشكل كبير من الحاجة إلى مجموعات بيانات كبيرة خاصة بالمهمة المستهدفة ومن الموارد الحاسوبية اللازمة للتدريب من الصفر.

4.5.2.1. تجهيز البيانات وزيادتها

تُشكل عملية تجهيز البيانات خطوة أولية حاسمة في تدريب نماذج التعلم العميق؛ حيث يتم تطبيق مجموعة من التحويلات (Transformations) باستخدام مكتبة PyTorch's torchvision.transforms. هذه التحويلات ضرورية لعدة أسباب: توحيد أبعاد المدخلات، زيادة تنوع البيانات التدريبية (Data Augmentation)، وتقليل الفرط في الملاءمة (Overfitting) [3]. تتضمن التحويلات النموذجية المطبقة ما يلي:

- **تغيير الحجم (Resizing):** يتم تغيير حجم جميع الصور إلى أبعاد موحدة (مثل 224×224 بكسل) لتتناسب مع متطلبات أبعاد المدخلات للشبكة العصبية. هذا يضمن أن النموذج يتلقى مدخلات ذات حجم ثابت.
- **القص العشوائي أو المركزي (Random or Central Cropping):** في مرحلة التدريب، يمكن تطبيق القص العشوائي لزيادة تنوع البيانات، حيث يتم أخذ أجزاء مختلفة من الصورة. أما في مرحلتي التحقق والاختبار، فيُفضل القص المركزي لضمان الاتساق والتركيز على الجزء الأكثر أهمية من الصورة.
- **القلب الأفقي (Horizontal Flip):** يُعد القلب الأفقي تحويلاً شائعاً لزيادة البيانات، حيث يتم عكس الصورة أفقياً بشكل عشوائي. هذا يساعد النموذج على أن يكون مقاوماً للاختلافات الطفيفة في توجيه الكائنات داخل الصورة.
- **التحويل إلى تنسيق موحد (ToTensor):** يتم تحويل صور PIL (Pillow) أو مصفوفات NumPy إلى تنسيق PyTorch Tensor. هذه الخطوة ضرورية لأن نماذج PyTorch تعمل مع هذا النوع من البيانات.
- **التطبيع (Normalization):** تُعد هذه العملية حاسمة لتحسين استقرار وسرعة عملية التدريب. يتم تطبيع قيم البكسلات في الصور باستخدام قيم المتوسط والانحراف المعياري المستمدة من مجموعة بيانات ImageNet [16]. هذا يضمن أن قيم المدخلات تقع ضمن نطاق موحد، مما يمنع بعض الميزات من السيطرة على عملية التعلم بسبب نطاق قيمها الأكبر.

بعد تطبيق هذه التحويلات، يتم تحميل مجموعات بيانات التدريب والتحقق بكفاءة باستخدام torch.utils.data.DataLoader. تُسهل هذه الفئة عملية تقسيم البيانات إلى دفعات (Batches) صغيرة، مما يُحسن من استخدام الذاكرة ويُسرع من عملية التدريب من خلال معالجة عدة عينات في وقت واحد. كما تدعم DataLoader التحميل المتعدد الخيوط (Multi-threading) والخلط العشوائي (Shuffling) للبيانات في كل حقبة، مما يقلل من التحيز ويحسن من تعميم النموذج.

4.5.2.2. نقل التعلم (Transfer Learning)

يستفيد الرماز المصدري (Code) بشكل كبير من تقنية نقل التعلم، وذلك بتحميل نموذج ResNet18 [17] المُدرب مسبقاً على مجموعة بيانات ImageNet. تُعد هذه النماذج المدربة مسبقاً بمثابة مستخلصات ميزات (Feature Extractors) قوية، حيث أنها تعلمت تمثيلات هرمية للميزات من كميات هائلة من الصور المتنوعة [18]. هذا يعني أن الطبقات الأولية في ResNet18 قد اكتسبت القدرة على التعرف على ميزات عامة مثل الحواف، الأنسجة، والأشكال، والتي يمكن تطبيقها على مهام رؤية حاسوبية جديدة.

يتم تكييف النموذج ليناسب مهمة تصنيف الخيوط الجديدة عن طريق استبدال الطبقة الخطية (Fully Connected Layer) النهائية للنموذج بطبقة جديدة تتوافق مع عدد الفئات المستهدفة (أربع فئات في هذه الحالة). يمكن تجميد(freeze) أوزان الطبقات السابقة في ResNet18 للحفاظ على الميزات المستخلصة، أو يمكن السماح بتعديلها(fine-tuning) بمعدل تعلم صغير لضبطها بشكل أدق على البيانات الجديدة. في هذه الحالة، يتم تكييف النموذج لمهام تصنيف محددة للخيوط القطنية.

تُشكل دالة الخسارة (Loss Function)، المتمثلة في nn.CrossEntropyLoss، الأساس الذي يعتمد عليه النموذج لتعلم تصنيف الصور. تُستخدم هذه الدالة على نطاق واسع في مشاكل التصنيف متعددة الفئات، حيث تُقيس مدى تباعد مخرجات النموذج (التوزيع الاحتمالي للفئات المتوقعة) عن التسميات الحقيقية (التوزيع الفعلي للفئات) الهدف من عملية التدريب هو تقليل هذه الخسارة إلى أدنى حد ممكن [19].

4.5.2.3. حلقة التدريب والتقييم

تُعد حلقة التدريب هي جوهر عملية التعلم. ففي كل دورة تدريب (Epoch)، يتم التكرار على مجموعة بيانات التدريب بأكملها، حيث تُطبّق الخطوات التالية لكل دفعة من الصور:

1. **التمرير الأمامي (Forward Pass):** يتم تغذية الصور من الدفعة الحالية إلى النموذج لإنتاج التوقعات(predictions) أو المخرجات. يقوم النموذج بمعالجة المدخلات عبر طبقاته المتتالية.
2. **حساب الخسارة (Loss Calculation):** تُحسب الخسارة (error) بين التوقعات التي أنتجها النموذج والتسميات الحقيقية (ground truth labels) باستخدام دالة .CrossEntropyLoss تُشير قيمة الخسارة إلى مدى سوء أداء النموذج في التنبؤ بالفئات الصحيحة.
3. **التمرير الخلفي (Backward Pass):** يتم حساب التدرجات (gradients) للخسارة بالنسبة لجميع أوزان النموذج القابلة للتدريب باستخدام loss.backward(). هذه الخطوة، المعروفة باسم الانتشار الخلفي (Backpropagation)، تحسب كيف تساهم كل معلمة في النموذج في الخطأ الكلي.
4. **تحديث الأوزان (Weight Update):** يستخدم مُحسن آدم optimizer.step() لتحديث أوزان النموذج بناءً على التدرجات المحسوبة. تُصفر التدرجات optimizer.zero_grad() قبل كل تمرير أمامي لمنع تراكمها من الدفعات السابقة، مما يضمن أن التحديثات تعتمد فقط على التدرجات الحالية.

بعد كل حلقة تدريب، تُجرى مرحلة التحقق (Validation) لتقييم أداء النموذج على بيانات لم يرها أثناء التدريب. تُختتم العملية برسم بياني لمنحنيات الخسارة والدقة (مثل خسارة التدريب مقابل دقة التحقق)، مما يوفّر تصوراً بصرياً لأداء النموذج على مدار التدريب، ويساعد في تحديد مشاكل مثل الفرط في الملاءمة (overfitting)أو النقص في الملاءمة (underfitting).

4.5.3. استدلال نماذج التعلم العميق لتصنيف الصور: معالجة البيانات وتقنيات التنبؤ

تُعد مرحلة الاستدلال (Inference) في نماذج التعلم العميق (Deep Learning) خطوة حاسمة لتوظيف النماذج المدربة في تطبيقات العالم الحقيقي، حيث تتضمن استخدام النموذج المُدرب مسبقاً لتصنيف بيانات جديدة غير مرئية[8] . يتركز هذا النهج على معالجة الصورة المدخلة وتكييفها لتناسب متطلبات النموذج، ثم استخراج التنبؤات وتفسيرها، مما يضمن دقة وفعالية النظام في بيئات التشغيل الحقيقية [17].

تُستهل عملية الاستدلال بتحميل النموذج (Model Loading)، لاستعادة بنية النموذج وأوزانه التي تم تعلمها أثناء مرحلة التدريب. بعد تحميل نموذج ResNet18 المُدرب على مجموعة بيانات ImageNet (التي تتضمن 1000 فئة)، يتم تُحمل الأوزان المُدربة خصيصاً لهذه الفئات من النماذج المدربة. بعد تحميل الأوزان، يُعين النموذج في وضع التقييم (model.eval())، لضمان سلوك ثابت وموثوق للنموذج أثناء الاستدلال [10] [23].

ثم تتم عملية تُعد معالجة الصورة المسبقة (Image Preprocessing) لضمان توافق الصورة المدخلة مع التنسيق الذي يتوقعه النموذج. حيث تُطبق سلسلة من التحويلات المتطابقة مع تلك المستخدمة في مرحلة التحقق أثناء التدريب باستخدام مكتبة torchvision.transforms. هذا التطابق حاسم لضمان أن النموذج يتلقى بيانات بنفس التوزيع والخصائص التي تدرب عليها، مما يمنع تدهور الأداء. تشمل هذه التحويلات: تغيير حجم الصورة (transforms.Resize)، قص الجزء المركزي منها (transforms.CenterCrop)، التحويل إلى Tensor (transforms.ToTensor())، والتطبيع (Normalization).

في الختام، يُمثل هذا النهج تطبيقاً عملياً لمبادئ التعلم العميق المتقدمة. تُساهم هذه المنهجية في بناء وتطوير نماذج تصنيف صور قوية، قابلة للتعميم، وذات كفاءة عالية، مما يجعلها أساساً متيناً لمختلف تطبيقات الرؤية الحاسوبية، وخاصة في مجال فحص جودة الخيوط القطنية. إن الجمع بين قوة النماذج المدربة مسبقاً والمحسّنات التكيفية وتقنيات زيادة البيانات يضمن تحقيق أداء عالٍ حتى مع مجموعات بيانات محدودة نسبياً.

5. القسم العملي (الدراسة التجريبيّة)

5.1. تمهيد

يواجه هذا البحث العديد من التحديات، وهي: توليد صور متعددة من صورة معيارية واحدة تمثل درجة جودة الخيط، تدريب نماذج ذكية للقيام بعملية تصنيف صورة الخيط المختبر، بناء منصة تعمل على الخادم المحلي للتأكد من تأمين تجربة مستخدم مثالية، رفع المنصة على خادم خارجي لجعل استثمار المنصة متاح للجميع. بناء على ما تقدم سيتم تنفيذ المذكورة في الفقرات التالية.

5.2. توليد الصور اللازمة لعملية تدريب النماذج

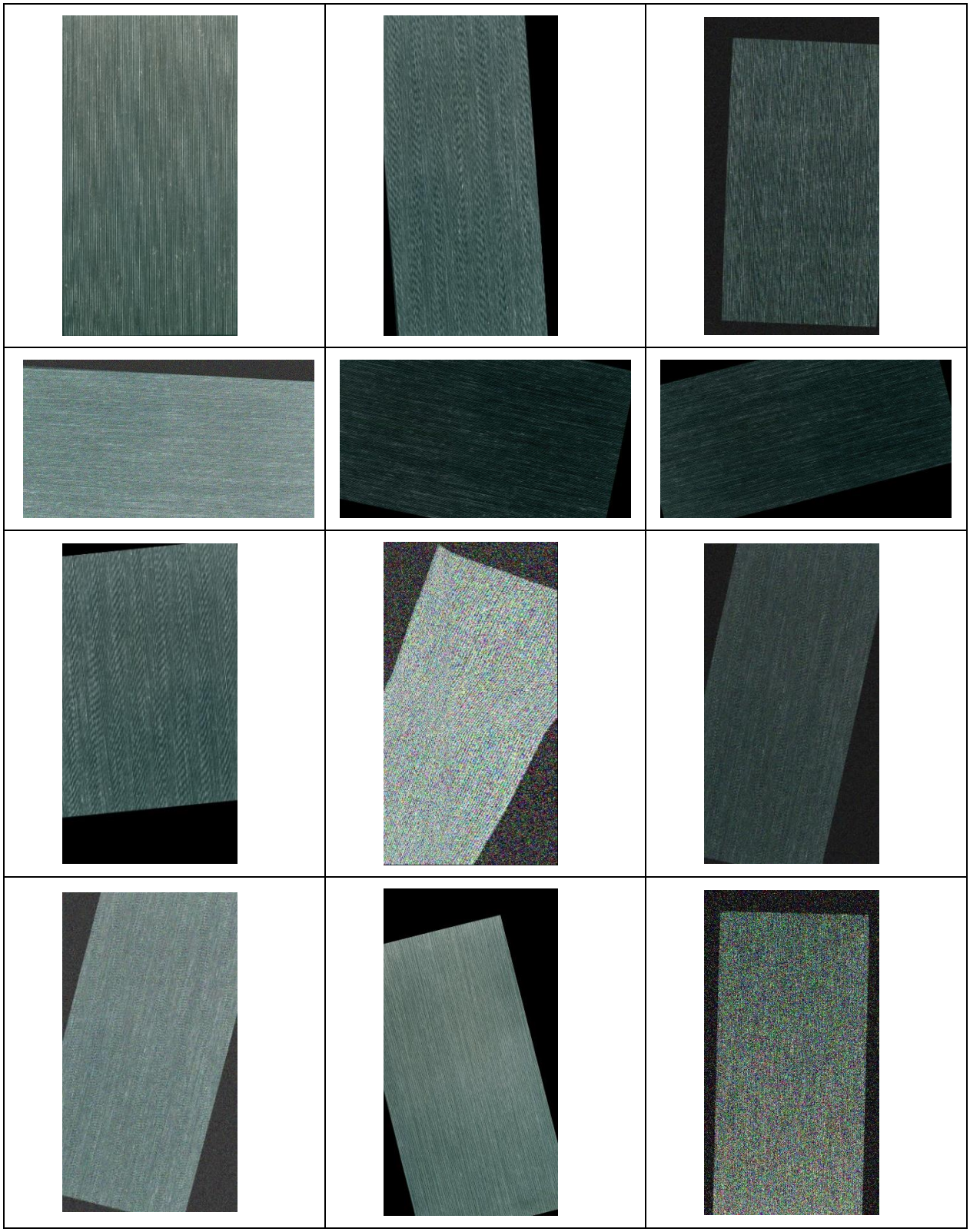
تتضمّن المواصفة القياسية لتقييم مظهرية الخيوط القطنية، خمس سلاسل من الصور المعيارية. كل سلسلة تُمثّل مجال نمر خيوط محدد حسب الجدول (2-1). تتكوّن كل سلسلة من أربع صور معيارية، كل صورة تمثل درجة مظهرية الخيط (A, B, C, D) حيث الدرجة (A) هي الأفضل و(D) هي الأسوأ. بالمجمل، ينتج 20 صورة معيارية معتمدة ومستخدمة في عملية تصنيف جودة الخيوط القطنية. وهنا تظهر الحاجة لتوليد صور جديدة، والتي تلزم لعملية تدريب خوارزميات التصنيف مظهرية الخيوط القطنية. تُعد توسيع البيانات (Data Augmentation) استراتيجية محورية في مجال الرؤية الحاسوبية (Computer Vision) لتعزيز أداء نماذج التعلم العميق، لا سيما في سيناريوهات محدودة لتوفير البيانات التدريبية [3].

في هذا البحث تم تطبيق منهجية توسيع البيانات بالاعتماد على مكتبات بايثون متخصصة لضمان الكفاءة والمرونة في عملية توليد البيانات، مما يُسهم بشكل مباشر في بناء مجموعات بيانات تدريبية أكثر قوة، وتنوعاً، وواقعية. يتمحور النهج المتبع حول الاستفادة القصوى من مكتبة **Albumentations**، وهي إطار عمل عالي الأداء لتوسيع بيانات الصور، معروف بقدرته على توفير مجموعة واسعة من التحويلات المدعومة بوحدة معالجة الرسوميات (GPU) لتحقيق سرعة تنفيذ فائقة [4]. يتضمن الرماز المصدري (Code) المستخدم، عمليات أساسية ومتقدمة تهدف إلى محاكاة الاختلافات الطبيعية والمعقدة في ظروف التقاط الصور والخصائص الهندسية للكائنات، وهي:

- **التحويلات الهندسية الأساسية:** تشمل القلب الأفقي (A.HorizontalFlip) والعمودي (A.VerticalFlip)، والتدوير العشوائي (A.RandomRotate90)، وتحويلات التحول والقياس والتدوير المتزامنة (A.ShiftScaleRotate). تُسهم هذه التحويلات في جعل النموذج مقاوماً للتغيرات في التوجيه، الحجم، والموضع داخل الصورة.
- **تعديلات الألوان والضوضاء:** تتضمن (A.RandomBrightnessContrast) تعديلات السطوع والتباين العشوائية، وإضافة الضوضاء الغاوسية (A.GaussNoise). هذه التحويلات تُحاكي الاختلافات في ظروف الإضاءة والتشويش التي قد تواجهها النماذج في البيئات الواقعية.
- **التحويلات التشوهية المتقدمة:** يشمل التحويل المرن (A.ElasticTransform)، وتشويه الشبكة (A.GridDistortion)، والتشويه البصري (A.OpticalDistortion). تُعد هذه التحويلات ذات أهمية خاصة لأنها تُحدث تشوهات هندسية دقيقة وغير خطية في الصورة، مما يزيد من قوة النموذج على التعميم ويجعله أكثر قدرة على التعامل مع التشوهات الطفيفة في شكل الكائنات أو الخلفية [6] [5].

من جهة أخرى، تم تضمين مكتبة **OpenCV (cv2)**، لقراءة الصور وتغيير أبعادها وحفظها بكفاءة. يتم تحجيم الصور المُولّدة إلى عرض مستهدف مع الحفاظ على نسبة الأبعاد، ممّا يضمن توحيد حجم الصور في مجموعة البيانات الجديدة. كما أنه تم استخدام مكتبتي os وglob للتعامل السلس مع نظام الملفات والبحث الفعال عن الصور المرجعية، ممّا يجعل العملية برمتها قابلة للتطوير والتكيف مع هياكل المجلدات المختلفة.

يظهر الشكل (5−1) صورة خيط قطني أصلية في الزاوية العلوية اليسرى، وبجانبها عدة صور لنفس الخيط بعد تطبيق تحويلات متقدمة من .Albumentations يجب أن تُظهر الصور تحويلات مثل القلب الأفقي والعمودي، والتدوير، وتغيير السطوع والتباين، بالإضافة إلى التحويلات التشوهية مثل التحويل المرن (ElasticTransform)أو تشويه الشبكة (GridDistortion) لإبراز قدرة المكتبة على توليد تنوع كبير وواقعي في البيانات.



18الشكل (5-1): بعض الأمثلة على التحويلات الناتجة عن توسيع البيانات المتقدمة باستخدام*Albumentations*

5.3. توليد نماذج التصنيف

5.3.1. تقسيم مجموعات البيانات لتدريب نماذج التعلم العميق وتحسين تقييم الأداء

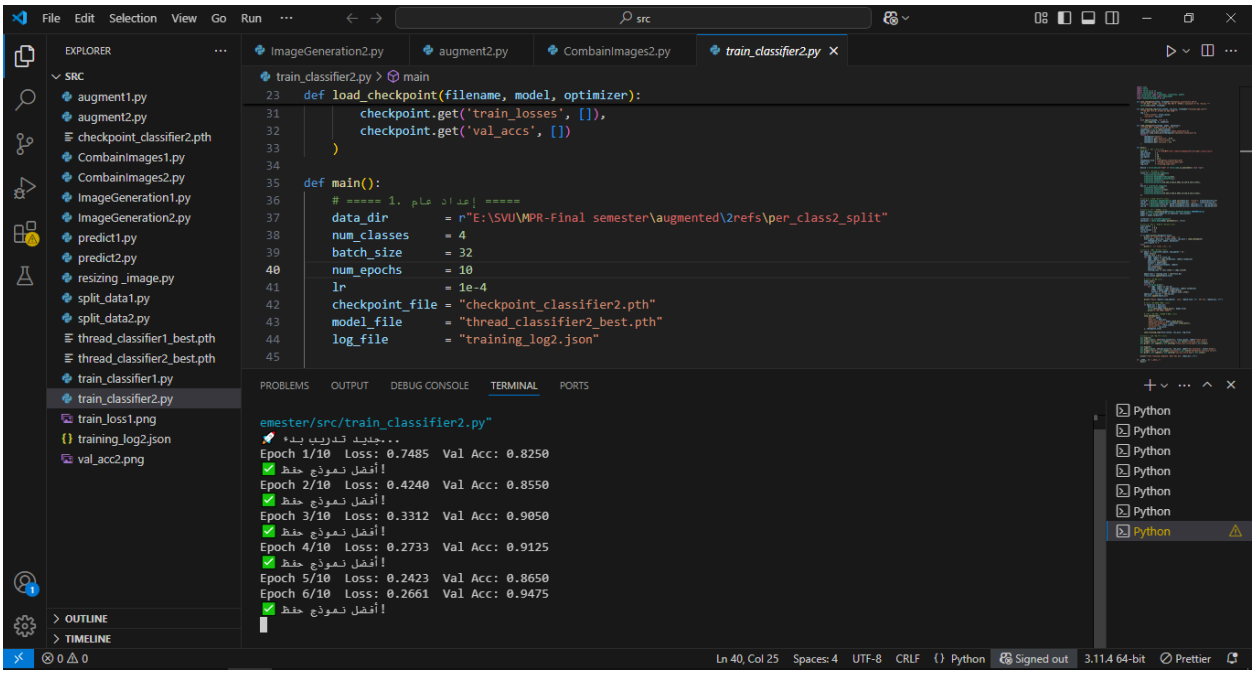
تُعد عملية تقسيم مجموعات البيانات (Dataset Splitting) خطوة أساسية في تطوير وتقييم نماذج التعلم الآلي (Machine Learning) والتعلم العميق (Deep Learning)، خاصة في سياق الرؤية الحاسوبية (Computer Vision) [12]. يهدف هذا التقسيم إلى إنشاء مجموعات فرعية متميزة: مجموعة تدريب (training set) لتعليم النموذج، ومجموعة تحقق (validation set) لضبط المعلمات الفائقة (hyperparameters) وتقييم الأداء أثناء التدريب، ومجموعة اختبار (test set) لتقييم الأداء النهائي للنموذج على بيانات غير مرئية تماماً [13]. يُقدّم الرماز المصدري (Code) المستخدم آلية فعالة ومنظمة لتقسيم مجموعة بيانات صور مُعدّة مسبقاً إلى مجموعتي تدريب وتحقّق بنسب محددة. حيث تم تقسيم البيانات بنسبة 80% للتدريب (training) و20% للتحقق (validation). تُسهم هذه الخطوات في بناء أساس متين لتدريب وتقييم نماذج التعلم العميق، مما يؤدّي إلى تطوير حلول رؤية حاسوبية أكثر موثوقية وقدرة على التعميم في التطبيقات الواقعية.

5.3.2. تدريب وتقييم نماذج التعلم العميق لتصنيف الصور

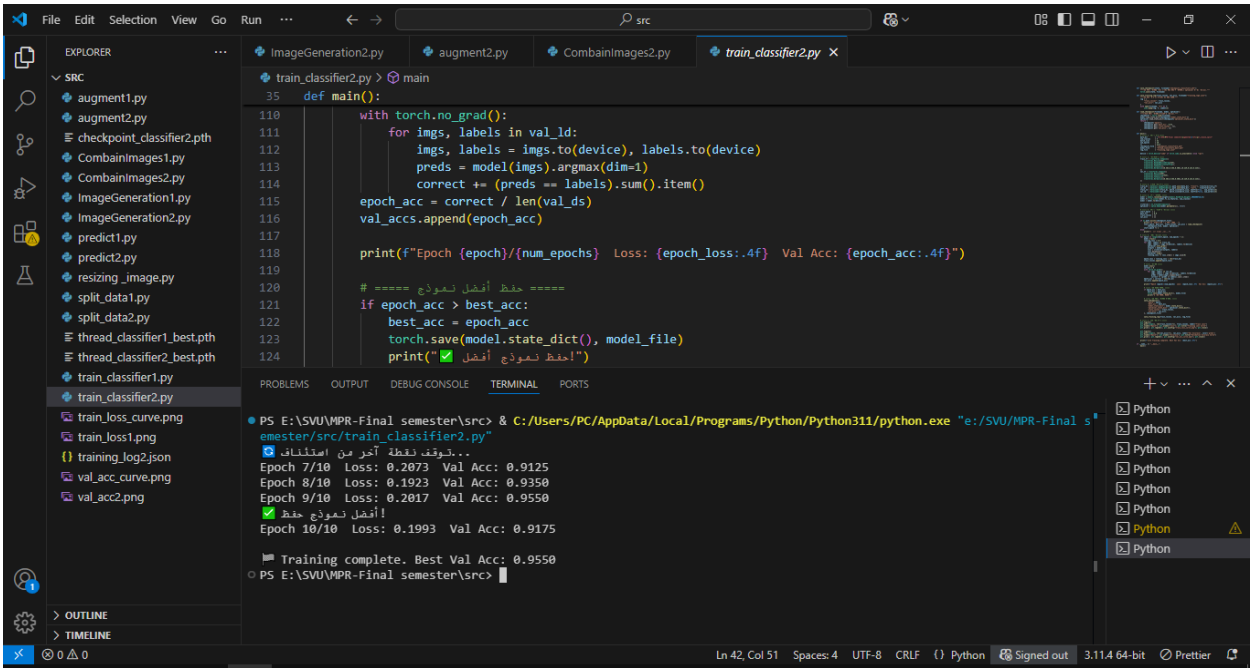
يُقدم الرماز المصدري (Code) المستخدم إطاراً شاملاً لتدريب شبكة عصبية تلافيفية (Convolutional Neural Network - CNN) باستخدام تقنية نقل التعلم (Transfer Learning)، مع التركيز على تحسين المعلمة(Parameter Optimization) بواسطة مُحسن آدم (Adam .Optimizer). حيث تم تطبيق مجموعة من التحويلات (Transformations) باستخدام مكتبة PyTorch's torchvision.transforms. هذه التحويلات ضرورية لتوحيد أبعاد المدخلات، زيادة تنوع البيانات التدريبية (Data Augmentation)، وتقليل الفرط في الملاءمة(Overfitting) [3]. بعد تطبيق هذه التحويلات، يتم تحميل مجموعات بيانات التدريب والتحقق بكفاءة باستخدام torch.utils.data.DataLoader. تُسهل هذه الفئة عملية تقسيم البيانات إلى دفعات (Batches) صغيرة، مما يُحسن من استخدام الذاكرة ويُسرع من عملية التدريب من خلال معالجة عدة عينات في وقت واحد. ثم تم تحميل نموذج ResNet18 [17] المستخدم بشكل كبير في تقنية نقل التعلم، والمُدرب مسبقاً على مجموعة بيانات ImageNet. يتم تكييف النموذج ليناسب مهمة تصنيف الخيوط الجديدة عن طريق استبدال الطبقة الخطية(Fully Connected Layer) النهائية للنموذج بطبقة جديدة تتوافق مع عدد الفئات المستهدفة

(أربع فئات في هذه الحالة). تُعد حلقة التدريب جوهر عملية التعلم. ففي كل دورة تدريب (Epoch)، يتم التكرار على مجموعة بيانات التدريب بأكملها. وبعد كل حلقة تدريب، تُجرى مرحلة التحقق (Validation) لتقييم أداء النموذج على بيانات لم يرها أثناء التدريب.

تُساهم هذه المنهجية في بناء نماذج تصنيف صور قوية، قابلة للتعميم، وذات كفاءة عالية، مما يجعلها أساساً متيناً في مجال فحص جودة الخيوط القطنية. يُظهر الشكل (5–2) آلية عمل خوارزمية التدريب للسلسلة الثانية علماً أنّه تم تحديد عدد دورات التدريب بعشر دورات. ويُظهر الشكل (5–3) آلية توقف خوارزمية التدريب، وإظهار النتيجة المثالية. تبيّن من خلال التجربة بأنه عند عدد دورات تدريب أكثر من ذلك تستغرق عملية التدريب الكثير من الوقت ولتجنب الوقوع في حالة (Overfitting)، وأقل من ذلك العدد لا يتم الحصول على النتائج المرجوة من عملية التصنيف. ويُلاحظ من الصور أن أفضل دقة لعملية التدريب تم الحصول عليها هي (0.955).

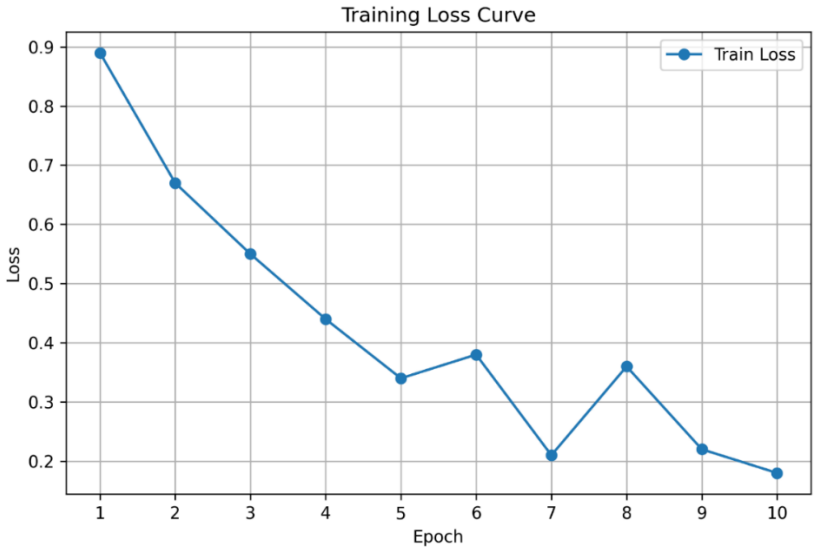


الشكل (5–2): البدء بعملية تدريب النموذج الخاص بتصنيف السلسلة الثانية 19

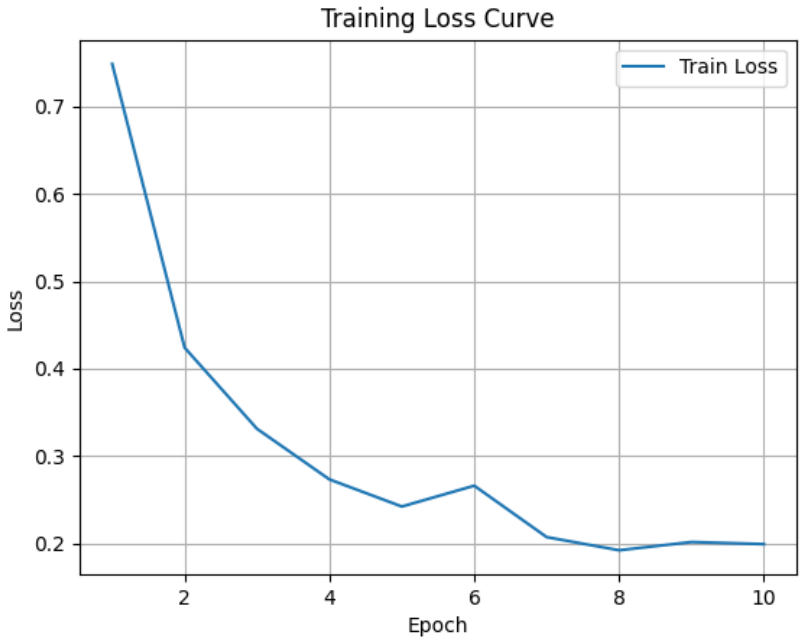


20الشكل (5-3): الانتهاء من عملية تدريب النموذج الخاص بتصنيف السلسلة الثانية

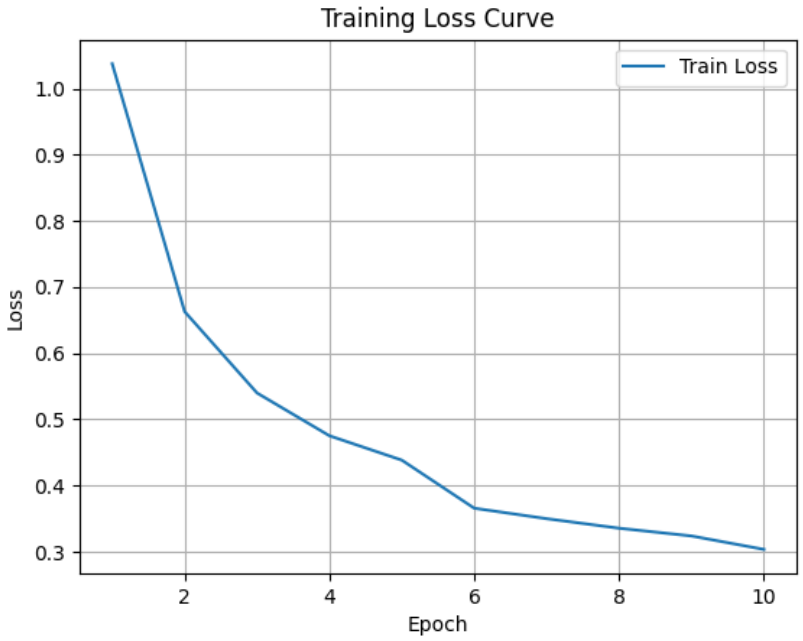
تُظهر الأشكال التالية (5-4)، (5-5)، (5-6)، (5-7)، (5-8)، منحنيات التدريب الخاصة بنماذج التصنيف للسلاسل الخمسة لنمر الخيوط.



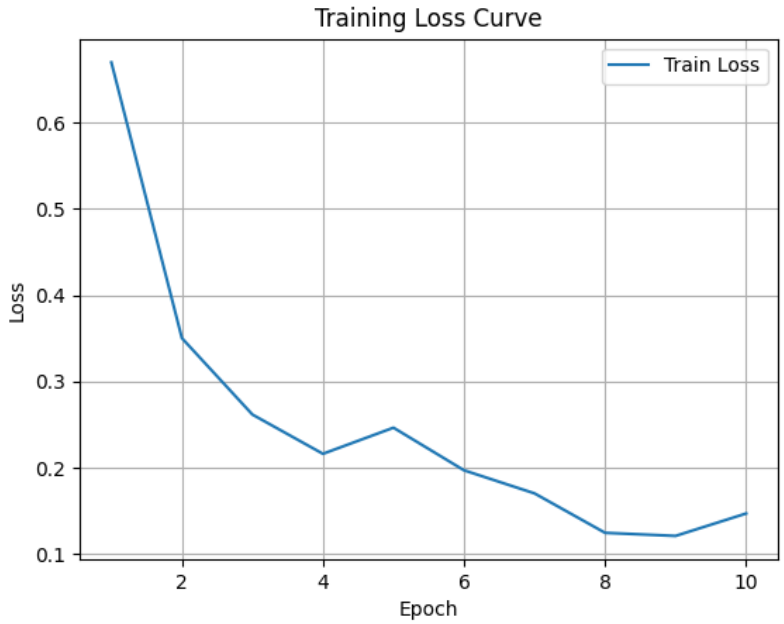
21الشكل (5-4): منحني تدريب السلسلة الأولى



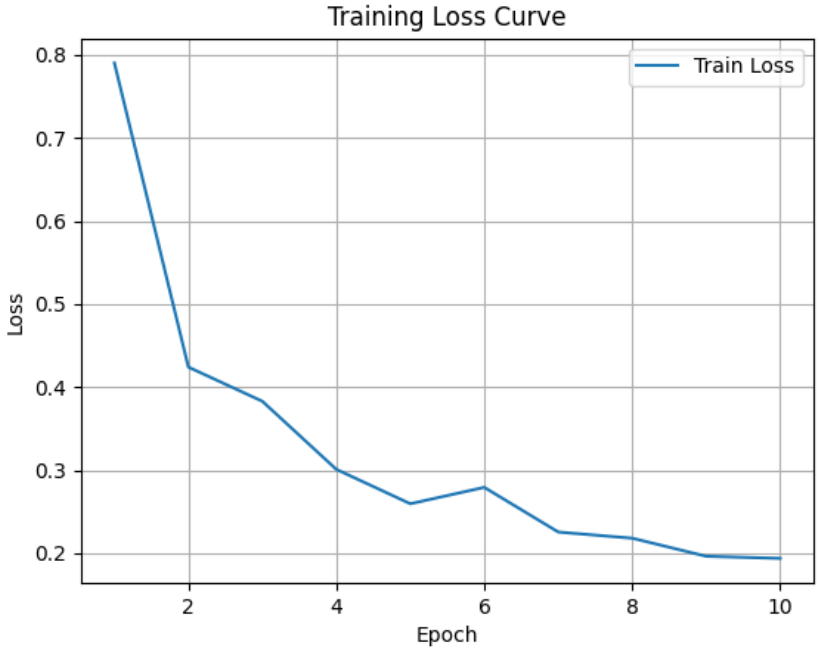
الشكل (5-5): منحني تدريب السلسلة الثانية22



الشكل (5-6): منحني تدريب السلسلة الثالثة23

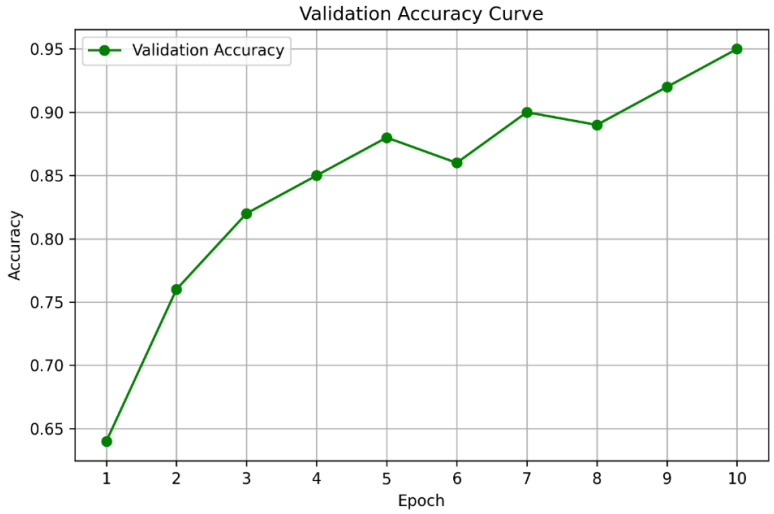


24الشكل (5-7): منحني تدريب السلسلة الرابعة

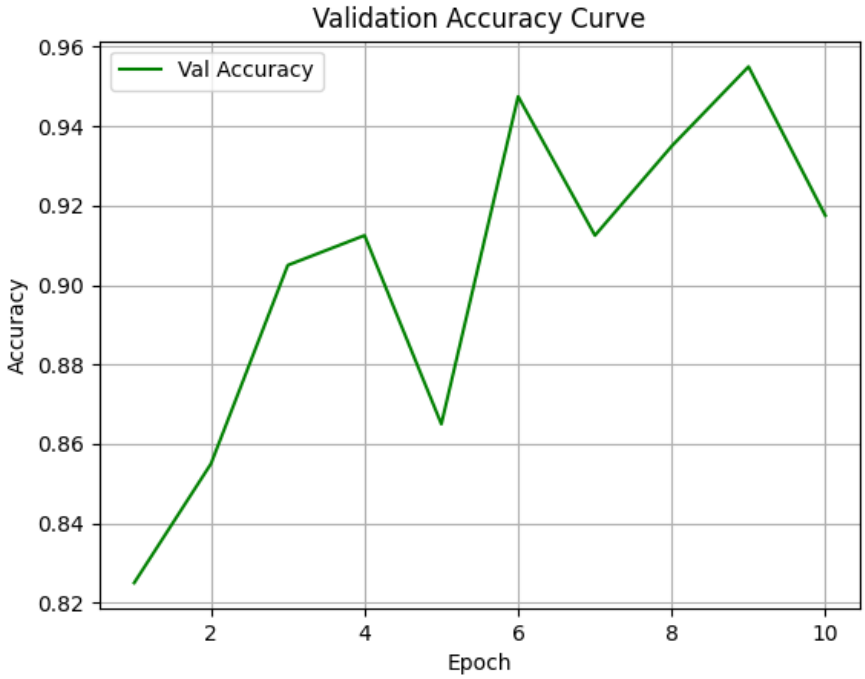


25الشكل (5-8): منحني تدريب السلسلة الخامسة

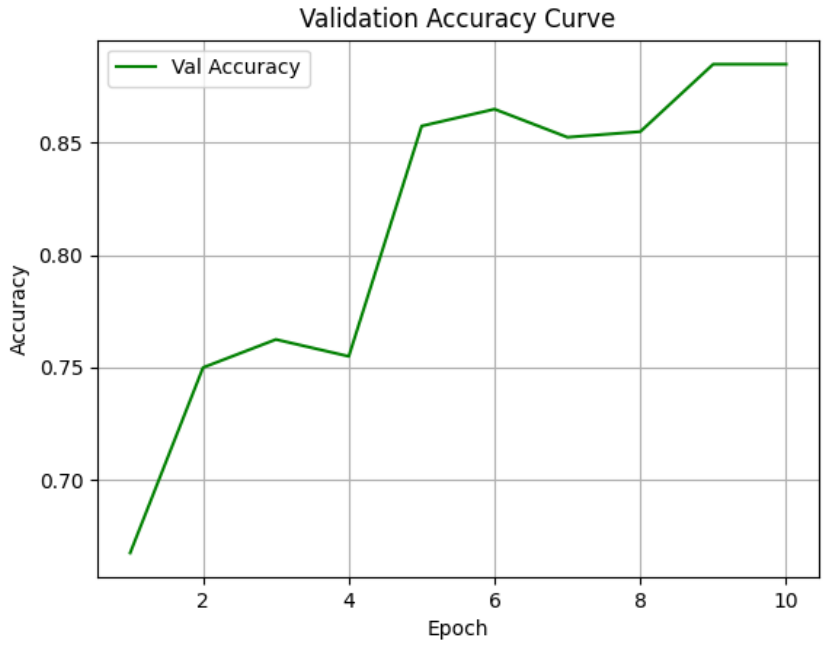
تُظهر الأشكال التالية (5-9)، (5-10)، (5-11)، (5-12)، (5-13)، منحنيات الدقة الخاصة بنماذج التصنيف للسلاسل الخمسة لنمر الخيوط.



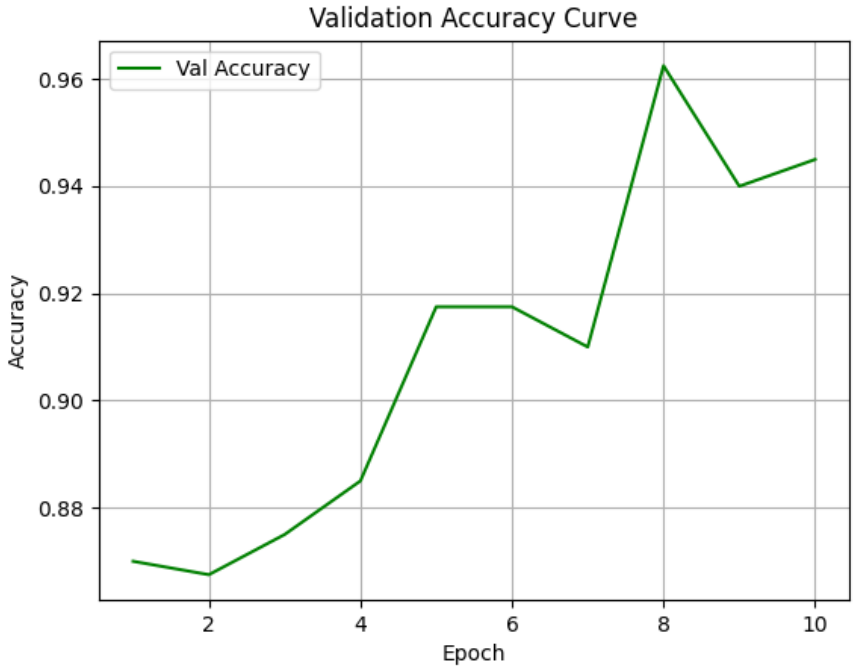
26الشكل (5–9): منحني دقة النموذج الأول



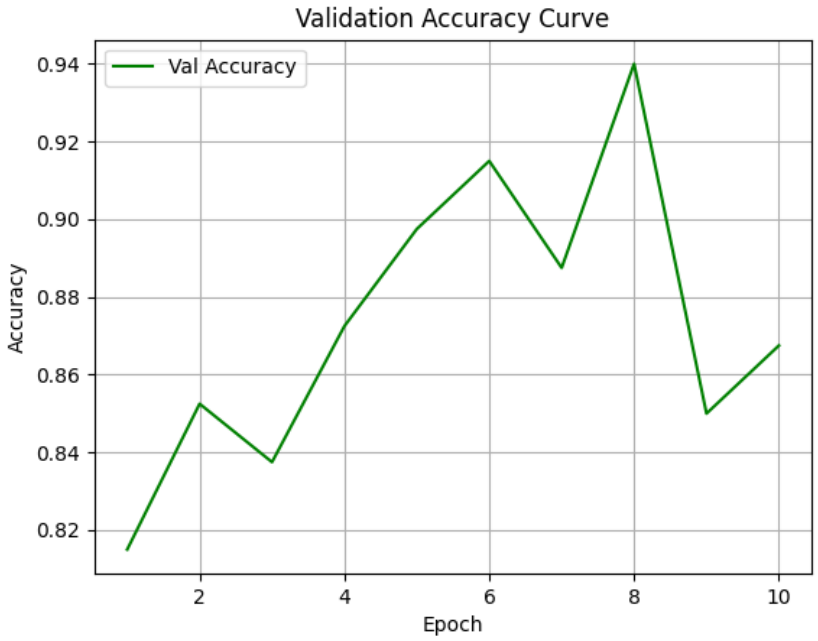
27الشكل (5–10): منحني دقة النموذج الثاني



الشكل (5–11): منحني دقة النموذج الثالث28



الشكل (5–12): منحني دقة النموذج الرابع29



30الشكل (5-13): منحني دقة النموذج الخامس

بعد الحصول على نماذج التصنيف المدربة وحفظها، تمّ اختبار النماذج بشكل مستقل. فيما يلي نتائج تجريب النموذج الثاني (الخاص بتصنيف السلسلة الثانية من الخيوط) على عينات عشوائية من السلسلة الثانية وكانت النتائج كما يلي:

```
    python predict2.py --image "E:/SVU/MPR-Final semester/test_images/A2_014.jpg" --weights
"E:/SVU/MPR-Final semester/src/thread_classifier2_best.pth"
    ✅Predicted class: A2  (Confidence: 100.0%)
    📊Probabilities:
     A2: 100.0%
     B2:   0.0%
     C2:   0.0%
     D2:   0.0%
    python predict2.py --image "E:/SVU/MPR-Final semester/test_images/B2_026.jpg" --weights
"E:/SVU/MPR-Final semester/src/thread_classifier2_best.pth"
    ✅Predicted class: B2  (Confidence: 71.7%)
    📊Probabilities:
     A2:   2.5%
     B2:  71.7%
     C2:  24.4%
     D2:   1.3%
    python predict2.py --image "E:/SVU/MPR-Final semester/test_images/C2_038.jpg" --weights
"E:/SVU/MPR-Final semester/src/thread_classifier2_best.pth"
    ✅Predicted class: A2  (Confidence: 50.1%)
    📊Probabilities:
     A2:  50.1%
     B2:   0.3%
```

```
    C2:  49.5%
    D2:   0.0%
  python predict2.py --image "E:/SVU/MPR-Final semester/test_images/D2_026.jpg" --weights "E:/SVU/MPR-Final semester/src/thread_classifier2_best.pth"
    ✅Predicted class: D2  (Confidence: 99.5%)
    📊Probabilities:
    A2:   0.3%
    B2:   0.0%
    C2:   0.2%
    D2:  99.5%
```

5.4. تقييم أداء نماذج التصنيف باستخدام مصفوفة الارتباك Confusion Matrix

5.4.1. تمهيد

غالباً ما تكون مقاييس تقييم أداء نماذج التعلم الآلي البسيطة مثل الدقة (Accuracy) غير كافية، خاصة في سياقات البيانات غير المتوازنة (Imbalanced Datasets) أو عندما تكون التكلفة المترتبة على أنواع معينة من الأخطاء غير متساوية. هنا تبرز أهمية **مصفوفة الارتباك (Confusion Matrix)** كأداة تحليلية قوية وشاملة، حيث توفر رؤية مفصلة حول أداء النموذج من خلال تلخيص نتائج التنبؤ بشكل منظم. تُستخدم مصفوفة الارتباك بشكل واسع في الأبحاث الأكاديمية والتطبيقات العملية لتشخيص نقاط القوة والضعف في النماذج، ومقارنة أداء النماذج المختلفة، واستخلاص مجموعة من مقاييس الأداء الدقيقة والموضوعية [25].

5.4.2. مكونات مصفوفة الارتباك

تتكون مصفوفة الارتباك من أربع مكونات رئيسية، يتم تمثيلها عادةً في جدول (2×2) [25، 26، 27]، كما هو موضح في الجدول (5-1). مكونات مصفوفة الارتباك هي:

- **الإيجابيات الحقيقية (True Positives - TP):** هي عدد الحالات الإيجابية التي تنبأ بها النموذج بشكل صحيح على أنها إيجابية. مثال: التنبؤ بأن شخصاً مصاب بمرض ما، وفعلياً هو مصاب به.
- **السلبيات الحقيقية (True Negatives - TN):** هي عدد الحالات السلبية التي تنبأ بها النموذج بشكل صحيح على أنها سلبية. مثال: التنبؤ بأن شخصاً غير مصاب بمرض ما، وفعلياً هو غير مصاب به.

- **الإيجابيات الكاذبة (False Positives - FP):** تُعرف أيضاً بالخطأ من النوع الأول .(Type I Error) هي عدد الحالات السلبية التي تنبأ بها النموذج بشكل خاطئ على أنها إيجابية. مثال: التنبؤ بأن شخصاً مصاب بمرض ما، وفعلياً هو غير مصاب به.
- **السلبيات الكاذبة (False Negatives - FN):** تُعرف أيضاً بالخطأ من النوع الثاني (Type II Error). هي عدد الحالات الإيجابية التي تنبأ بها النموذج بشكل خاطئ على أنها سلبية. مثال: التنبؤ بأن شخصاً غير مصاب بمرض ما، وفعلياً هو مصاب به.

الجدول (5-1): البنية الأساسية لمصفوفة الارتباك

	تنبأ النموذج بأنه إيجابي (Predicted Positive)	**تنبأ النموذج بأنه سلبي (Predicted Negative)**
القيم الفعلية إيجابية (Actual Positive)	الإيجابيات الحقيقية (TP)	السلبيات الكاذبة (FN)
القيم الفعلية سلبية (Actual Negative)	الإيجابيات الكاذبة (FP)	السلبيات الحقيقية (TN)

5.4.3. مقاييس الأداء المستخلصة من مصفوفة الارتباك

تتيح لنا مصفوفة الارتباك حساب مجموعة واسعة من المقاييس التي تقدم فهماً أعمق لأداء النموذج. فيما يلي أبرز هذه المقاييس:

- الدقة (Accuracy): تُعد الدقة هي أبسط مقياس، حيث تقيس نسبة التنبؤات الصحيحة من إجمالي عدد التنبؤات. تُعبر الدقة عن مدى نجاح النموذج في التنبؤ الصحيح بشكل عام.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- الدقّة النوعية (Precision): تُعرّف الدقة أيضاً بالقيمة التنبؤية الإيجابية (Positive Predictive Value). تقيس الدقة نسبة التنبؤات الإيجابية التي كانت صحيحة بالفعل. هذا المقياس مهم عندما تكون تكلفة الإيجابيات الكاذبة (FP) عالية.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- الاستدعاء (Recall): يُعرّف الاستدعاء أيضاً بالحساسية (Sensitivity) أو معدل الإيجابيات الحقيقية (True Positive Rate).يقيس الاستدعاء نسبة الحالات الإيجابية الفعلية التي تمكن النموذج من تحديدها بشكل صحيح. هذا المقياس مهم عندما تكون تكلفة السلبيات الكاذبة (FN) عالية.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- النوعية (Specificity): تُعرّف النوعية أيضاً بمعدل السلبيات الحقيقية (True Negative Rate). تقيس النوعية نسبة الحالات السلبية الفعلية التي تمكن النموذج من تحديدها بشكل صحيح. هذا المقياس مهم لتجنب التنبؤات الخاطئة للحالات السلبية.

$$\text{Specificity} = \frac{TN}{TN + FP}$$

- مقياس F1 (F1-Score): يُعد مقياس F1 هو المتوسط التوافقي (Harmonic Mean) للدقة (Precision) والاستدعاء (Recall). يُستخدم هذا المقياس لتقييم أداء النموذج عندما يكون هناك حاجة لتحقيق توازن بين الدقة والاستدعاء، خاصة في حالات عدم توازن البيانات.

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.4.4. نتائج تطبيق مصفوفة الارتباك على نماذج التصنيف الناتجة

توضّح الجداول: (5–2)، (5–3)، (5–4)، (5–5)، (5–6)، نتائج تقييم النماذج لكل سلسلة من السلاسل الخمسة المستخدمة في عملية تقييم جودة الخيوط. وتوضّح الأشكال (5–14)، (5–15)، (5–16)، (5–17)، (5–18)، رسومات مصفوفات الارتباك الخاصة بكل نموذج من نماذج التصنيف.

3الجدول (5–2): نتائج مقاييس مصفوفة الارتباك لنموذج التصنيف الخاص بالسلسلة الأولى

	precision	recall	F1-score	support
A1	0.82	0.86	0.84	100
B1	0.66	0.53	0.59	100
C1	0.67	0.80	0.73	100
D1	0.96	0.92	0.94	100
accuracy			0.78	400
macro avg	0.78	0.78	0.77	400
weighted avg	0.78	0.78	0.77	400

4الجدول (5–3): نتائج مقاييس مصفوفة الارتباك لنموذج التصنيف الخاص بالسلسلة الثانية

	precision	recall	F1-score	support
A2	0.95	0.99	0.97	100
B2	0.92	0.97	0.94	100
C2	0.99	0.86	0.92	100
D2	0.97	1.00	0.99	100
accuracy			0.95	400
macro avg	0.96	0.95	0.95	400
weighted avg	0.96	0.95	0.95	400

5الجدول (5–4): نتائج مقاييس مصفوفة الارتباك لنموذج التصنيف الخاص بالسلسلة الثالثة

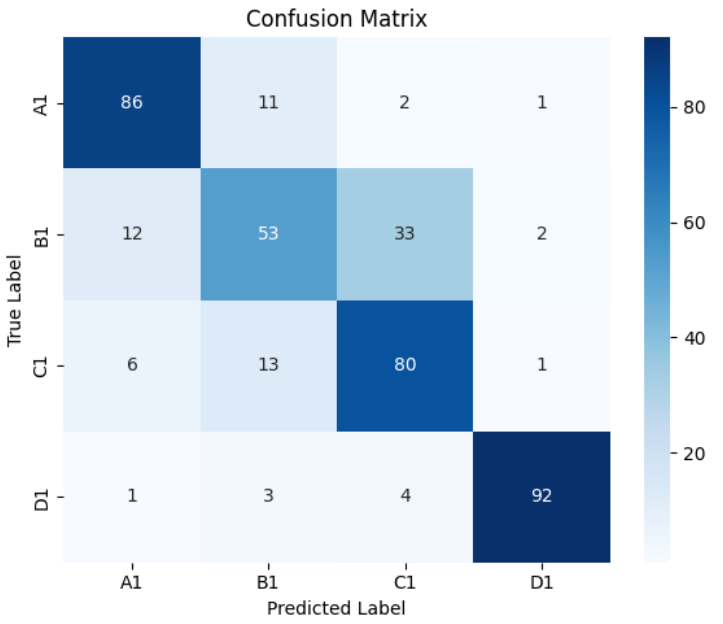
	precision	recall	F1-score	support
A3	0.96	0.81	0.88	100
B3	0.79	0.95	0.86	100
C3	0.87	0.98	0.92	100
D3	0.96	0.80	0.87	100
accuracy			0.89	400
macro avg	0.90	0.89	0.88	400
weighted avg	0.90	0.89	0.88	400

6الجدول (5–5): نتائج مقاييس مصفوفة الارتباك لنموذج التصنيف الخاص بالسلسلة الرابعة

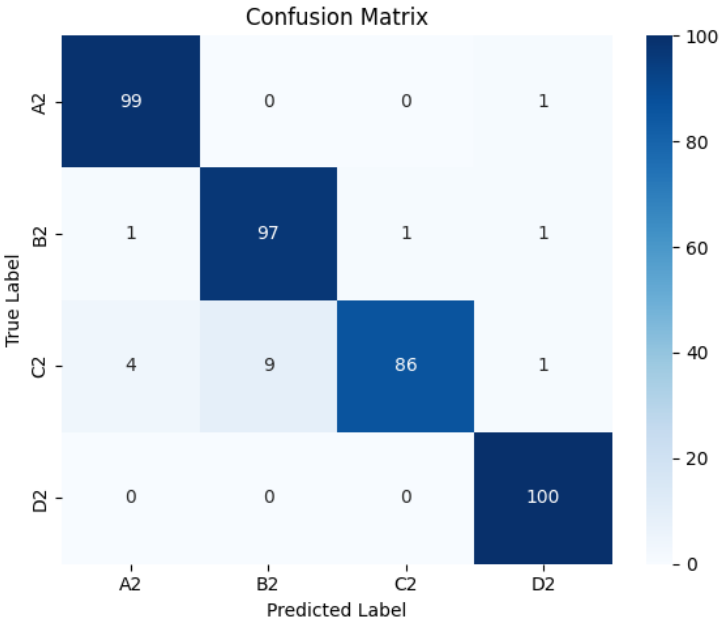
	precision	recall	F1-score	support
A4	1.00	0.87	0.93	100
B4	1.00	1.00	1.00	100
C4	0.92	0.98	0.95	100
D4	0.94	1.00	0.97	100
accuracy			0.96	400
macro avg	0.96	0.96	0.96	400
weighted avg	0.96	0.96	0.96	400

7الجدول (5–6): نتائج مقاييس مصفوفة الارتباك لنموذج التصنيف الخاص بالسلسلة الخامسة

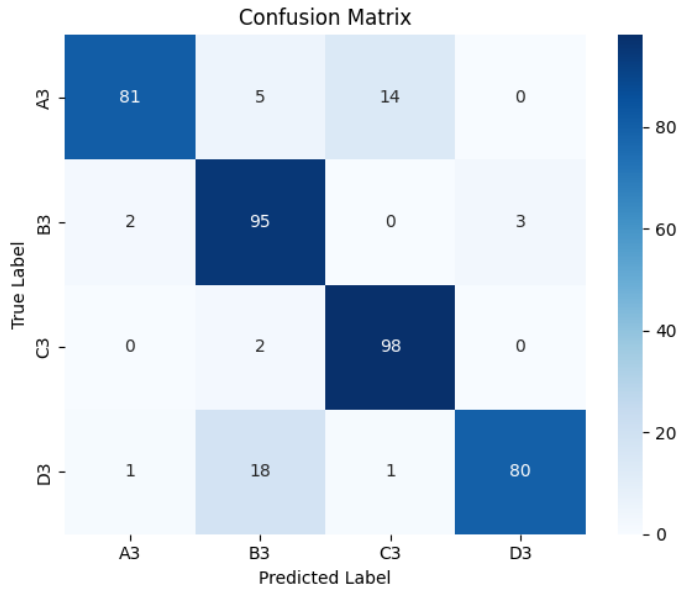
	precision	recall	F1-score	support
A5	0.97	1.00	0.99	100
B5	0.84	0.97	0.90	100
C5	0.97	0.93	0.95	100
D5	1.00	0.86	0.92	100
accuracy			0.94	400
macro avg	0.95	0.94	0.94	400
weighted avg	0.95	0.94	0.94	400



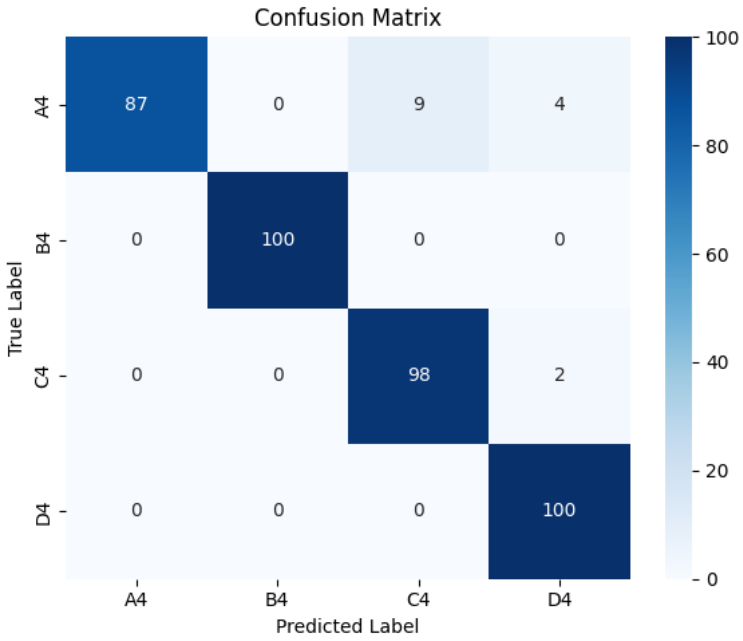
الشكل (5-14): مصفوفة الارتباك الخاص بنموذج التصنيف للسلسلة الأولى31



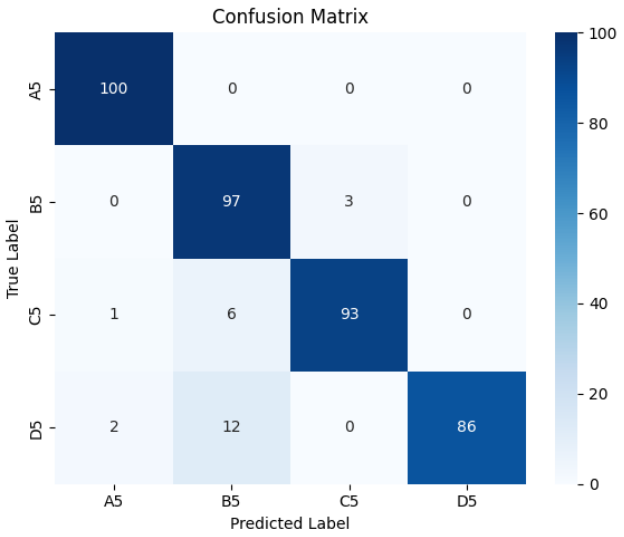
الشكل (5-15): مصفوفة الارتباك الخاص بنموذج التصنيف للسلسلة الثانية32



33الشكل (5–16): مصفوفة الارتباك الخاص بنموذج التصنيف للسلسلة الثالثة



34الشكل (5–17): مصفوفة الارتباك الخاص بنموذج التصنيف للسلسلة الرابعة



*35***الشكل (5–18): مصفوفة الارتباك الخاص بنموذج التصنيف للسلسلة الخامسة**

5.5. الدراسة التحليلية والتصميمية للمنصة المطلوبة

5.5.1. تحليل النظام Analysis

5.5.1.1. المتطلبات الوظيفية (Functional Requirements)

يبين الجدول (5–7) المتطلبات الوظيفية اللازمة لإنشاء المنصة المطلوبة [29, 28].

*8***الجدول (5–7): المتطلبات الوظيفية للنظام**

المتطلب	الوصف
تسجيل الدخول	يمكن للمستخدم الدخول إلى النظام باستخدام اسم المستخدم وكلمة المرور.
إنشاء حساب	يمكن للمستخدم التسجيل بحساب جديد وإدخال بياناته الشخصية.
رفع صورة	يستطيع المستخدم رفع صورة لوحة المظهرية وتحديد نمرة الخيط (Ne)
تصنيف الصورة	يقوم النظام بتحليل الصورة وتحديد النتيجة باستخدام النموذج المناسب.
عرض النتيجة	يظهر النظام للمستخدم نتيجة التصنيف والثقة المصاحبة بها.
عرض الملف الشخصي	يمكن للمستخدم عرض وتعديل بياناته الشخصية.
لوحة المستخدم	يمكن للمستخدم استعراض سجل الصور المرفوعة ونتائج التصنيف.
تحميل تقرير PDF	يسمح النظام بتحميل تقرير شامل يحتوي على كل التصنيفات السابقة.
تسجيل الخروج	يمكن للمستخدم الخروج من النظام بأمان.

5.5.1.2. *متطلبات الأداء* (Performance Requirements)

يبين الجدول (5–8) متطلبات الأداء اللازمة لإنشاء المنصة المطلوبة.

الجدول (5–8): المتطلبات الأداء الخاصة بالنظام *9*

المتطلب	الوصف
الاستجابة الفورية	يتم تصنيف الصورة وعرض النتيجة خلال أقل من 3 ثوانٍ.
دعم تعدد المستخدمين	يدعم النظام عدداً غير محدود من المستخدمين المسجلين.
تخزين دائم	تُخزّن البيانات والصور والتصنيفات في قاعدة بيانات آمنة.
التعامل مع الصور الكبيرة	يجب أن يدعم النظام ملفات صور حتى 5 ميغابايت دون مشاكل.
التوافق	يعمل النظام عبر جميع المتصفحات الحديثة (Chrome, Firefox, Edge).

5.5.1.3. *قيود التصميم* (Design Constraints)

- استخدام **Django** إطاراً للواجهة الخلفية.
- استخدام **Bootstrap 5** لتحسين المظهر والتجاوب.
- تُخزن الصور في مجلد media/uploaded_images/
- لا يُسمح باستخدام واجهات خارجية غير موثوقة.

5.5.1.4. *متطلبات الواجهة* (Interface Requirements)

يبين الجدول (5–9) متطلبات الواجهة اللازمة لإنشاء المنصة المطلوبة.

الجدول (5–9): متطلبات واجهة النظام *10*

العنصر	المتطلب
الواجهة الأمامية	HTML5, CSS3, Bootstrap
الواجهة الخلفية	Django Framework
قاعدة البيانات على الخادم المحلي	SQLite3
قاعدة البيانات على الخادم الخارجي	MySQL
واجهة المستخدم	يدعم العربية والإنكليزية، تصميم متجاوب
الأجهزة المدعومة	سطح المكتب، الحواسيب المحمولة، الأجهزة اللوحية
الصور المدخلة	JPG, PNG

5.5.2. حالات الاستخدام Use Cases

5.5.2.1. *الممثلون* (Actors)

- **المستخدم المسجّل (User):** يرفع الصور، يُدخل نمرة الخيط المطلوب تصنيفه، يطلب التصنيف، يستعرض التاريخ.
- **مدير النظام (Admin):** يشرف على المنصة عبر (/admin/).

5.5.2.2. *قائمة حالات الاستخدام الرئيسية*

يبين الجدول (5–10) حالات الاستخدام اللازمة لإنشاء المنصة المطلوبة.

***11*الجدول (5–10): المتطلبات الوظيفية للنظام**

اسم الحالة	الممثل	الهدف
تسجيل الدخول	المستخدم	الوصول إلى المنصة
التسجيل	المستخدم	إنشاء حساب
رفع صورة	المستخدم	إدخال صورة التصنيف
تصنيف الصورة	النظام	استخدام النموذج المناسب للتصنيف
عرض النتيجة	المستخدم	مشاهدة نتائج التصنيف
استعراض التاريخ	المستخدم	استعراض سجله من التصنيفات
تحميل التقرير PDF	المستخدم	تنزيل تقرير شامل
تحديث الملف الشخصي	المستخدم	تعديل بياناته

5.5.2.3. *صلاحيات المستخدمين*

يبين الجدول (5–11) المتطلبات الوظيفة اللازمة لإنشاء المنصة المطلوبة.

***12*الجدول (5–11): المتطلبات الوظيفية للنظام**

الدور	صلاحيات	الوصف
المستخدم العادي	رفع، تصنيف، طباعة تقارير، تعديل ملفه الشخصي	يرفع الصور ويشاهد نتائج التصنيف
المشرف(admin)	التحكم في المستخدمين، عرض جميع البيانات	مسؤول عن النظام بالكامل

5.5.2.4. *عمليات التصنيف* (Workflow)

- يسجل المستخدم الدخول.
- يرفع صورة ويُدخل نمرة الخيط.
- يحدد النظام تلقائياً النموذج المناسب حسب النمرة.
- يتم تصنيف الصورة وعرض النتيجة.
- تُحفظ النتيجة في قاعدة البيانات.
- يمكن للمستخدم طباعة تقرير PDF بالنشاطات.

5.5.2.5. *مواصفات حالات الاستخدام* (Use Case Specifications)

1) رفع صورة:

- **الممثل**: المستخدم
- **المدخلات**: صورة + نمرة الخيط
- العمليات:
  - يتحقق من صلاحية البيانات.
  - يحفظ الصورة في قاعدة البيانات.
  - يربطها بالمستخدم.
- **النواتج**: عرض نتيجة التصنيف.

2) تصنيف الصورة:

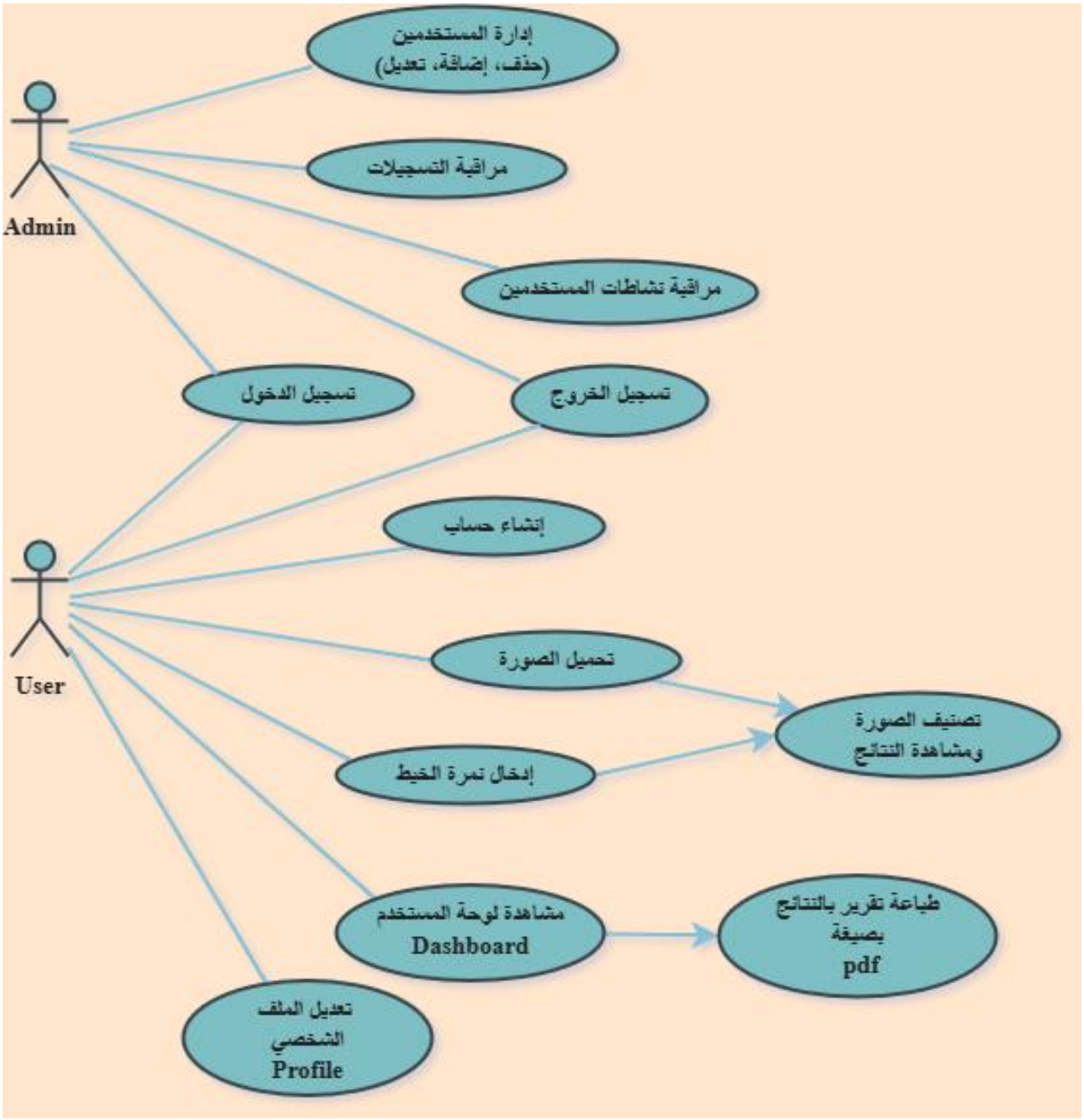
- الممثل: النظام
- العمليات:
  - تحديد النموذج المناسب حسب نمرة الخيط.
  - تحميل النموذج حسب انتماء نمرة الخيط لسلسلة التصنيف المناسبة.
  - تمرير الصورة.
  - حساب النتيجة والثقة.

3) استعراض التاريخ:

- الممثل: المستخدم
- المدخلات: لا شيء
- **المخرجات:** جدول يتضمن (الصورة، النمرة، النتيجة، التاريخ)

5.5.2.6. رسم مخطط حالات الاستخدام (Use Cases Diagram)

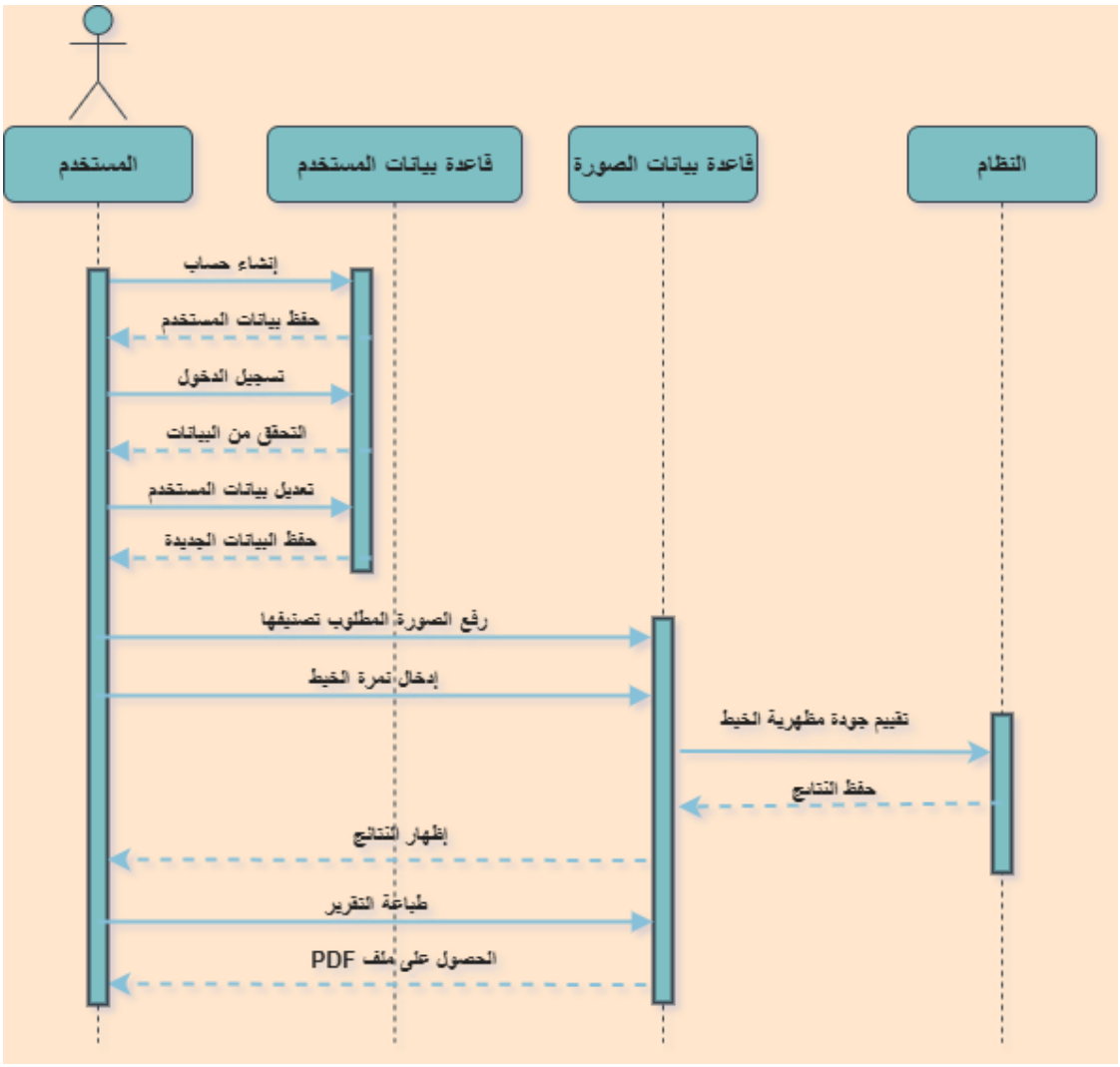
يبيّن الشكل (5-19) مخطط حالات الاستخدام للمنصة المطلوبة. وفيه يظهر المشرف والمستخدم مع المتطلبات الوظيفية الخاصة بكل منهما. علماً بأنه تم رسم مخطط حالات الاستخدام بواسطة موقع (draw.io).



36الشكل (5-19): مخطط حالات الاستخدام

7.5.2.5. *مخطط التسلسل* (sequence) *لأنشطة المستخدم*

يبيّن الشكل (5–20) مخطط التسلسل (sequence) للعمليات التي يقوم بها المستخدم. إنشاء حساب، تسجيل دخول، تعديل بياناته، رفع الصورة المطلوب تصنيفها، إدخال نمرة الخيط، ثم يقم النظام بعملية التصنيف ويتم حفظ النتائج ضمن قاعدة بيانات الصورة. يمكن للمستخدم البحث في قاعدة بيانات العمليات السابقة ومشاهدة الصور المرافقة وطباعة تقرير عن أنشطته بصيغة (PDF). تم رسم مخطط التسلسل (sequence) الخاص بأنشطة المستخدم بواسطة موقع (draw.io).



37الشكل (5–20): مخطط التسلسل لعمليات المستخدم

5.6. إنشاء قاعدة بيانات المنصة

5.6.1. توصيف قاعدة البيانات (Database Specification)

يمكن تحديد سمات كل كيان وإضافة القيود اللازمة وكذلك تحديد السمات التي تلعب دور المفتاح الرئيسي والسمات التي تلعب دور المفتاح الخارجي ومصدر المفتاح الخارجي [30]. يبيّن الجدول (5-12) نوع كيان "**User**" وسماته ونمط البيانات المستعملة وطبيعة قيمها المحتملة والقيود إن وُجدت.

13الجدول (5-12): مواصفات قاعدة بيانات كيان المستخدم *User*

اسم الكيان: User						
نوع الكيان	**السمة (الحقل)**	**نمط البيانات**	**وحيدة القيمة**	**متعددة القيمة**	**القيد (إن وُجد)**	**FK Source**
قوي	معرّف المستخدم user_id	Int	1	0	PK	–
	اسم المستخدم username	varchar	1	0	NOT NULL	–
	البريد الإلكتروني email	varchar	1	0	UNIQUE NOT NULL	–
	كلمة المرور password	varchar	1	0	ENCRYPTED NOT NULL	–
	هل المستخدم مشرف أم لا is_superuser	bol	1	0	DEFAULT (0)	–

يبيّن الجدول (5-13) نوع كيان "**UserProfile** " وسماته ونمط البيانات المستعملة وطبيعة قيمها المحتملة والقيود إن وُجدت.

14الجدول (5-13): مواصفات قاعدة بيانات الكيان *UserProfile*

اسم الكيان: UserProfile						
نوع الكيان	***السمة (الحقل)***	***نمط البيانات***	***وحيدة القيمة***	***متعددة القيمة***	***القيد (إن وُجد)***	***FK Source***
قوي	معرّف حساب المستخدم UserProfile_Id	Int	1	0	PK	-
	معرّف المستخدم user_id	varchar	1	0	FK	user_id
	الاسم الكامل full_name	varchar	1	0	UNIQUE NOT NULL	-
	المؤسسة/الجهة organization	varchar	1	0	NOT NULL	-
	رقم الهاتف phone_number	BIGINT	0	1	UNIQUE NOT NULL	-

يبيّن الجدول (5-14) نوع كيان "**UploadedImage**" وسماته ونمط البيانات المستعملة وطبيعة قيمها المحتملة والقيود

15الجدول (5-14): مواصفات قاعدة بيانات الكيان *UploadedImage*

اسم الكيان: UploadedImage						
نوع الكيان	***السمة (الحقل)***	***نمط البيانات***	***وحيدة القيمة***	***متعددة القيمة***	***القيد (إن وُجد)***	***FK Source***
ضعيف	معرّف الصورة المُحمّلة UploadedImage_id	Int	1	0	PK	-
	معرّف المستخدم user_id	varchar	1	0	FK	user_id
	ملف الصورة	image	0	1	UNIQUE NOT NULL	-
	نمرة الخيط thread_number	Float	1	0	NOT NULL	-
	النتيجة المتوقعة predicted_class	Float	1	0	NOT NULL	-
	درجة الثقة (نسبة مئوية) confidence	Float	1	0	NOT NULL	-
	تاريخ ووقت الرفع uploaded_at	DateTime2	1	0	NOT NULL	-

5.6.2. تحديد العلاقات بين الكيانات:

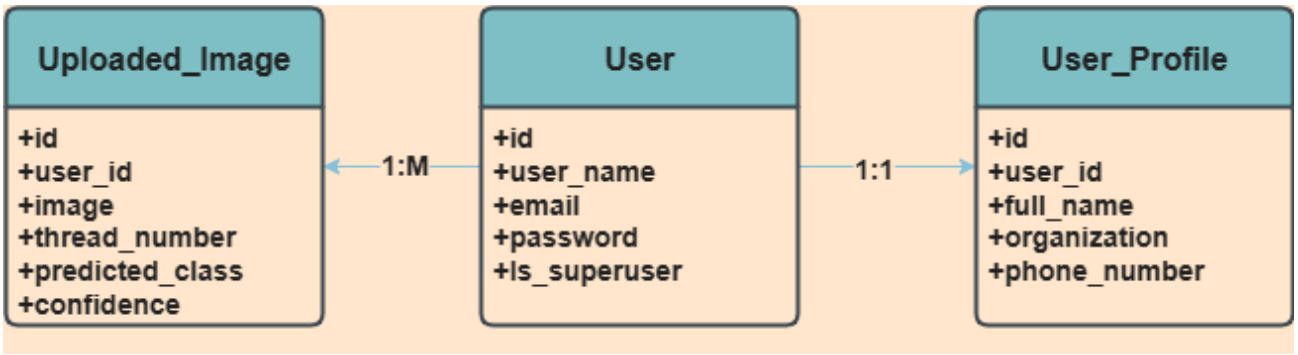
يوضح الجدول (5-15) طبيعة العلاقة بين الكيانات مع إضافة شرح عن طبيعة العلاقة بين كل كيانين مرتبطين.

16الجدول (5-15): العلاقة بين الكيانات

الكيانات المرتبطة	نوع العلاقة	الشرح
User ⟷ UserProfile	1:1	لكل مستخدم معلومات شخصية خاصة به
User ⟷ UploadedImage	1:M	يمكن للمستخدم رفع عدة صور للقيام بتصنيفها

5.6.3. مخطط كيان-علاقة (ERD):

تم رسم مخطط كيان-علاقة يدوياً باستخدام موقع (draw.io). لتوضيح قاعدة البيانات المطلوبة. يُظهر الشكل (5-21) نتيجة عملية الرسم والكيانات الموجودة في قاعدة البيانات وطبيعة العلاقة بين الكيانات.

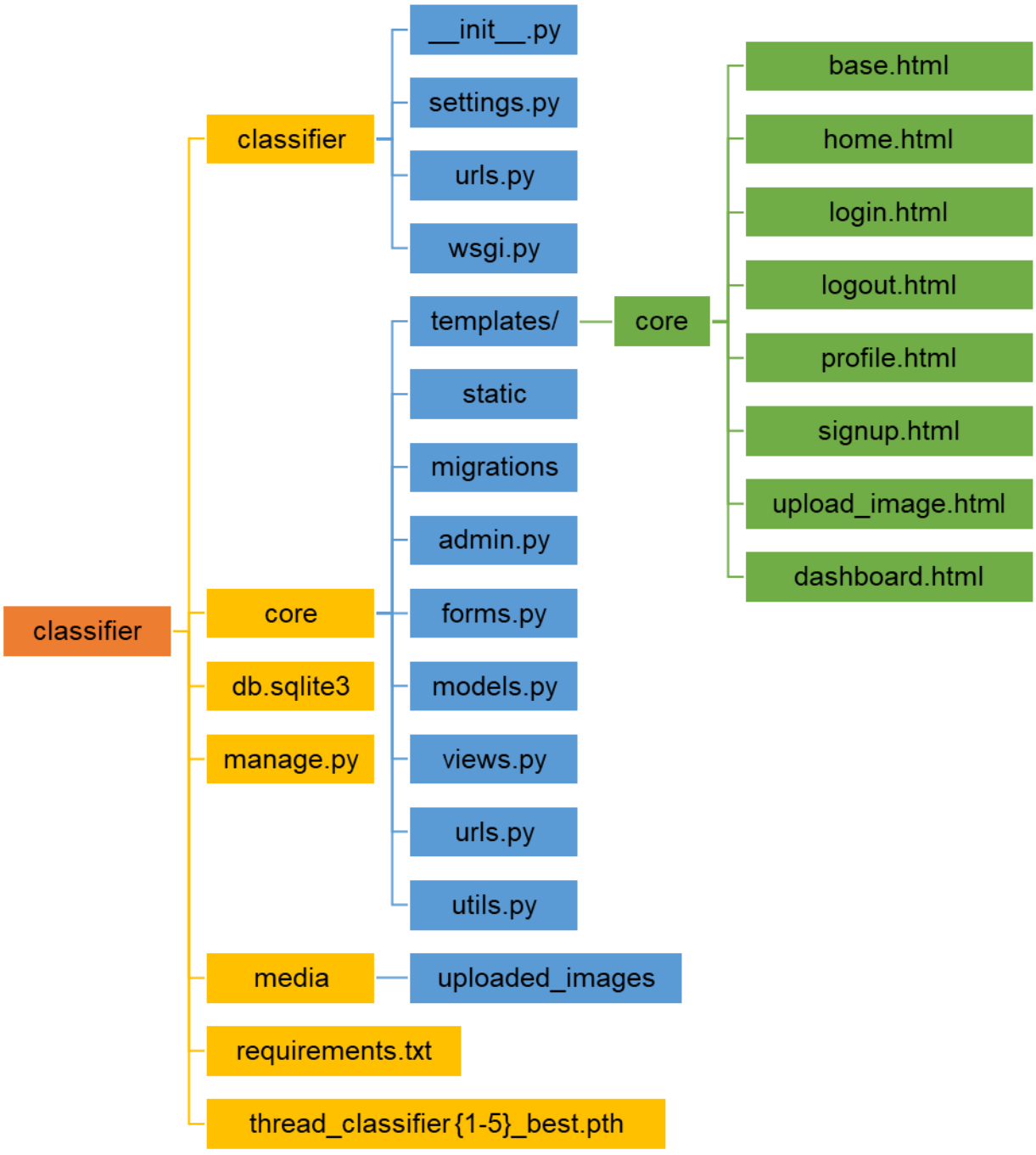


38الشكل (5-21): مخطط كيان-علاقة للمنصة المطلوبة

5.7. إنشاء المنصة على الخادم المحلي

5.7.1. شجرة الموقع

يظهر الشكل (5-22) توضّع ملفات المشروع في شجرة الموقع [31، 32، 33].



الشكل (5-22): الهيكل العام للمشروع39*(Project Structure)*

5.7.2. توصيف ملفات الموقع

1) __init__.py

- الوظيفة: يُستخدم لجعل المجلد الذي يحتوي عليه وحدة قابلة للاستيراد (package) في بايثون.
- الوصف: لا يحتوي عادة على منطق، بل هو ملف فارغ يشير إلى أن المجلد هو وحدة برمجية.

2) settings.py

- الوظيفة: يحتوي على إعدادات المشروع الأساسية.
- محتوياته: تعريف التطبيقات المثبتة، إعدادات قاعدة البيانات، مسارات الملفات الثابتة والوسائط، إعدادات اللغة والتوقيت، إعدادات الأمان.

3) urls.py في مجلد (classifier) وفي مجلد (core)

- الوظيفة: يحدد خريطة التوجيه (Routing) في الموقع، أي ربط روابط URL بالوظائف أو الصفحات المناسبة.
- مثال: عند دخول المستخدم على /upload/ يتم استدعاء upload_image في views.py.

4) wsgi.py

- الوظيفة: نقطة الدخول لتشغيل التطبيق في بيئة خوادم الإنتاج مثل Gunicorn أو uWSGI.
- الوصف: يستخدمه PythonAnywhere و Heroku لبدء تشغيل التطبيق.

5) admin.py

- الوظيفة: يسمح بتسجيل النماذج (Models) في لوحة تحكم Django الإدارية.
- التفاصيل: يتم من خلاله تخصيص طريقة عرض البيانات في لوحة المشرف.

6) forms.py

- الوظيفة: يحتوي على تعريف نماذج إدخال البيانات مثل نموذج تسجيل المستخدم أو رفع الصورة.
- الفائدة: يسهل التحقق من صحة البيانات (Validation) قبل تخزينها.

7) models.py

- الوظيفة: يحتوي على تعريفات قواعد البيانات باستخدام Django ORM.
- أمثلة:
  - UploadedImage: لحفظ صور المستخدمين.
  - UserProfile: لتخزين معلومات إضافية عن المستخدم.

8) views.py

- الوظيفة: يحتوي على المنطق البرمجي لتنفيذ الوظائف استجابة لطلبات المستخدم.
- أمثلة: رفع الصورة، تنفيذ التصنيف، عرض لوحة المستخدم.

9) utils.py

- الوظيفة: يحتوي على دوال مساعدة مثل:
  - تحميل النموذج المدرب.
  - تنفيذ التصنيف.
  - حساب الثقة والنتائج.

10) thread_classifier{1-5}_best.pth

- الوظيفة: ملفات تحتوي على النماذج المدربة باستخدام PyTorch.
- الوصف: يتم تحميلها تلقائياً من utils.py عند تنفيذ التصنيف لكل نمرة.

11) db.sqlite3

- الوظيفة: قاعدة بيانات SQLite يتم إنشاؤها تلقائياً لتخزين المستخدمين والصور المصنفة.
- الوصف: تحتوي على الجداول المرتبطة بـ models.py.

12) manage.py

- الوظيفة: أداة سطر الأوامر لإدارة المشروع.
- أمثلة للاستخدام:
  - تشغيل الخادم المحلي python manage.py runserver
  - إجراء الهجرات python manage.py migrate
  - إنشاء مستخدم مشرف createsuperuser

13) ملف requirements.txt

- الوظيفة: يحتوي على قائمة بجميع المكتبات المطلوبة لتشغيل المشروع.
- الوصف: يُستخدم عند تنصيب المشروع على خادم خارجي مثل PythonAnywhere.

14) مجلد uploaded_images

- الوظيفة: يُخزّن فيه الصور التي يرفعها المستخدم.
- الوصف: يتم ضبط مسار التخزين من خلال إعدادات MEDIA_ROOT وMEDIA_URL في settings.py.

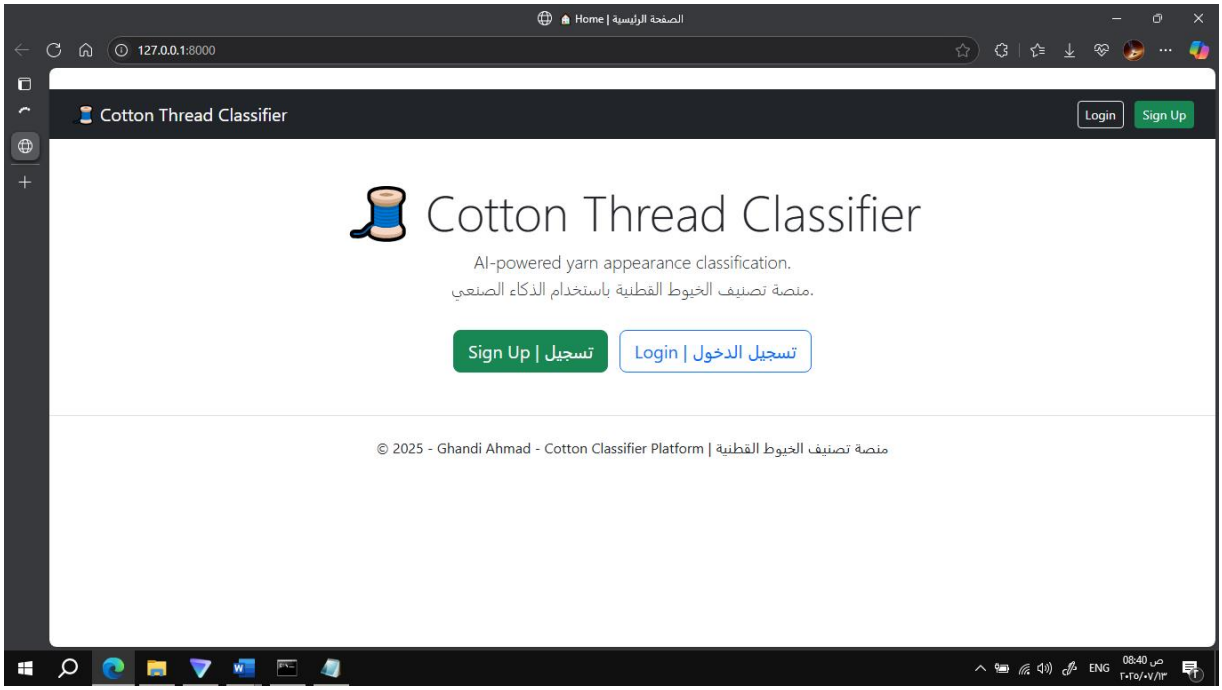
15) ملفات HTML (base.html, home.html, ...)

يبيّن الجدول (5–16) الوصف العام لملفات HTML التي تمثل الواجهة الأمامية للمنصة وتُعرض للمستخدم.

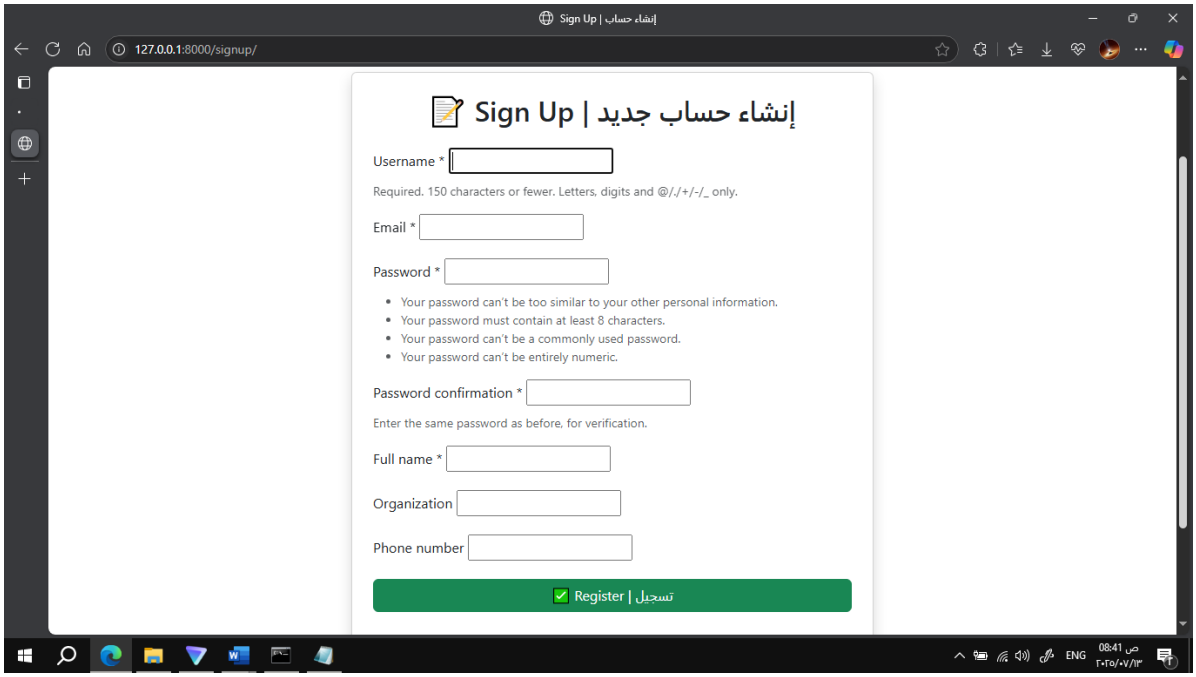
17الجدول (5–16): توصيف وظائف ملفات *HTML* المستخدمة في المنصة

الملف	الوظيفة
base.html	القالب الرئيسي الذي يُورّث في باقي الصفحات.
home.html	الصفحة الرئيسية التي ترحب بالمستخدم وتعرض خيارات التسجيل والدخول.
login.html	صفحة تسجيل الدخول.
logout.html	صفحة إعلام المستخدم بالخروج.
profile.html	صفحة عرض وتعديل معلومات المستخدم.
signup.html	صفحة إنشاء حساب جديد.
upload_image.html	صفحة رفع صورة خيط وتنفيذ التصنيف.
dashboard.html	صفحة لوحة النتائج التي تعرض التصنيفات السابقة وتحميل التقرير.

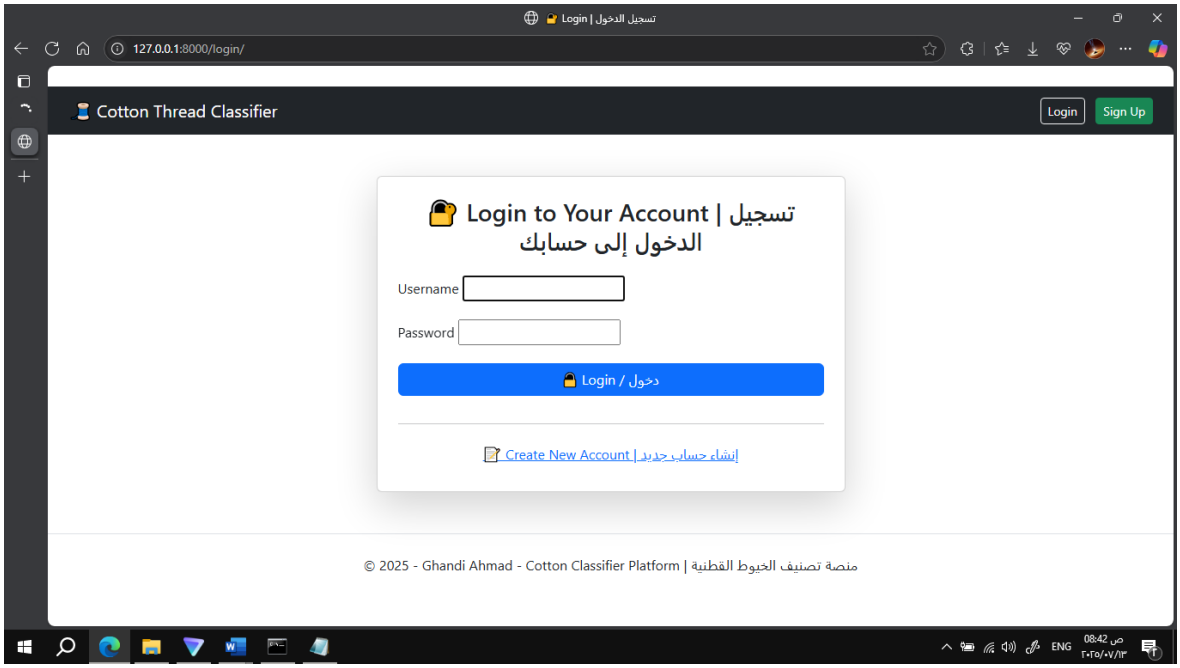
تُبيّن الأشكال من (5–23) إلى (5–40) صوراً لآلية استخدام المنصة المخصصة لتصنيف الخيوط القطنية.



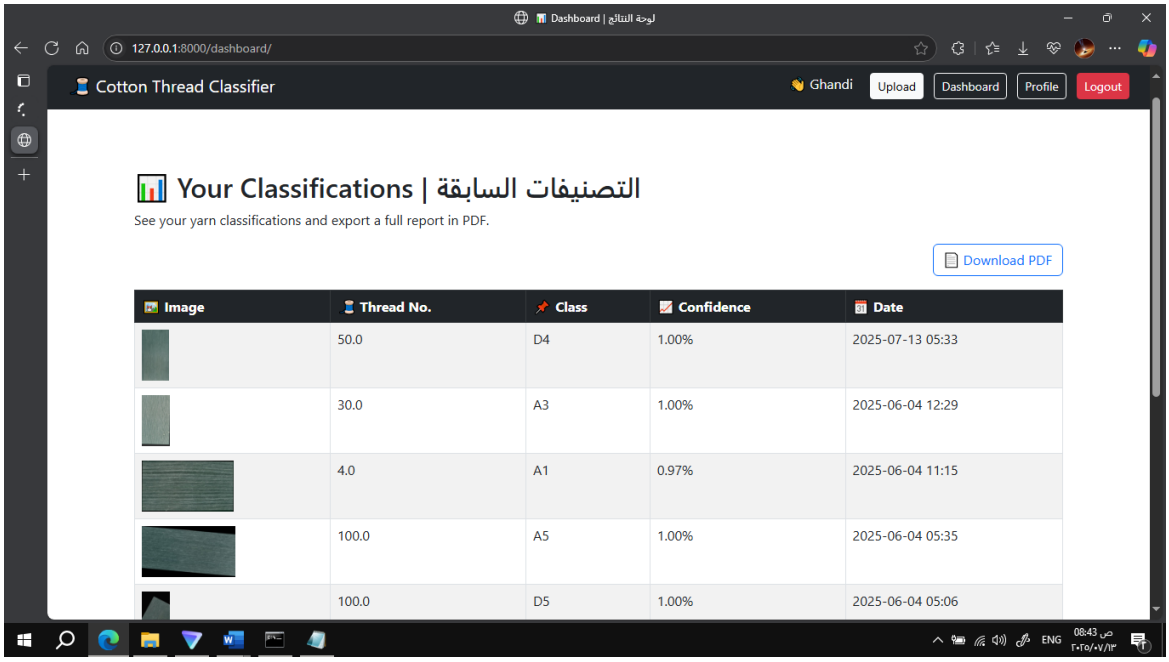
40الشكل (5–23): الصفحة الرئيسية للمنصة



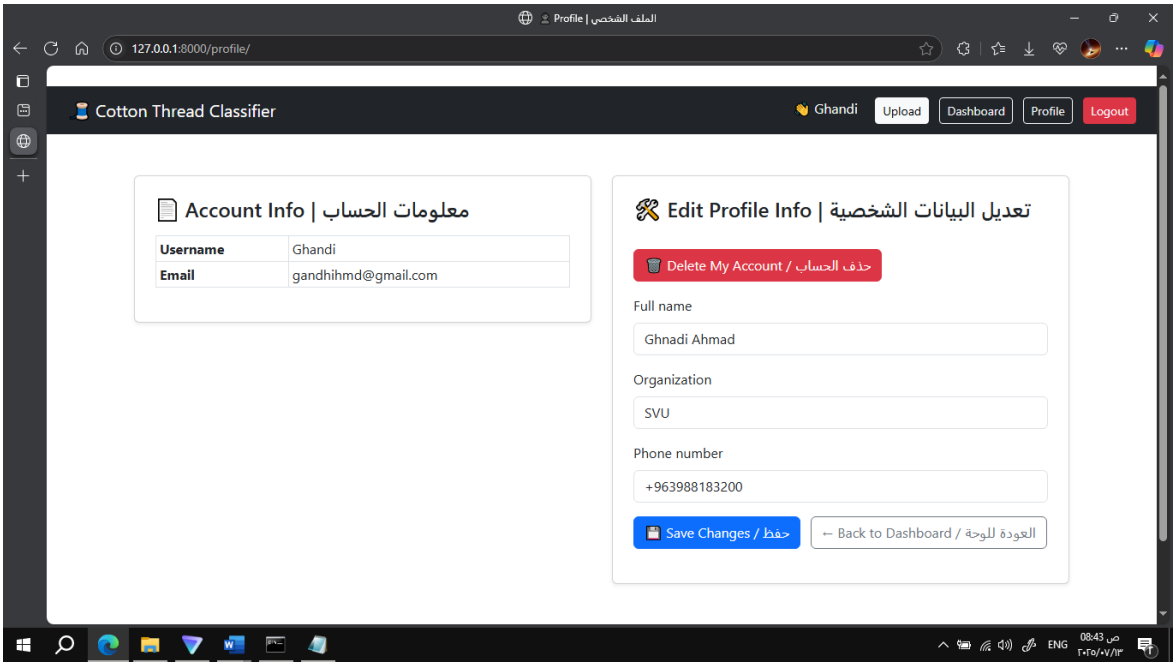
41الشكل (5-24): صفحة إنشاء حساب جديد



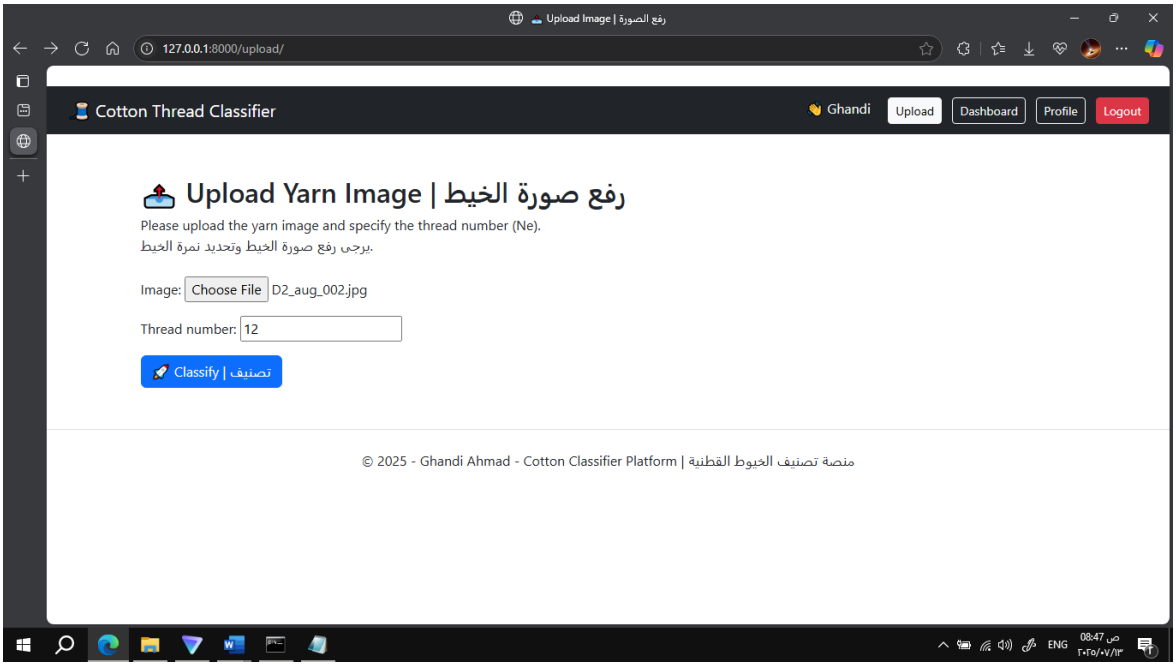
42الشكل (5-25): صفحة تسجيل الدخول



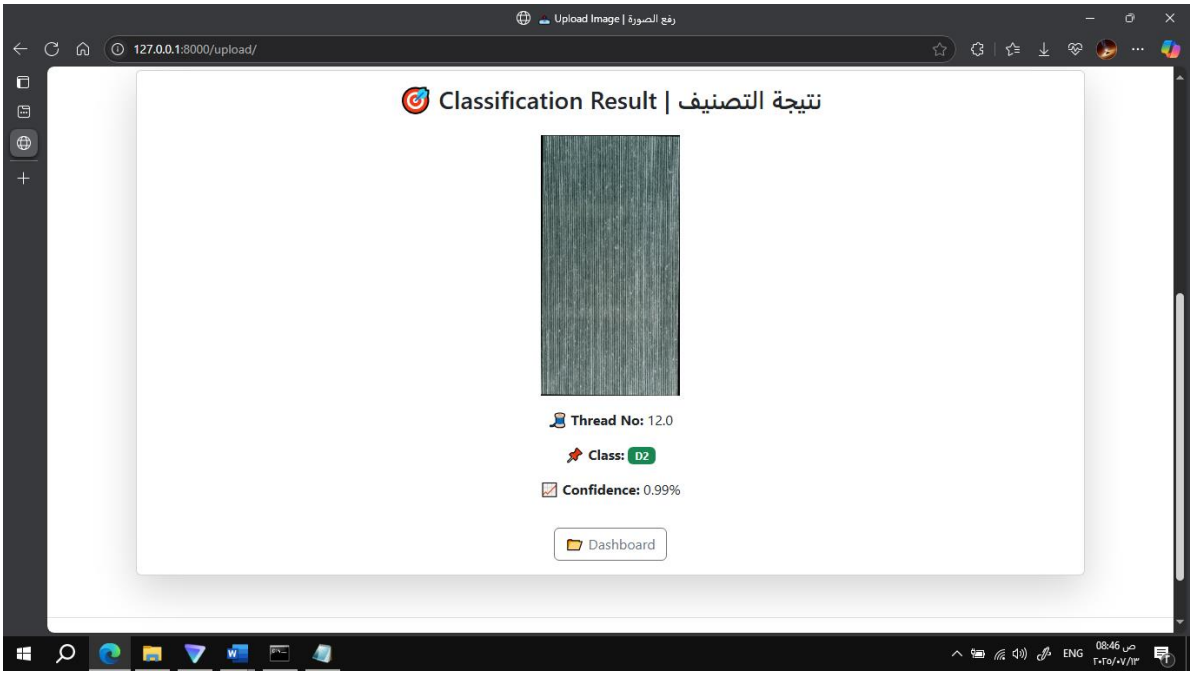
الشكل (5-26): صفحة *Dashboard* المستخدم 43



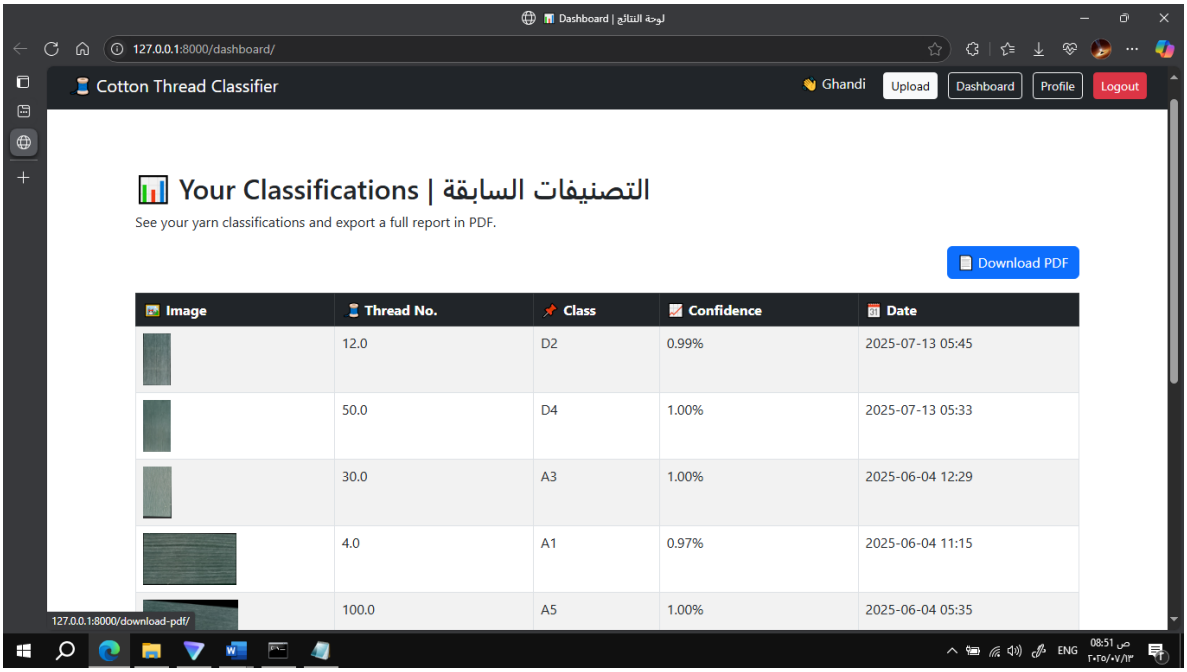
الشكل (5-27): صفحة *Profile* المستخدم 44



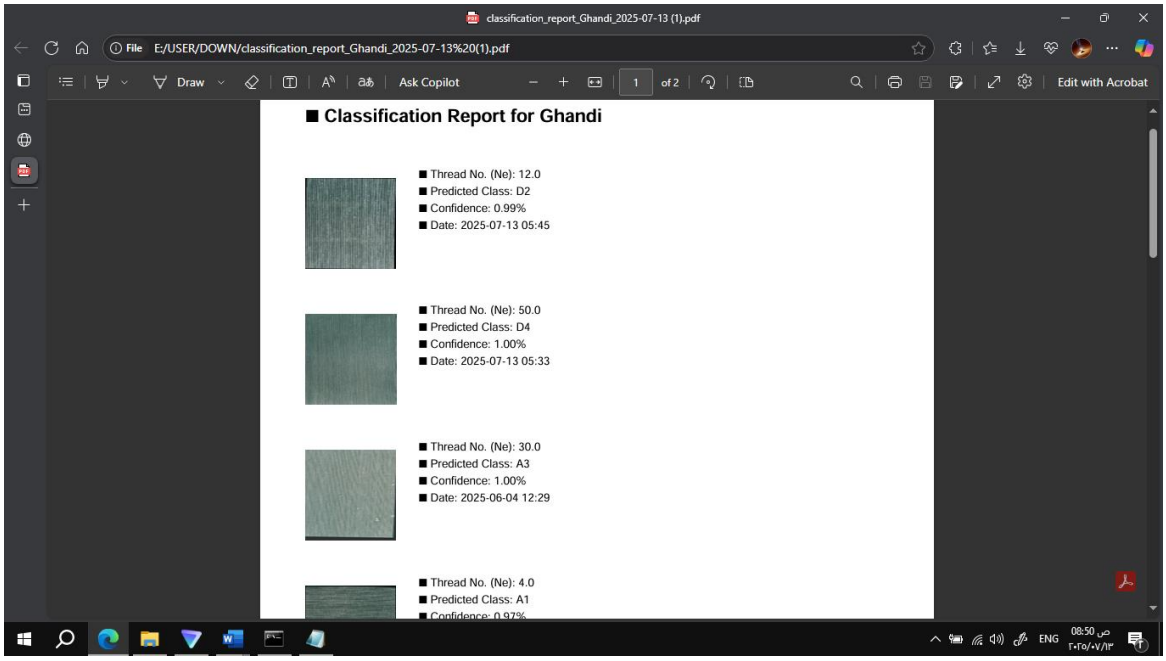
45الشكل (5-28): صفحة رفع الصورة وإدخال النمرة



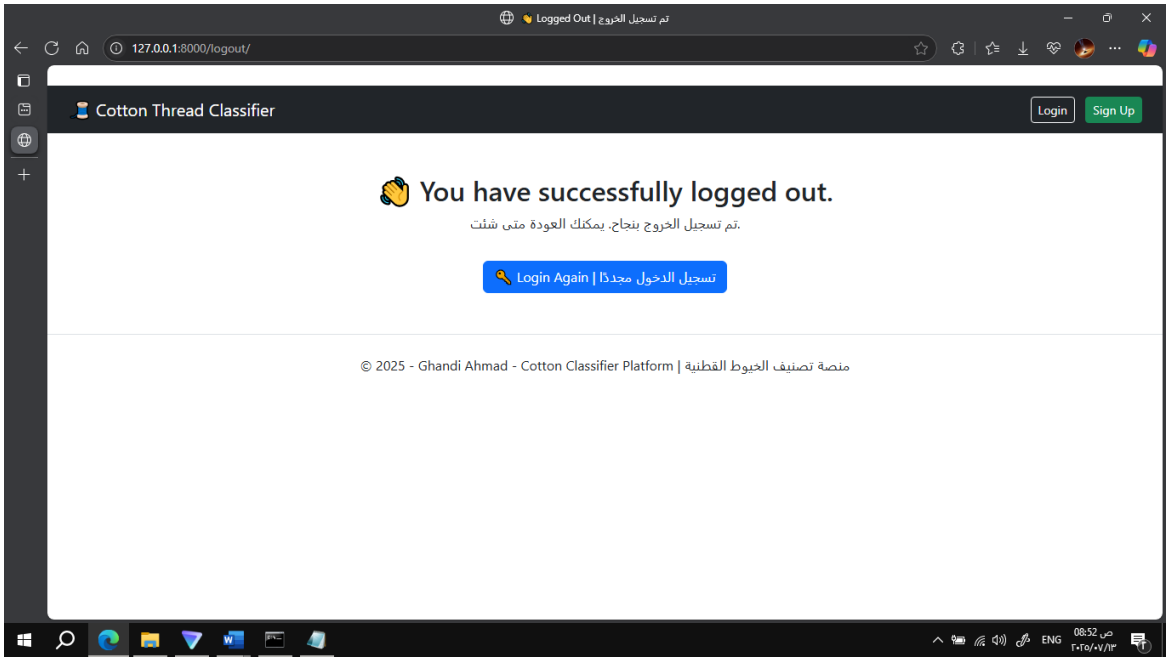
46الشكل (5-29): صفحة عرض النتيجة



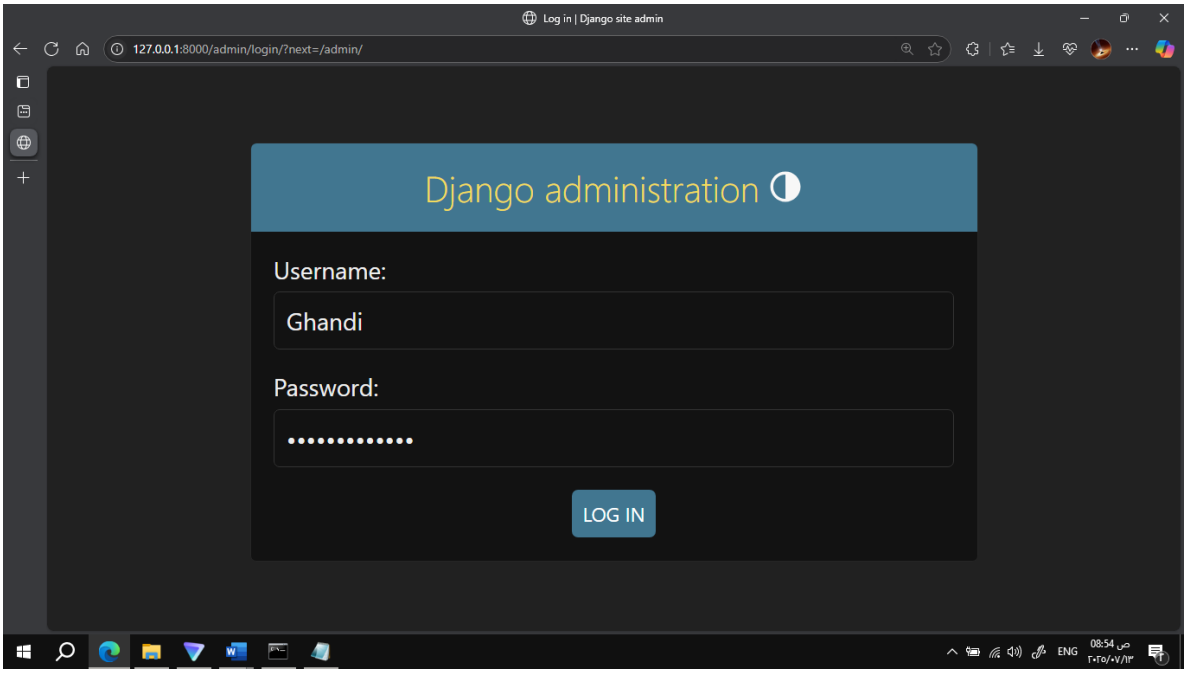
47الشكل (5-30): صفحة تظهر إضافة النتيجة الجديدة إلى*Dashboard*



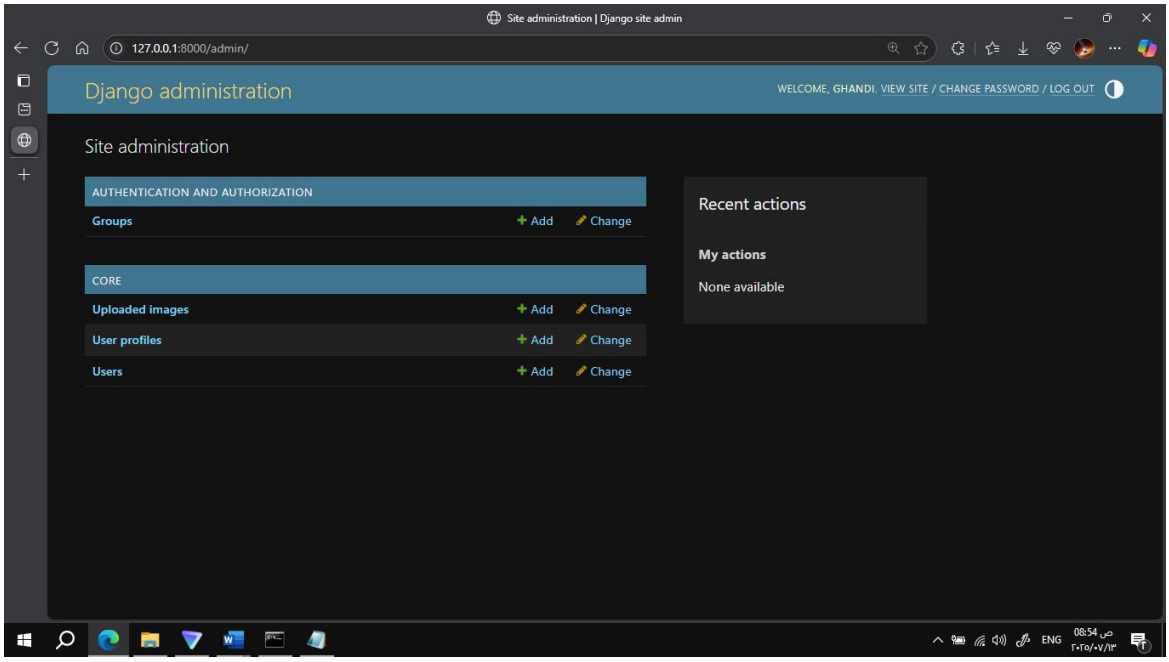
48الشكل (5-31): صفحة تظهر النتائج بشكل ملف*pdf*



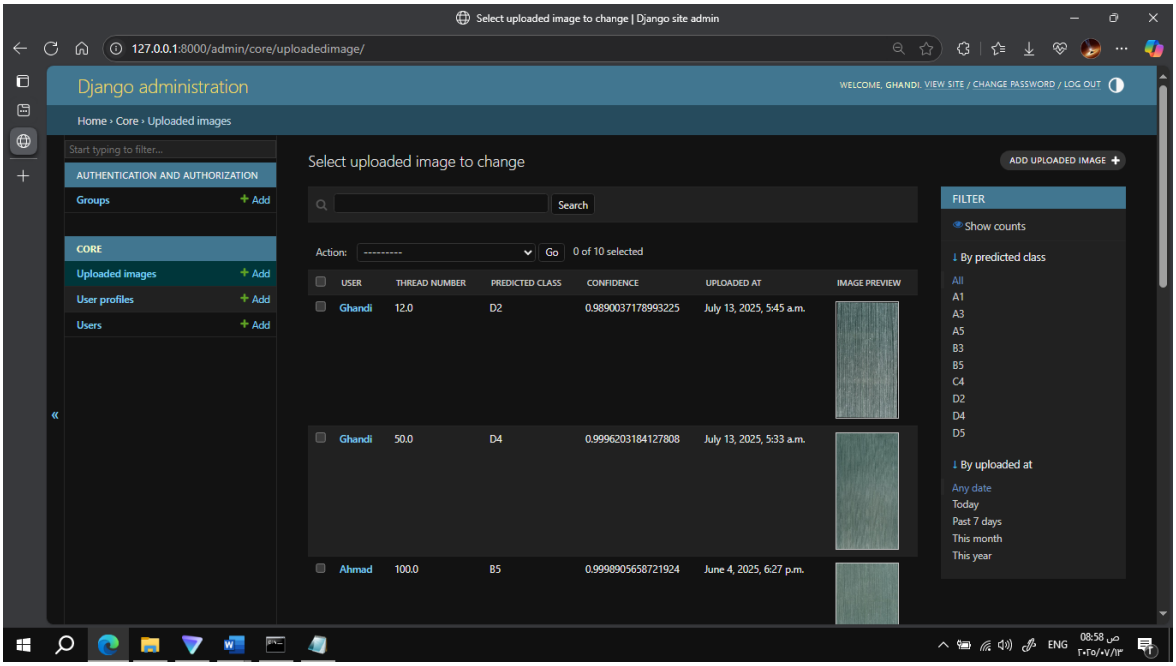
الشكل (5-32): صفحة تسجيل الخروج 49



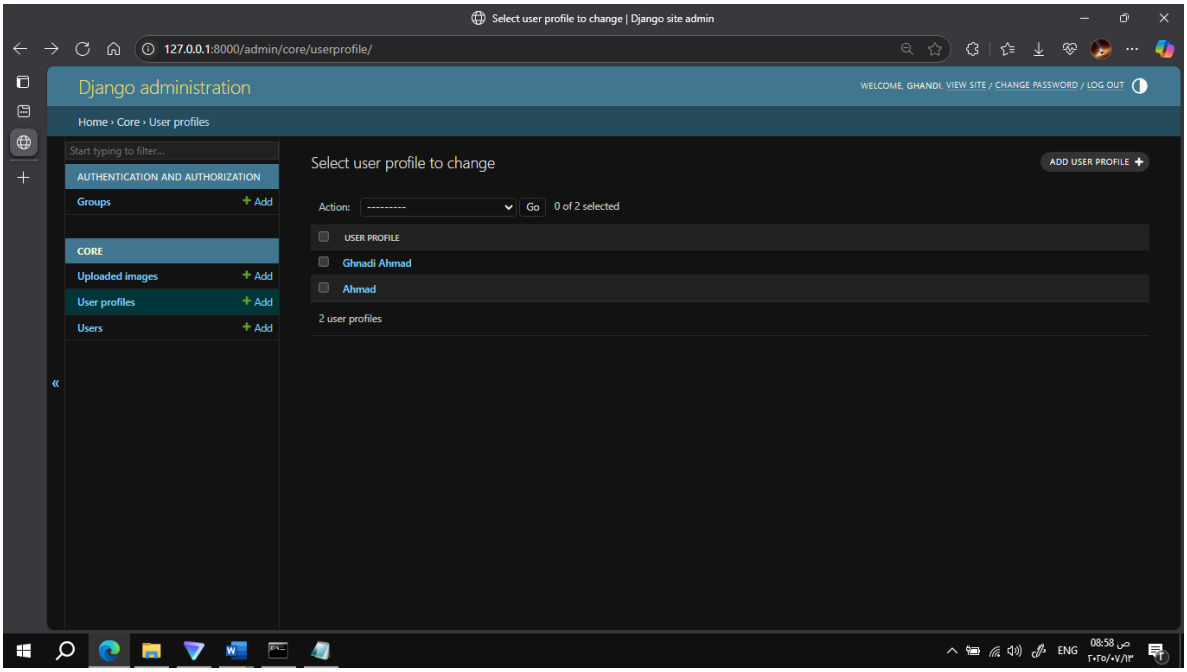
الشكل (5-33): صفحة مشرف الموقع 50



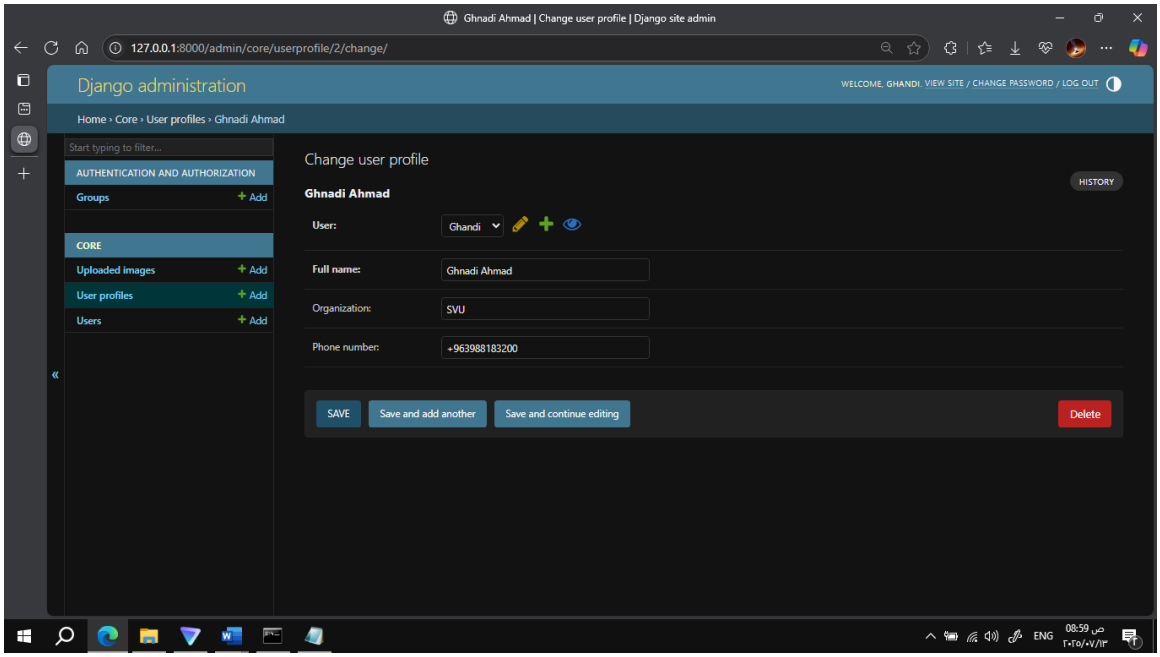
51الشكل (5-34): صفحة إدارة الموقع



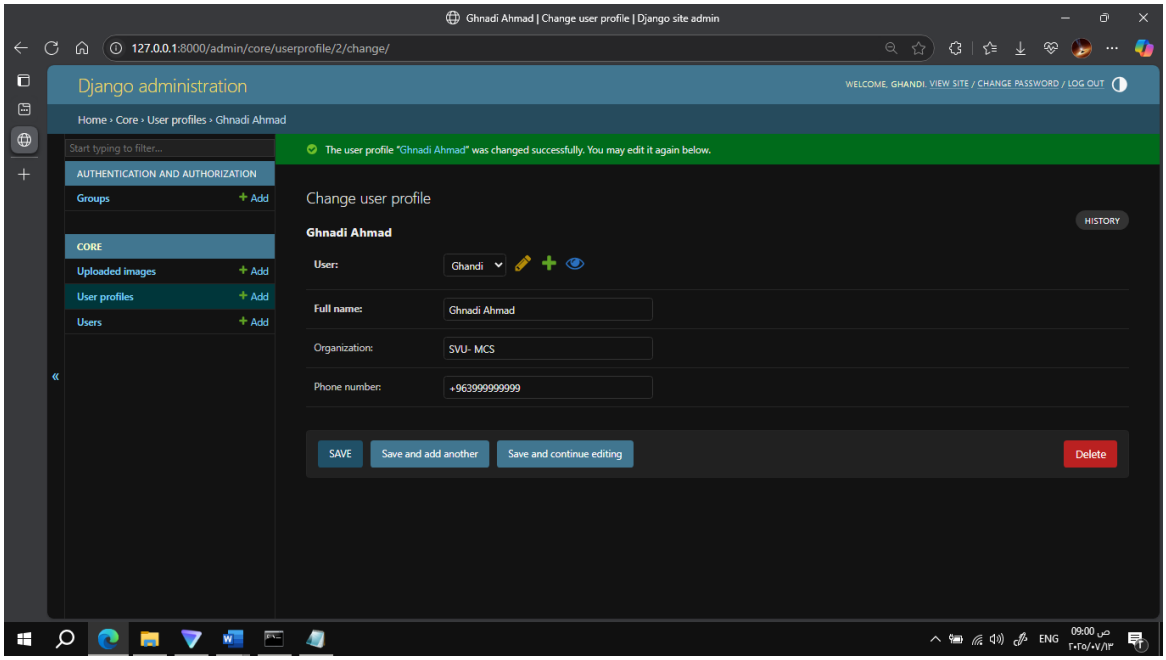
52الشكل (5-35): صفحة تُظهر الصور المُحمّلة



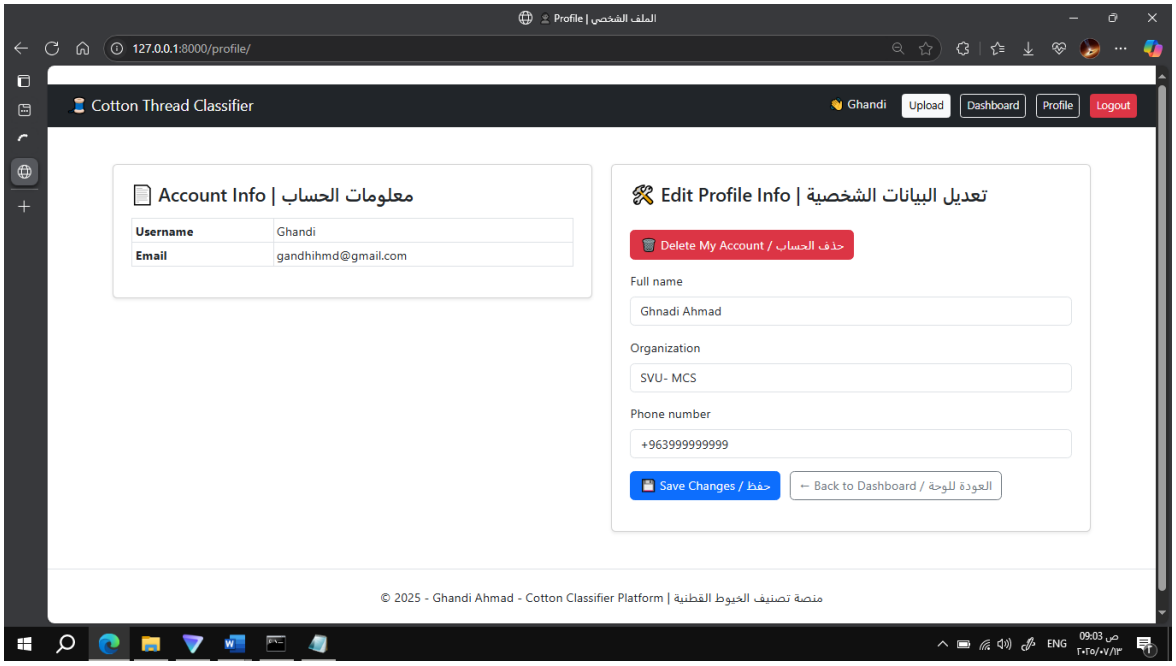
53الشكل (5-36): صفحة تظهر *profile* المستخدمين المسجلين



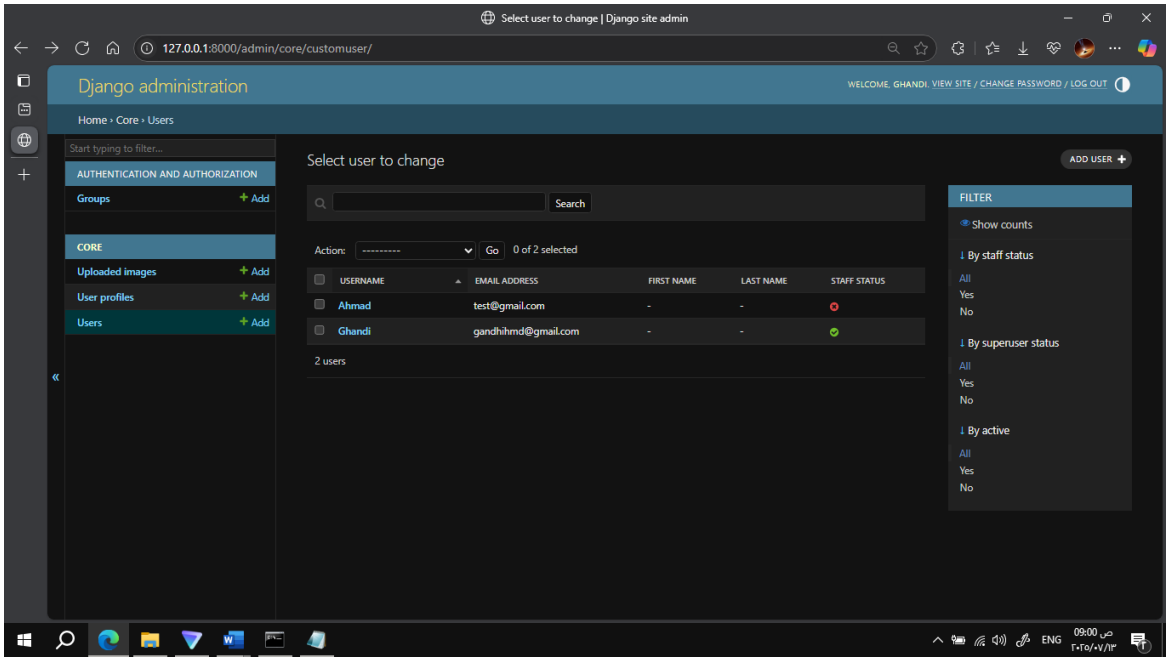
54الشكل (5-37): صفحة لتعديل بيانات المستخدمين



55الشكل (5-38): صفحة تظهر تأكيد عملية تعديل بيانات المستخدم



56الشكل (5-39): صفحة تظهر بيانات المستخدم بعد تعديلها من قبل مشرف الموقع



57الشكل (5–40): صفحة تظهر المستخدمين المسجلين في الموقع

5.7.3. آلية الربط المتزامن بين الواجهة الأمامية والخلفية في منصة Django الذكية

تعتمد منصة تصنيف مظهرية الخيوط القطنية، والمصممة باستخدام إطار عمل Django، على نموذج العميل–الخادم(Client-Server Architecture) لتحقيق التفاعل بين الواجهة الأمامية والواجهة الخلفية. يتم هذا التفاعل بشكل أساسي عبر آلية الطلبات والاستجابات المتزامنة (Synchronous HTTP Requests) التي تُعالج بواسطة بنيةDjango المعمارية القائمة على نمط النموذج–الواجهة–القالب (Model-View-Template - MVT) [80]. تتولى الواجهات (Views) منطق العمل، وتتولى القوالب (Templates) عرض البيانات، وتتولى النماذج (Models) إدارة البيانات الدائمة وقواعد البيانات. في هذا النهج، عند كل تفاعل يتطلب بيانات جديدة من الخادم أو إرسال بيانات إليه (مثل تحميل صورة أو تسجيل الدخول)، يُرسل المتصفح طلباً كاملاً إلى الخادم، ويُعاد تحميل الصفحة أو يُعاد توجيه المستخدم إلى صفحة أخرى بعد تلقي الاستجابة. هذا يختلف عن التفاعلات غير المتزامنة (AJAX) التي تستخدمJavaScript لتحديث أجزاء من الصفحة دون إعادة تحميلها بالكامل[81].

5.7.3.1. *الواجهة الأمامية القوالب* (Django Templates) *ودورها في إرسال الطلبات*

تُصمم الواجهة الأمامية للموقع باستخدام ملفات HTML التي تتضمن لغة قوالب **Django (Django Template Language)**. هذه القوالب ليست مجرد ملفات HTML ثابتة، بل هي ديناميكية، حيث يمكنها استقبال بيانات من الواجهة الخلفية وعرضها، وإنشاء نماذج لإرسال البيانات، وتوجيه المستخدمين إلى صفحات مختلفة.

فيما يلي شرح لملفات الموقع ودورها في عملية الربط بين واجهة الموقع وخلفية الموقع [82]:

a) **قالب base.html:** لبناء الروابط الديناميكية وإظهار حالة المستخدم. هذا القالب هو الأساس الذي تُورث منه جميع القوالب الأخرى. يحتوي على الهيكل العام للصفحة (مثل شريط التنقل العلوي navbarوالتذييل footer)

b) **قالب upload_image.html:** إرسال البيانات والملفات ومعالجة الاستجابة. هذا القالب هو المثال الأبرز على كيفية إرسال البيانات (بما في ذلك الملفات) من الواجهة الأمامية إلى الواجهة الخلفية بشكل متزامن، وكيفية عرض الواجهة الأمامية للاستجابة الواردة من الخادم.

5.7.3.2. *الواجهة الخلفية* (Django – urls.py, views.py, models.py)

تُعد الواجهة الخلفية هي المحرك الذي يستقبل الطلبات من الواجهة الأمامية، ويعالجها باستخدام منطق العمل ونماذج التعلّم العميق، ويتفاعل مع قاعدة البيانات، ثم يُعيد استجابات إلى الواجهة الأمامية. فيما يلي شرح لملفات واجهة الموقع الخلفية [82]:

a) **urls.py:** نظام توجيه الطلبات. هذا الملف (سواء في التطبيق نفسه أو في المشروع الرئيسي) هو الذي يُحدد كيف تُربط أنماط عناوين URL بـ view functions/classes إنه بمثابة خارطة طريق لـDjango لتحديد أي جزء من الكود يجب تنفيذه بناءً على الـ URL المطلوب.

b) **views.py:** يمثل منطق العمل والمعالجة الذكية وهو يقوم بالمعالجة الفعلية للطلبات. كل دالة view تستقبل كائن HttpRequest كمعامل أول. هذا الكائن يحتوي على جميع المعلومات حول الطلب الوارد (البيانات المرسلة، نوع الطلب، معلومات المستخدم، إلخ).

c) **models.py:** طبقة التخزين الدائم للبيانات. يُعرّف هذا الملف نماذج Django (Models)، والتي تُمثل جداول في قاعدة البيانات. هذه النماذج هي الواجهة بين كود Python وقاعدة البيانات الفعلية، مما يسمح بالتعامل مع البيانات ككائنات Python بدلاً من كتابة استعلامات SQL يدوياً.

5.7.3.3. *مسار البيانات خلال عملية التصنيف (دورة الطلب–الاستجابة)*

بناءً على الأجزاء البرمجية المذكورة، فيما يلي المسار الكامل لعملية تصنيف الخيط:

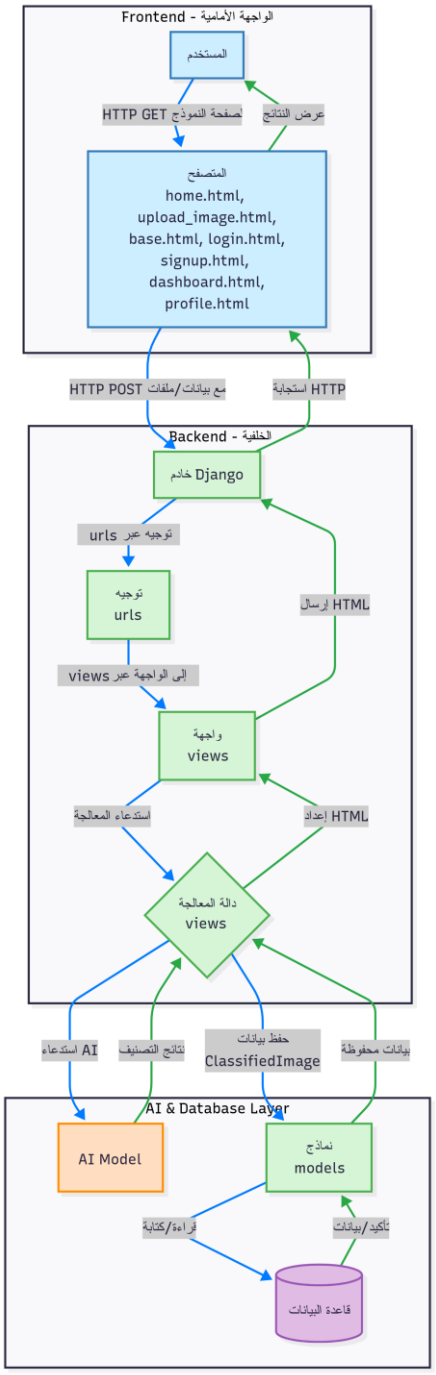
1. يزور المستخدم صفحة الرفع (/upload/):
   - يُرسل المتصفح طلب **GET HTTP** إلى المسار /upload/.
   - يُحل urls.py الطلب ويُوجهه إلى دالة upload_image_view في views.py.
   - تكتشف upload_image_view أن الطلب هو GET ، فتُنشئ نموذج UploadImageForm فارغاً.
   - تُعيد upload_image_view استجابة render() التي تدمج النموذج الفارغ مع قالب upload_image.html.
   - يتلقى المتصفح صفحة HTML ويعرض النموذج الفارغ للمستخدم.
2. المستخدم يُرفع صورة ويُدخل نمرة الخيط ويضغط زر "**تصنيف**":
   - يقوم المتصفح بجمع البيانات من نموذج:

     HTML (<input type="file">, <input type="text">, و csrf_token).
   - يُرسل المتصفح طلب **POST HTTP** إلى نفس المسار /upload/ (بما أن action غير محدد).
   - يُحل urls.py الطلب مرة أخرى ويُوجهه إلى دالة upload_image_view.
   - تكتشف upload_image_view أن الطلب هو POST.
   - تُنشئ الدالة UploadImageForm، وتُمرر لها request.POST و request.FILES.
   - يُنفذ form.is_valid() للتحقق من صحة البيانات.
   - إذا كان النموذج صالحاً:
     - تُستخلص الصورة والبيانات من form.cleaned_data.
     - يُستدعى منطق التعلّم العميق: تُمرر الصورة التي تم رفعها إلى نموذج PyTorch (الذي تم تحميله مسبقاً في الذاكرة عند بدء التطبيق)، ويُجرى التصنيف.
     - تُحفظ نتائج التصنيف (الفئة، الثقة) والصورة والبيانات الأخرى في قاعدة البيانات عبر ClassifiedImage.save().
     - تُعيد upload_image_view استجابة render() التي تدمج قالب upload_image.html مرة أخرى، ولكن هذه المرة تُمرر إليه أيضاً كائن result (الذي يحتوي على تفاصيل التصنيف).
   - إذا كان النموذج غير صالح:
     - تُعيد upload_image_view استجابة render() لنفس القالب، ولكن مع عرض أخطاء النموذج

(إذا وجدت) للمستخدم.

- يتلقى المتصفح صفحة HTML المعاد تحميلها. نظراً لوجود متغير result في السياق، سيتم عرض قسم "Classification Result" في upload_image.html، مما يُظهر للمستخدم نتائج التصنيف.

يبيّن الشكل (5–41) آلية تدفق الطلبات والاستجابات بين واجهة الموقع وخلفية الموقع



58الشكل (5–41): آلية الربط بين واجهة وخلفية الموقع

6. مناقشة النتائج، التوصيات، والآفاق المستقبلية

يُعد هذا الفصل بمثابة تتويج للجهد البحثي المبذول في هذه الأطروحة، حيث يقدم تحليلاً شاملاً وتقييماً نقدياً للنتائج التي تم التوصل إليها من خلال تطبيق تقنيات الذكاء الاصطناعي ومعالجة الصور الرقمية لتصنيف مظهرية الخيوط القطنية. بناءً على هذه النتائج، سيتم تقديم مجموعة من التوصيات العملية والمقترحات البحثية المستقبلية التي تهدف إلى تعزيز وتوسيع نطاق هذا المشروع. كما سيتناول الفصل الآفاق المستقبلية المحتملة التي يمكن أن تفتحها هذه الدراسة في مجال فحص جودة المنسوجات وتطبيقات الذكاء الاصطناعي الصناعية.

6.1. مناقشة النتائج

لقد تضمنت الأطروحة، كما هو مفصل في الفصول السابقة، معالجة مشكلة جوهرية في صناعة الغزل والنسيج تتمثل في التقييم الذاتي وغير المتسق لمظهرية الخيوط القطنية. تمثلت الأهداف الرئيسية للبحث في تطوير نظام آلي وموضوعي لتصنيف مظهرية الخيوط باستخدام الذكاء الاصطناعي، وبناء منصة ويب لتسهيل الوصول إلى هذه التقنية.

من خلال القسم العملي والدراسة التجريبية المفصلة في الفصل الخامس، تم تحقيق الأهداف المحددة بنجاح، وتجلت النتائج في النقاط الرئيسية التالية:

1. **التصنيف الآلي لمظهرية الخيوط باستخدام الشبكات العصبية التلافيفية (CNNs):**
   - تم بنجاح بناء وتدريب نموذج تعلم عميق يعتمد على معماريات الشبكات العصبية التلافيفية (CNNs)، وبالتحديد نموذج ResNet، كما هو موضح في الفصل الرابع. وقد أثبت هذا النموذج قدرته الفائقة على التعرف على الأنماط المعقدة والميزات الدقيقة في صور الخيوط القطنية، مما أتاح تصنيفها بدقة إلى فئات مظهرية مختلفة مثل (A, B, C, D) وفقاً لمعايير الجودة الصناعية.
   - اعتمدت العملية على تقنيات معالجة الصور الرقمية لتحضير الصور، بالإضافة إلى استخدام torchvision.transforms و albumentations زيادة البيانات بشكل فعال، مما عزّز من قوة النموذج وقدرته على التعميم، وقلل من مشكلة الانحدار الزائد (overfitting) [32, 17].

- تجاوز الأداء المحقق باستخدام هذا النهج الموضوعية التي تعاني منها طرق الفحص اليدوية، مما يوفر حلاً أكثر اتساقاً وموثوقية لتقييم الجودة.

2. أداء النموذج وتقييمه باستخدام مصفوفة الارتباك:

- تم تقييم أداء نموذج التصنيف بشكل شامل باستخدام مصفوفة الارتباك (Confusion Matrix) ومقاييس الأداء المستخلصة منها، كما تم تفصيله في الفصل الثالث. سمحت مصفوفة الارتباك بتحليل دقيق لأنواع الأخطاء التي يرتكبها النموذج (الإيجابيات الكاذبة والسلبيات الكاذبة)، مما وفّر رؤى أعمق من مجرد الاعتماد على مقياس الدقة الإجمالية (Accuracy) [25].
- أظهرت النتائج أن النموذج حقق مستويات عالية من الدقة (Precision) والاستدعاء (Recall) ومقياس F1-Score للفئات المختلفة. على سبيل المثال (بناءً على المثال في الفصل الثالث): إذا كانت دقة النموذج 90.9% واستدعاؤه 83.3%، فهذا يعني أن معظم التنبؤات الإيجابية كانت صحيحة، ولكن هناك مجال لتحسين القدرة على اكتشاف جميع الحالات الإيجابية الفعلية. مثل هذا التحليل الدقيق ضروري لتحديد مجالات التحسين المستقبلي.

3. تطوير منصة الويب باستخدام Django:

- تم تصميم وتطوير منصة ويب متكاملة باستخدام إطار عمل Django، مما وفر واجهة سهلة الاستخدام لعملية تصنيف مظهرية الخيوط. تضمنت المنصة وظائف أساسية مثل:

- **إدخال الصورة ونمرة الخيط:** السماح للمستخدمين بتحميل صور الخيوط وإدخال البيانات الوصفية ذات الصلة.
- **استدعاء عملية التصنيف:** دمج نموذج التعلم العميق في الواجهة الخلفية للمنصة لإجراء عملية التصنيف تلقائياً عند طلب المستخدم.
- **إدارة المستخدمين:** توفير نظام متكامل لإنشاء الحسابات، تسجيل الدخول، تعديل البيانات الشخصية، وتسجيل الخروج، مما يضمن أمان وخصوصية المستخدمين.
- **إنشاء تقارير PDF:** تمكين المستخدمين من طباعة تقارير PDF مفصلة عن عمليات التصنيف التي قاموا بها، مما يوفر توثيقاً قيماً لعمليات مراقبة الجودة.

- لقد أثبتت هذه المنصة جدواها كأداة عملية لنشر نموذج الذكاء الاصطناعي وجعله متاحاً للمستخدمين النهائيين دون الحاجة إلى خبرة تقنية عميقة، مما يحقق التكامل بين البحث النظري والتطبيق الصناعي.

4. **التحديات والقيود:** خلال مرحلة التطوير، تم مواجهة بعض التحديات التي أثرت على جوانب معينة من البحث:

- **حجم وتنوع مجموعة البيانات:** على الرغم من استخدام تقنيات زيادة البيانات، قد يكون توفر مجموعة بيانات أكبر وأكثر تنوعاً لجميع فئات مظهرية الخيوط، خاصة الفئات الأقل تمثيلاً، قد يؤدي إلى تحسينات إضافية في أداء النموذج.
- **التعقيد الحسابي:** يتطلب تدريب النماذج العميقة (CNNs) موارد حاسوبية كبيرة (GPUs)، مما قد يشكل قيداً على إمكانية إعادة الإنتاج أو التوسع في بيئات ذات موارد محدودة.

6.2. التوصيات والمقترحات

بناءً على النتائج التي تم التوصل إليها والتحديات التي تمت مواجهتها، تُقدم الدراسة الحالية مجموعة من التوصيات والمقترحات التي يمكن أن توجه الأبحاث المستقبلية وتساهم في تعزيز فعالية المشروع:

1. **توسيع وتنويع مجموعة البيانات:**

- **جمع المزيد من البيانات:** يُوصى بجمع كميات أكبر من صور الخيوط من مصادر متنوعة وظروف إضاءة مختلفة لزيادة قوة النموذج وقدرته على التعميم.
- **التركيز على الفئات النادرة:** إيلاء اهتمام خاص لجمع وتسمية صور الفئات النادرة من عيوب المظهرية، أو استخدام تقنيات توليد البيانات المتقدمة مثل الشبكات التوليدية التنافسية(GANs) لإنشاء أمثلة اصطناعية لهذه الفئات [22].

2. **استكشاف معماريات نماذج متقدمة وتقنيات التعلم المعزز:**

- **تجارب مع معماريات أحدث:** تجربة معماريات CNNs أحدث وأكثر كفاءة مثل EfficientNet [38] لتقييم ما إذا كان يمكن تحقيق أداء أفضل بكفاءة حاسوبية مماثلة أو أقل.

- **تطبيق تعلم المجموعات (Ensemble Learning):** يمكن دمج تنبؤات نماذج متعددة (سواء كانت من نفس النوع أو أنواع مختلفة) باستخدام تقنيات التجميع (Bagging) أو التعزيز (Boosting) أو التكديس (Stacking) لتعزيز دقة وموثوقية التصنيف الكلية وتقليل التباين أو التحيز [27].

6.3. الآفاق المستقبلية للمشروع

تفتح الدراسة الحالية آفاقاً واسعة للبحث والتطوير المستقبلي في مجال فحص جودة المنسوجات وتطبيقات الذكاء الاصطناعي الصناعية:

1. **التصنيف متعدد العيوب وتحديد المواقع (Multi-defect Classification and Localization):**
توسيع نطاق النظام ليشمل أنواع محددة من العيوب (مثل العقد، الألياف السائبة، تكتلات)، وتحديد موقعها الدقيق داخل الصورة باستخدام تقنيات اكتشاف الكائنات (Object Detection) أو التجزئة (Segmentation).

2. **الدمج مع أنظمة التحكم في الإنتاج (Integration with Production Control Systems):**
تطوير المنصة لدمجها بشكل مباشر مع أنظمة إدارة المصانع (MES) أو أنظمة تخطيط موارد المؤسسات (ERP)، مما يتيح مراقبة الجودة في الوقت الفعلي على خطوط الإنتاج واتخاذ إجراءات تصحيحية فورية.

3. **النشر على الحافة (Edge Deployment):**
تحسين النموذج ليصبح خفيف الوزن وقابلاً للنشر على الأجهزة الطرفية (Edge Devices) مباشرة في بيئة الإنتاج. هذا يقلل من زمن الاستجابة، ويقلل من الحاجة إلى نقل البيانات الكبيرة إلى السحابة، ويزيد من استقلالية النظام.

4. **إنشاء قاعدة بيانات معيارية مفتوحة المصدر:**
المساهمة في بناء قاعدة بيانات عامة ومُعلّمة لمظهرية الخيوط القطنية وفقاً لمعايير ASTM D2255–09. ستكون هذه القاعدة مصدراً قيماً للأبحاث المستقبلية وتطوير حلول جديدة في هذا المجال.

5. تطوير واجهة مستخدم متقدمة للتفاعل مع النماذج:

إنشاء واجهات تفاعلية تمكن المستخدمين من استكشاف خرائط الميزات التي تعلمها النموذج (Feature Maps) أو فهم العصبونات التي تتنشط لأنواع معينة من العيوب، مما يوفر رؤى عميقة للمتخصصين في الجودة.

6. التعلم المستمر والتكيف (Continuous Learning and Adaptation):

تطوير آليات تسمح للنموذج بالتعلم المستمر والتكيف مع البيانات الجديدة أو التغيرات في أنواع الخيوط أو العيوب التي قد تظهر بمرور الوقت، مما يضمن أن يظل النظام فعالاً وذا صلة على المدى الطويل.

7. تطوير آليات كشف الشذوذ (Anomaly Detection):

استخدام المرمزات التلقائية (Autoencoders) أو أنواعها المتطورة للكشف عن عيوب أو شذوذات في مظهرية الخيوط لم يتم تدريب النموذج عليها بشكل صريح. يمكن تدريب هذه النماذج على الخيوط السليمة، ثم يتم تمييز أي انحراف كبير عن النمط الطبيعي كعيب [34].

8. تحسين قابلية التفسير (Explainability) للنموذج:

تطبيق تقنيات الذكاء الاصطناعي القابلة للتفسير (XAI) واستخدام أدوات مثل Grad-CAM أو SHAP لفهم كيفية اتخاذ النموذج لقرارات التصنيف، وتحديد الميزات البصرية في صور الخيوط التي تؤثر أكثر في التنبؤ. هذا يعزز ثقة المستخدمين الخبراء في النظام ويوفر رؤى قيمة لتحسين جودة الخيوط.

6.4. الخاتمة

لقد أثبتت هذه الأطروحة الإمكانات التحويلية للذكاء الاصطناعي، وخاصة الشبكات العصبية التلافيفية، في أتمتة وتحسين عملية تصنيف مظهرية الخيوط القطنية. من خلال دمج أحدث تقنيات التعلم العميق مع منصة ويب عملية، تم تقديم حل موضوعي وفعّال يتجاوز القيود المتأصلة في طرق الفحص التقليدية. إن النتائج التي تم التوصل إليها تضع أساساً متيناً لمستقبل مراقبة الجودة في صناعة النسيج، وتفتح الباب أمام العديد من الابتكارات والتطبيقات المستقبلية التي يمكن أن تسهم في تعزيز الكفاءة، الدقة، والإنتاجية في هذا القطاع الحيوي.

7. المـــراجـــع العـــلـــمـــيـــة

[1] ASTM Standards., D2255/D2255M-09. (2013). *Standard Test Method for Grading Spun Yarns for Appearance*. USA.

[2] Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing. USA-New Jersey: Prentice-Hall. P954.

[3] Adibpour, M., Shakeri, M., & Norouzi, M. (2018). Fabric defect classification using convolutional neural networks. *Journal of Textile Science and Technology*, 8(2), 107-118.

[4] Liu, Y., Li, Q., & Li, S. (2020). A review of fabric defect detection based on deep learning. *Textile Research Journal*, 90(13-14), 1461-1476.

[5] Zhang, L., Wang, Y., & Li, X. (2019). Yarn defect classification based on image processing and machine learning. *Journal of Engineered Fibers and Fabrics*, 14, 1-10.

[6] Chen, H., Wang, W., & Li, C. (2021). Artificial intelligence in textile industry: A review. *Journal of Fiber Bioengineering and Informatics*, 14(1), 1-15.

[7] Koschan, A., & Abidi, M. (2008). *Digital Color Image Processing*. USA-New Jersey: John Wiley & Sons. P375.

[8] Jahne, B. (2005). *Digital Image Processing*. Germany: Springer. P639.

[9] Yadav, A., & Yadav, P. (2009). *Digital Image Processing*. India: Laxmi Publications. P244.

[10] Vassiliadis, S. (2011). *Advances in Modern Woven Fabrics Technology*. Croatia: InTech. P240.

[11] Burge, M. J., & Burger, W. (2016). *Digital Image Processing: An Algorithmic Introduction Using Java*. USA: Springer. P811.

[12] Solomon, C., & Breckon, T. (2011). *Fundamentals of Digital Image Processing*. UK: John Wiley & Sons, Ltd. P328.

[13] Jayaraman, S., Esakkirajan, S., & Veerakumar, T. (2011). Digital Image Processing. USA: Tata McGraw-Hill Education. P 723.

[14] Russell, S. J., & Norvig, P. (2020). Artificial intelligence: A modern approach (4th ed.). Pearson.

[15] McCorduck, P. (2004). Machines who think: A personal inquiry into the history and prospects of artificial intelligence. CRC Press.

[16] Dreyfus, H. L. (1992). What computers still can't do: A critique of artificial reason. MIT press.

[17] Nilsson, N. J. (2009). The quest for artificial intelligence: A history of ideas and achievements. Cambridge University Press.

[18] Luger, G. F. (2009). Artificial intelligence: Structures and strategies for complex problem solving. Pearson Education.

[19] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 3(3), 210-229.

[20] Topol, E. J. (2019). Deep medicine: How artificial intelligence can make healthcare human again. Basic Books.
[21] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533-536.
[22] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
[23] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
[24] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
[25] Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113.
[26] Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD international conference on Management of data (pp. 207-216).
[27] Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.
[28] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
[29] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (Vol. 27).
[30] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.
[31] Hochreiter, S., & Schmidhuber, J. J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
[32] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (Vol. 25).
[33] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
[34] Mienye, D. I., & Swart, T. (2025). Deep Autoencoder Neural Networks: A Comprehensive Review and New Perspectives. Archives of Computational Methods in Engineering, 32(3), 1-27.
[35] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.
[36] Saxena, A., Prasad, M., Gupta, A., Bharill, N., Janardhanan, M. N., Satapathy, S. C., & Hota, C. (2017). A review of clustering techniques and developments. In Advances in computing and communications: Proceedings of the second international conference on computing and communications (ACC 2012), Part 1 (pp. 433-442). Springer Berlin Heidelberg.
[37] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT Press.
[39] Yu, Y., Si, X., Hu, C., & Zhang, J. (2017). A review of recurrent neural networks: Architectures, applications, and future directions. International Journal of Automation and Computing, 14(5), 517-531.

[40] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.
[41] Google Cloud documentation on Supervised Learning:

- Google Cloud. (n.d.). "Supervised & Unsupervised Learning." Google Cloud Documentation. (https://cloud.google.com/discover/supervised-vs-unsupervised-learning)
- Google Cloud. (n.d.). " Learning." Google Cloud Documentation. (https://developers.google.com/machine-learning/intro-to-ml/supervised)
- Google Cloud. (n.d.). " glossary." Google Cloud Documentation. (https://developers.google.com/machine-learning/glossary)

[42] IBM documentation on Reinforcement Learning:

- IBM. (n.d.). "Neural Networks." IBM Documentation. (https://www.ibm.com/cloud/learn/neural-networks)
- IBM. (n.d.). "Deep Learning." IBM Documentation. (https://www.ibm.com/cloud/learn/deep-learning)
- IBM. (n.d.). "Ensemble Learning." IBM Documentation. (https://www.ibm.com/cloud/learn/ensemble-learning)
- IBM. (n.d.). "Bagging." IBM Documentation. (https://www.ibm.com/cloud/learn/bagging)

[43] Seldon.io. (n.d.). "Machine Learning Regression." Seldon.io Blog. (https://www.seldon.io/blog/machine-learning-regression)
[44] DataCamp. (n.d.). "Classification Machine Learning." DataCamp Tutorials. (https://www.datacamp.com/blog/classification-machine-learning)
[45] Brownlee, J. (n.d.). "Classification Algorithms." Machine Learning Mastery. (https://machinelearningmastery.com/category/python-for-machine-learning/)
[46] Google Developers. (n.d.). "Clustering Algorithms." Google Developers Documentation. (https://developers.google.com/machine-learning/clustering)
[47] Neptune.ai. (n.d.). "Dimensionality Reduction." Neptune.ai Blog. (https://neptune.ai/blog/dimensionality-reduction)
[48] GeeksforGeeks:

- GeeksforGeeks. (n.d.). "Dimensionality Reduction." GeeksforGeeks. (https://www.geeksforgeeks.org/dimensionality-reduction/)
- GeeksforGeeks. (n.d.). "Perceptron Algorithm." GeeksforGeeks. (https://www.geeksforgeeks.org/perceptron-algorithm-in-machine-learning/)

[49] Web of Proceedings. (n.d.). "Convolutional Neural Networks." Web of Proceedings. (https://www.webofproceedings.org/)
[50] ResearchGate:

- ResearchGate. (n.d.). "Generative Adversarial Networks." ResearchGate. (https://www.researchgate.net/topic/Generative-Adversarial-Networks)
- ResearchGate. (n.d.). "Autoencoders." ResearchGate. (https://www.researchgate.net/topic/Autoencoders)

[51] MDPI. (n.d.). "Generative Adversarial Networks." MDPI Journals. (https://www.mdpi.com/2504-4990/7/3/63)

[52] Taylor & Francis. (n.d.).
(https://taylorandfrancis.com/knowledge/Engineering_and_technology/Artificial_intelligence/Bootstrap_aggregating/)
[53] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 60.
[54] Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and flexible image augmentations. Information, 11(2), 125.
[55] Tran, L., Yin, X., & Liu, X. (2017). Disentangled Representation Learning for 3D Face Shape and Identity. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1721-1730.
[56] Wang, J., & Perez, L. (2017). The effectiveness of data augmentation in image classification using deep learning. In proceedings of the 2017 10th International Conference on Machine Learning and Applications (ICMLA), 909-915. IEEE.
[57] Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149.
[58] Shahar, T & ,.Dekel, T. (2020). Image Generation from a Single Image .In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops
[59] McKinney, W .(2012) .Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython .O'Reilly Media
[60] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G & ,.Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library .Advances in Neural Information Processing Systems
[61] Géron, A .(2019) .Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd ed.). O'Reilly Media.
[62] Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection .In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI).1143-1137,
[63] Olah, C. (2015). Understanding LSTMs .Colah's Blog .Available at : http://colah.github.io/posts/2015-08-Understanding-LSTMs /
[64] Russell, S. J & ,.Norvig, P .(2010) .Artificial Intelligence: A Modern Approach (3rd ed.). Prentice Hall.
[65] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 248-255.

[66] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems (NeurIPS), 27, 3320-3328.

[67] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.

[68] Li, Jing & Gao, Guozhong. (2023). Digital construction of geophysical well logging curves using the LSTM deep-learning network. Frontiers in Earth Science. 10. 1041807. 10.3389/feart.2022.1041807.

[69] Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In International Conference on Machine Learning (ICML), 448-456.

[70] Bridle, J. S. (1990). Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. In F. F. Fogelman Soulie & J. Hérault (Eds.), Neurocomputing: Algorithms, Architectures and Applications (pp. 227-236). Springer.

[71] Tharwat, A; Classification assessment methods. Applied Computing and Informatics 4 January 2021; 17 (1): 168–192. https://doi.org/10.1016/j.aci.2018.08.003

[72] Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning. Springer.

[73] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1-score and accuracy in binary classification tasks. BMC Genomics, 21(1), 6.

[74] Ryan, T. (2022). Introduction to Software Engineering. Amazon Digital Services LLC - Kdp.

[75] Leach, R. J. (2018). Introduction to Software Engineering. United States: CRC Press.

[76] Coronel, C. Morris, S. and Rob, P. (2011), "Database Systems: Design, Implementation, and Management, Ninth Edition", Cengage Learning, Boston, USA.

[77] Ranjan, A. (2021). Building Websites with Django: Build and Deploy Professional Websites with Python Programming and the Django Framework. India: Bpb Publications.

[78] Rubio, D. (2017). Beginning Django: Web Application Development and Deployment with Python. United States: Apress.

[79] Django for Beginners, Fifth Edition: Build Modern Web Applications with Python. (2024). (n.p.): Still River Press.

[80] Django Software Foundation. (n.d.). The Django web framework. Retrieved from https://docs.djangoproject.com/en/stable/

[81] Django Software Foundation. (n.d.). Handling uploaded files. Django documentation. Retrieved from https://docs.djangoproject.com/en/stable/topics/http/file-uploads/

[82] Django Software Foundation. (n.d.). Cross Site Request Forgery protection. Django documentation. Retrieved from https://docs.djangoproject.com/en/stable/ref/csrf/

8. المـــلاحـــق

8.1. الملحق (1): توليد 250 صورة جديدة من صورة معيارية واحدة

```
import cv2
import os
import albumentations as A
import glob

# إعداد المسارات النسبية
BASE_DIR = os.path.dirname(os.path.abspath(__file__))
REF_DIR  = os.path.join(BASE_DIR, "..", "refs", "5refs")
SAVE_DIR = os.path.join(BASE_DIR, "..", "augmented", "5refs", "classic")

os.makedirs(SAVE_DIR, exist_ok=True)

print("Reading from:", REF_DIR)
print("Saving to :", SAVE_DIR)

# خريطة الفئات إلى أسماء المجلدات الفرعية
grade_map = {
   "A5": "A",
   "B5": "B",
   "C5": "C",
   "D5": "D",
}

# تحميل الصور المرجعية والتأكد من وجودها
refs = {}
for grade, subfolder in grade_map.items():
   folder_path = os.path.join(REF_DIR, subfolder)
   candidates = glob.glob(os.path.join(folder_path, "*.jpg")) + \
            glob.glob(os.path.join(folder_path, "*.png"))
   if not candidates:
      raise FileNotFoundError(f"لم أجد أي صورة في: {folder_path}")
   img_path = candidates[0]
   img = cv2.imread(img_path)
   if img is None:
      raise FileNotFoundError(f"لم أتمكن من قراءة الصورة: {img_path}")
   refs[grade] = img

print("Loaded images:", list(refs.keys()))
```

```
    # إعداد Data Augmentation عبر Albumentations
    transform = A.Compose([
        A.HorizontalFlip(p=0.5),
        A.RandomRotate90(p=0.5),
        A.ShiftScaleRotate(shift_limit=0.1, scale_limit=0.2, rotate_limit=15, p=0.7),
        A.RandomBrightnessContrast(p=0.5),
        A.GaussNoise(var_limit=(10.0, 50.0), p=0.3),
    ])

    # إعداد باراميترات الضغط وتصغير الأبعاد
    TARGET_WIDTH    = 512
    JPEG_QUALITY    = 85
    USE_GRAYSCALE   = False

    for grade, img in refs.items():
        out_grade_dir = os.path.join(SAVE_DIR, grade)
        os.makedirs(out_grade_dir, exist_ok=True)

        for i in range(250):
            aug = transform(image=cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
            out = cv2.cvtColor(aug["image"], cv2.COLOR_RGB2BGR)

            h, w = out.shape[:2]
            scale = TARGET_WIDTH / float(w)
            new_h = int(h * scale)
                            out_small      =      cv2.resize(out,      (TARGET_WIDTH,      new_h),
interpolation=cv2.INTER_AREA)

            if USE_GRAYSCALE:
                out_small = cv2.cvtColor(out_small, cv2.COLOR_BGR2GRAY)



            filename = os.path.join(out_grade_dir, f"{grade}_aug_{i:03d}.jpg")
            params = [cv2.IMWRITE_JPEG_QUALITY, JPEG_QUALITY]
            cv2.imwrite(filename, out_small, params)

        print(f'Generated 250 images for grade {grade} → {out_grade_dir}")
```

8.2. الملحق (2): توليد 250 صورة جديدة من صورة معيارية واحدة

```
import os
import cv2
import glob
from albumentations import (
    Compose, HorizontalFlip, VerticalFlip, RandomRotate90,
    ShiftScaleRotate, RandomBrightnessContrast, GaussNoise,
    ElasticTransform, GridDistortion, OpticalDistortion
)

def build_transform():
    return Compose([
        HorizontalFlip(p=0.5),
        VerticalFlip(p=0.3),
        RandomRotate90(p=0.5),
        ShiftScaleRotate(shift_limit=0.1, scale_limit=0.2, rotate_limit=30, p=0.7),
        RandomBrightnessContrast(brightness_limit=0.2, contrast_limit=0.2, p=0.5),
        GaussNoise(var_limit=(5.0, 25.0), p=0.4),
        ElasticTransform(alpha=1, sigma=50, alpha_affine=50, p=0.3),
        GridDistortion(num_steps=5, distort_limit=0.3, p=0.2),
        OpticalDistortion(distort_limit=0.2, shift_limit=0.1, p=0.2),
    ])
def augment_category(grade, ref_dir, out_dir, n_samples=250):


    sub = grade[0]
    folder = os.path.join(ref_dir, sub)


        candidates = glob.glob(os.path.join(folder, "*.jpg")) + glob.glob(os.path.join(folder,
"*.png"))
    if not candidates:
        raise FileNotFoundError(f"لم أجد أي صورة في: {folder}")
    ref_path = candidates[0]

    img = cv2.imread(ref_path)
    if img is None:
        raise FileNotFoundError(f"تعذّر قراءة الصورة: {ref_path}")


    h, w = img.shape[:2]
    scale = 512 / max(h, w)
    if scale < 1:
        img = cv2.resize(img, (int(w*scale), int(h*scale)), interpolation=cv2.INTER_AREA)
    transform = build_transform()
    os.makedirs(out_dir, exist_ok=True)
```

```
    for i in range(n_samples):
        augmented = transform(image=img)['image']
        save_path = os.path.join(out_dir, f"{grade}_aug_{i:03d}.jpg")
        cv2.imwrite(save_path, augmented, [cv2.IMWRITE_JPEG_QUALITY, 85])
    print(f"✅ Augmented {n_samples} images for {grade} → {out_dir}")
def main():
    BASE_DIR = os.path.dirname(os.path.abspath(__file__))
    REF_DIR  = os.path.join(BASE_DIR, "..", "refs", "5refs")
    OUT_BASE = os.path.join(BASE_DIR, "..", "augmented", "5refs", "singan")
    grades = ["A5", "B5", "C5", "D5"]
    for grade in grades:
        out_dir = os.path.join(OUT_BASE, grade)
        augment_category(grade, REF_DIR, out_dir, n_samples=250)
    print("🎉 Done augmenting all categories.")

if __name__ == "__main__":
    main()
```

8.3. الملحق (3): دمج الصور الناتجة عن الكودين السابقين

```
import os
import shutil

base_dir = r"E:\SVU\MPR-Final semester\augmented\5refs"
classic_dir = os.path.join(base_dir, "classic")
singan_dir = os.path.join(base_dir, "singan")
output_dir = os.path.join(base_dir, "per_class5")

categories = ["A5", "B5", "C5", "D5"]

os.makedirs(output_dir, exist_ok=True)

for cat in categories:
    print(f"Processing category: {cat}")

    cat_output_dir = os.path.join(output_dir, cat)
    os.makedirs(cat_output_dir, exist_ok=True)

    src_classic = os.path.join(classic_dir, cat)
    src_singan = os.path.join(singan_dir, cat)

    image_paths = []

    for src_folder in [src_classic, src_singan]:
```

```
        for filename in sorted(os.listdir(src_folder)):
            if filename.lower().endswith(".jpg"):
                full_path = os.path.join(src_folder, filename)
                image_paths.append(full_path)

    for idx, img_path in enumerate(image_paths):
        new_filename = f"{cat}_{idx+1:03}.jpg"
        dest_path = os.path.join(cat_output_dir, new_filename)
        shutil.copy(img_path, dest_path)

    print(f"  -> Done. Total images: {len(image_paths)}")

print("✅ All categories processed and merged into per_class5.")
```

8.4. الملحق (4): تقسيم الصور للقيام بعملية التدريب

```
import os, shutil, random

SRC = r"E:\SVU\MPR-Final semester\augmented\5refs\per_class5"
DST = r"E:\SVU\MPR-Final semester\augmented\5refs\per_class5_split"
RATIO = 0.8
random.seed(42)

for phase in ["train","val"]:
    for cat in ["A5","B5","C5","D5"]:
        os.makedirs(os.path.join(DST,phase,cat), exist_ok=True)

for cat in ["A5","B5","C5","D5"]:
    imgs = [f for f in os.listdir(os.path.join(SRC,cat)) if f.lower().endswith(".jpg")]
    random.shuffle(imgs)
    cut = int(len(imgs)*RATIO)
    for phase,subset in [("train", imgs[:cut]), ("val", imgs[cut:])]:
        for fname in subset:
            src_path = os.path.join(SRC,cat,fname)
            dst_path = os.path.join(DST,phase,cat,fname)
            shutil.copy(src_path, dst_path)

print("Data split done.")
```

8.5. الملحق (5): تدريب نماذج التصنيف الخمسة

```
import os
import json
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms, models
from torch.utils.data import DataLoader
import matplotlib.pyplot as plt

def save_checkpoint(state, filename="checkpoint_classifier5.pth"):
    """حفظ نقطة التوقف التي تحتوي على أوزان النموذج، optimizer وحالة التدريب."""
    torch.save(state, filename)

def save_training_log(train_losses, val_accs, filename="training_log5.json"):
    """حفظ قيم الخسارة والدقة في ملف JSON."""
    log = {
        "train_losses": train_losses,
        "val_accs": val_accs
    }
    with open(filename, "w") as f:
        json.dump(log, f, indent=2)

def load_checkpoint(filename, model, optimizer):
    """تحميل نقطة التوقف لاستئناف التدريب."""
    checkpoint = torch.load(filename)
    model.load_state_dict(checkpoint['model_state_dict'])
    optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
    return (
        checkpoint['epoch'],
        checkpoint.get('best_acc', 0.0),
        checkpoint.get('train_losses', []),
        checkpoint.get('val_accs', [])
    )

def main():
    # إعداد عام
    data_dir       = r"E:\SVU\MPR-Final semester\augmented\5refs\per_class5_split"
    num_classes    = 4
    batch_size     = 32
    num_epochs     = 10
    lr             = 1e-4
    checkpoint_file = "checkpoint_classifier5.pth"
```

```
model_file     = "thread_classifier5_best.pth"
log_file       = "training_log5.json"

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# التحويلات
train_tf = transforms.Compose([
   transforms.Resize(256),
   transforms.RandomResizedCrop(224),
   transforms.RandomHorizontalFlip(),
   transforms.ToTensor(),
   transforms.Normalize([0.485,0.456,0.406],[0.229,0.224,0.225]),
])
val_tf = transforms.Compose([
   transforms.Resize(256),
   transforms.CenterCrop(224),
   transforms.ToTensor(),
   transforms.Normalize([0.485,0.456,0.406],[0.229,0.224,0.225]),
])

# تحميل البيانات
train_ds = datasets.ImageFolder(os.path.join(data_dir, "train"), transform=train_tf)
val_ds   = datasets.ImageFolder(os.path.join(data_dir, "val"),   transform=val_tf)
train_ld = DataLoader(train_ds, batch_size=batch_size, shuffle=True,  num_workers=4)
val_ld   = DataLoader(val_ds,   batch_size=batch_size, shuffle=False, num_workers=4)

# بناء النموذج
model = models.resnet18(weights=models.ResNet18_Weights.IMAGENET1K_V1)
model.fc = nn.Linear(model.fc.in_features, num_classes)
model = model.to(device)

criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=lr)

# إعداد استئناف التدريب
start_epoch  = 1
best_acc     = 0.0
train_losses = []
val_accs     = []

if os.path.exists(checkpoint_file):
   print("🔄 استئناف من آخر نقطة توقف...")
   start_epoch, best_acc, train_losses, val_accs = load_checkpoint(
      checkpoint_file, model, optimizer)
```

```
        start_epoch += 1
    else:
        print("🚀 بدء تدريب جديد...")

    # حلقة التدريب
    for epoch in range(start_epoch, num_epochs + 1):
        model.train()
        running_loss = 0.0
        for imgs, labels in train_ld:
            imgs, labels = imgs.to(device), labels.to(device)
            optimizer.zero_grad()
            outputs = model(imgs)
            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()
            running_loss += loss.item() * imgs.size(0)

        epoch_loss = running_loss / len(train_ds)
        train_losses.append(epoch_loss)

        # التحقق
        model.eval()
        correct = 0
        with torch.no_grad():
            for imgs, labels in val_ld:
                imgs, labels = imgs.to(device), labels.to(device)
                preds = model(imgs).argmax(dim=1)
                correct += (preds == labels).sum().item()
        epoch_acc = correct / len(val_ds)
        val_accs.append(epoch_acc)

                print(f"Epoch {epoch}/{num_epochs} Loss: {epoch_loss:.4f} Val Acc: {epoch_acc:.4f}")

        # حفظ أفضل نموذج
        if epoch_acc > best_acc:
            best_acc = epoch_acc
            torch.save(model.state_dict(), model_file)
            print("✅ حفظ نموذج أفضل!")

        #  حفظ نقطة التوقف
        save_checkpoint({
            'epoch': epoch,
            'best_acc': best_acc,
```

```
            'model_state_dict': model.state_dict(),
            'optimizer_state_dict': optimizer.state_dict(),
            'train_losses': train_losses,
            'val_accs': val_accs,
        }, checkpoint_file)

        save_training_log(train_losses, val_accs, log_file)

    # رسم المخططات
    plt.figure()
    plt.plot(range(1, len(train_losses)+1), train_losses, label="Train Loss")
    plt.xlabel("Epoch"); plt.ylabel("Loss"); plt.title("Training Loss Curve")
    plt.grid(); plt.legend(); plt.savefig("train_loss_curve5.png"); plt.close()

    plt.figure()
    plt.plot(range(1, len(val_accs)+1), val_accs, label="Val Accuracy", color='green')
    plt.xlabel("Epoch"); plt.ylabel("Accuracy"); plt.title("Validation Accuracy Curve")
    plt.grid(); plt.legend(); plt.savefig("val_acc_curve5.png"); plt.close()

    print(f"\n🏁 Training complete. Best Val Acc: {best_acc:.4f}")

if __name__ == "__main__":
    main()
```

8.6. الملحق (6): خوارزمية تقييم نماذج التدريب والمستخدمة في البحث

```
import os
import argparse
import torch
import torch.nn as nn
from torchvision import transforms, models
from PIL import Image

def load_model(weights_path: str, device: torch.device, num_classes: int = 4):
    model = models.resnet18(pretrained=False)
    model.fc = nn.Linear(model.fc.in_features, num_classes)
    state = torch.load(weights_path, map_location=device)
    model.load_state_dict(state)
    model.to(device).eval()
    return model

def predict_image(model, img_path: str, device: torch.device, class_names):
    if not os.path.isfile(img_path):
```

```
        raise FileNotFoundError(f"لم أجد الصورة: {img_path}")

    tf = transforms.Compose([
        transforms.Resize(256),
        transforms.CenterCrop(224),
        transforms.ToTensor(),
        transforms.Normalize([0.485,0.456,0.406],
                             [0.229,0.224,0.225]),
    ])

    img = Image.open(img_path).convert('RGB')
    x = tf(img).unsqueeze(0).to(device)

    with torch.no_grad():
        logits = model(x)
        probs  = torch.softmax(logits, dim=1)[0]
        conf, idx = probs.max(0)

    return idx.item(), conf.item(), probs.cpu().tolist()

def main():
    parser = argparse.ArgumentParser(description="تصنيف صورة جديدة بمصنف السلسلة الأولى")
    parser.add_argument('--image',   required=True,
                        help="مسار الصورة لتصنيفها")
    parser.add_argument('--weights', default="thread_classifier5_best.pth",
                        help="مسار ملف الأوزان المحفوظ")
    args = parser.parse_args()

    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

    class_names = ["A5", "B5", "C5", "D5"]

    model = load_model(args.weights, device, num_classes=len(class_names))

    idx, confidence, all_probs = predict_image(model, args.image, device, class_names)

    print(f"✅ Predicted class: {class_names[idx]}  (Confidence: {confidence*100:.1f}%)\n")
    print("📊 Probabilities:")
    for name, p in zip(class_names, all_probs):
        print(f"  {name}: {p*100:5.1f}%")

if __name__ == "__main__":
    main()
```

8.7. الملحق (7): الخوارزمية المستخدمة في رسم مصفوفة الارتباك لكل نموذج

```
import os
import torch
import torch.nn as nn
from torchvision import models, transforms, datasets
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt
# إعداد المسارات
model_path = r"E:\SVU\MPR-Final semester\classifier\thread_classifier5_best.pth"
data_dir   = r"E:\SVU\MPR-Final semester\augmented\5refs\per_class5_split\val"
output_dir = r"E:\SVU\MPR-Final semester\classifier"
# التحويلات
val_tf = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize([0.485,0.456,0.406],[0.229,0.224,0.225]),
])
#  تحميل البيانات
val_ds = datasets.ImageFolder(data_dir, transform=val_tf)
val_loader = torch.utils.data.DataLoader(val_ds, batch_size=32, shuffle=False)
class_names = val_ds.classes
num_classes = len(class_names)
# تحميل النموذج
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = models.resnet18(weights=None)
model.fc = nn.Linear(model.fc.in_features, num_classes)
model.load_state_dict(torch.load(model_path, map_location=device))
model = model.to(device)
model.eval()
# التنبؤ
all_preds = []
all_labels = []
with torch.no_grad():
    for imgs, labels in val_loader:
        imgs, labels = imgs.to(device), labels.to(device)
        outputs = model(imgs)
        _, preds = torch.max(outputs, 1)
        all_preds.extend(preds.cpu().numpy())
        all_labels.extend(labels.cpu().numpy())
```

```
# مصفوفة الارتباك
cm = confusion_matrix(all_labels, all_preds)

print("\n📄 Classification Report:")
print(classification_report(all_labels, all_preds, target_names=class_names))

plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=class_names, yticklabels=class_names)
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.tight_layout()

confusion_matrix_path = os.path.join(output_dir, "confusion_matrix5.png")
plt.savefig(confusion_matrix_path)
plt.show()

print(f"✅ تم حفظ مصفوفة الارتباك في: {confusion_matrix_path}")
```

Abstract

This thesis addresses a fundamental problem in the spinning and textile industry: the subjective and inconsistent evaluation of cotton yarn appearance according to the ASTM D2255-09 standard. The traditional assessment method, which relies on human expertise and visual comparison, leads to a lack of consistency, errors in quality and price determination, and significant waste of time and resources.

This research aims to develop a fully intelligent and objective system for evaluating yarn appearance by leveraging the analytical capabilities of computer vision and artificial intelligence. To achieve complete independence from human intervention and improve assessment accuracy, five models were designed and trained using Convolutional Neural Networks (CNNs)—a key deep learning technique for image analysis—to perform accurate yarn appearance classification.

Following a comprehensive evaluation of the models, an interactive web platform was developed to enable factories to conduct professional and objective quality assessments. This system is expected to provide substantial savings in resources, time, and cost, in addition to ensuring a continuous improvement in the final product's quality. This research serves as a practical application of modern technologies in evaluating yarn properties and provides a solid foundation for future research in the textile sector.

Keywords: Cotton Yarn Appearance, Digital Image Processing, Machine Learning, Deep Learning, Automated Classification, Web Platform.

Syrian Arab Republic

Ministry of Higher Education & Scientific researches

Syrian Virtual University

Program name: Master in Computer Science

Building a Smart System to Evaluate the Quality of Cotton Yarns

Thesis Submitted in the Requirements for Master Degree in Computer Science

Prepared by

Ghandi Ghazi Ahmad

Supervisor

Dr. Raouf Hamdan

Year/2025