

Faculty of Engineering
Department of Informatics Engineering
Artificial Intelligence and Data Science



Design of an Integrated Automated System for Proactive Anti-Cheating

A senior 2 project report - submitted to complete the requirements for obtaining a bachelor's degree in informatics engineering - Artificial Intelligence and Data Science Engineering

Prepared by

Ghena Omar-Hussain

Supervised by

Dr. Mouhib Al-Noukari

Eng. Aya Al-Aswad

JAN - 2026

SUPERVISION CERTIFICATION

I Certify that the preparation of this project entitled

[Design of an Integrated Automated System for Proactive Anti-Cheating]

Prepared by [**Ghena Omar-Hussain**]

was made under my supervision at Faculty of Informatics Engineering in partial Fulfillment of
the Requirements for the Degree of Bachelor of Artificial Intelligence and Data Science

Academic Rank:

Name:

Signature:

Date:

الملخص

فهرس المحتويات

1 الفصل الأول: المقدمة

2.....	خلفية الموضوع	1.1
2.....	أهمية نزاهة الامتحانات في العملية التعليمية	1.1.1
2.....	التطور التقني في مجال الرؤية الحاسوبية	1.1.2
3.....	الذكاء الاصطناعي في دعم القرار	1.1.3
4.....	المشكلة البحثية	1.2
4.....	تحديات المراقبة التقليدية للامتحانات	1.2.1
5.....	صعوبة التوثيق الموضوعي للسلوكيات	1.2.2
6.....	الإرهاق البشري وضعف التغطية	1.2.3
6.....	أهمية المشروع	1.3
7.....	الأهمية العلمية	1.3.1
7.....	الأهمية العملية	1.3.2
8.....	أهداف المشروع	1.4
8.....	الهدف الرئيسي	1.4.1
9.....	الأهداف الفرعية	1.4.2
11.....	نطاق المشروع وحدوده	1.5
11.....	السلوكيات المشمولة بالكشف	1.5.1
12.....	بيئة العمل المستهدفة	1.5.2
12.....	محددات المشروع	1.5.3
14.....	منهجية العمل	1.6
14.....	مراحل تطوير المشروع	1.6.1
15.....	الأدوات والتقنيات المستخدمة	1.6.2
17.....	هيكل التقرير	1.7

19 الفصل الثاني: الدراسة النظرية

20.....	مقدمة الفصل	2.1
20.....	الرؤية الحاسوبية	2.2

21	مفهوم الرؤية الحاسوبية.....	2.2.1
21	تطبيقات الرؤية الحاسوبية في المراقبة	2.2.2
22	كشف الأشخاص في الصور والفيديو.....	2.3
22	الشبكات العصبية التلافيفية (CNN)	2.3.1
23	خوارزمية YOLO	2.3.2
23	إصدارات YOLO وتطورها	2.3.3
23	YOLOv8 المستخدم في المشروع	2.3.4
24	تتبع الأشخاص عبر إطارات الفيديو.....	2.4
24	مفهوم التتبع المتعدد (MOT)	2.4.1
25	مرشح كالمان (KALMAN FILTER)	2.4.2
25	خوارزمية DEEPSORT	2.4.3
26	ميزة إعادة التعرف (RE-ID)	2.4.4
27	استخراج ميزات الوجه والجسم.....	2.5
27	إطار عمل MEDIAPIPE	2.5.1
28	FACE MESH ونقاط الوجه	2.5.2
28	POSE ESTIMATION ونقاط الجسم	2.5.3
29	حساب اتجاه الرأس (YAW, PITCH)	2.5.4
30	كشف السلوكيات من الإشارات	2.6
30	تحويل الإشارات إلى سلوكيات	2.6.1
31	آلية HYSTERESIS لمنع التذبذب	2.6.2
32	السلوكيات المستهدفة	2.6.3
33	تقييم المخاطر واتخاذ القرار	2.7
33	حساب درجة الخطورة (RISK SCORE)	2.7.1
34	التمهيد باستخدام EWMA	2.7.2
34	نظام التنبيهات	2.7.3
35	الدراسات السابقة	2.8
36	أنظمة كشف الغش المبنية على تحليل الفيديو.....	2.8.1
36	تطبيقات التعلم العميق في مراقبة السلوك	2.8.2
37	أنظمة المراقبة عن بُعد والامتحانات الإلكترونية.....	2.8.3
37	تحليل مقارن للمنهجيات	2.8.4
38	الفجوات والتحديات القائمة	2.8.5
38	موقع المشروع الحالي	2.8.6

39 الفصل الثالث: بيئة العمل والأدوات

40 مقدمة الفصل	3.1
41 البيئة البرمجية	3.2
41 لغة PYTHON وإصدارها	3.2.1
41 البيئة الافتراضية (VIRTUAL ENVIRONMENT)	3.2.2
42 إدارة الحزم والمكتبات	3.2.3
43 المكتبات والأطر المستخدمة	3.3
43 PYTORCH للتعلم العميق	3.3.1
44 OPENCV لمعالجة الفيديو	3.3.2
44 ULTRALYTICS لـ YOLOV8	3.3.3
45 MEDIAPIPE لاستخراج الميزات	3.3.4
46 DASH لواجهة المستخدم	3.3.5
46 متطلبات العتاد	3.4
47 المعالج والذاكرة	3.4.1
47 كرت الشاشة (GPU)	3.4.2
48 الكاميرات وجودة الفيديو	3.4.3
49 هيكل المشروع	3.5
49 تنظيم المجلدات	3.5.1
50 الوحدات الرئيسية	3.5.2
51 ملفات الإعدادات (YAML)	3.5.3
52 تجهيز البيانات	3.6
53 الفيديوهات المستخدمة	3.6.1
53 صيغ الملفات (JSONL)	3.6.2

56 الفصل الرابع: الدراسة التنفيذية

57 مقدمة الفصل	4.1
58 مرحلة الكشف عن الأشخاص	4.2
58 تحميل نموذج YOLOV8	4.2.1
58 معالجة إطارات الفيديو	4.2.2
59 استخراج صناديق الإحاطة	4.2.3
59 تصفية النتائج	4.2.4

60	مرحلة التتبع	4.3
60	تهيئة DEEPSORT	4.3.1
61	ربط الاكتشافات بالمسارات	4.3.2
61	الحفاظ على هوية كل شخص	4.3.3
61	التعامل مع الاختفاء المؤقت	4.3.4
62	مرحلة استخراج الميزات	4.4
62	تحديد منطقة الاهتمام (ROI)	4.4.1
63	معالجة الوجه بـ MEDIAPIPE	4.4.2
63	معالجة الجسم بـ MEDIAPIPE	4.4.3
63	حساب إشارات السلوك	4.4.4
64	مرحلة كشف السلوكيات	4.5
64	تطبيق آلية HYSTERESIS	4.5.1
65	كشف النظر للأسفل (LOOK_DOWN)	4.5.2
65	كشف الالتفاتات (HEAD_TURN)	4.5.3
65	كشف اليد للوجه (HAND_TO_FACE)	4.5.4
66	تسجيل أحداث البداية والنهاية	4.5.5
66	مرحلة حساب المخاطر	4.6
66	المعادلة المستخدمة	4.6.1
67	أوزان السلوكيات	4.6.2
67	التنعيم الزمني	4.6.3
67	تصنيف مستويات الخطورة	4.6.4
68	مرحلة التنبيهات والقرارات	4.7
68	شروط إطلاق التنبيه	4.7.1
68	تسجيل التنبيهات	4.7.2
69	ملخص القرارات	4.7.3
69	مرحلة المخرجات	4.8
70	توليد الفيديو التوضيحي	4.8.1
70	استخراج مقاطع الأدلة	4.8.2
71	توليد تقرير PDF	4.8.3
71	الإحصائيات والرسوم البيانية	4.8.4
72	واجهة المستخدم (DASHBOARD)	4.9
72	صفحة عرض التشغيلات	4.9.1

72 صفحة التفاصيل	4.9.2
73 عرض الفيديو والإحصائيات	4.9.3

74 الفصل الخامس: الاختبار والنتائج

75 مقدمة الفصل	5.1
75 بيانات الاختبار	5.2
76 وصف الفيديوهاستخدمة	5.2.1
76 مدة الفيديو وعدد الإطارات	5.2.2
76 عدد الأشخاص في المشهد	5.2.3
76 اختبار وحدة الكشف	5.3
77 دقة كشف الأشخاص	5.3.1
77 التعامل مع الحجب الجزئي	5.3.2
77 ملاحظات الأداء	5.3.3
77 اختبار وحدة التتبع	5.4
77 استقرار هوية الأشخاص	5.4.1
78 عدد تبديلات الهوية (ID SWITCHES)	5.4.2
78 التعامل مع الاختفاء والظهور	5.4.3
78 اختبار كشف السلوكيات	5.5
78 دقة كشف النظر للأسفل	5.5.1
79 دقة كشف الالتفات	5.5.2
79 دقة كشف اليد للوجه	5.5.3
79 الإنذارات الكاذبة	5.5.4
80 اختبار الأداء	5.6
80 سرعة المعالجة (FPS)	5.6.1
80 استهلاك الذاكرة	5.6.2
80 زمن المعالجة الكلي	5.6.3
81 نتائج التشغيل الكامل	5.7
81 ملخص السلوكيات المكتشفة	5.7.1
81 التنبيهات الصادرة	5.7.2
82 عينات من المخرجات	5.7.3
82 تحليل النتائج	5.8

82	نقاط القوة.....	5.8.1
83	نقاط الضعف	5.8.2
83	العوامل المؤثرة على الدقة.....	5.8.3

84 الفصل السادس: الخاتمة والتوصيات

85	ملخص المشروع	6.1
85	ملخص الإنجازات.....	6.2
85	الأهداف المحققة	6.2.1
86	المساهمة العلمية	6.2.2
87	التحديات والحلول	6.3
87	التحديات التقنية	6.3.1
87	الحلول المُطبَّقة	6.3.2
88	العمل المستقبلي	6.4
88	تحسينات ممكنة	6.4.1
88	ميزات إضافية مقترحة	6.4.2
88	توسيع نطاق المشروع	6.4.3
89	التوصيات	6.5

92 REFERENCES

الفصل الأول: المقدمة

1.1 خلفية الموضوع

تُشكل الامتحانات ركيزة أساسية في المنظومة التعليمية، إذ تُمثل الوسيلة الرئيسية لتقييم تحصيل الطلاب وقياس مدى إتقانهم للمعارف والمهارات المطلوبة. ومع تزايد أعداد الطلاب في المؤسسات التعليمية وتنوع أساليب الغش وتطورها، برزت الحاجة إلى أنظمة مراقبة أكثر كفاءة ودقة وموضوعية. يستعرض هذا القسم السياق العام للمشروع من خلال ثلاثة محاور: أهمية نزاهة الامتحانات، والتطور في مجال الرؤية الحاسوبية، ودور الذكاء الاصطناعي في دعم القرار.

1.1.1 أهمية نزاهة الامتحانات في العملية التعليمية

تُعَدُّ نزاهة الامتحانات من الركائز الجوهرية التي يقوم عليها النظام التعليمي برمته. فالامتحانات ليست مجرد أداة لقياس المعرفة، بل هي آلية لضمان العدالة بين الطلاب، ووسيلة لمنح الشهادات والمؤهلات التي تعكس الكفاءة الحقيقية لحاملها. عندما تنتهك هذه النزاهة من خلال الغش، تتأثر منظومة كاملة من القيم والمعايير التي يُفترض أن تُميّز الأكفاء عن غيرهم. من منظور الطالب الملتزم، يُمثل الغش ظلماً صريحاً يُهدر جهده ويُساوي بينه وبين من لم يبذل الجهد ذاته. هذا الشعور بالظلم قد يؤدي إلى إحباط واسع وفقدان الثقة في عدالة النظام التعليمي. على المستوى المؤسسي، تتضرر سمعة المؤسسة التعليمية عندما تنتشر ظاهرة الغش فيها، مما يقلل من قيمة الشهادات الصادرة عنها في سوق العمل والمجتمع الأكاديمي. على المستوى المجتمعي الأوسع، تترتب آثار خطيرة على تخريج كوادر غير مؤهلة حصلت على شهاداتها بطرق غير مشروعة. ففي المجالات الحساسة كالطب والهندسة والقانون، قد يُعرض عدم الكفاءة حياة الناس وسلامتهم للخطر. لذلك فإن حماية نزاهة الامتحانات ليست مسألة أكاديمية بحتة، بل هي مسألة ذات أبعاد اجتماعية واقتصادية وأخلاقية عميقة.

1.1.2 التطور التقني في مجال الرؤية الحاسوبية

شهد مجال الرؤية الحاسوبية (Computer Vision) تطوراً هائلاً خلال العقد الأخير، مدفوعاً بثلاثة عوامل رئيسية: التقدم في خوارزميات التعلم العميق، وتوفر كميات ضخمة من البيانات للتدريب، والتطور في قدرات العتاد الحاسوبي وخاصة وحدات معالجة الرسومات (GPU) [5] [1].

الرؤية الحاسوبية هي فرع من علوم الحاسوب يهدف إلى تمكين الآلات من "رؤية" وفهم المحتوى البصري بطريقة مشابهة للإنسان. تشمل مهامها الأساسية: تصنيف الصور، وكشف الأشياء وتحديد مواقعها، والتعرف على الوجوه، وتقدير وضعية الجسم، وتتبع الأشياء المتحركة، وغيرها من المهام التي كانت حتى وقت قريب حكراً على الإدراك البشري.

أحدثت الشبكات العصبية التلافيفية (Convolutional Neural Networks – CNN) ثورة في هذا المجال منذ عام 2012، عندما حققت شبكة AlexNet نتائج غير مسبوقة في مسابقة ImageNet. منذ ذلك الحين، تطورت البنى المعمارية للشبكات بشكل متسارع: من VGGNet إلى ResNet إلى EfficientNet، مع تحسينات مستمرة في الدقة والكفاءة الحسابية.

في مجال كشف الأشياء (Object Detection)، برزت خوارزمية YOLO (You Only Look Once) كنقطة نوعية في سرعة المعالجة [6]. بخلاف الخوارزميات السابقة التي كانت تُعالج الصورة على مراحل متعددة، تُعالج YOLO الصورة بالكامل في مرور واحد، مما يُتيح الكشف في الوقت الحقيقي بمعدلات تصل إلى عشرات الإطارات في الثانية.

تطور أيضاً مجال تتبع الأشياء المتعددة (Multiple Object Tracking – MOT) بشكل ملحوظ. خوارزميات مثل DeepSORT تجمع بين التنبؤ الحركي باستخدام مرشح كالمان وبين الميزات المظهرية العميقة، مما يُتيح تتبعاً مستقراً للأشخاص عبر تسلسلات فيديو طويلة حتى مع وجود حجب مؤقت أو تقاطع بين المسارات [2][7].

أصبح بالإمكان أيضاً استخراج معلومات تفصيلية عن وضعية الجسم البشري من خلال أدوات مثل MediaPipe من Google، التي تُوفر نقاطاً معلمية دقيقة للوجه والجسم واليدين في الوقت الحقيقي [3][8]. هذه القدرات فتحت آفاقاً جديدة لتطبيقات تحليل السلوك البشري.

1.1.3 الذكاء الاصطناعي في دعم القرار

يُستخدم مصطلح "نظام دعم القرار" (Decision Support System) للإشارة إلى الأنظمة الحاسوبية التي تُساعد صانعي القرار البشريين من خلال توفير معلومات وتحليلات وتوصيات، دون أن تحل محلهم في اتخاذ القرار النهائي. هذا التمييز جوهري في السياقات التي تتطلب حكماً بشرياً، حيث يبقى الإنسان مسؤولاً عن القرار ويستخدم النظام كأداة مساعدة.

في مجال المراقبة والأمن، أثبتت أنظمة دعم القرار المبنية على الذكاء الاصطناعي فعاليتها في عدة تطبيقات: مراقبة حركة المرور، والكشف عن السلوكيات المشبوهة في الأماكن العامة، ومراقبة المنشآت الحيوية. في كل هذه التطبيقات، يقوم النظام بتحليل كميات ضخمة من البيانات البصرية واستخلاص الأنماط المهمة وتنبيه المشغلين البشريين عند الحاجة.

يُتيح هذا النهج الاستفادة من نقاط قوة كل من الآلة والإنسان. الآلة تتفوق في: معالجة كميات كبيرة من البيانات بسرعة، والحفاظ على الانتباه المستمر دون إرهاق، وتطبيق معايير موحدة على جميع الحالات، والتوثيق الدقيق للأحداث. في المقابل، يتفوق الإنسان في: فهم السياق الكامل للموقف، والتمييز بين الحالات المتشابهة ظاهرياً، واتخاذ قرارات أخلاقية، والتعامل مع المواقف غير المتوقعة.

في سياق مراقبة الامتحانات تحديداً، يُمكن لنظام دعم القرار أن يُساعد المراقب البشري من خلال: الكشف التلقائي عن السلوكيات التي قد تُشير إلى محاولة غش، وتقييم درجة الخطورة لكل سلوك، وتوجيه انتباه المراقب نحو الحالات الأكثر إثارة

للكشك، وتوثيق الأحداث بطريقة منهجية تصلح كأدلة. يبقى المراقب البشري هو صاحب القرار النهائي في تحديد ما إذا كان السلوك يُشكّل غشاً فعلياً يستدعي تدخلاً.

هذا النهج يُعالج أيضاً المخاوف الأخلاقية المرتبطة بالأتمتة الكاملة لعمليات الحكم على سلوك الأفراد. فالنظام لا يتهم أحداً بالغش، بل يُشير إلى سلوكيات تستحق المراجعة البشرية. القرار النهائي يبقى للإنسان الذي يستطيع مراعاة الظروف والسياق الذي قد لا يدركه النظام الآلي بالكامل.

1.2 المشكلة البحثية

على الرغم من الجهود المبذولة لضمان نزاهة الامتحانات، تواجه المؤسسات التعليمية تحديات جوهرية في مراقبة قاعات الامتحان بالطرق التقليدية. تتمحور هذه التحديات حول ثلاثة محاور أساسية: محدودية قدرات المراقبة البشرية، وصعوبة التوثيق الموضوعي للسلوكيات المشبوهة، والإرهاق الذي يُصيب المراقبين ويُضعف تغطيتهم. يستعرض هذا القسم هذه التحديات بالتفصيل لتأطير المشكلة التي يسعى المشروع لمعالجتها.

1.2.1 تحديات المراقبة التقليدية للامتحانات

تعتمد المراقبة التقليدية للامتحانات على وجود مراقبين بشريين يتجولون في قاعة الامتحان أو يجلسون في مواقع تُتيح لهم رؤية الطلاب. هذا النهج، رغم أنه لا يزال الأساس في معظم المؤسسات التعليمية، يُعاني من قيود بنيوية متعددة تُضعف فعاليته.

القيد الأول هو محدودية مجال الرؤية البشرية. المراقب الواحد لا يستطيع مراقبة جميع الطلاب في آن واحد، خاصة في القاعات الكبيرة التي قد تضم عشرات أو مئات الطلاب. حتى مع تعدد المراقبين، تبقى هناك مناطق وزوايا قد لا تحظى بالتغطية الكافية. هذه الثغرات المعروفة قد يستغلها بعض الطلاب الراغبين في الغش.

القيد الثاني هو صعوبة الكشف عن أساليب الغش الحديثة. مع تطور التقنية، ظهرت أجهزة صغيرة يصعب اكتشافها بالعين المجردة: سماعات أذن لاسلكية بحجم حبة البازلاء، وكاميرات مُخفاة في أزرار الملابس أو النظارات، وساعات ذكية تعرض المعلومات. الكشف عن هذه الأجهزة يتطلب تفتيشاً دقيقاً قد يُعتبر انتهاكاً للخصوصية، كما أن المراقب قد لا ينتبه لاستخدامها خلال الامتحان.

القيد الثالث هو التباين في معايير الحكم بين المراقبين المختلفين. ما يُعتبره مراقب سلوكاً مشبوهاً قد لا يلفت انتباه مراقب آخر. هذا التباين يُنتج حالة من عدم الاتساق في تطبيق قواعد المراقبة، مما يُثير تساؤلات حول العدالة. قد يُعاقب طالب على سلوك معين بينما يُفلت طالب آخر من المساءلة رغم ارتكابه السلوك ذاته في قاعة أخرى مع مراقب مختلف.

القيد الرابع هو محدودية الذاكرة البشرية. خلال امتحان يستمر ساعتين أو ثلاثاً، قد يلاحظ المراقب سلوكيات عديدة لطلاب مختلفين، لكنه قد لا يتذكر التفاصيل الدقيقة عند الحاجة لتوثيقها لاحقاً. هذه المشكلة تتفاقم عندما يُطلب من المراقب الإدلاء بشهادته في لجان تحقيق تُعقد بعد أيام من الامتحان.

1.2.2 صعوبة التوثيق الموضوعي للسلوكيات

يُشكل التوثيق الموضوعي للسلوكيات المشبوهة تحدياً كبيراً في منظومة مراقبة الامتحانات. عندما يشتبه المراقب بحدوث غش، يواجه صعوبة في تقديم أدلة قاطعة تُثبت ما رآه، خاصة إذا لم يكن هناك تسجيل مرئي للحدث.

التوثيق الكتابي وحده غير كافٍ في كثير من الأحيان. وصف المراقب لما رآه يبقى شهادة شخصية قابلة للطعن، خاصة إذا أنكر الطالب المتهم ما نُسب إليه. في غياب أدلة موضوعية، قد تجد لجان التحقيق نفسها أمام كلمة ضد كلمة، مما يُصعب إصدار حكم عادل.

حتى عند توفر كاميرات مراقبة في قاعات الامتحان، يبقى التحدي قائماً في استخراج الأدلة منها. مراجعة ساعات من التسجيلات للبحث عن لحظات محددة من سلوك مشبوه تتطلب وقتاً وجهداً كبيرين. غالباً ما تكون زوايا الكاميرات غير مثالية، فلا تُظهر تفاصيل دقيقة كاتجاه نظر الطالب أو ما يخفيه في يده.

يُضاف إلى ذلك أن مفهوم "السلوك المشبوه" ذاته يحمل درجة من الذاتية. النظر إلى الجانب قد يكون لتأمل السؤال أو لمحاولة النظر في ورقة الجار. الميل إلى الأمام قد يكون للتركيز أو لقراءة ورقة مخفية. هذا التداخل بين السلوكيات الطبيعية والمشبوهة يجعل التوثيق والإثبات أكثر تعقيداً.

تترتب على صعوبة التوثيق عدة نتائج سلبية. قد يتردد المراقبون في الإبلاغ عن حالات مشتبهاً بها خوفاً من عدم القدرة على إثباتها. قد يُفلت الغشاشون من العقوبة بسبب عدم كفاية الأدلة. وفي بعض الحالات، قد يُتهم أبرياء بناءً على سوء فهم لسلوكيات طبيعية، دون وجود دليل موضوعي يُثبت براءتهم.

1.2.3 الإرهاق البشري وضعف التغطية

يُمثل الإرهاق البشري تحدياً جوهرياً في فعالية المراقبة التقليدية لامتحانات. الانتباه البشري مورد محدود، ومع مرور الوقت ينخفض مستوى اليقظة بشكل طبيعي. هذه الظاهرة موثقة علمياً في دراسات علم النفس الإدراكي، وتُعرف بانخفاض الأداء عبر الزمن (Vigilance Decrement).

المراقب الذي يُراقب قاعة امتحان لساعات متواصلة يمر بمراحل متعاقبة من الانتباه. في البداية، يكون مستوى التركيز عالياً والقدرة على ملاحظة التفاصيل جيدة. مع مرور الوقت، يبدأ الانتباه بالتشتت والتركيز بالتراجع. بعد ساعة أو ساعتين، قد يفوت المراقب سلوكيات كان سيلاحظها في بداية الفترة.

يتفاقم الإرهاق في الفترات المكثفة من الامتحانات، كفترة الامتحانات النهائية حيث قد يُكلف المراقب بعدة فترات في يوم واحد أو أيام متتالية. التراكم المتواصل للإرهاق يُضعف الأداء بشكل ملحوظ، وقد يؤدي إلى تسرب حالات غش لم تُلاحظ.

تبرز أيضاً مشكلة التغطية غير المتكافئة. بعض أجزاء القاعة تحظى بانتباه أكثر من غيرها بحسب موقع المراقب وطريقة تجوله. الصفوف الأمامية القريبة من المراقب تكون تحت مراقبة أشد من الصفوف الخلفية أو الزوايا البعيدة. هذا التفاوت في التغطية يخلق فرصاً غير متكافئة للغش.

علاوة على ذلك، يواجه المراقبون أحياناً ضغوطاً تُشتت انتباههم عن مهمة المراقبة: الإجابة على استفسارات الطلاب، والتعامل مع مشكلات تقنية في أوراق الأسئلة، وإدارة الدخول والخروج من القاعة. كل هذه المهام الإضافية تستهلك من طاقة المراقب وتُقلل من قدرته على التركيز على المهمة الأساسية.

في ضوء هذه التحديات المتعددة، يتضح أن المراقبة التقليدية وحدها غير كافية لضمان نزاهة الامتحانات بالمستوى المطلوب. هناك حاجة إلى أدوات تقنية تُعزز قدرات المراقب البشري وتُعوّض عن قيوده الطبيعية، دون أن تحل محله في الحكم على السلوكيات واتخاذ القرارات. هذه الفجوة هي ما يسعى المشروع الحالي لسدها من خلال تطوير نظام ذكي لدعم قرار المراقبة.

1.3 أهمية المشروع

ينبثق المشروع الحالي من الفجوة بين تحديات المراقبة التقليدية لامتحانات والإمكانيات التي تُتيحها التقنيات الحديثة في الرؤية الحاسوبية والذكاء الاصطناعي. تتجلى أهمية المشروع في بُعدين متكاملين: الأهمية العلمية التي تتعلق بالمساهمة في المعرفة والبحث الأكاديمي، والأهمية العملية التي تتعلق بالفوائد المباشرة للمؤسسات التعليمية والمستفيدين.

1.3.1 الأهمية العلمية

يساهم المشروع الحالي في عدة جوانب من المعرفة العلمية في مجال تطبيقات الذكاء الاصطناعي والرؤية الحاسوبية. أولاً، يُقدّم نموذجاً تطبيقياً لبناء نظام دعم قرار متكامل يجمع بين عدة تقنيات متقدمة في خط معالجة موحد. هذا التكامل بين الكشف (YOLOv8) والتتبع (DeepSORT) واستخراج الميزات (MediaPipe) وتحليل السلوك (Hysteresis) وتقييم المخاطر (EWMA) يُوفّر نموذجاً قابلاً للتعميم على تطبيقات مشابهة.

ثانياً، يُعالج المشروع مسألة تحويل الإشارات المستمرة إلى أحداث سلوكية منفصلة، وهي مسألة ذات أهمية في مجالات عديدة من تحليل السلوك. استخدام آلية Hysteresis مع عتبات مُعايرة يُوفّر منهجية قابلة للتكرار لتحقيق التوازن بين الحساسية والاستقرار في كشف السلوكيات.

ثالثاً، يتناول المشروع التحدي الأخلاقي للأتمتة في سياقات الحكم على سلوك الأفراد. اختيار نهج "دعم القرار" بدلاً من "اتخاذ القرار" يُقدّم نموذجاً للتعامل مع هذا التحدي، حيث تُستخدم التقنية لتعزيز القدرات البشرية لا لاستبدالها. هذا النهج له دلالات مهمة في النقاش الأكاديمي حول أخلاقيات الذكاء الاصطناعي [13].

رابعاً، يساهم المشروع في توثيق تجربة عملية لتطبيق تقنيات الرؤية الحاسوبية في البيئة التعليمية العربية. معظم الدراسات السابقة في هذا المجال أُجريت في سياقات غربية، والتوثيق العربي للتجارب المحلية يُثري الأدبيات العلمية ويُوفّر مرجعاً للباحثين في المنطقة.

خامساً، يُوفّر المشروع أساساً للدراسات المستقبلية في مجالات ذات صلة: تحسين دقة كشف السلوكيات، ودراسة العوامل المؤثرة في الأداء، ومقارنة الخوارزميات المختلفة، واستكشاف سلوكيات إضافية يُمكن كشفها. المنهجية المعيارية والتوثيق المفصل للمشروع يُسهّلان البناء عليه في أبحاث لاحقة.

1.3.2 الأهمية العملية

على الصعيد العملي، يُقدّم المشروع قيمة مباشرة للمؤسسات التعليمية من خلال عدة جوانب. الجانب الأول هو تعزيز كفاءة المراقبة. بدلاً من أن يحاول المراقب مراقبة جميع الطلاب بشكل متساوٍ، يُمكنه الاعتماد على النظام لتوجيه انتباهه نحو الحالات الأكثر إثارة للشك. هذا التوزيع الذكي للانتباه يُحسن من فعالية المراقبة دون زيادة في الموارد البشرية.

الجانب الثاني هو التوثيق الموضوعي للأحداث. النظام يُسجّل كل ما يكتشفه بطوابع زمنية دقيقة ودرجات كمية ومقاطع فيديو موثقة. هذا التوثيق يُوفّر أدلة موضوعية يُمكن الرجوع إليها في لجان التحقيق، ويُقلل من الاعتماد على الذاكرة البشرية والشهادات الشفهية.

الجانب الثالث هو توفير الوقت في مراجعة التسجيلات. بدلاً من مشاهدة ساعات من تسجيلات الفيديو للبحث عن لحظات مشبوهة، يُمكن للمراجع التركيز مباشرة على المقاطع التي حدّدها النظام كعالية الخطورة. هذا التوفير في الوقت يُتيح مراجعة عدد أكبر من الحالات بالموارد المتاحة ذاتها.

الجانب الرابع هو المعيارية في التعامل مع جميع الطلاب. النظام يُطبّق المعايير ذاتها على جميع الطلاب في جميع القاعات، مما يُقلل من التباين الناتج عن اختلاف المراقبين. هذه المعيارية تُعزز الشعور بالعدالة لدى الطلاب وتُقلل من احتمالية الطعن في قرارات المراقبة.

الجانب الخامس هو توفير تقارير وإحصائيات شاملة. النظام يُنتج تقارير PDF احترافية تتضمن ملخصات ورسوماً بيانية وتفاصيل الأحداث، يُمكن استخدامها للتوثيق الإداري أو لتحليل المؤسسي لأنماط السلوكيات عبر فترات زمنية متعددة.

الجانب السادس هو سهولة الاستخدام والتكامل. النظام مبني بتقنيات مفتوحة المصدر ومُوثّق بشكل جيد، مما يُسهّل تشغيله وصيانته وتخصيصه حسب احتياجات المؤسسة. لوحة التحكم التفاعلية تُوفّر واجهة سهلة للمراقبين والإداريين دون الحاجة لخبرة تقنية متقدمة.

من المهم التأكيد أن الأهمية العملية للمشروع تنبع من كونه أداة مساعدة لا بديلاً عن المراقبة البشرية. القيمة الحقيقية تتحقق عندما يُستخدم النظام بشكل متكامل مع العملية البشرية، حيث يُوفّر النظام المعلومات والتحليلات ويُوفّر الإنسان الحكم والقرار. هذا التكامل يُنتج منظومة أكثر فعالية وعدالة من أي من المكونين منفرداً.

1.4 أهداف المشروع

يسعى المشروع الحالي إلى تحقيق مجموعة من الأهداف المحددة والقابلة للقياس، تنطلق من فهم عميق للمشكلة البحثية وتستجيب للحاجات الفعلية في مجال مراقبة الامتحانات. تُصاغ هذه الأهداف في مستويين: هدف رئيسي يُحدد الغاية العامة للمشروع، وأهداف فرعية تُفصّل المهام المطلوب إنجازها لتحقيق الهدف الرئيسي.

1.4.1 الهدف الرئيسي

يتمثّل الهدف الرئيسي للمشروع في تصميم وتنفيذ نظام ذكي لدعم قرار مراقبة الامتحانات، يقوم على تحليل فيديوهات قاعات الامتحان باستخدام تقنيات الرؤية الحاسوبية والتعلم العميق، بهدف الكشف التلقائي عن السلوكيات التي قد تُشير إلى محاولات غش، وتقديم هذه المعلومات للمراقب البشري بشكل منظم ومُوثّق ليتخذ القرار المناسب.

ينطوي هذا الهدف على عدة عناصر جوهرية تُحدد هوية المشروع وتوجهه. العنصر الأول هو أن النظام "نظام دعم قرار" وليس "نظام اتخاذ قرار"، بمعنى أنه يُؤَقَر المعلومات والتحليلات لكنه لا يُصدر حكماً نهائياً بوجود غش أو عدمه. هذا التمييز جوهري من الناحية الأخلاقية والقانونية، ويضمن بقاء المسؤولية في يد الإنسان.

العنصر الثاني هو الاعتماد على تحليل الفيديو وليس مصادر أخرى. النظام يعمل على فيديوهات مُسجَّلة أو مباشرة من كاميرات المراقبة، دون الحاجة لأجهزة استشعار إضافية أو تعديلات في بيئة قاعة الامتحان. هذا يجعل النظام قابلاً للتطبيق في المؤسسات التي تمتلك بنية تحتية أساسية للمراقبة بالفيديو.

العنصر الثالث هو استخدام تقنيات متقدمة ومُثبتة الفعالية. النظام يعتمد على خوارزميات رائدة في مجالاتها: YOLOv8 للكشف عن الأشخاص، وDeepSORT للتتبع، وMediaPipe لاستخراج الميزات. هذا الاختيار يضمن مستوى عالياً من الأداء والموثوقية.

العنصر الرابع هو التركيز على الكشف عن سلوكيات محددة وليس الغش بشكل مباشر. النظام يكشف سلوكيات مثل النظر للأسفل والالتفات ووضع اليد على الوجه، والتي قد تُشير إلى محاولة غش لكنها ليست دليلاً قاطعاً بحد ذاتها. هذا النهج أكثر موضوعية وأقل عرضة للخطأ من محاولة الحكم المباشر على نوايا الأفراد.

1.4.2 الأهداف الفرعية

لتحقيق الهدف الرئيسي، يُحدد المشروع الأهداف الفرعية التالية، كل منها يُمثّل مهمة قابلة للتنفيذ والتقييم:

الهدف الأول: بناء وحدة كشف وتتبع الأشخاص

تصميم وتنفيذ وحدة قادرة على كشف جميع الأشخاص في كل إطار من إطارات الفيديو باستخدام نموذج YOLOv8، وتتبعهم عبر الإطارات المتتالية باستخدام خوارزمية DeepSORT، مع الحفاظ على هوية فريدة ومستقرة لكل شخص طوال مدة الفيديو. يجب أن تتعامل الوحدة مع التحديات الشائعة كالحجب الجزئي والتقاطع بين مسارات الأشخاص.

الهدف الثاني: بناء وحدة استخراج الميزات

تصميم وتنفيذ وحدة تستخدم إطار عمل MediaPipe لاستخراج نقاط معلمية للوجه والجسم من كل شخص مُتتَبَع في كل إطار، ثم حساب إشارات ذات دلالة سلوكية منها: درجة النظر للأسفل (من وضعية الرأس والجسم)، ودرجة الالتفات الجانبي (من زاوية Yaw للرأس)، ودرجة اليد نحو الوجه (من المسافة بين اليد والوجه).

الهدف الثالث: بناء وحدة كشف السلوكيات

تصميم وتنفيذ وحدة تُحوّل الإشارات المستمرة إلى أحداث سلوكية منفصلة (بداية ونهاية) باستخدام آلية Hysteresis ذات العتبتين. يجب أن تُحقق الوحدة توازناً بين الحساسية للسلوكيات الفعلية والمناعة ضد التذبذبات العابرة والضوضاء في الإشارات.

الهدف الرابع: بناء نظام تقييم المخاطر

تصميم وتنفيذ نظام يحسب درجة خطورة لكل شخص في كل لحظة، بناءً على مجموع موزون لدرجات السلوكيات المختلفة. يجب أن يتضمن النظام تمهيداً زمنياً باستخدام EWMA لضمان استقرار الدرجات، وتصنيفاً للمخاطر إلى مستويات (آمن، مشبوه، عالي الخطورة، مؤكد).

الهدف الخامس: بناء نظام التنبيهات

تصميم وتنفيذ نظام يُصدر تنبيهات عند تجاوز درجة الخطورة لعتبات محددة، مع آليات لمنع التنبيهات المتكررة (Cooldown) وضمان استمرار الحالة قبل التنبيه (Hold Time). يجب أن تتضمن التنبيهات معلومات كافية لفهم سياقها: هوية الشخص، والسلوكيات المُسببة، والطابع الزمني، ودرجة الخطورة.

الهدف السادس: توليد المخرجات المتعددة

تصميم وتنفيذ أدوات لتوليد مخرجات متنوعة تخدم أغراضاً مختلفة: فيديو توضيحي مُشرح يُظهر حالة كل شخص، ومقاطع فيديو قصيرة للأحداث عالية الخطورة، وتقرير PDF احترافي يتضمن إحصائيات ورسوماً بيانية، وملفات بيانات بصيغة JSONL للتحليل الإضافي.

الهدف السابع: بناء لوحة تحكم تفاعلية

تصميم وتنفيذ واجهة ويب تفاعلية تُتيح للمستخدمين استعراض نتائج التشغيل السابقة، ومشاهدة الفيديوهات التوضيحية، والاطلاع على الإحصائيات والتقارير، دون الحاجة لتعامل مباشر مع الملفات أو سطر الأوامر.

الهدف الثامن: التوثيق الشامل

إعداد توثيق تفصيلي للمشروع يشمل: الوصف التقني للبنية والخوارزميات، وإرشادات التشغيل، وشرح ملفات الإعدادات وكيفية تخصيصها، وتوثيق المخرجات وكيفية تفسيرها. هذا التوثيق ضروري لتمكين الآخرين من استخدام النظام والبناء عليه.

1.5 نطاق المشروع وحدوده

يُحدد هذا القسم بدقة ما يشمل المشروع وما يقع خارج نطاقه. هذا التحديد ضروري لوضع توقعات واقعية، ولتوجيه جهود التطوير نحو الأهداف الأساسية، ولتمكين التقييم الموضوعي للنتائج. يتناول القسم ثلاثة محاور: السلوكيات المشمولة بالكشف، وبيئة العمل المستهدفة، ومحددات المشروع وقيوده.

1.5.1 السلوكيات المشمولة بالكشف

يستهدف النظام الحالي كشف ثلاث فئات رئيسية من السلوكيات التي تُعدّ مؤشرات شائعة لمحاولات الغش في بيئة الامتحانات. اختيرت هذه السلوكيات بناءً على معيارين: إمكانية كشفها بشكل موثوق من كاميرات المراقبة النموذجية، وارتباطها الموثق بمحاولات الغش في الدراسات السابقة.

السلوك الأول: النظر للأسفل (Look Down)

يُشير هذا السلوك إلى توجيه الرأس والنظر نحو الأسفل بشكل ملحوظ، باتجاه منطقة الحوض أو أسفل الطاولة. في سياق الامتحان، قد يُشير هذا السلوك إلى قراءة ورقة غش مخفية في الحوض أو تحت الطاولة، أو استخدام هاتف محمول مخفي، أو قراءة معلومات مكتوبة على الفخذ أو اليد. يُحسب هذا السلوك من زاوية إمالة الرأس (Pitch) ومن العلاقة بين موقع الرأس والكتفين. تُستخدم عتبات Hysteresis: 0.48 للتفعيل و 0.42 للإيقاف.

السلوك الثاني: الالتفات الجانبي (Head Turn)

يُشير هذا السلوك إلى توجيه الرأس يميناً أو يساراً بزاوية ملحوظة عن الوضع الطبيعي الموجه نحو ورقة الامتحان. في سياق الامتحان، قد يُشير هذا السلوك إلى محاولة النظر في ورقة زميل مجاور، أو التواصل البصري مع شخص آخر في القاعة، أو مراقبة موقع المراقب قبل القيام بفعل مشبوه. يُقسم هذا السلوك إلى نوعين فرعيين (الالتفات لليمين والالتفات لليسار) لتمكين تحليل أكثر تفصيلاً. يُحسب من زاوية الالتفات الأفقي (Yaw). تُستخدم عتبات Hysteresis: 0.42 للتفعيل و 0.32 للإيقاف.

السلوك الثالث: وضع اليد على الوجه (Hand to Face)

يُشير هذا السلوك إلى تقريب اليد من منطقة الوجه، خاصة الفم والأذن. في سياق الامتحان، قد يُشير هذا السلوك إلى الهمس لزميل قريب (تغطية الفم أثناء الكلام)، أو استخدام سماعة أذن صغيرة للتواصل مع طرف خارجي، أو قراءة معلومات مكتوبة على راحة اليد. يُحسب من المسافة بين معصم اليد والأنف. تُستخدم عتبات Hysteresis: 0.45 للتفعيل و 0.35 للإيقاف، مع فترة ثبات إضافية (300 مللي ثانية) لتجنب التسجيل الخاطئ للحركات الطبيعية العابرة كحكّ الوجه.

سلوكيات خارج النطاق الحالي:

لا يشمل النظام الحالي كشف بعض السلوكيات التي قد تكون ذات صلة بالغش لكنها تتطلب قدرات تقنية إضافية أو بيانات غير متوفرة: الكشف عن الأجهزة الإلكترونية الصغيرة (يتطلب دقة عالية جداً)، وتحليل محتوى ما يقرأه الطالب، والتعرف على الكلام أو الهمس، والكشف عن تبادل الأوراق بين الطلاب.

1.5.2 بيئة العمل المستهدفة

صُمم النظام للعمل في بيئة محددة تتوفر فيها شروط معينة. فهم هذه البيئة المستهدفة مهم لتقييم مدى ملاءمة النظام لسياق استخدام معين.

خصائص الفيديو المدخل:

يعمل النظام على فيديوهات بدقة p720 على الأقل، ويُفضّل p1080 أو أعلى لضمان دقة كافية في استخراج الميزات. معدل الإطارات المتوقع هو 25-30 إطاراً في الثانية، وهو المعدل الشائع في كاميرات المراقبة. يُفترض أن تكون الكاميرا ثابتة وليست متحركة، وأن تكون زاوية التصوير من الأمام أو من الأعلى قليلاً (النمذجة لكاميرات CCTV).

خصائص بيئة القاعة:

يُفترض وجود إضاءة كافية ومستقرة نسبياً في القاعة. الإضاءة السيئة أو المتقلبة تؤثر سلباً على دقة الكشف. يُفترض أن يكون الطلاب جالسين ومواجهين للأمام بشكل عام، وأن يكون جزء كافٍ من أجسامهم مرئياً (على الأقل الرأس والكتفين). يتعامل النظام مع قاعات تضم عدداً معقولاً من الطلاب المرئيين في الإطار (حتى 25 شخصاً تقريباً في الإطار الواحد حسب الإعدادات الافتراضية).

متطلبات العتاد:

يتطلب النظام حاسوباً بمواصفات معقولة لتشغيل نماذج التعلم العميق. وجود وحدة معالجة رسومات (GPU) يُحسن الأداء بشكل كبير ويُتيح المعالجة في الوقت الحقيقي. بدون GPU، يمكن للنظام العمل لكن بسرعة أقل (مناسب للمعالجة اللاحقة وليس المباشرة). الذاكرة المطلوبة تتراوح بين 8-16 جيجابايت حسب حجم الفيديو وعدد الأشخاص.

1.5.3 محددات المشروع

يُقرّ المشروع بوجود محددات وقيود ينبغي فهمها لاستخدام النظام بشكل صحيح وتقييم نتائجه بموضوعية.

المحددات التقنية:

أولاً، دقة الكشف ليست مطلقة. النظام قد يُنتج إنذارات كاذبة (سلوكيات طبيعية تُصنّف كمشبوكة) أو يفوّت سلوكيات فعلية (سلوكيات مشبوكة لا تُكتشف). معدلات هذه الأخطاء تعتمد على جودة الفيديو وظروف التصوير وضبط العتبات. ثانياً، الحجب يُؤثر على الأداء. عندما يحجب شخص أو عائق رؤية طالب آخر بشكل كامل، يفقد النظام تتبعه مؤقتاً. الحجب الجزئي للوجه يُؤثر على دقة استخراج ميزات الوجه. ثالثاً، الأداء يتأثر بعدد الأشخاص في المشهد. كلما زاد عدد الأشخاص المرئيين، زادت متطلبات المعالجة وقد تنخفض سرعة النظام. رابعاً، النظام لا يفهم السياق الكامل. قد ينظر طالب للأسفل لقراءة سؤال في ورقة الامتحان (سلوك طبيعي) بنفس الطريقة التي ينظر بها لقراءة ورقة غش (سلوك مشبوكة). النظام يُبلّغ عن السلوك لكنه لا يُميّز بين السياقين.

المحددات المنهجية:

أولاً، النظام يكشف سلوكيات وليس غشاً. السلوكيات المكتشفة هي مؤشرات تستحق المراجعة وليست أدلة قاطعة على الغش. كثير من السلوكيات المشبوكة قد يكون لها تفسيرات بريئة تماماً. ثانياً، النظام لا يتخذ قرارات. هو أداة مساعدة للمراقب البشري وليس بديلاً عنه. القرار النهائي بشأن وجود غش واتخاذ إجراء يبقى مسؤولية الإنسان. ثالثاً، العتبات والأوزان تحتاج ضبطاً. القيم الافتراضية المستخدمة في النظام قد لا تكون مثالية لكل بيئة. قد يتطلب الاستخدام الفعلي تعديل هذه المعاملات بناءً على خصائص البيئة المحلية.

ما لا يفعله النظام:

لا يُصدر النظام حكماً نهائياً بالغش. لا يتخذ قرارات تأديبية أو إدارية. لا يُخزّن بيانات بيومترية دائمة للطلاب (ما لم تُفعّل ميزة التعرف على الهوية الاختيارية). لا يعمل بشكل مستقل دون إشراف بشري. لا يضمن كشف جميع حالات الغش أو عدم وجود إنذارات كاذبة. هذه المحددات ليست عيوباً بقدر ما هي تحديد واعي لنطاق عمل النظام ودوره كأداة مساعدة ضمن منظومة أوسع لمراقبة الامتحانات.

1.6 منهجية العمل

يتبع المشروع منهجية تطوير منظمة تُقسّم العمل إلى مراحل متتابعة ومتراصة، مع الاعتماد على أدوات وتقنيات مُختارة بعناية لتحقيق الأهداف المحددة. يستعرض هذا القسم مراحل تطوير المشروع والأدوات والتقنيات المستخدمة فيه.

1.6.1 مراحل تطوير المشروع

مرّ تطوير المشروع بست مراحل رئيسية، كل منها بنت على مخرجات المرحلة السابقة وأسست للمرحلة التالية.

المرحلة الأولى: الدراسة والتحليل

بدأ المشروع بمرحلة دراسة مكثفة تضمنت عدة أنشطة. أولاً، مراجعة الأدبيات العلمية في مجال كشف السلوكيات باستخدام الرؤية الحاسوبية، وأنظمة مراقبة الامتحانات، وتطبيقات التعلم العميق في تحليل الفيديو. ثانياً، دراسة التقنيات المتاحة وتقييم مدى ملاءمتها للمشروع: مقارنة خوارزميات الكشف المختلفة (YOLO, Faster R-CNN, SSD)، ومقارنة خوارزميات التتبع (DeepSORT, ByteTrack, SORT)، ومقارنة أدوات استخراج الميزات (MediaPipe, OpenPose, DensePose). ثالثاً، تحليل متطلبات النظام الوظيفية وغير الوظيفية بناءً على فهم المشكلة البحثية واحتياجات المستخدمين المحتملين.

المرحلة الثانية: التصميم المعماري

بناءً على نتائج الدراسة والتحليل، صُممت البنية المعمارية للنظام. اتُخذ قرار باعتماد بنية خط أنابيب (Pipeline) معيارية، حيث تُعالج البيانات عبر سلسلة من المراحل المستقلة نسبياً. هذا التصميم يُوفّر مرونة عالية: يُمكن تحسين أو استبدال أي مرحلة دون التأثير على بقية النظام، كما يُسهّل الاختبار والتصحيح لأن كل مرحلة تُنتج مخرجات قابلة للفحص. حُدّدت المراحل الأساسية للخط: الكشف عن الأشخاص، ثم التتبع عبر الإطارات، ثم استخراج الميزات، ثم كشف السلوكيات، ثم تقييم المخاطر، ثم توليد التنبيهات، وأخيراً إنتاج المخرجات. صُمم أيضاً هيكل الملفات والمجلدات، وحُدّدت صيغ البيانات المتبادلة بين المراحل (JSONL للبيانات التسلسلية، JSON للملخصات، YAML للإعدادات).

المرحلة الثالثة: التنفيذ الأساسي

نُفذت الوحدات الأساسية للنظام بلغة Python مع الاستفادة من المكتبات المتخصصة. بدأ التنفيذ بوحدة الكشف باستخدام YOLOv8 من مكتبة Ultralytics، ثم وحدة التتبع باستخدام DeepSORT، ثم وحدة استخراج الميزات باستخدام MediaPipe. كل وحدة طُوّرت واختُبرت بشكل مستقل قبل دمجها في خط المعالجة الموحد.

اتُبع نهج التطوير التكراري: تُنفَّذ نسخة أولية بسيطة، ثم تُختبر على بيانات حقيقية، ثم تُحسَّن بناءً على النتائج. هذا النهج أتاح اكتشاف المشكلات مبكراً ومعالجتها قبل أن تتراكم.

المرحلة الرابعة: تطوير نظام القرارات

بعد استقرار وحدات الإدراك (الكشف والتتبع والميزات)، طُوِّر نظام القرارات الذي يُحوِّل الإشارات إلى معلومات قابلة للتنفيذ. شمل ذلك: تنفيذ آلية Hysteresis لكشف السلوكيات، وتنفيذ حاسب المخاطر (RiskScorer) بأوزان قابلة للضبط، وتنفيذ التمهيد الزمني باستخدام EWMA، وتنفيذ مدير التنبيهات (AlertManager) مع آليات Cooldown و Hold Time.

المرحلة الخامسة: تطوير المخرجات والواجهات

طُوِّرت أدوات توليد المخرجات المتنوعة: أداة استخراج مقاطع الأحداث عالية الخطورة، وأداة توليد تقرير PDF مع الرسوم البيانية. طُوِّرت أيضاً لوحة التحكم التفاعلية باستخدام إطار عمل Dash، لتوفير واجهة سهلة الاستخدام لاستعراض النتائج.

المرحلة السادسة: الاختبار والتوثيق

أُجريت اختبارات شاملة على النظام باستخدام فيديوهات متنوعة للتحقق من صحة عمل جميع المكونات وتكاملها. وُثِّق المشروع بشكل تفصيلي: ملفات README، ووثيقة المشروع الشاملة، وتعليقات في الشيفرة المصدرية، وأمثلة على الاستخدام.

1.6.2 الأدوات والتقنيات المستخدمة

يعتمد المشروع على مجموعة متكاملة من الأدوات والتقنيات، اختيرت بناءً على معايير الأداء والموثوقية والتوافق وتوفر التوثيق.

لغة البرمجة:

اختيرت لغة Python (الإصدار 3.10) كلغة التطوير الرئيسية لعدة أسباب: توفر مكتبات غنية للتعلم العميق ومعالجة الصور والفيديو، وسهولة النمذجة السريعة والتجريب، والمجتمع الكبير والدعم المتوفر، والتوافق مع جميع الأدوات المختارة.

مكتبات التعلم العميق:

تُستخدم مكتبة PyTorch كإطار التعلم العميق الأساسي، وتعمل عليه نماذج YOLOv8 و DeepSORT. اختيرت PyTorch لمرونتها وأدائها وتوافقها مع GPU.

أدوات الرؤية الحاسوبية:

تُستخدم مكتبة Ultralytics للوصول إلى نموذج YOLOv8 بواجهة برمجية سهلة وأداء مُحسَّن [1]. تُستخدم مكتبة OpenCV لمعالجة الفيديو والصور: قراءة الفيديو، وتحويل الصيغ، ورسم التعليقات، وحفظ النتائج [5]. تُستخدم مكتبة MediaPipe من Google لاستخراج النقاط المعلمية للوجه والجسم [3].

أدوات التتبع:

تُستخدم تنفيذ DeepSORT المتوفرة كمكتبة Python، مع تعديلات طفيفة لتتكامل مع بقية النظام [2]. تتضمن المكتبة نموذج استخراج الميزات المظهرية المُدرَّب مسبقاً.

أدوات واجهة المستخدم:

تُستخدم مكتبة Dash (المبنية على Flask و React) لإنشاء لوحة التحكم التفاعلية [4]. تُستخدم مكتبة Plotly لإنشاء الرسوم البيانية التفاعلية. تُستخدم مكتبة dash-bootstrap-components للتصميم المتجاوب.

أدوات توليد التقارير:

تُستخدم مكتبة ReportLab لإنشاء ملفات PDF. تُستخدم مكتبة Matplotlib لإنشاء الرسوم البيانية الثابتة المُضمَّنة في التقارير.

إدارة الإعدادات:

تُستخدم صيغة YAML لملفات الإعدادات لسهولة قراءتها وتعديلها. تُستخدم مكتبة OmegaConf لتحميل الإعدادات ودمجها مع القيم الافتراضية.

إدارة البيانات:

تُستخدم صيغة JSONL (JSON Lines) لملفات البيانات التسلسلية، حيث كل سطر يُمثَّل سجلاً مستقلاً. هذه الصيغة تُتيح القراءة التدفقية والإضافة السهلة. تُستخدم صيغة JSON للملخصات والبيانات الوصفية.

هذه المجموعة من الأدوات والتقنيات تُشكّل بنية تحتية متينة للمشروع، تجمع بين الأداء العالي والمرونة وسهولة الصيانة والتطوير المستقبلي.

1.7 هيكل التقرير

يُنظَّم هذا التقرير في ستة فصول متكاملة، يُغطي كل منها جانباً محدداً من جوانب المشروع. صُمِّمَ هذا الهيكل لتقديم المعلومات بتسلسل منطقي يبدأ من السياق العام وينتهي بالنتائج والتوصيات، مروراً بالأسس النظرية والتفاصيل التنفيذية.

الفصل الأول: المقدمة

يُقدِّم هذا الفصل (الفصل الحالي) الإطار العام للمشروع. يبدأ بعرض خلفية الموضوع التي تُوضِّح السياق الذي ينطلق منه المشروع: أهمية نزاهة الامتحانات، والتطور في مجال الرؤية الحاسوبية، ودور الذكاء الاصطناعي في دعم القرار. ثم يُحدد المشكلة البحثية بتحدياتها المتعددة، ويُبرز أهمية المشروع علمياً وعملياً، ويُصيغ الأهداف الرئيسية والفرعية. كما يُحدد نطاق المشروع وحدوده بوضوح، ويصف منهجية العمل المتبعة، وينتهي بهذا القسم الذي يُقدِّم خارطة طريق لقراءة التقرير.

الفصل الثاني: الدراسة النظرية

يُقدِّم هذا الفصل الأساس المعرفي الذي يقوم عليه المشروع. يبدأ بمقدمة تُحدد إطار الدراسة النظرية، ثم يتناول مفهوم الرؤية الحاسوبية وتطبيقاتها في المراقبة. يشرح بالتفصيل تقنية كشف الأشخاص باستخدام الشبكات العصبية التلافيفية وخوارزمية YOLO بإصداراتها المختلفة مع التركيز على YOLOv8 المستخدم في المشروع. يتناول أيضاً تقنية التتبع المتعدد للأشخاص: مرشح كالمان وخوارزمية DeepSORT وميزة إعادة التعرف. يشرح استخراج الميزات باستخدام MediaPipe للوجه والجسم وحساب اتجاه الرأس. يُوضِّح كيفية تحويل الإشارات إلى سلوكيات باستخدام آلية Hysteresis، وكيفية تقييم المخاطر وإصدار التنبيهات. وينتهي الفصل بمراجعة الدراسات السابقة في المجال وتحديد موقع المشروع الحالي منها.

الفصل الثالث: تجهيز بيئة العمل والأدوات

يُوثِّق هذا الفصل البيئة التقنية للمشروع بشكل تفصيلي. يبدأ بوصف البيئة البرمجية: لغة Python وإصدارها، والبيئة الافتراضية، وإدارة الحزم والمكتبات. يستعرض المكتبات والأطر المستخدمة بالتفصيل: PyTorch للتعلم العميق، وOpenCV لمعالجة الفيديو، وUltralytics لـ YOLOv8، وMediaPipe لاستخراج الميزات، وDash لواجهة المستخدم. يُحدد متطلبات العتاد: المعالج والذاكرة ووحدة معالجة الرسومات ومتطلبات الكاميرات. يصف هيكل المشروع: تنظيم المجلدات والوحدات الرئيسية وملفات الإعدادات. وينتهي بشرح تجهيز البيانات: الفيديوهات المستخدمة وصيغ الملفات.

الفصل الرابع: الدراسة التنفيذية

يُعدّ هذا الفصل جوهر التقرير من الناحية التقنية، إذ يصف التنفيذ الفعلي لجميع مكونات النظام. يتناول كل مرحلة من مراحل خط المعالجة بالتفصيل: مرحلة الكشف عن الأشخاص (تحميل النموذج، ومعالجة الإطارات، واستخراج صناديق الإحاطة، وتصفية النتائج)، ومرحلة التتبع (تهيئة DeepSORT، وربط الاكتشافات بالمسارات، والحفاظ على الهوية، والتعامل مع الاختفاء). ثم مرحلة استخراج الميزات (تحديد منطقة الاهتمام، ومعالجة الوجه والجسم، وحساب الإشارات). ومرحلة كشف السلوكيات (تطبيق Hysteresis، وكشف كل سلوك، وتسجيل الأحداث). ومرحلة حساب المخاطر (المعادلة والأوزان والتمهيد والتصنيف). ومرحلة التنبيهات (شروط الإطلاق والتسجيل). ومرحلة المخرجات (الفيديو التوضيحي ومقاطع الأدلة وتقرير PDF). وينتهي بوصف واجهة المستخدم التفاعلية.

الفصل الخامس: الاختبار والنتائج

يُقدّم هذا الفصل تقييماً شاملاً لأداء النظام. يبدأ بوصف بيانات الاختبار: الفيديوهات المستخدمة وخصائصها. ثم يُقدّم نتائج اختبار كل وحدة: دقة كشف الأشخاص والتعامل مع الحجب، واستقرار هوية التتبع وعدد تبديلات الهوية، ودقة كشف كل سلوك ومعدل الإنذارات الكاذبة. يُقدّم أيضاً نتائج اختبار الأداء: سرعة المعالجة واستهلاك الذاكرة والزمن الكلي. يعرض نتائج التشغيل الكامل: ملخص السلوكيات المكتشفة والتنبيهات الصادرة مع عينات من المخرجات. وينتهي بتحليل النتائج: نقاط القوة ونقاط الضعف والعوامل المؤثرة على الدقة.

الفصل السادس: الخاتمة والتوصيات

يُختتم التقرير بهذا الفصل الذي يُلخّص المشروع وإنجازاته ويُقدّم نظرة مستقبلية. يبدأ بملخص شامل للمشروع، ثم يُحدد الإنجازات المتحققة بمقابلة الأهداف الموضوعية بالنتائج المحققة. يستعرض التحديات التي واجهت المشروع والحلول التي طُبِّقت للتغلب عليها. يُقدّم رؤية للعمل المستقبلي: تحسينات ممكنة على النظام الحالي، وميزات إضافية يُمكن إضافتها، وآفاق لتوسيع نطاق المشروع. وينتهي بتوصيات عملية لمن يرغب في استخدام النظام أو البناء عليه.

المراجع

يُذيل التقرير بقائمة المراجع العلمية والتقنية التي استند إليها، مُرتبة حسب ترقيمها في النص. تشمل المراجع: الوثائق الرسمية للأدوات المستخدمة، والأوراق البحثية الأصلية للخوارزميات، والدراسات السابقة في مجال مراقبة الامتحانات.

الفصل الثاني: الدراسة النظرية

2.1 مقدمة الفصل

يُعدّ هذا الفصل الأساس النظري الذي يركز عليه المشروع، إذ يستعرض المفاهيم والتقنيات الجوهرية المستخدمة في بناء نظام دعم القرار الاستباقي لمراقبة الامتحانات. يهدف الفصل إلى تزويد القارئ بفهم عميق للآليات التقنية التي تُمكن النظام من تحليل مقاطع الفيديو واستخلاص المعلومات منها بشكل آلي.

تتنوع المحاور التي يتناولها هذا الفصل لتشمل عدة مجالات متكاملة تُشكّل في مجموعها البنية التقنية للنظام. يبدأ الفصل بتقديم نظرة شاملة على مجال الرؤية الحاسوبية الذي يُمثّل حجر الأساس في معالجة البيانات المرئية وتحليلها. ثم ينتقل إلى استعراض تقنيات كشف الأشخاص في الصور والفيديو، مع التركيز على الشبكات العصبية التلافيفية وخوارزمية YOLO التي أحدثت نقلة نوعية في هذا المجال.

يتناول الفصل أيضاً موضوع تتبع الأشخاص عبر إطارات الفيديو المتتالية، وهو جانب حيوي يضمن الحفاظ على هوية كل شخص طوال مدة التسجيل. كما يستعرض آليات استخراج الميزات من الوجه والجسم باستخدام إطار عمل MediaPipe، والتي تُوفّر معلومات دقيقة عن وضعية الرأس وحركات الجسم.

يُعالج الفصل كذلك موضوع تحويل الإشارات الخام إلى سلوكيات قابلة للتفسير، مستخدماً آليات متقدمة لضمان استقرار الكشف ومنع الإنذارات الكاذبة. وأخيراً، يستعرض الفصل نماذج تقييم المخاطر وآليات إصدار التنبيهات، إضافة إلى مراجعة الدراسات السابقة في مجال مراقبة الامتحانات باستخدام الذكاء الاصطناعي.

2.2 الرؤية الحاسوبية

يتناول هذا القسم مفهوم الرؤية الحاسوبية وأساسياتها، ثم يستعرض تطبيقاتها في مجال المراقبة بشكل عام ومراقبة الامتحانات بشكل خاص. يُعدّ فهم هذه الأساسيات ضرورياً لاستيعاب آليات عمل النظام المُطوّر في هذا المشروع.

2.2.1 مفهوم الرؤية الحاسوبية

تُعرّف الرؤية الحاسوبية (Computer Vision) بأنها فرع من فروع الذكاء الاصطناعي يهتم بتمكين الحواسيب من تفسير وفهم المحتوى المرئي من العالم الحقيقي [5]. يشمل هذا المجال مجموعة واسعة من المهام تتراوح بين التعرف على الأشياء وتصنيفها، وصولاً إلى فهم المشاهد المعقدة واستخلاص المعلومات منها بشكل آلي.

تعتمد الرؤية الحاسوبية على محاكاة قدرة الإنسان على الإبصار والإدراك البصري، إذ يستقبل النظام البصري البشري الضوء عبر العين ويُحوّله إلى إشارات عصبية يُعالجها الدماغ لتكوين فهم شامل للمشاهد. وبالمثل، تستقبل أنظمة الرؤية الحاسوبية الصور أو مقاطع الفيديو كمدخلات رقمية، ثم تُطبّق عليها سلسلة من الخوارزميات لاستخلاص المعلومات المطلوبة.

شهد مجال الرؤية الحاسوبية تطوراً هائلاً خلال العقد الأخير بفضل التقدم في تقنيات التعلم العميق (Deep Learning) وتوفر كميات ضخمة من البيانات وزيادة القدرات الحاسوبية. أصبحت الأنظمة الحديثة قادرة على تحقيق دقة تفوق الأداء البشري في العديد من المهام، مما فتح آفاقاً واسعة للتطبيقات العملية في شتى المجالات.

تتضمن المهام الأساسية في الرؤية الحاسوبية: تصنيف الصور (Image Classification) الذي يهدف إلى تحديد محتوى الصورة، وكشف الأشياء (Object Detection) الذي يحدد موقع وهوية الأشياء داخل الصورة، وتجزئة الصور (Image Segmentation) التي تُقسّم الصورة إلى مناطق ذات معنى، إضافة إلى تتبع الأشياء (Object Tracking) الذي يتابع حركة الأشياء عبر إطارات الفيديو المتتالية.

2.2.2 تطبيقات الرؤية الحاسوبية في المراقبة

تُعدّ أنظمة المراقبة الذكية من أبرز التطبيقات العملية للرؤية الحاسوبية، إذ تُوظّف هذه التقنيات لتحليل تدفقات الفيديو من كاميرات المراقبة بشكل آلي [11]. تتراوح هذه التطبيقات بين المراقبة الأمنية في الأماكن العامة وصولاً إلى مراقبة البيئات التعليمية كقاعات الامتحانات.

في سياق المراقبة الأمنية، تُستخدم الرؤية الحاسوبية للكشف عن السلوكيات غير الطبيعية أو المشبوهة، مثل التجمعات غير المعتادة أو الحركات السريعة أو دخول مناطق محظورة. تعتمد هذه الأنظمة على تحليل أنماط الحركة والسلوك لتمييز الأحداث العادية عن تلك التي تستدعي الانتباه.

أما في مجال مراقبة الامتحانات، فقد برزت الحاجة إلى أنظمة ذكية قادرة على مساعدة المراقبين البشريين في رصد السلوكيات المشبوهة [10][12]. تواجه المراقبة التقليدية تحديات عديدة منها: محدودية قدرة المراقب على متابعة أعداد كبيرة من الطلاب في آن واحد، والإرهاق الناتج عن فترات المراقبة الطويلة، وصعوبة توثيق الأدلة بشكل موضوعي ومنهجي.

تُقدّم الرؤية الحاسوبية حلاً فعّالاً لهذه التحديات من خلال القدرة على مراقبة جميع الطلاب بشكل متزامن ومستمر دون تعب أو تشتت. كما تُوفّر إمكانية التوثيق الآلي للأحداث مع طوابع زمنية دقيقة، وتُنبّه المراقب البشري عند رصد سلوكيات تستدعي الانتباه، مما يُمكنه من التركيز على الحالات ذات الأولوية.

تجدر الإشارة إلى أن دور الرؤية الحاسوبية في هذا السياق يبقى مساعداً وداعماً للقرار البشري، إذ يظل المراقب البشري صاحب الكلمة الفصل في تقييم السلوكيات واتخاذ القرارات المناسبة. هذا التكامل بين الذكاء الاصطناعي والحكم البشري يضمن الموازنة بين الكفاءة والعدالة في عملية المراقبة.

2.3 كشف الأشخاص في الصور والفيديو

يُمثّل كشف الأشخاص (Person Detection) الخطوة الأولى والأساسية في أي نظام مراقبة ذكي، إذ يتوجب على النظام تحديد موقع كل شخص في الإطار قبل أن يتمكن من تحليل سلوكه. يستعرض هذا القسم الأسس النظرية لتقنيات الكشف، بدءاً من الشبكات العصبية التلافيفية وصولاً إلى خوارزمية YOLO التي تُستخدم في المشروع الحالي.

2.3.1 الشبكات العصبية التلافيفية (CNN)

تُعَدّ الشبكات العصبية التلافيفية (Convolutional Neural Networks – CNN) من أهم الإنجازات في مجال التعلم العميق، وقد أحدثت ثورة في معالجة الصور وتحليلها [6]. تتميز هذه الشبكات بقدرتها على استخلاص الميزات (Features) من الصور بشكل تلقائي وهرمي، بدءاً من الحواف والأشكال البسيطة في الطبقات الأولى وصولاً إلى الأنماط المعقدة في الطبقات العميقة.

تتكون الشبكة العصبية التلافيفية من عدة أنواع من الطبقات: طبقات الالتفاف (Convolutional Layers) التي تُطبّق مرشحات على الصورة لاستخراج الميزات، وطبقات التجميع (Pooling Layers) التي تُقلّل من حجم البيانات مع الحفاظ على المعلومات المهمة، وطبقات التسطّيح والربط الكامل (Fully Connected Layers) التي تُحوّل الميزات المستخرجة إلى تصنيفات أو تنبؤات.

تعتمد عملية التدريب على خوارزمية الانتشار العكسي (Backpropagation) حيث يتم تعديل أوزان الشبكة بشكل تدريجي لتقليل الفرق بين المخرجات المتوقعة والفعلية. يتطلب تدريب الشبكات العميقة كميات كبيرة من البيانات المُعنونة وقدرة حاسوبية عالية، لكن النتائج تكون دقيقة بشكل ملفت.

2.3.2 خوارزمية YOLO

تُعدّ خوارزمية YOLO (You Only Look Once) نقلة نوعية في مجال كشف الأشياء، إذ قدّمها Redmon وزملاؤه عام 2016 كبديل سريع وفعال للطرق التقليدية [6]. تكمن الفكرة الأساسية في معالجة الصورة كاملةً مرة واحدة للتنبؤ بجميع صناديق الإحاطة (Bounding Boxes) واحتمالات الفئات في آن واحد، على خلاف الطرق السابقة التي كانت تعتمد على مسح الصورة بنوافذ متعددة.

تُقسّم خوارزمية YOLO الصورة إلى شبكة من الخلايا، وتكون كل خلية مسؤولة عن التنبؤ بالأشياء التي يقع مركزها داخلها. لكل خلية، تتنبأ الشبكة بعدة صناديق إحاطة مع درجات الثقة واحتمالات الانتماء لكل فئة. ثم تُطبّق تقنية قمع غير الأقصى (Non-Maximum Suppression – NMS) لإزالة الصناديق المتكررة والإبقاء على الأفضل.

تتميز YOLO بسرعتها الفائقة التي تُمكنها من معالجة الفيديو في الوقت الحقيقي، إذ يمكن للإصدارات الحديثة معالجة عشرات الإطارات في الثانية. كما تتميز بقدرتها على فهم السياق العام للصورة لأنها تتنظر إليها ككل، مما يُقلّل من الأخطاء الناتجة عن الخلفية.

2.3.3 إصدارات YOLO وتطورها

شهدت خوارزمية YOLO تطوراً مستمراً منذ إطلاقها، حيث صدرت عدة إصدارات حسّنت من الدقة والسرعة والقدرة على كشف الأشياء الصغيرة. بدأت الرحلة مع YOLOv1 عام 2016 التي أثبتت جدوى المفهوم، ثم جاء YOLOv2 (YOLO9000) بتحسينات في البنية والتدريب، تلاه YOLOv3 الذي قدّم الكشف متعدد المقاييس لتحسين اكتشاف الأشياء الصغيرة.

استمر التطوير مع YOLOv4 و YOLOv5 اللذين قدّما تحسينات كبيرة في الأداء وسهولة الاستخدام. أصبحت هذه الإصدارات معياراً في الصناعة للتطبيقات التي تتطلب كشفاً سريعاً ودقيقاً. ومع إصدار YOLOv8 من شركة Ultralytics [1]، وصلت الخوارزمية إلى مستوى جديد من النضج مع واجهة استخدام محسّنة وأداء متفوق.

2.3.4 YOLOv8 المستخدم في المشروع

يعتمد المشروع الحالي على إصدار YOLOv8 من Ultralytics [1] لكشف الأشخاص في إطارات الفيديو. يتوفر هذا الإصدار بعدة أحجام تتناسب مع مختلف المتطلبات: النسخة النانو (n) والصغيرة (s) والمتوسطة (m) والكبيرة (l) والضخمة (x)، حيث تزداد الدقة مع الحجم على حساب السرعة.

في هذا المشروع، تم اختيار النسخة الصغيرة (YOLOv8s) لتحقيق التوازن الأمثل بين الدقة والسرعة في بيئة مراقبة الامتحانات. يُدرَّب النموذج مسبقاً على مجموعة بيانات COCO التي تضم 80 فئة من الأشياء، ويتم استخدام فئة "الشخص" (person) فقط لأغراض هذا المشروع.

تشمل المعاملات الأساسية المستخدمة: عتبة الثقة (confidence threshold) التي تُحدد الحد الأدنى لقبول الكشف، وعتبة التقاطع (IoU threshold) لقمع غير الأقصى، والحد الأقصى لعدد الاكتشافات في الإطار الواحد. يتم ضبط هذه المعاملات بما يتناسب مع طبيعة فيديوهات قاعات الامتحان من حيث الإضاءة وزاوية الكاميرا وكثافة الطلاب.

يُخرج النموذج لكل إطار قائمة بصناديق الإحاطة التي تُحدد موقع كل شخص مكتشف، مع درجة الثقة لكل صندوق. تُشكّل هذه المخرجات المدخلات الأساسية لمرحلة التتبع التي ستناقش في القسم التالي.

2.4 تتبع الأشخاص عبر إطارات الفيديو

يُشكّل تتبع الأشخاص (Person Tracking) المرحلة الحيوية التي تلي عملية الكشف، إذ لا يكفي معرفة أن هناك أشخاصاً في الإطار، بل يجب ربط كل شخص بهويته عبر الإطارات المتتالية للفيديو. تضمن هذه العملية أن الشخص الذي يظهر في الإطار رقم 100 هو نفسه الذي ظهر في الإطار رقم 1، مما يُتيح تحليل سلوكه على مدار الوقت واستخلاص أنماط ذات معنى. يتناول هذا القسم المفاهيم الأساسية للتتبع المتعدد، ومرشح كالمان، وخوارزمية DeepSORT المُستخدمة في المشروع.

2.4.1 مفهوم التتبع المتعدد (MOT)

يُعرف التتبع المتعدد للأشياء (Multiple Object Tracking – MOT) بأنه مهمة تحديد مواقع وهويات عدة أشياء متحركة في تسلسل فيديو [7]. تكمن الصعوبة الرئيسية في الحفاظ على هوية ثابتة لكل شخص رغم التحديات العديدة: الحركة المستمرة، والاختفاء المؤقت خلف عوائق، والتقاطع بين مسارات الأشخاص، والتغيرات في المظهر بسبب الإضاءة أو الزاوية.

يعتمد النهج السائد في التتبع المتعدد على مبدأ "التتبع بالكشف" (Tracking-by-Detection)، حيث يُجرى الكشف عن الأشخاص في كل إطار بشكل مستقل باستخدام كاشف مثل YOLO، ثم تُربط هذه الاكتشافات عبر الإطارات لتشكيل مسارات (Tracks). تتطلب عملية الربط هذه خوارزميات ذكية قادرة على التعامل مع الغموض والضوضاء في البيانات.

تُقاس جودة أنظمة التتبع بعدة مقاييس أهمها: دقة التتبع المتعدد (MOTA) التي تأخذ في الاعتبار الكشوفات الفائتة والإيجابيات الكاذبة وتبديلات الهوية، وعدد تبديلات الهوية (ID Switches) الذي يُحصى المرات التي تغيرت فيها الهوية المُسندة لنفس الشخص، ونسبة المسارات المكتملة التي تتبع الشخص من ظهوره الأول حتى اختفائه.

في سياق مراقبة الامتحانات، يكتسب التتبع أهمية خاصة لأن تحليل السلوك يتطلب متابعة الشخص نفسه عبر الزمن. فإذا كان الطالب ينظر للأسفل بشكل متكرر، يجب أن تُنسب كل هذه الملاحظات لنفس الشخص لحساب درجة المخاطر بشكل صحيح. أي خطأ في التتبع قد يؤدي إلى نسب سلوكيات شخص لآخر، مما يُفقد النظام مصداقيته.

2.4.2 مرشح كالمان (Kalman Filter)

يُعدّ مرشح كالمان (Kalman Filter) من الأدوات الرياضية الأساسية في أنظمة التتبع، إذ يُوفّر طريقة فعّالة للتنبؤ بموقع الشيء في الإطار التالي بناءً على حالته الحالية وتاريخ حركته [7]. طُوّر هذا المرشح في الستينيات على يد رودولف كالمان، واستُخدم أصلاً في أنظمة الملاحة الفضائية قبل أن يجد تطبيقات واسعة في مجالات عديدة.

يعمل مرشح كالمان على مرحلتين متكررتين: مرحلة التنبؤ (Prediction) ومرحلة التحديث (Update). في مرحلة التنبؤ، يستخدم المرشح نموذجاً للحركة لتقدير موقع الشخص في الإطار التالي بناءً على موقعه وسرعته الحاليين. يفترض هذا النموذج عادةً حركة خطية منتظمة، أي أن الشخص سيستمر في نفس الاتجاه بنفس السرعة تقريباً.

في مرحلة التحديث، يُقارن الموقع المتوقع به مع الموقع الفعلي المكتشف، ويُعدّل تقدير الحالة بناءً على هذه المقارنة. يُحسب المرشح أيضاً مقدراً يُسمى مصفوفة التغاير (Covariance Matrix) التي تُعبّر عن درجة عدم اليقين في التقديرات. كلما زاد عدم اليقين، أعطى المرشح وزناً أكبر للقياسات الجديدة.

تتمثل إحدى المزايا الرئيسية لمرشح كالمان في قدرته على التعامل مع الضوضاء في القياسات، فحتى لو كانت مخرجات الكاشف غير دقيقة تماماً أو مفقودة أحياناً، يستطيع المرشح تقديم تقديرات معقولة للموقع. هذه الخاصية مفيدة جداً عندما يخفي الشخص مؤقتاً خلف عائق أو يفشل الكاشف في رصده لبضعة إطارات.

في سياق التتبع المتعدد، يُخصص مرشح كالمان مستقل لكل مسار نشط، حيث يحتفظ كل مرشح بتقديره الخاص لموقع وسرعة الشخص المُتتبع. عند وصول اكتشافات جديدة، تُستخدم المواقع المتنبأ بها لمطابقة الاكتشافات مع المسارات الموجودة.

2.4.3 خوارزمية DeepSORT

تُمثّل خوارزمية DeepSORT (Deep Simple Online and Realtime Tracking) تطويراً مهماً لخوارزمية SORT الأصلية، حيث تُضيف معلومات المظهر (Appearance Information) لتحسين دقة التتبع وتقليل تبديلات الهوية [7]. قُدِّمت هذه الخوارزمية من قبل Wojke وزملائه عام 2017، وأصبحت من أكثر خوارزميات التتبع استخداماً في التطبيقات العملية.

تجمع DeepSORT بين ثلاثة مكونات أساسية: مرشح كالمان للتنبؤ بالحركة، وخوارزمية المطابقة الهنغارية (Hungarian Algorithm) لربط الاكتشافات بالمسارات، وشبكة عصبية عميقة لاستخراج ميزات المظهر. هذا الدمج يُمكن الخوارزمية من التعامل مع الحالات الصعبة التي تفشل فيها الطرق المعتمدة على الحركة وحدها.

تعمل الخوارزمية على النحو التالي: عند وصول إطار جديد مع اكتشافاته، يُستخدم مرشح كالمان للتنبؤ بمواقع جميع المسارات النشطة. ثم تُحسب مسافتان لكل زوج من (اكتشاف، مسار): مسافة ماهالانوبيس (Mahalanobis Distance) التي تقيس التوافق في الموقع والسرعة، ومسافة الكوساين (Cosine Distance) التي تقيس التشابه في المظهر. تُدمج هاتان المسافتان لتشكيل تكلفة المطابقة.

تُحل مشكلة المطابقة باستخدام الخوارزمية الهنغارية التي تجد التعيين الأمثل الذي يُقلل التكلفة الإجمالية. الاكتشافات التي لا تُطابق أي مسار تُنشئ مسارات جديدة، والمسارات التي لا تُطابق أي اكتشاف تُعتبر "مفقودة" وتبقى نشطة لعدد محدود من الإطارات قبل أن تُحذف.

تتضمن المعاملات الرئيسية لـ DeepSORT: max_age الذي يُحدد عدد الإطارات المسموح بها لمسار مفقود قبل حذفه، و n_init الذي يُحدد عدد الاكتشافات المتتالية اللازمة لتأكيد مسار جديد، و max_cosine_distance الذي يُحدد الحد الأقصى للمسافة المقبولة في مطابقة المظهر.

2.4.4 ميزة إعادة التعرف (Re-ID)

تُعد ميزة إعادة التعرف (Re-Identification) من أهم ما يُميز DeepSORT عن خوارزميات التتبع البسيطة [7]. تتمثل الفكرة في استخدام شبكة عصبية عميقة مُدربة على مهمة التعرف على الأشخاص لاستخراج متجه ميزات (Feature Vector) يُمثل مظهر كل شخص. هذا المتجه يكون مميزاً لكل شخص ومستقراً نسبياً رغم التغيرات في الوضعية والإضاءة. تُدرب شبكة استخراج الميزات على مجموعات بيانات كبيرة لإعادة التعرف على الأشخاص، حيث تتعلم استخراج ميزات تكون متشابهة لنفس الشخص ومختلفة بين الأشخاص المختلفين. في مرحلة الاستخدام، تُمرر صورة كل شخص مكتشف عبر هذه الشبكة للحصول على متجه ميزات يُخزن مع المسار.

يحتفظ كل مسار بمجموعة من متجهات الميزات الأخيرة (Gallery) لاستخدامها في المطابقة. عند وصول اكتشاف جديد، يُقارن متجه ميزاته مع متجهات الميزات المخزنة لكل مسار باستخدام مسافة الكوساين. هذه المقارنة تُمكن النظام من إعادة ربط الشخص بمساره الصحيح حتى بعد اختفائه لعدة إطارات.

تبرز أهمية ميزة إعادة التعرف في حالات عديدة: عندما يختفي الشخص مؤقتاً خلف عائق ثم يظهر مجدداً، أو عندما يتقاطع مسار شخصين فيصعب التمييز بينهما بالموقع وحده، أو عندما يفشل الكاشف في رصد الشخص لعدة إطارات متتالية. في كل هذه الحالات، تُساعد ميزات المظهر في الحفاظ على الهوية الصحيحة.

في سياق مراقبة الامتحانات، تكتسب هذه الميزة أهمية خاصة لأن الطلاب قد يتحركون أو يميلون بطرق تُخفيهم جزئياً عن الكاميرا. كما أن وجود عدد كبير من الطلاب في مساحة محدودة يزيد من احتمال التقاطع والتداخل. تضمن ميزة إعادة التعرف أن سجل السلوك لكل طالب يبقى مرتبطاً به بشكل صحيح طوال مدة الامتحان.

يُخرج نظام التتبع لكل إطار قائمة بالمسارات النشطة، حيث يتضمن كل مسار معرفاً فريداً (Track ID) وصندوق إحاطة مُحَدَّث وحالة المسار (مؤكد أو مؤقت أو مفقود). تُشكل هذه المخرجات المدخلات لمرحلة استخراج الميزات التي ستناقش في القسم التالي.

2.5 استخراج ميزات الوجه والجسم

بعد كشف الأشخاص وتتبعهم، تأتي مرحلة استخراج الميزات التفصيلية من كل شخص لتحليل وضعيته وحركاته. تُوفّر هذه الميزات المعلومات اللازمة لفهم ما يفعله الشخص: هل ينظر للأمام أم للأسفل؟ هل يلتفت يميناً أم يساراً؟ هل يرفع يده نحو وجهه؟ يتناول هذا القسم إطار عمل MediaPipe المستخدم لاستخراج هذه الميزات، والتقنيات المحددة لتحليل الوجه والجسم، وكيفية حساب الإشارات المفيدة منها.

2.5.1 إطار عمل MediaPipe

يُعدّ MediaPipe إطار عمل مفتوح المصدر طوّره شركة Google لبناء خطوط معالجة الوسائط المتعددة [3][8]. صُمم هذا الإطار ليكون مرناً وقابلاً للتوسيع، مع التركيز على الأداء العالي والقدرة على العمل في الوقت الحقيقي على أجهزة متنوعة تتراوح بين الهواتف المحمولة والحواسيب المكتبية.

يُقدّم MediaPipe مجموعة من الحلول الجاهزة (Solutions) لمهام شائعة في الرؤية الحاسوبية، من أهمها: Face Detection لكشف الوجوه، و Face Mesh لاستخراج النقاط المعلمية للوجه، و Pose لتقدير وضعية الجسم، و Hands لتتبع اليدين، و Holistic الذي يجمع عدة حلول معاً. تعتمد هذه الحلول على نماذج تعلم آلي خفيفة الوزن ومُحسّنة للعمل بكفاءة.

تتميز بنية MediaPipe بمفهوم الرسم البياني (Graph) حيث تُعرّف خطوط المعالجة كعقد مترابطة، تستقبل كل عقدة مدخلات وتنتج مخرجات تتدفق للعقد التالية. هذا التصميم يُسهّل بناء أنظمة معقدة من مكونات بسيطة، ويُتيح تشغيل أجزاء مختلفة من الخط بشكل متوازٍ لتحسين الأداء.

في المشروع الحالي، يُستخدم MediaPipe لاستخراج معلومات تفصيلية عن وضعية الرأس والجسم واليدين لكل شخص مُتتبع. تُعالج صورة كل شخص (المقتطعة من الإطار بناءً على صندوق الإحاطة) بشكل مستقل، وتُخزّن الميزات المستخرجة مع معرف المسار والطابع الزمني للتحليل اللاحق.

2.5.2 Face Mesh ونقاط الوجه

يُوفّر حل Face Mesh من MediaPipe القدرة على استخراج 468 نقطة معلمية ثلاثية الأبعاد (D Landmarks3) من الوجه في الوقت الحقيقي [3][8]. تُغطي هذه النقاط ملامح الوجه بالكامل: محيط الوجه، والحاجبين، والعينين، والأنف، والشففتين. تُقدّم إحداثيات كل نقطة بالأبعاد الثلاثة (X, Y, Z)، حيث يُشير البعد الثالث إلى العمق النسبي.

يعتمد Face Mesh على شبكة عصبية خفيفة الوزن مُدرّبة على ملايين الصور لوجوه متنوعة. تعمل الشبكة بمرحلتين: أولاً كشف الوجه وتحديد منطقة الاهتمام، ثم استخراج النقاط المعلمية من هذه المنطقة. هذا التقسيم يُحسّن الكفاءة لأن المرحلة الأولى تعمل على الصورة كاملة بينما المرحلة الثانية تركز على منطقة صغيرة.

تُستخدم نقاط الوجه في المشروع الحالي لحساب عدة إشارات مهمة. أولاً، يُمكن تقدير اتجاه النظر من خلال تحليل موقع بؤبؤ العين بالنسبة لمحيط العين. ثانياً، يُمكن حساب زوايا الرأس (الانحناء والإمالة) من الترتيب الهندسي للنقاط المعلمية. ثالثاً، يُمكن الكشف عن تعابير الوجه التي قد تُشير إلى حالات معينة.

من المعاملات المهمة في استخدام Face Mesh: الحد الأدنى لحجم الوجه المقبول للمعالجة (لتجنب النتائج غير الدقيقة مع الوجوه الصغيرة جداً)، ودرجة الثقة المطلوبة للكشف، وخيار تتبع الوجه عبر الإطارات لتحسين الاستقرار. يُضبط كل معامل بما يتناسب مع ظروف التصوير في قاعة الامتحان.

2.5.3 Pose Estimation ونقاط الجسم

يُوفّر حل Pose من MediaPipe تقدير وضعية الجسم من خلال استخراج 33 نقطة معلمية تُغطي الجسم من الرأس حتى القدمين [3][8]. تشمل هذه النقاط: الأنف والعينين والأذنين، والكتفين والمرفقين والرسغين، والوركين والركبتين والكاحلين، إضافة إلى نقاط أصابع اليدين والقدمين.

يتميز هذا الحل بقدرته على العمل حتى عندما يكون جزء من الجسم مخفياً، إذ يُقدّر النموذج مواقع النقاط غير المرئية بناءً على السياق. هذه الخاصية مفيدة جداً في بيئة الامتحان حيث قد يجلس الطلاب خلف طاولات تُخفي الجزء السفلي من أجسامهم.

تُستخدم نقاط الجسم في المشروع لحساب إشارات حيوية عديدة. يُحسب انحناء الرأس للأمام من العلاقة بين موقع الأنف وموقع الكتفين، فعندما ينظر الشخص للأسفل ينخفض أنفه بالنسبة لخط الكتفين. كما يُحسب التقاف الجذع من التناظر بين الكتفين، فعندما يلتفت الشخص يظهر أحد الكتفين أقرب للكاميرا من الآخر.

تُحسب أيضاً إشارة "اليد نحو الوجه" من المسافة بين معصم اليد والأنف. عندما تقترب اليد من الوجه (كأن يضع الشخص يده على فمه أو أذنه)، تنخفض هذه المسافة دون عتبة معينة مما يُطلق الإشارة. هذه الحركة قد تُشير إلى الهمس أو استخدام سماعة أذن صغيرة.

يُوفّر MediaPipe أيضاً درجة رؤية (Visibility Score) لكل نقطة، تُشير إلى مدى وضوح هذه النقطة في الصورة. تُستخدم هذه الدرجة لتجاهل النقاط ذات الرؤية المنخفضة والتي قد تكون غير دقيقة، مما يُحسن موثوقية الإشارات المحسوبة.

2.5.4 حساب اتجاه الرأس (Yaw, Pitch)

يُعدّ تقدير اتجاه الرأس (Head Pose Estimation) من المهام الأساسية في تحليل السلوك، إذ يُوفّر معلومات دقيقة عن وجهة نظر الشخص [8]. يُعبّر عن اتجاه الرأس عادةً بثلاث زوايا: الالتفات الأفقي (Yaw) حول المحور العمودي، والإمالة العمودية (Pitch) حول المحور الأفقي، والدوران (Roll) حول المحور الأمامي-الخلفي.

يُحسب زاوية Yaw من الموقع الأفقي للأنف بالنسبة لمركز الوجه. عندما ينظر الشخص للأمام مباشرة، يكون الأنف في منتصف الوجه تقريباً. وعندما يلتفت يميناً، يتحرك الأنف نحو يسار الصورة (من منظور الكاميرا)، والعكس عند الالتفات يساراً. يُمكن حساب هذه الزاوية باستخدام دالة arctangent للنسبة بين الإزاحة الأفقية للأنف وعمق الوجه.

تُحسب زاوية Pitch من الموقع العمودي للأنف بالنسبة لمستوى العينين. عندما ينظر الشخص للأمام، يكون الأنف أسفل العينين بمسافة نموذجية. وعندما يُميل رأسه للأسفل (للنظر في حضنه مثلاً)، يرتفع الأنف بالنسبة للعينين في الصورة. يُمكن تطبيق هذه المسافة بارتفاع الوجه للحصول على قيمة مستقلة عن بُعد الشخص عن الكاميرا.

في المشروع الحالي، يُستخدم مزيج من معلومات Face Mesh و Pose لحساب اتجاه الرأس بشكل موثوق. عندما يكون الوجه مرئياً بوضوح، تُستخدم نقاط Face Mesh لأنها أكثر دقة. وعندما يكون الوجه مخفياً جزئياً أو مُلتقياً بشكل كبير، تُستخدم نقاط Pose كبديل.

تُحوّل الزوايا المحسوبة إلى إشارات سلوكية قابلة للاستخدام. فمثلاً، تُشير قيمة Pitch السالبة الكبيرة إلى النظر للأسفل، وتُحوّل إلى درجة (score) بين 0 و 1 تُعبّر عن شدة السلوك. وبالمثل، تُشير قيم Yaw الكبيرة (موجبة أو سالبة) إلى الالتفات للجانب، وتُحوّل إلى درجة التفات يمين أو يسار.

تُجمع الميزات المستخرجة من كل إطار في سجل (Record) يتضمن: معرّف المسار، والطابع الزمني، ورقم الإطار، وصندوق الإحاطة، ودرجات الإشارات المختلفة (النظر للأسفل، الالتفات، اليد نحو الوجه)، إضافة إلى معلومات إضافية عن جودة الكشف. تُخزّن هذه السجلات بتنسيق JSONL (سطر JSON لكل سجل) للمعالجة اللاحقة في مرحلة كشف السلوكيات.

2.6 كشف السلوكيات من الإشارات

تمثّل الإشارات المستخرجة من MediaPipe قيماً رقمية خاماً تتغير من إطار لآخر، وقد تتذبذب بشكل سريع بسبب الضوضاء أو عدم الاستقرار في الكشف. لتحويل هذه الإشارات إلى معلومات مفيدة عن السلوك، يجب تطبيق آليات تُحدد متى يبدأ سلوك معين ومتى ينتهي، مع ضمان الاستقرار وتجنب الإنذارات الكاذبة الناتجة عن التذبذبات العابرة. يتناول هذا القسم منهجية تحويل الإشارات إلى أحداث سلوكية، وآلية Hysteresis المستخدمة لتحقيق الاستقرار، والسلوكيات المحددة التي يستهدفها النظام.

2.6.1 تحويل الإشارات إلى سلوكيات

تختلف الإشارات الخام عن السلوكيات في طبيعتها الجوهرية. الإشارة هي قيمة رقمية مستمرة تتغير في كل إطار، مثل "درجة النظر للأسفل = 0.45". أما السلوك فهو حدث له بداية ونهاية ومدة زمنية، مثل "بدأ الشخص بالنظر للأسفل في الثانية 5.2 وانتهى في الثانية 8.4". هذا التحويل ضروري لأن تحليل السلوك يعتمد على فهم الأنماط الزمنية لا مجرد القيم اللحظية.

تتطلب عملية التحويل تحديد عتبة (Threshold) تفصل بين الحالة الطبيعية والحالة السلوكية. فمثلاً، يُمكن اعتبار أن درجة النظر للأسفل أكبر من 0.5 تُشير إلى أن الشخص ينظر للأسفل فعلاً. لكن استخدام عتبة واحدة يُسبب مشكلة خطيرة: عندما تتذبذب الإشارة حول العتبة، يتم تسجيل عدد كبير من أحداث البداية والنهاية المتتالية، مما يُشوّش على التحليل ويُنتج نتائج غير واقعية.

تبرز هذه المشكلة بشكل خاص في البيانات الواقعية حيث تتأثر الإشارات بعوامل عديدة: تغيرات الإضاءة، وحركة الكاميرا الطفيفة، وعدم دقة الكشف في بعض الإطارات، والحركات الطبيعية السريعة للشخص. كل هذه العوامل تُسبب تذبذباً في قيم الإشارات حتى عندما لا يتغير السلوك الفعلي للشخص.

لذلك، يتطلب التحويل الناجح آلية تُميز بين التغيرات الحقيقية في السلوك والتذبذبات العابرة. يجب أن تكون هذه الآلية حساسة بما يكفي لالتقاط السلوكيات الفعلية، وفي الوقت نفسه مُستقرة بما يكفي لتجاهل الضوضاء. تُحقق آلية Hysteresis هذا التوازن بشكل فعال.

2.6.2 آلية Hysteresis لمنع التذبذب

تعتمد آلية Hysteresis على استخدام عتبتين بدلاً من عتبة واحدة: عتبة التفعيل (ON Threshold) وعتبة الإيقاف (OFF Threshold)، حيث تكون عتبة التفعيل أعلى من عتبة الإيقاف. هذا الفارق بين العتبتين يُسمى "نطاق Hysteresis" أو "المنطقة الميتة"، ويُوفر استقراراً كبيراً في الكشف.

تعمل الآلية على النحو التالي: عندما يكون السلوك غير نشط (الحالة الابتدائية)، يجب أن تتجاوز الإشارة عتبة التفعيل العليا لتُفعّل الحالة السلوكية وتُسجّل بداية الحدث. بمجرد تفعيل السلوك، يبقى نشطاً حتى تتخفض الإشارة دون عتبة الإيقاف السفلى، عندها تُسجّل نهاية الحدث وتُعاد الحالة للوضع الابتدائي.

لتوضيح الفائدة، لنفترض أن عتبة التفعيل هي 0.48 وعتبة الإيقاف هي 0.42 للنظر للأسفل. إذا كانت الإشارة تتذبذب بين 0.44 و 0.46، فلن يُفعّل السلوك أبداً لأنها لم تتجاوز 0.48. وإذا فُعل السلوك ثم تذبذبت الإشارة بين 0.43 و 0.47، فسيبقى نشطاً لأنها لم تتخفض دون 0.42. فقط التغيرات الكبيرة والمستمرة في الإشارة تؤدي إلى تغيير حالة السلوك.

يُشتق مصطلح Hysteresis من الفيزياء حيث يصف ظاهرة اعتماد حالة النظام على تاريخه السابق. في الدوائر الإلكترونية، تُستخدم هذه الآلية في مقارنات Schmitt Trigger لتحويل الإشارات التماثلية المُشوَّشة إلى إشارات رقمية نظيفة. وبالمثل، تُستخدم هنا لتحويل الإشارات المستمرة المُشوَّشة إلى أحداث سلوكية واضحة.

يتطلب ضبط عتبات Hysteresis توازناً دقيقاً. إذا كان الفارق بين العتبتين كبيراً جداً، سيتأخر النظام في اكتشاف بداية السلوك وسيستمر في تسجيله بعد انتهائه الفعلي. وإذا كان الفارق صغيراً جداً، لن تكون الحماية من التذبذب كافية. يُضبط هذا الفارق تجريبياً بناءً على خصائص الإشارات في البيئة المستهدفة.

2.6.3 السلوكيات المستهدفة

يستهدف النظام ثلاث فئات رئيسية من السلوكيات التي قد تُشير إلى محاولة غش أو سلوك مشبوه في بيئة الامتحان [10][12][14]. اختيرت هذه السلوكيات بناءً على دراسات سابقة وخبرة عملية في مجال مراقبة الامتحانات، مع مراعاة إمكانية كشفها بشكل موثوق من كاميرات المراقبة النموذجية.

السلوك الأول هو النظر للأسفل (Look Down)، ويُشير إلى توجيه النظر نحو منطقة الحضن أو أسفل الطاولة. قد يُشير هذا السلوك إلى قراءة ورقة غش مخفية أو استخدام هاتف محمول. تُحسب درجة هذا السلوك من زاوية إمالة الرأس (Pitch) ومن انخفاض الأنف بالنسبة لخط الكتفين. تُستخدم عتبات نموذجية: 0.48 للتفعيل و 0.42 للإيقاف.

السلوك الثاني هو الالتفات الجانبي (Head Turn)، ويُشير إلى توجيه الرأس يميناً أو يساراً بشكل ملحوظ. قد يُشير هذا السلوك إلى محاولة النظر في ورقة زميل مجاور. يُقسم هذا السلوك إلى نوعين فرعيين: الالتفات لليمين والالتفات لليسار، لتمكين تحليل أكثر تفصيلاً. تُحسب درجة هذا السلوك من زاوية الالتفات الأفقي (Yaw). تُستخدم عتبات نموذجية: 0.42 للتفعيل و 0.32 للإيقاف.

السلوك الثالث هو وضع اليد على الوجه (Hand to Face)، ويُشير إلى تقريب اليد من منطقة الفم أو الأذن. قد يُشير هذا السلوك إلى الهمس لزميل قريب أو استخدام سماعة أذن صغيرة للتواصل مع طرف خارجي. تُحسب درجة هذا السلوك من المسافة بين معصم اليد والأنف مقارنة بطول الذراع. تُستخدم عتبات نموذجية: 0.45 للتفعيل و 0.35 للإيقاف، مع فترة ثبات إضافية (Hold Time) تبلغ 300 مللي ثانية لتجنب التسجيل الخاطئ للحركات العابرة كحكّ الوجه.

يُنتج كشف السلوكيات سجلاً من الأحداث بتنسيق JSONL، حيث يتضمن كل حدث: الطابع الزمني بالمللي ثانية، ورقم الإطار، ومعرّف الشخص، ونوع السلوك، ونوع الحدث (بداية أو نهاية)، ودرجة الإشارة عند لحظة الحدث. يُشكّل هذا السجل المدخل الأساسي لمرحلة تقييم المخاطر التي ستناقش في القسم التالي.

من المهم التأكيد أن اكتشاف هذه السلوكيات لا يعني بالضرورة وجود غش، فكثير منها قد يحدث لأسباب بريئة تماماً. الطالب قد ينظر للأسفل ليقراً سؤالاً في ورقة الامتحان، أو يلتفت ليسأل المراقب، أو يضع يده على وجهه من التعب أو التفكير. لذلك يُعامل النظام هذه السلوكيات كمؤشرات تستحق الانتباه لا كأدلة قاطعة، ويبقى القرار النهائي للمراقب البشري بعد مراجعة السياق الكامل.

2.7 تقييم المخاطر واتخاذ القرار

بعد كشف السلوكيات الفردية، تأتي مرحلة تجميع هذه المعلومات في تقييم شامل لدرجة خطورة كل شخص. لا يُعامل كل سلوك بالتساوي، فالالتفاتات المتكررة قد يكون أكثر إثارة للقلق من النظر للأسفل العابر، وتزامن عدة سلوكيات يرفع درجة الشك أكثر من سلوك منفرد. يتناول هذا القسم كيفية حساب درجة الخطورة الموزونة، وأساليب تمهيد الإشارات لتحقيق الاستقرار، ونظام التنبيهات الذي يُحوّل درجات الخطورة إلى إشعارات قابلة للتنفيذ.

2.7.1 حساب درجة الخطورة (Risk Score)

تُحسب درجة الخطورة لكل شخص في كل إطار كمجموع موزون لدرجات السلوكيات المختلفة. لكل نوع سلوك وزن يعكس أهميته النسبية في الدلالة على احتمال الغش. يُتيح هذا النهج مرونة في ضبط حساسية النظام لكل نوع سلوك بشكل مستقل، ويُمكن من التكيف مع سياسات مؤسسية مختلفة.

تُعبّر صيغة حساب درجة الخطورة الفورية رياضياً على أنها مجموع حاصل ضرب درجة كل سلوك في وزنه المقابل. فإذا كانت درجة النظر للأسفل 0.7 ووزنه 0.35، ودرجة الالتفاتات 0.4 ووزنه 0.40، ودرجة اليد للوجه 0.0 ووزنها 0.25، فإن درجة الخطورة الفورية تساوي حاصل جمع هذه المنتجات. تُطبّع النتيجة لتبقى في المدى من 0 إلى 1.

يُراعى في تحديد الأوزان عدة عوامل. أولاً، مدى ارتباط السلوك بالغش فعلياً استناداً إلى الدراسات السابقة والخبرة العملية. ثانياً، موثوقية كشف السلوك تقنياً، فالسلوكيات التي تُكشف بدقة عالية يُمكن إعطاؤها أوزاناً أعلى. ثالثاً، معدل حدوث السلوك بشكل طبيعي، فالسلوكيات النادرة في الظروف الطبيعية تكون أكثر دلالة عند حدوثها.

تُضبط الأوزان الافتراضية في النظام بحيث يحصل الالتفاتات الجانبي على الوزن الأعلى (0.40) لأنه الأكثر ارتباطاً بمحاولة النظر في ورقة الآخرين، يليه النظر للأسفل (0.35) الذي قد يُشير لقراءة مواد مخفية، ثم اليد للوجه (0.25) الذي قد يُشير لاستخدام أجهزة اتصال. هذه الأوزان قابلة للتعديل من ملفات الإعداد دون تغيير الشيفرة المصدرية.

تُحسب أيضاً درجات فرعية لكل فئة سلوكية، مما يُتيح تحليلاً أكثر تفصيلاً. يُمكن مثلاً تتبع درجة "الانتباه البصري" التي تجمع النظر للأسفل والالتفاتات، مقابل درجة "الحركة الجسدية" التي تركز على اليد للوجه. هذا التقسيم يُساعد المراقب على فهم طبيعة السلوك المشبوه بشكل أفضل.

2.7.2 التمهيد باستخدام EWMA

تتذبذب درجة الخطورة الفورية بشكل ملحوظ من إطار لآخر، حتى مع استخدام آلية Hysteresis في كشف السلوكيات. هذا التذبذب يجعل من الصعب تحديد الحالة الفعلية للشخص: هل هو في وضع طبيعي أم مشبوه؟ لذلك يُطبَّق تمهيد زمني على درجة الخطورة باستخدام خوارزمية المتوسط المتحرك الأسّي الموزون (Exponential Weighted Moving Average - EWMA).

تعمل خوارزمية EWMA على حساب متوسط متحرك يُعطي وزناً أكبر للقيم الحديثة ووزناً متناقصاً أسياً للقيم القديمة. تُعبّر الصيغة على أن القيمة الممهدة الحالية تساوي حاصل ضرب معامل التمهيد (α) في القيمة الفورية الحالية، مضافاً إليه حاصل ضرب مكمل معامل التمهيد ($1-\alpha$) في القيمة الممهدة السابقة.

يتحكم معامل التمهيد α في سرعة استجابة النظام للتغيرات. قيمة α قريبة من 1 تجعل النظام سريع الاستجابة لكن أقل استقراراً، بينما قيمة α قريبة من 0 تجعل النظام مستقرًا جداً لكن بطيء الاستجابة. يُستخدم عادةً معامل بين 0.1 و 0.3 لتحقيق توازن مناسب. في المشروع الحالي، يُستخدم معامل افتراضي قدره 0.15.

يُوفّر EWMA عدة مزايا مقارنة بأنواع المتوسطات المتحركة الأخرى. أولاً، لا يتطلب تخزين نافذة من القيم السابقة، بل يكفي تخزين القيمة الممهدة الأخيرة فقط، مما يُوفّر الذاكرة. ثانياً، يستجيب بسلاسة للتغيرات دون قفزات مفاجئة عند خروج قيم قديمة من نافذة المتوسط. ثالثاً، يُعطي وزناً طبيعياً متدرجاً للتاريخ، فالماضي القريب أهم من البعيد.

تُطبَّق EWMA بشكل مستقل لكل شخص مُتتبع، مع الحفاظ على حالة التمهيد عبر الإطارات. عندما يُفقد شخص من التتبع ثم يُعاد اكتشافه، تُعاد تهيئة حالة التمهيد لتجنب تأثير القيم القديمة غير ذات الصلة.

2.7.3 نظام التنبيهات

يُحوّل نظام التنبيهات (Alert Manager) درجات الخطورة الممهدة إلى إشعارات منظمة تُوجّه للمراقب البشري. لا يكفي مجرد عرض درجة الخطورة، بل يجب تحديد متى تستحق الموقف انتبهاً خاصاً، وتجنب إغراق المراقب بتنبيهات متكررة لنفس الشخص.

يعمل نظام التنبيهات بناءً على ثلاث عتبات تُحدد مستويات الخطورة. العتبة الأولى (Low) تفصل بين الحالة الطبيعية والمنخفضة الخطورة، وقيمتها الافتراضية 0.3. العتبة الثانية (Medium) تفصل بين الخطورة المنخفضة والمتوسطة، وقيمتها الافتراضية 0.5. العتبة الثالثة (High) تفصل بين الخطورة المتوسطة والعالية، وقيمتها الافتراضية 0.7.

يُصدر النظام تنبيهاً عندما تتجاوز درجة الخطورة الممهدة إحدى العتبات صعوداً، ويُسجل انتهاء التنبيه عندما تنخفض دون العتبة السفلى لذلك المستوى. يُطبَّق مبدأ Hysteresis هنا أيضاً لتجنب التذبذب عند حدود العتبات.

يتضمن النظام آلية فترة التهدئة (Cooldown) التي تمنع إصدار تنبيهات متكررة لنفس الشخص في فترة زمنية قصيرة. بعد إصدار تنبيه لشخص ما، لا يُصدر تنبيه جديد لنفس الشخص على نفس المستوى أو أدنى إلا بعد انقضاء فترة التهدئة (افتراضياً 10 ثوانٍ). هذا يمنع إزعاج المراقب بتنبيهات متكررة مع إتاحة الفرصة لملاحظة تصاعد الخطورة لمستوى أعلى.

يتضمن النظام أيضاً آلية فترة الثبات (Hold Time) التي تتطلب استمرار تجاوز العتبة لفترة زمنية قبل إصدار التنبيه. هذا يمنع التنبيهات الناتجة عن ارتفاعات لحظية عابرة في درجة الخطورة. الفترة الافتراضية هي 500 ميلي ثانية، ويُمكن ضبطها حسب الحاجة.

يُنتج نظام التنبيهات سجلاً يتضمن لكل تنبيه: الطابع الزمني، ومعرّف الشخص، ومستوى الخطورة، ودرجة الخطورة الدقيقة، وأهم السلوكيات المساهمة في الدرجة، وصندوق إحاطة الشخص في ذلك الإطار. تُستخدم هذه المعلومات لتوليد إشعارات مرئية في الفيديو المُعالج، ولإنتاج مقاطع فيديو قصيرة للأحداث عالية الخطورة، ولتغذية لوحة المراقبة الحية.

من المبادئ الأساسية في تصميم نظام التنبيهات أنه أداة مساعدة لا بديل عن الحكم البشري. يهدف النظام إلى توجيه انتباه المراقب نحو الحالات التي تستحق المراجعة، لكنه لا يُصدر أحكاماً نهائية. المراقب هو من يُقرر ما إذا كان السلوك المُبلغ عنه يستدعي تدخلاً فعلياً، آخذاً في الاعتبار السياق الذي لا يستطيع النظام الآلي فهمه بالكامل.

2.8 الدراسات السابقة

شهد مجال كشف الغش في الامتحانات باستخدام الرؤية الحاسوبية اهتماماً بحثياً متزايداً في السنوات الأخيرة، مدفوعاً بالحاجة إلى أنظمة مراقبة أكثر كفاءة وموضوعية، وبالتطور الملحوظ في تقنيات التعلم العميق وتحليل الفيديو. يستعرض هذا القسم أبرز الدراسات والأنظمة المقترحة في هذا المجال، مع تحليل منهجياتها ونتائجها ونقاط قوتها وضعفها، وصولاً إلى تحديد موقع المشروع الحالي ضمن هذا السياق البحثي.

2.8.1 أنظمة كشف الغش المبنية على تحليل الفيديو

قدّم الباحثون في [16] نظاماً لكشف الغش في قاعات الامتحان في الوقت الحقيقي باستخدام الرؤية الحاسوبية. اعتمد النظام على تحليل حركات الرأس واتجاه النظر لتحديد السلوكيات المشبوهة. استخدم الباحثون تقنيات كشف الوجه التقليدية مع مصنّفات الحركة لتصنيف السلوك إلى طبيعي أو مشبوه. أظهرت النتائج دقة جيدة في البيئات المضبوطة، لكن الأداء تراجع في ظروف الإضاءة المتغيرة وعند وجود إخفاء جزئي للوجه.

في دراسة أخرى [14]، طُوّر نظام لمراقبة الامتحانات يجمع بين تحليل الصورة وتتبع العين. ركّز النظام على كشف حالات النظر في ورقة الجار من خلال تحليل اتجاه بؤبؤ العين والموقع النسبي للطلاب بالنسبة لزملائه. استخدم الباحثون شبكات عصبية التلافيفية لاستخراج ميزات الوجه وتقدير اتجاه النظر. حقق النظام نتائج واعدة لكنه تطلّب كاميرات عالية الدقة وقريبة من الطلاب لضمان وضوح تفاصيل العين.

اقترح [12] إطار عمل شامل لمراقبة سلوك الطلاب في الامتحانات الإلكترونية والتقليدية. تميّز الإطار بدمج عدة مصادر للمعلومات: تحليل الفيديو لكشف الحركات الجسدية، وتحليل الصوت للكشف عن الهمس أو الحديث، ومراقبة نشاط الحاسوب في الامتحانات الإلكترونية. أظهر هذا النهج متعدد الوسائط تحسناً ملحوظاً في دقة الكشف مقارنة بالأنظمة أحادية المصدر، لكنه زاد من تعقيد النظام ومتطلبات الحوسبة.

2.8.2 تطبيقات التعلم العميق في مراقبة السلوك

تناولت دراسة [11] استخدام الشبكات العصبية العميقة لتصنيف سلوك الطلاب في بيئات التعلم. على الرغم من تركيزها على البيئة الصفية العامة وليس الامتحانات تحديداً، إلا أنها قدّمت منهجية قابلة للتطبيق في سياق المراقبة. استخدم الباحثون بنية شبكة التلافيفية ثلاثية الأبعاد (D-CNN3) لتحليل تسلسلات الفيديو واستخراج ميزات زمنية-مكانية. أظهرت النتائج قدرة النموذج على التمييز بين حالات الانتباه والتشتت والتفاعل.

في [18]، قدّم نظام ذكي لمراقبة قاعات الامتحان يعتمد على تقنيات الرؤية الحاسوبية الحديثة. استخدم النظام نموذج YOLO لكشف الأشخاص والأجهزة الإلكترونية (كالهواتف المحمولة)، مع خوارزمية تتبع للحفاظ على هوية كل طالب عبر الإطارات. ركّز النظام على كشف الأجهزة المحظورة أكثر من تحليل السلوك، وحقق دقة عالية في كشف الهواتف لكنه واجه صعوبات مع الأجهزة الصغيرة كالسماعات اللاسلكية.

استكشفت دراسة [15] إمكانية استخدام نماذج تقدير الوضعية البشرية لتحليل سلوك الطلاب. استخدم الباحثون نموذج OpenPose لاستخراج الهيكل العظمي للجسم، ثم حللوا الزوايا بين المفاصل لتصنيف الوضعيات إلى فئات محددة. أظهر

هذا النهج مرونة في التعامل مع الإخفاء الجزئي لأن النموذج يستطيع تقدير المفاصل غير المرئية، لكن دقة التصنيف اعتمدت بشكل كبير على جودة استخراج الوضعية.

2.8.3 أنظمة المراقبة عن بُعد والامتحانات الإلكترونية

مع انتشار التعليم عن بُعد، ظهرت دراسات عديدة تركز على مراقبة الامتحانات الإلكترونية. في [10]، قُدِّمَ نظام مراقبة يعمل من خلال كاميرا الويب الخاصة بالطالب. يُحلل النظام وجه الطالب للتحقق من هويته، ويراقب اتجاه نظره وحركاته للكشف عن السلوكيات المشبوهة كالنظر خارج الشاشة بشكل متكرر أو اختفاء الوجه من الإطار. استخدم النظام تقنيات خفيفة الوزن تعمل على جهاز الطالب دون الحاجة لخوادم قوية.

تناولت [13] تحديات مراقبة الامتحانات الإلكترونية من منظور الخصوصية والأخلاقيات. ناقشت الدراسة التوازن بين فعالية المراقبة واحترام خصوصية الطلاب، واقترحت مجموعة من المبادئ التوجيهية لتصميم أنظمة مراقبة عادلة وشفافة. أكدت الدراسة على أهمية إعلام الطلاب بما يُراقب وكيف تُستخدم البيانات، وضرورة إتاحة آلية للطعن في قرارات النظام.

2.8.4 تحليل مقارن للمنهجيات

يُمكن تصنيف المنهجيات المستخدمة في الدراسات السابقة إلى عدة فئات. الفئة الأولى تعتمد على التصنيف المباشر للإطارات، حيث يُحلل كل إطار بشكل مستقل ويُصنَّف إلى فئات سلوكية. هذا النهج بسيط لكنه يفتقر للسياق الزمني ويُنتج نتائج متذبذبة.

الفئة الثانية تعتمد على تحليل التسلسلات الزمنية، حيث تُحلل مجموعة من الإطارات المتتالية معاً باستخدام شبكات عصبية تكرارية (RNN) أو شبكات التفاضلية ثلاثية الأبعاد. هذا النهج يلتقط الأنماط الزمنية لكنه يتطلب موارد حوسبة أكبر وبيانات تدريب أكثر.

الفئة الثالثة، التي ينتمي إليها المشروع الحالي، تعتمد على خط أنابيب متعدد المراحل: كشف ثم تتبع ثم استخراج ميزات ثم تحليل سلوك. يُوفّر هذا النهج مرونة عالية لأن كل مرحلة يُمكن تحسينها أو استبدالها بشكل مستقل، كما يُتيح فهماً أفضل لكيفية وصول النظام لقراراته.

من حيث السلوكيات المستهدفة، ركّزت معظم الدراسات على: حركات الرأس والالتفات، واتجاه النظر، وكشف الأجهزة الإلكترونية، والتواصل اللفظي. قليل من الدراسات تناولت حركات اليد بشكل مفصّل، وهو ما يُميّز المشروع الحالي بتضمين إشارة "اليد نحو الوجه".

2.8.5 الفجوات والتحديات القائمة

كشفت مراجعة الدراسات السابقة عن عدة فجوات وتحديات لم تُعالج بشكل كافٍ. التحدي الأول هو العمل في بيئات واقعية غير مضبوطة، حيث أُجريت معظم التجارب في ظروف مثالية (إضاءة جيدة، زوايا كاميرا مناسبة، عدد محدود من الأشخاص) لا تعكس واقع قاعات الامتحان الفعلية.

التحدي الثاني هو قابلية التوسع، حيث قليل من الأنظمة أثبتت قدرتها على التعامل مع قاعات كبيرة تضم عشرات أو مئات الطلاب في وقت واحد. معظم الأنظمة صُممت لمراقبة طالب واحد أو عدد صغير من الطلاب.

التحدي الثالث هو التكامل مع سير عمل المراقبين البشريين. كثير من الأنظمة تُنتج تنبيهات دون توفير السياق الكافي للمراقب لاتخاذ قرار، أو تُغرقه بتنبيهات كثيرة تُفقد الثقة في النظام.

التحدي الرابع هو غياب مجموعات بيانات معيارية للتقييم. تستخدم كل دراسة بياناتها الخاصة، مما يُصعب المقارنة الموضوعية بين الأنظمة المختلفة.

2.8.6 موقع المشروع الحالي

يُعالج المشروع الحالي عدة جوانب من الفجوات المذكورة. من حيث البنية، يعتمد على خط أنابيب معياري يستخدم أدوات مفتوحة المصدر ومثبتة الفعالية (YOLOv8 للكشف، DeepSORT للتتبع، MediaPipe لاستخراج الميزات)، مما يُسهل الصيانة والتطوير المستقبلي.

من حيث قابلية التوسع، صُمم النظام للتعامل مع عدة أشخاص في وقت واحد مع الحفاظ على هوية كل منهم، وتُعالج كل شخص بشكل مستقل مما يُتيح التوازي.

من حيث دعم المراقب، يُوفّر النظام مخرجات متعددة: فيديو مُشروح يُظهر حالة كل طالب، ومقاطع قصيرة للأحداث عالية الخطورة، وتقرير PDF مفصل، ولوحة مراقبة تفاعلية. هذا التنوع يُمكن المراقب من اختيار مستوى التفصيل المناسب لاحتياجاته.

من حيث الشفافية، يُنتج النظام سجلات تفصيلية لكل مرحلة من مراحل المعالجة، مما يُتيح فهم كيفية وصوله لأي تنبيه أو تقييم. هذه الشفافية ضرورية لبناء الثقة في النظام ولتمكين مراجعة قراراته.

يبقى المشروع الحالي نظاماً لدعم القرار لا نظاماً للحكم الآلي، متسقاً مع التوجه الأخلاقي الذي أكدت عليه الدراسات الحديثة في هذا المجال. الهدف هو مساعدة المراقب البشري على أداء مهمته بكفاءة أعلى، لا استبداله.

الفصل الثالث: بيئة العمل والأدوات

3.1 مقدمة الفصل

يُشكّل تجهيز بيئة العمل المناسبة الأساس الذي يُبنى عليه أي مشروع برمجي ناجح، وخاصة في مشاريع الذكاء الاصطناعي والرؤية الحاسوبية التي تتطلب تكاملاً دقيقاً بين مكونات برمجية متعددة ومتطلبات عتادية خاصة. يهدف هذا الفصل إلى توثيق البيئة التقنية الكاملة للمشروع، بما يُمكن القارئ من فهم المتطلبات اللازمة لتشغيل النظام، ويُوفّر مرجعاً للمطورين الراغبين في إعادة إنتاج البيئة أو البناء عليها.

يتطلب نظام Proctor AI تضافر عدة تقنيات متخصصة تعمل معاً في انسجام: نماذج تعلم عميق للكشف عن الأشخاص، وخوارزميات تتبع متقدمة، وأدوات استخراج ميزات للوجه والجسم، ومحركات معالجة فيديو، وواجهات مستخدم تفاعلية. كل من هذه المكونات له متطلباته الخاصة من حيث إصدارات المكتبات والتوافقات والموارد الحاسوبية.

يتناول هذا الفصل ستة محاور أساسية تُغطي جميع جوانب البيئة التقنية. يبدأ المحور الأول بوصف البيئة البرمجية الأساسية: لغة Python المستخدمة وإصدارها، وكيفية إنشاء بيئة افتراضية معزولة، وآلية إدارة الحزم والمكتبات. يستعرض المحور الثاني المكتبات والأطر المستخدمة في المشروع بالتفصيل، مع شرح دور كل مكتبة ومبررات اختيارها.

يُحدد المحور الثالث متطلبات العتاد اللازمة لتشغيل النظام بكفاءة: المعالج والذاكرة، ووحدة معالجة الرسومات، ومتطلبات الكاميرات وجودة الفيديو. يصف المحور الرابع هيكل المشروع: كيفية تنظيم المجلدات، والوحدات البرمجية الرئيسية، وملفات الإعدادات ودورها. يتناول المحور الخامس تجهيز البيانات: الفيديوهات المستخدمة في الاختبار والتطوير، وصيغ الملفات المعتمدة لتبادل البيانات بين مكونات النظام.

يُراعي التوثيق في هذا الفصل مبدأ قابلية إعادة الإنتاج (Reproducibility)، حيث تُحدّد إصدارات جميع المكونات بدقة، وتُوصف خطوات التجهيز بوضوح كافٍ لتمكين أي مطور من بناء بيئة مطابقة. هذا المبدأ جوهرى في المشاريع العلمية والتقنية، إذ يُتيح للآخرين التحقق من النتائج والبناء على العمل المُنجز.

تجدر الإشارة إلى أن البيئة الموصوفة في هذا الفصل تُمثّل الإعداد الذي طُوّر واختُبر عليه النظام. قد تختلف بعض التفاصيل عند التثبيت على أنظمة تشغيل مختلفة أو عتاد مختلف، لكن المبادئ العامة والمتطلبات الأساسية تبقى ثابتة. حيثما تكون هناك اعتبارات خاصة بنظام تشغيل معين أو تكوين عتادي محدد، تُذكر صراحةً في السياق المناسب.

3.2 البيئة البرمجية

تُشكل البيئة البرمجية الطبقة الأساسية التي تعمل فوقها جميع مكونات النظام. يتناول هذا القسم العناصر الثلاثة الرئيسية للبيئة البرمجية: لغة البرمجة المستخدمة وإصدارها، والبيئة الافتراضية التي تعزل المشروع عن بقية النظام، وآلية إدارة الحزم والمكتبات التي تضمن التوافق والاتساق.

3.2.1 لغة Python وإصدارها

اختيرت لغة Python كلغة التطوير الرئيسية للمشروع، وتحديدًا الإصدار 3.10. يعكس هذا الاختيار عدة اعتبارات تقنية وعملية جعلت Python الخيار الأمثل لمشاريع الذكاء الاصطناعي والرؤية الحاسوبية. تتميز Python بنظام بيئي غني جداً في مجال التعلم الآلي والتعلم العميق. جميع الأطر والمكتبات الرئيسية في هذا المجال (PyTorch, TensorFlow, scikit-learn) مكتوبة بـ Python أو توفر واجهات Python أصلية. هذا يعني أن المطور يستطيع الوصول لأحدث الخوارزميات والنماذج دون الحاجة للتعامل مع لغات منخفضة المستوى. تُوفّر Python أيضاً إنتاجية عالية في التطوير. صياغتها الواضحة والمقروءة تُسرّع كتابة الشيفرة ومراجعتها. الطبيعة التفسيرية (Interpreted) تُتيح تجريباً سريعاً ودورات تطوير قصيرة. هذا مهم خاصة في مراحل البحث والتطوير حيث تتكرر التجارب والتعديلات.

اختير الإصدار 3.10 تحديداً لعدة أسباب. أولاً، هو إصدار مستقر ومدعوم على المدى الطويل. ثانياً، يدعم جميع المكتبات المطلوبة للمشروع دون مشاكل توافق. ثالثاً، يتضمن تحسينات في الأداء مقارنة بالإصدارات الأقدم. رابعاً، يدعم ميزات لغوية حديثة مثل Pattern Matching و Type Hints المحسّنة.

للتحقق من إصدار Python المُثبّت على النظام، يُستخدم الأمر `python --version` أو `python3 --version` في الطرفية. يجب أن يُظهر الناتج إصداراً x.3.10 أو أحدث. في حال عدم توفر الإصدار المطلوب، يُمكن تثبيته من الموقع الرسمي python.org أو باستخدام مدير حزم النظام.

3.2.2 البيئة الافتراضية (Virtual Environment)

البيئة الافتراضية هي مجلد معزول يحتوي على تثبيت Python مستقل مع مكتباته الخاصة، منفصلاً عن تثبيت Python العام على النظام. استخدام البيئات الافتراضية ممارسة أساسية في تطوير Python، وتكتسب أهمية خاصة في مشاريع التعلم العميق حيث تتعدد المكتبات وتتشابك تبعياتها.

تُوفّر البيئة الافتراضية عدة فوائد جوهرية. الفائدة الأولى هي العزل: كل مشروع له مكتبته الخاصة بإصداراتها المحددة، دون تعارض مع مشاريع أخرى على نفس الجهاز. قد يتطلب مشروع ما إصداراً قديماً من مكتبة بينما يتطلب مشروع آخر إصداراً حديثاً، والبيئات الافتراضية تُتيح هذا التعايش.

الفائدة الثانية هي قابلية إعادة الإنتاج: يُمكن تصدير قائمة المكتبات المثبتة في البيئة مع إصداراتها الدقيقة، ثم إعادة إنشاء بيئة مطابقة على جهاز آخر. هذا ضروري للعمل الجماعي ولنشر المشروع.

الفائدة الثالثة هي النظافة: عند انتهاء العمل على مشروع، يُمكن حذف بيئته الافتراضية بالكامل دون التأثير على النظام. هذا يمنع تراكم المكتبات غير المستخدمة.

يُوفّر Python أداة مدمجة لإنشاء البيئات الافتراضية تُسمى `venv`. لإنشاء بيئة افتراضية جديدة للمشروع، يُنفَّذ الأمر `python -m venv venv` في مجلد المشروع. ينشئ هذا مجلداً باسم `venv` يحتوي على البيئة الافتراضية.

لتفعيل البيئة الافتراضية على نظام Windows، يُنفَّذ الأمر `venv\Scripts\activate`. على أنظمة Linux و macOS، يُنفَّذ الأمر `source venv/bin/activate`. بعد التنغيع، تتغير مُحثّة الطرفية لتُظهر اسم البيئة، وتُصبح جميع أوامر Python و pip تعمل ضمن البيئة المعزولة.

يُمكن أيضاً استخدام أداة `conda` كبديل لـ `venv`، وهي شائعة في مجتمع علوم البيانات. تُوفّر `conda` إدارة أشمل تتضمن حزمًا غير Python أيضاً، لكنها أثقل حجماً. للمشروع الحالي، `venv` كافية وأخف.

3.2.3 إدارة الحزم والمكتبات

تُستخدم أداة `pip` (Python Package Installer) لإدارة الحزم والمكتبات في المشروع. هي الأداة القياسية في Python لتثبيت الحزم من مستودع (PyPI (Python Package Index) الذي يحتوي على مئات الآلاف من الحزم المتاحة.

يعتمد المشروع على ملف `requirements.txt` لتوثيق جميع المكتبات المطلوبة مع إصداراتها. هذا الملف النصي يحتوي على سطر لكل مكتبة بصيغة اسم_المكتبة==الإصدار. عند تثبيت المشروع على بيئة جديدة، يكفي تنفيذ الأمر `pip install requirements.txt -r` لتثبيت جميع المكتبات دفعة واحدة.

تحديد الإصدارات بشكل صريح (Version Pinning) أمر بالغ الأهمية في مشاريع التعلم العميق. تحديث مكتبة إلى إصدار جديد قد يُغيّر سلوكها أو يكسر التوافق مع مكتبات أخرى. بتثبيت الإصدارات، نضمن أن البيئة ستعمل بنفس الطريقة على أي جهاز وفي أي وقت.

يتضمن ملف requirements.txt الأساسي للمشروع المكتبات التالية مع إصداراتها: torch للتعلم العميق، و ultralytics لنموذج YOLOv8، و opencv-python لمعالجة الفيديو، و mediapipe لاستخراج الميزات، و dash و plotly لواجهة المستخدم، و numpy و pandas للعمليات الحسابية ومعالجة البيانات، و PyYAML و omegaconf لقراءة ملفات الإعدادات، و reportlab لتوليد تقارير PDF.

للمطورين الراغبين في المساهمة في تطوير المشروع، يُوفّر ملف إضافي requirements-dev.txt يتضمن أدوات التطوير: pytest للاختبارات، و black و flake8 لتنسيق الشيفرة وفحصها، و mypy للتحقق من الأنواع.

لتنصيب المشروع، تُتبع الخطوات التالية بالترتيب: أولاً، استنساخ مستودع المشروع أو تحميله. ثانياً، الانتقال إلى مجلد المشروع. ثالثاً، إنشاء بيئة افتراضية بالأمر python -m venv venv. رابعاً، تفعيل البيئة الافتراضية. خامساً، تنصيب المكتبات بالأمر pip install -r requirements.txt. بعد هذه الخطوات، تكون البيئة جاهزة لتشغيل النظام.

يُنصح بتحديث pip نفسها قبل تنصيب المكتبات بالأمر pip install --upgrade pip، لضمان الحصول على أحدث ميزات التنصيب وحل التبعية. كما يُنصح بتنصيب wheel بالأمر pip install wheel لتسريع تنصيب الحزم التي توفر توزيعات مُجمّعة مسبقاً.

3.3 المكتبات والأطر المستخدمة

يعتمد المشروع على مجموعة متكاملة من المكتبات والأطر البرمجية، كل منها يُقدّم وظيفة محددة ضمن منظومة النظام. يستعرض هذا القسم المكتبات الرئيسية الخمس المستخدمة، مع شرح تفصيلي لدور كل مكتبة ومبررات اختيارها وكيفية استخدامها في المشروع.

3.3.1 PyTorch للتعلم العميق

يُعدّ PyTorch إطار العمل الأساسي للتعلم العميق في المشروع [1]. طوّره شركة Meta (فيسبوك سابقاً) ويحظى بشعبية واسعة في الأوساط البحثية والصناعية. يُستخدم PyTorch في المشروع كمحرك أساسي تعمل عليه نماذج الكشف والتتبع.

يتميز PyTorch بعدة خصائص تجعله الخيار المفضل لمشاريع الرؤية الحاسوبية. الخاصية الأولى هي التنفيذ الفوري (Eager Execution) الذي يُتيح تنفيذ العمليات فوراً دون الحاجة لبناء رسم حسابي مسبقاً، مما يُسهّل التصحيح والتجريب. الخاصية الثانية هي الدعم المتميز لوحدات معالجة الرسومات GPU، مع انتقال سلس للعمليات بين CPU و GPU. الخاصية الثالثة هي المجتمع الكبير والنشط الذي يُوفّر نماذج مُدرّبة مسبقاً وأدوات ودروس تعليمية.

في المشروع الحالي، لا يُستخدم PyTorch مباشرة في الشيفرة بشكل كبير، لكنه يعمل كأساس تعمل عليه مكتبات أخرى. نماذج YOLOv8 مبنية على PyTorch، ونموذج استخراج الميزات المظهرية في DeepSORT يستخدم PyTorch. لذلك فإن تثبيت PyTorch بشكل صحيح (مع دعم CUDA إن توفر GPU) شرط أساسي لعمل النظام.

يُنَبِّت PyTorch عادةً باستخدام أمر pip مُخصص يعتمد على نظام التشغيل وتوفر GPU. للأنظمة التي تملك بطاقة NVIDIA مع دعم CUDA، يُستخدم إصدار PyTorch المُصمَّم لإصدار CUDA المُثَبَّت على النظام. للأنظمة التي لا تملك GPU متوافقة، يُنَبِّت إصدار CPU-only الذي يعمل لكن بسرعة أقل.

3.3.2 OpenCV لمعالجة الفيديو

مكتبة OpenCV (Open Source Computer Vision Library) هي المكتبة الأساسية لمعالجة الصور والفيديو في المشروع [5]. نشأت في مختبرات Intel وأصبحت المكتبة الأكثر استخداماً في مجال الرؤية الحاسوبية، مع دعم لأكثر من 2500 خوارزمية.

تُستخدم OpenCV في المشروع لعدة مهام أساسية. المهمة الأولى هي قراءة ملفات الفيديو إطاراً بإطار باستخدام فئة VideoCapture. تدعم OpenCV جميع صيغ الفيديو الشائعة (MP4, AVI, MOV, إلخ) من خلال تكاملها مع مكتبات الترميز مثل FFmpeg. المهمة الثانية هي تحويل صيغ الألوان، فالفيديو عادةً يُقرأ بصيغة BGR بينما تتطلب بعض النماذج صيغة RGB. المهمة الثالثة هي عمليات معالجة الصور الأساسية: تغيير الحجم، والاقتصاص، والتدوير.

تُستخدم OpenCV أيضاً في مرحلة توليد المخرجات. كتابة الفيديو التوضيحي تتم باستخدام فئة VideoWriter. رسم التعليقات على الإطارات (صناديق الإحاطة، والنصوص، والأشرطة) يتم باستخدام دوال الرسم مثل rectangle و putText و line. اقتصاص مقاطع الأدلة من الفيديو الأصلي يستخدم أيضاً VideoCapture و VideoWriter.

تُنَبِّت OpenCV في Python باستخدام حزمة opencv-python التي تتضمن الوحدات الأساسية. تتوفر أيضاً حزمة opencv-contrib-python التي تتضمن وحدات إضافية، لكنها غير مطلوبة للمشروع الحالي. التثبيت بسيط عبر الأمر `pip install opencv-python`.

3.3.3 Ultralytics لـ YOLOv8

مكتبة Ultralytics هي المكتبة الرسمية لتشغيل نماذج YOLO بإصداراتها الحديثة، وخاصة YOLOv8 المستخدم في المشروع [1]. تُوفّر المكتبة واجهة برمجية بسيطة وقوية للكشف عن الأشياء والتصنيف وتقدير الوضعية وغيرها من المهام.

تتميز مكتبة Ultralytics بسهولة الاستخدام الاستثنائية. تحميل نموذج وتشغيله يتطلب سطرين فقط من الشيفرة: سطر لإنشاء كائن النموذج وسطر لتشغيله على صورة أو فيديو. هذه البساطة لا تأتي على حساب المرونة، إذ تُتيح المكتبة ضبط جميع المعاملات المهمة: عتبة الثقة، وعتبة NMS، وحجم الصورة، والفئات المستهدفة، وغيرها.

تُوفّر المكتبة عدة إصدارات من YOLOv8 تختلف في الحجم والسرعة والدقة: yolov8n (nano) الأصغر والأسرع، و yolov8s (small)، و yolov8m (medium)، و yolov8l (large)، و yolov8x (extra large) الأكبر والأدق. يستخدم المشروع إصدار yolov8s الذي يُوفّر توازناً جيداً بين السرعة والدقة.

النماذج مُدرّبة مسبقاً على مجموعة بيانات COCO التي تتضمن 80 فئة من الأشياء الشائعة. في المشروع، نستخدم فقط فئة "person" (معرف 0) ونتجاهل بقية الفئات. يُمكن تحديد الفئات المستهدفة عند تشغيل النموذج لتحسين الأداء.

تُنَبِّت المكتبة بالأمر `pip install ultralytics`. عند أول استخدام لنموذج معين، تقوم المكتبة بتحميله تلقائياً وحفظه محلياً للاستخدامات اللاحقة. يُمكن أيضاً تحميل النماذج يدوياً مسبقاً لضمان توفرها دون اتصال بالإنترنت.

MediaPipe 3.3.4 لاستخراج الميزات

MediaPipe هي مكتبة مفتوحة المصدر من Google لبناء خطوط معالجة الوسائط المتعددة [3]. تُوفّر حلاً جاهزة ومُحسّنة لمهام شائعة في الرؤية الحاسوبية، مع أداء عالٍ يُتيح المعالجة في الوقت الحقيقي.

يستخدم المشروع حلّين من MediaPipe: Face Mesh لاستخراج النقاط المعلمية للوجه، و Pose لاستخراج النقاط المعلمية للجسم. حل Face Mesh يُوفّر 468 نقطة ثلاثية الأبعاد تُغطي جميع ملامح الوجه بدقة عالية. حل Pose يُوفّر 33 نقطة تُغطي الجسم من الرأس حتى القدمين، مع قدرة على تقدير مواقع النقاط غير المرئية.

تُستخدم هذه النقاط المعلمية لحساب إشارات السلوك: زاوية الرأس الأفقية (Yaw) والعمودية (Pitch) من نقاط الوجه، ودرجة النظر للأسفل من علاقة الرأس بالكتفين، ودرجة اليد نحو الوجه من المسافة بين المعصم والأنف. هذه الحسابات الهندسية تُحوّل النقاط الخام إلى مقاييس ذات دلالة سلوكية.

تتميز MediaPipe بكفاءة استثنائية. النماذج المستخدمة خفيفة الوزن ومُحسّنة للعمل على أجهزة متواضعة. يُمكنها معالجة عشرات الإطارات في الثانية حتى على CPU فقط. هذه الكفاءة حاسمة لأن استخراج الميزات يُجرى لكل شخص مُتتبع في كل إطار، فالعدد الإجمالي للمعالجات قد يكون كبيراً.

تُنَبِّت MediaPipe بالأمر `pip install mediapipe`. المكتبة مستقلة ولا تتطلب TensorFlow أو PyTorch، مما يُبسّط التبعيات. تتوفر إصدارات لأنظمة Windows و Linux و macOS، مع بعض الاختلافات في الأداء بينها.

3.3.5 Dash لواجهة المستخدم

Dash هو إطار عمل Python لبناء تطبيقات ويب تفاعلية، طوّرت شركة Plotly [4]. يُتيح Dash إنشاء واجهات مستخدم غنية بالكامل من شيفرة Python، دون الحاجة لكتابة HTML أو CSS أو JavaScript بشكل مباشر.

يُستخدم Dash في المشروع لبناء لوحة التحكم التفاعلية (Dashboard) التي تُتيح للمستخدمين استعراض نتائج التشغيلات ومشاهدة الفيديوها والاطلاع على الإحصائيات. اختير Dash لعدة أسباب: تكامله الطبيعي مع Python والمكتبات العلمية، ومرونته في بناء واجهات معقدة، والمجتمع النشط والتوثيق الجيد.

يعتمد Dash على عدة مكتبات تحت الغطاء: Flask كخادم ويب، و React لواجهة المستخدم، و Plotly للرسوم البيانية التفاعلية. المطور يتعامل مع واجهة Python فقط، والإطار يتولى توليد HTML و JavaScript تلقائياً.

تتضمن لوحة التحكم عدة صفحات: صفحة قائمة التشغيلات التي تعرض جميع الـ runs المُنفّذة مع معلوماتها الأساسية، و صفحة تفاصيل التشغيل التي تعرض إحصائيات ورسوم بيانية لتشغيل محدد، وإمكانية مشاهدة الفيديو التوضيحي مباشرة في المتصفح.

تُستخدم مكتبة dash-bootstrap-components لإضافة تصميم Bootstrap للوحة التحكم، مما يُوفّر مظهراً احترافياً وتصميماً متجاوباً يعمل على الشاشات المختلفة. تُستخدم أيضاً مكتبة plotly لإنشاء الرسوم البيانية التفاعلية: خطوط زمنية لدرجات المخاطر، ومخططات شريطية لتوزيع السلوكيات، ورسوم دائرية للنسب.

تُنشأ المكتبات بالأوامر: `pip install dash` للإطار الأساسي، و `pip install dash-bootstrap-components` لمكونات Bootstrap، و `pip install plotly` لمكتبة الرسوم البيانية. تشغيل لوحة التحكم يتم بتنفيذ `python -m dashboard.app` ثم فتح المتصفح على العنوان <http://localhost:8050>.

3.4 متطلبات العتاد

تُحدد متطلبات العتاد الحد الأدنى والموصى به من الموارد الحاسوبية اللازمة لتشغيل النظام بكفاءة. تختلف هذه المتطلبات بحسب حالة الاستخدام: المعالجة اللاحقة للفيديوهات المُسجّلة تتطلب موارد أقل من المعالجة المباشرة في الوقت الحقيقي. يتناول هذا القسم متطلبات المعالج والذاكرة، ووحدة معالجة الرسومات، ومتطلبات الكاميرات وجودة الفيديو.

3.4.1 المعالج والذاكرة

يُشكّل المعالج المركزي (CPU) العمود الفقري لتشغيل النظام، حتى في حالة توفر وحدة معالجة رسومات. عمليات عديدة تعمل حصرياً على CPU: قراءة الفيديو وفك ترميزه، وتحويلات الصور، والحسابات الهندسية، وإدارة البيانات، وكتابة المخرجات.

للحد الأدنى من الأداء المقبول في المعالجة اللاحقة، يُنصح بمعالج رباعي النوى على الأقل بتردد 2.5 جيجاهرتز أو أعلى. معالجات Intel Core i5 أو AMD Ryzen 5 من الأجيال الحديثة تُلبّي هذا المتطلب. الأداء المحدود على معالجات أضعف سيُترجم إلى بطء في المعالجة لكن النظام سيعمل.

للأداء الموصى به، خاصة مع المعالجة المباشرة أو الفيديوهات عالية الدقة، يُنصح بمعالج سداسي أو ثماني النوى بتردد 3.0 جيجاهرتز أو أعلى. معالجات Intel Core i7/i9 أو AMD Ryzen 7/9 تُوفّر أداءً ممتازاً. تعدد النوى يُفيد في تشغيل العمليات المتوازية والاستفادة من خيوط التنفيذ المتعددة في بعض المكتبات.

بالنسبة للذاكرة العشوائية (RAM)، الحد الأدنى المطلوب هو 8 جيجابايت. هذا يكفي لتحميل النماذج وتخزين إطارات الفيديو وتشغيل العمليات الأساسية. لكن مع فيديوهات طويلة أو عدد كبير من الأشخاص في المشهد، قد تُصبح الذاكرة عنق زجاجة.

الذاكرة الموصى بها هي 16 جيجابايت أو أكثر. هذا يُوفّر مساحة كافية لتحميل نماذج متعددة، وتخزين مؤقت للإطارات، وتشغيل لوحة التحكم وأدوات أخرى بالتوازي مع المعالجة. للاستخدام المكثف أو التطوير، 32 جيجابايت تُوفّر راحة إضافية.

استهلاك الذاكرة يتأثر بعدة عوامل: حجم نموذج YOLO المستخدم (yolov8n أخف من yolov8x)، وعدد الأشخاص المُتنبّعين (كل مسار يستهلك ذاكرة لتخزين حالته)، ودقة الفيديو (الإطارات عالية الدقة تستهلك ذاكرة أكثر)، وحجم التخزين المؤقت في بعض العمليات.

3.4.2 كرت الشاشة (GPU)

وحدة معالجة الرسومات (GPU) ليست إلزامية لتشغيل النظام، لكنها تُحسّن الأداء بشكل جذري. نماذج التعلم العميق (YOLOv8، نموذج الميزات المظهرية في DeepSORT) صُمّمت للاستفادة من التوازي الضخم الذي توفره GPU. على CPU فقط، هذه النماذج تعمل لكن بسرعة أقل بكثير.

للتشغيل بدون GPU، يُمكن للنظام معالجة الفيديو لكن بمعدل إطارات منخفض، ربما 2-5 إطارات في الثانية بحسب قوة المعالج. هذا مقبول للمعالجة اللاحقة حيث الوقت ليس حرجاً: معالجة فيديو مدته 10 دقائق قد تستغرق ساعة، لكن النتائج ستكون صحيحة.

للتشغيل مع GPU، يدعم النظام بطاقات NVIDIA المتوافقة مع CUDA. الحد الأدنى الموصى به هو بطاقة ذاكرة 4 جيجابايت VRAM ودعم CUDA 11.x أو أحدث. بطاقات مثل NVIDIA GTX 1650 أو RTX 3050 تُلبي هذا المتطلب وتُوفّر تحسيناً ملموساً في السرعة.

للأداء الأفضل، يُنصح بطاقة ذاكرة 8 جيجابايت VRAM أو أكثر. بطاقات مثل RTX 3060 أو RTX 3070 أو ما يعادلها تُتيح معالجة في الوقت الحقيقي (25+ إطار في الثانية) مع جميع مكونات النظام نشطة. ذاكرة VRAM الأكبر تُتيح أيضاً استخدام نماذج أكبر (yolov8l, yolov8x) للحصول على دقة أعلى.

لتفعيل دعم GPU، يجب تثبيت: تعريفات NVIDIA الحديثة، و CUDA Toolkit بالإصدار المتوافق مع PyTorch، و cuDNN لتسريع عمليات الشبكات العصبية. يجب أيضاً تثبيت إصدار PyTorch المُصمّم لإصدار CUDA المُثبت. يُمكن التحقق من تفعيل GPU في Python بتنفيذ `import torch; print(torch.cuda.is_available())` الذي يجب أن يُعيد `.True`.

بطاقات AMD GPU غير مدعومة بشكل كامل حالياً في المكتبات المستخدمة. بطاقات Intel المدمجة لا تُقدّم تحسيناً يُذكر. لذلك، إذا كان الأداء أولوية، فإن بطاقات NVIDIA هي الخيار الواضح.

3.4.3 الكاميرات وجودة الفيديو

جودة الفيديو المُدخل تؤثر بشكل مباشر على دقة الكشف واستخراج الميزات. فيديو منخفض الجودة يُنتج نتائج أقل دقة حتى مع أفضل الخوارزميات. يُحدد هذا القسم المواصفات المطلوبة للفيديو ومتطلبات الكاميرات.

بالنسبة لدقة الفيديو (Resolution)، الحد الأدنى المقبول هو p720 (720×1280 بكسل). هذه الدقة تُوفّر تفاصيل كافية لكشف الأشخاص والتعرف على ملامح الوجه الأساسية عندما يكون الشخص قريباً نسبياً من الكاميرا. الدقة الموصى بها هي p1080 (1080×1920 بكسل) أو أعلى. هذه الدقة تُوفّر تفاصيل أفضل تحسّن دقة استخراج النقاط المعلمية، خاصة للأشخاص البعيدين عن الكاميرا.

بالنسبة لمعدل الإطارات (Frame Rate)، المعدل القياسي في كاميرات المراقبة هو 25-30 إطاراً في الثانية، وهو كافٍ تماماً للمشروع. معدلات أعلى (60 fps) لا تُضيف فائدة تُذكر لتحليل السلوكي وتُضاعف متطلبات المعالجة. معدلات أقل (15 fps) مقبولة لكنها قد تُفوّت بعض الحركات السريعة.

بالنسبة لصيغة الفيديو (Format)، يدعم النظام جميع الصيغ الشائعة التي تدعمها OpenCV عبر FFmpeg: MP4, AVI, MOV, MKV وغيرها. صيغة MP4 مع ترميز H.264 هي الأكثر شيوعاً وتوافقاً. الترميزات الحديثة مثل H.265/HEVC مدعومة أيضاً لكن قد تتطلب تثبيت مكتبات إضافية على بعض الأنظمة.

بالنسبة للإضاءة، تُعدّ الإضاءة الجيدة والمستقرة من أهم عوامل جودة الكشف. الإضاءة الخافتة تُزيد الضوضاء في الصورة وتُصعّب كشف الملامح الدقيقة. الإضاءة المتقلبة (مثل الشمس المتسللة من النوافذ) تُسبب تغيرات مفاجئة تُربك الخوارزميات. يُفضّل الإضاءة الصناعية المنتظمة والكافية في قاعات الامتحان.

بالنسبة لزاوية الكاميرا وموقعها، الزاوية المثالية هي من الأمام والأعلى قليلاً، بحيث تُظهر وجوه الطلاب وأجسادهم العلوية بوضوح. الكاميرات الجانبية تُقلل من دقة كشف اتجاه الرأس. الكاميرات الخلفية لا تُوفّر معلومات كافية عن الوجه. يجب أن تُغطي الكاميرا عدداً معقولاً من الطلاب (5-15) لضمان دقة كافية لكل شخص. قاعة كبيرة قد تتطلب كاميرات متعددة. بالنسبة لاستقرار الكاميرا، يُفترض النظام أن الكاميرا ثابتة. حركة الكاميرا تُربك خوارزمية التتبع وتُسبب فقدان المسارات. في حال استخدام كاميرا متحركة (PTZ)، يجب معالجة كل وضع ثابت على حدة.

3.5 هيكل المشروع

يتبع المشروع تنظيماً هرمياً واضحاً يفصل بين المكونات المختلفة ويُسهّل الصيانة والتطوير. يتناول هذا القسم تنظيم المجلدات العام، والوحدات البرمجية الرئيسية، وملفات الإعدادات ودورها في ضبط سلوك النظام.

3.5.1 تنظيم المجلدات

يتكون المشروع من عدة مجلدات رئيسية، كل منها يضم نوعاً محدداً من الملفات ويخدم غرضاً واضحاً. هذا الفصل بين الاهتمامات (Separation of Concerns) يُسهّل فهم البنية والعمل على أجزاء محددة دون التأثير على البقية.

- **مجلد src:** يحتوي على الكود المصدري الأساسية للنظام. داخله مجلد `proctor_ai` الذي يُشكّل الحزمة الرئيسية للمشروع. هذا الفصل بين `src` والمجلدات الأخرى يتبع الممارسات المعيارية في مشاريع Python ويُسهّل التعبئة والتوزيع.
- **مجلد tools:** يحتوي على سكريبتات الأدوات المساعدة التي تُنفَّذ من سطر الأوامر. كل أداة تُنجز مهمة محددة: `detect_behaviors.py` لكشف السلوكيات، و `compute_decisions.py` لحساب المخاطر، و `render_demo_video.py` لإنشاء الفيديو التوضيحي، و `extract_top_clips.py` لاستخراج مقاطع الأدلة، و `generate_report.py` لتوليد التقرير، وغيرها.

- **مجلد configs:** يحتوي على ملفات الإعدادات بصيغة YAML. يتضمن مجلدات فرعية: experiments للإعدادات التجريبية المختلفة، و system لإعدادات النظام العامة وقواعد القرار، و policy لسياسات التنبيهات، وغيرها. هذا التنظيم يُتيح إدارة إعدادات متعددة لسيناريوهات مختلفة.
- **مجلد dashboard:** يحتوي على شيفرة لوحة التحكم التفاعلية المبنية بـ Dash. يتضمن ملف التطبيق الرئيسي app.py، ومجلدات فرعية للصفحات pages والمكونات المشتركة components والأدوات المساعدة utils والأصول الثابتة assets.
- **مجلد data:** يحتوي على البيانات المستخدمة في المشروع. يتضمن مجلدات فرعية: raw للفيديوهات الأصلية، و students لقاعدة بيانات الطلاب (إن استُخدمت ميزة التعرف على الهوية)، و samples للأمثلة والعينات.
- **مجلد outputs:** يحتوي على مخرجات تشغيل النظام. كل تشغيل (run) يُنشئ مجلداً فرعياً خاصاً به يتضمن جميع نتائجه: الميزات المستخرجة، والأحداث المكتشفة، ودرجات المخاطر، والتنبيهات، والفيديو التوضيحي، والتقارير.
- **مجلد tests:** يحتوي على اختبارات النظام. يتضمن اختبارات وحدة (unit tests) لاختبار المكونات الفردية، واختبارات تكامل (integration tests) لاختبار تفاعل المكونات معاً.
- **مجلد docs:** يحتوي على التوثيق بصيغة Markdown. يتضمن وثائق الاستخدام والتنصيب والتطوير والإسهام.
- **ملفات الجذر:** يحتوي جذر المشروع على ملفات مهمة: README.md للتعريف بالمشروع، و requirements.txt للمكتبات المطلوبة، و pyproject.toml لبيانات المشروع، و Makefile لأتمتة المهام الشائعة.

3.5.2 الوحدات الرئيسية

يتوزع الكود الأساسي في مجلد src/proctor_ai على عدة وحدات (modules)، كل منها يُغلف وظيفة محددة ويُوفّر واجهة واضحة للتفاعل معها.

- **وحدة perception:** تتولى مهام الإدراك البصري: كشف الأشخاص وتتبعهم. تتضمن مجلدين فرعيين: detector للكاشفات (يحتوي على yolov8.py و factory.py)، و tracker للمتتبعات (يحتوي على deepsort.py و bytetrack.py). استخدام نمط المصنع (Factory Pattern) يُتيح التبديل السهل بين الكاشفات والمتتبعات المختلفة.

- **وحدة features:** تتولى استخراج الميزات من كل شخص مُتتَبَع. الملف الرئيسي `mediapipe_signals.py` يحتوي على منطق استخراج النقاط المعلمية وحساب الإشارات السلوكية. يُغَلَّف التفاعل مع مكتبة `MediaPipe` ويُوفَّر واجهة بسيطة لبقية النظام.
- **وحدة decisions:** تتولى منطق اتخاذ القرار وتقييم المخاطر. تتضمن عدة ملفات: `risk_scoring.py` لحساب درجة المخاطر باستخدام الأوزان و `EWMA`، و `decision_engine.py` لمحرك القرارات وتصنيف مستويات الخطورة، و `alert_manager.py` لإدارة التنبيهات مع `Cooldown` و `Hold Time`.
- **وحدة pipeline:** تتولى تنسيق خط المعالجة وربط المراحل المختلفة. ملف `runner.py` يحتوي على المنفذ الرئيسي الذي يقرأ الفيديو ويُمرر الإطارات عبر مراحل الكشف والتتبع واستخراج الميزات.
- **وحدة config:** تتولى تحميل ملفات الإعدادات ودمجها مع القيم الافتراضية. تستخدم مكتبة `OmegaConf` للتعامل مع ملفات `YAML` وتوفير وصول مرّن للإعدادات.
- **وحدة core:** تحتوي على وظائف أساسية مشتركة: التسجيل (`logging`)، وإدارة سياق التشغيل (`run context`)، والبذور العشوائية (`random seeds`) لضمان قابلية إعادة الإنتاج.
- **وحدة zones:** تتولى إدارة المناطق في القاعة وربطها بالمقاعد. هذه وحدة اختيارية تُستخدم عند الحاجة لتحديد مواقع الطلاب في القاعة.

3.5.3 ملفات الإعدادات (YAML)

يعتمد المشروع بشكل كبير على ملفات الإعدادات بصيغة `YAML` لضبط سلوك النظام دون تعديل الكود الأساسي. هذا الفصل بين الكود والإعدادات يُسهّل تخصيص النظام لسيناريوهات مختلفة ويُتيح التجريب السريع بمعاملات متنوعة. صيغة `YAML` اختيرت لعدة أسباب: سهولة القراءة والكتابة للبشر، ودعم التعليقات التوضيحية، والتركيب الهرمي الطبيعي، والتوافق الجيد مع `Python`. ملفات `YAML` أكثر قابلية للقراءة من `JSON` وأقل عرضة للأخطاء.

- **ملف التجربة (Experiment Config):** يُحدد إعدادات تشغيل معين. مثال: `configs/experiments/simple_deepsort.yaml`. يتضمن: اسم التجربة، وإعدادات النظام العامة (الحد الأقصى للإطارات، معدل الإطارات المفترض)، وإعدادات نموذج الكشف (اسم النموذج، عتبة الثقة، عتبة `NMS`)، وإعدادات المتتبع (الخوارزمية، المعاملات)، وإعدادات استخراج الميزات (تفعيل `MediaPipe`، نمط `ROI`).

- **ملف قواعد القرار (Decision Rules):** يُحدد معاملات نظام القرار. مثال: `configs/system/decision_rules.yaml`. يتضمن: مستويات المخاطر وعتباتها، وعتبات كشف السلوكيات (ON و OFF لكل سلوك)، وأوزان السلوكيات في حساب المخاطر، ومعاملات EWMA والنوافذ الزمنية، وإعدادات التنبيهات (عتبات التفعيل، Cooldown، Hold Time).
- **ملف سياسة التنبيهات (Alert Policy):** يُحدد سياسات إصدار التنبيهات. يتضمن: الحد الأقصى للتنبيهات لكل شخص، وفترات التهدة بين التنبيهات، وأولويات مستويات الخطورة.
- **آلية تحميل الإعدادات:** عند تشغيل النظام، تُحمّل الإعدادات من الملف المحدد وتُدمج مع القيم الافتراضية. هذا يعني أن ملف الإعدادات لا يحتاج لتحديد جميع المعاملات، بل فقط تلك التي تختلف عن الافتراضي. أي معامل غير محدد يأخذ قيمته الافتراضية من النظام.
- **حفظ الإعدادات المُحلّلة:** عند كل تشغيل، تُحفظ نسخة من الإعدادات الفعلية المستخدمة (بعد دمج القيم الافتراضية) في ملف `config_resolved.json` داخل مجلد التشغيل. هذا يُوثّق بدقة المعاملات التي استُخدمت، مما يُسهّل إعادة إنتاج النتائج أو فهم سبب سلوك معين.
- **تجاوز الإعدادات من سطر الأوامر:** يدعم النظام تجاوز إعدادات محددة من سطر الأوامر دون تعديل ملف YAML. هذا مفيد للتجريب السريع بمعاملات مختلفة. مثلاً:

```
proctor-ai run --config config.yaml models.person_detector.conf=0.5
```

لتغيير عتبة الثقة في تشغيل واحد.

3.6 تجهيز البيانات

تُشكّل البيانات المادة الخام التي يعمل عليها النظام، سواء الفيديوهات المُدخلة التي تُحلّل أو البيانات الوسيطة والمخرجات التي تُنتج خلال المعالجة. يتناول هذا القسم الفيديوهات المستخدمة في التطوير والاختبار، وصيغ الملفات المعتمدة لتخزين البيانات وتبادلها بين مكونات النظام.

3.6.1 الفيديوهاست المستخدمة

تُخزن الفيديوهاست الأصلية في مجلد `data/raw`. لأغراض التطوير والاختبار، استُخدمت فيديوهاست تحاكي بيئة قاعة الامتحان، تتضمن أشخاصاً يؤدون سلوكيات متنوعة (طبيعية ومشبوهة) لاختبار قدرات النظام على الكشف والتمييز. عند اختيار فيديو للتحليل، يجب مراعاة عدة معايير. المعيار الأول هو التوافق مع متطلبات الجودة المذكورة سابقاً: دقة 720p على الأقل، ومعدل إطارات 25-30 fps، وإضاءة كافية ومستقرة. المعيار الثاني هو ملائمة زاوية التصوير: يجب أن تُظهر الكاميرا وجوه الأشخاص وأجسادهم العلوية. المعيار الثالث هو مدة الفيديو المعقولة: فيديوهاست قصيرة جداً (ثوانٍ) لا تُتيح اختبار التتبع المستمر، وفيديوهاست طويلة جداً (ساعات) تستهلك وقتاً ومساحة تخزين كبيرين. قبل استخدام فيديو للتحليل، يُنصح بفحصه يدوياً للتأكد من: عدم وجود تلف أو انقطاع في الملف، ووضوح الصورة وثبات الكاميرا، ووجود أشخاص مرئيين بوضوح في جزء كافٍ من الفيديو. بعض الفيديوهاست قد تبدأ أو تنتهي بمشاهد فارغة (قبل دخول الطلاب أو بعد خروجهم) يُمكن قصّها لتوفير وقت المعالجة. يدعم النظام تحديد نطاق معين من الفيديو للمعالجة دون الحاجة لقصّه فعلياً. يُمكن تحديد إطار البداية وإطار النهاية في الإعدادات أو من سطر الأوامر. هذا مفيد للتركيز على جزء معين من فيديو طويل أو لتقسيم المعالجة إلى أجزاء. عند العمل مع فيديوهاست حقيقية من قاعات امتحان فعلية، يجب مراعاة اعتبارات الخصوصية والأخلاقيات. يجب الحصول على الموافقات اللازمة قبل تسجيل أو استخدام مثل هذه الفيديوهاست. يُفضّل استخدام فيديوهاست تمثيلية (مع ممثلين يُحاكون سيناريوهاست الامتحان) للتطوير والاختبار، وقصر استخدام الفيديوهاست الحقيقية على بيئة الإنتاج مع الضوابط المناسبة.

3.6.2 صيغة الملفات (JSONL)

يعتمد النظام بشكل أساسي على صيغة (JSON Lines) JSONL لتخزين البيانات المُهيكلية. في هذه الصيغة، كل سطر من الملف هو كائن JSON مستقل وصالح بذاته. هذا يختلف عن صيغة JSON التقليدية التي تُخزن الملف كاملاً ككائن واحد أو مصفوفة واحدة. توفر صيغة JSONL عدة مزايا جعلتها الخيار المفضل للمشروع. الميزة الأولى هي القراءة التدفقية (Streaming): يُمكن قراءة الملف سطرًا بسطر ومعالجة كل سجل فور قراءته، دون الحاجة لتحميل الملف كاملاً في الذاكرة. هذا حاسم للملفات الكبيرة التي قد تحتوي ملايين السجلات. الميزة الثانية هي الإضافة السهلة (Append): يُمكن إضافة سجلات جديدة لنهاية الملف دون الحاجة لإعادة كتابته كاملاً. هذا مثالي للكتابة المستمرة أثناء المعالجة، حيث يُكتب كل سجل فور إنتاجه.

الميزة الثالثة هي المرونة في التعامل مع الأخطاء: إذا تلف جزء من الملف أو توقفت المعالجة فجأة، تبقى السجلات السابقة صالحة ويُمكن قراءتها. في JSON التقليدي، أي تلف يُفسد الملف كاملاً.

الميزة الرابعة هي التوافق مع أدوات معالجة النصوص: يُمكن استخدام أدوات مثل grep و head و tail و wc لفحص ملفات JSONL ومعالجتها، مما يُسهّل الاستكشاف والتصحيح.

تُنتج مراحل المعالجة المختلفة ملفات JSONL متعددة، كل منها يحتوي نوعاً محدداً من البيانات:

- **ملف features/mediapipe.jsonl:** يحتوي الميزات المستخرجة لكل شخص في كل إطار. كل سجل يتضمن: رقم الإطار، والطابع الزمني بالمللي ثانية، ومعرّف المسار، وصندوق الإحاطة، وعلامات الكشف (هل كُشف الوجه؟ الجسم؟)، ودرجات الإشارات (look_down_score, head_turn_score, hand_to_face_score)، ومعلومات إضافية عن زوايا الرأس وحجم الوجه.
 - **ملف events/behaviors.jsonl:** يحتوي أحداث السلوكيات المكتشفة. كل سجل يُمثّل بداية أو نهاية سلوك ويتضمن: الطابع الزمني، ورقم الإطار، ومعرّف الشخص، ونوع السلوك، ونوع الحدث (start أو end)، ودرجة الإشارة عند الحدث.
 - **ملف decisions/risk_scores.jsonl:** يحتوي درجات المخاطر لكل شخص في كل إطار. كل سجل يتضمن: الطابع الزمني، ورقم الإطار، ومعرّف الشخص، ودرجة المخاطر الفورية، ودرجة المخاطر الممهدة (EWMA)، ومتوسط النافذة الزمنية، ومستوى الخطورة المُصنّف.
 - **ملف decisions/alerts.jsonl:** يحتوي التنبيهات الصادرة. كل سجل يتضمن: معرف الشخص، وإطاري البداية والنهاية، والطابع الزمني، وأعلى درجة مخاطر خلال الفترة، والسلوكيات المساهمة، ومستوى التنبيه.
- بالإضافة إلى ملفات JSONL، يُنتج النظام ملفات JSON عادية للملخصات والبيانات الوصفية التي تُقرأ دفعة واحدة:
- **ملف meta.json:** يحتوي معلومات وصفية عن التشغيل: معرف التشغيل، والطابع الزمني، ومسار الفيديو المُدخل، وعدد الإطارات المُعالجة، وزمن المعالجة.
 - **ملف behavior_stats.json:** يحتوي إحصائيات مُجمّعة عن السلوكيات: عدد أحداث كل سلوك، والمدة الإجمالية، والمتوسطات، لكل شخص ولمجموع الأشخاص.
 - **ملف summary.json:** يحتوي ملخصاً شاملاً للتشغيل: عدد الأشخاص المُتتبعين، وتوزيع مستويات المخاطر، وعدد التنبيهات بكل مستوى، والإحصائيات العامة.

هذا التنظيم المعياري للمخرجات يُسهّل تحليل النتائج والبناء عليها. يُمكن لأدوات خارجية قراءة هذه الملفات وإجراء تحليلات إضافية، أو استيرادها في قواعد بيانات، أو عرضها في واجهات مخصصة. لوحة التحكم المُضمّنة تقرأ هذه الملفات وتعرضها بشكل تفاعلي.

الفصل الرابع: الدراسة التنفيذية

4.1 مقدمة الفصل

يُمثل هذا الفصل جوهر التقرير من الناحية التقنية، إذ يصف التنفيذ الفعلي لجميع مكونات نظام Proctor AI. بينما تناول الفصل الثاني الأسس النظرية والخوارزميات بشكل عام، يُركّز هذا الفصل على كيفية تطبيق هذه المفاهيم في سياق المشروع، مع توضيح القرارات التصميمية والمعاملات المستخدمة والتدفق الفعلي للبيانات عبر مراحل النظام.

يتبع النظام بنية خط الأنابيب (Pipeline Architecture) حيث تمر البيانات عبر سلسلة من المراحل المتتالية، كل مرحلة تستقبل مخرجات المرحلة السابقة وتُنتج مخرجات للمرحلة التالية. هذا التصميم يُوفّر عدة مزايا: إمكانية اختبار كل مرحلة بشكل مستقل، وسهولة تحديد مصدر أي مشكلة، ومرونة في تحسين أو استبدال مراحل محددة دون التأثير على البقية.

يتكون خط المعالجة من ثماني مراحل رئيسية تُنفَّذ بالتتابع. تبدأ العملية بمرحلة الكشف عن الأشخاص التي تُحدد مواقع جميع الأشخاص في كل إطار من إطارات الفيديو باستخدام نموذج YOLOv8. تليها مرحلة التتبع التي تُحافظ على هوية كل شخص عبر الإطارات المتتالية باستخدام خوارزمية DeepSORT. ثم مرحلة استخراج الميزات التي تستخلص معلومات تفصيلية عن وضعية الوجه والجسم لكل شخص باستخدام MediaPipe.

بعد استخراج الميزات، تأتي مرحلة كشف السلوكيات التي تُحوّل الإشارات المستمرة إلى أحداث سلوكية منفصلة باستخدام آلية Hysteresis. تليها مرحلة حساب المخاطر التي تُجمّع درجات السلوكيات في درجة خطورة شاملة مع تمهيد زمني. ثم مرحلة التنبيهات التي تُصدر إشعارات عند تجاوز العتبات المحددة. وأخيراً مرحلة توليد المخرجات التي تُنتج الفيديو التوضيحي والتقارير والمقاطع.

يتناول كل قسم من هذا الفصل مرحلة واحدة بالتفصيل، موضحاً: المدخلات المتوقعة والمخرجات المُنتجة، والخطوات التنفيذية بالترتيب، والمعاملات القابلة للضبط وقيمها الافتراضية، والاعتبارات العملية والتحديات المحتملة. هذا المستوى من التفصيل يُمكن القارئ من فهم كيفية عمل النظام بدقة، ويُوفّر مرجعاً للمطورين الراغبين في تعديل أو توسيع النظام.

يُختتم الفصل بقسم عن واجهة المستخدم (Dashboard) التي تُوفّر طريقة تفاعلية لاستعراض نتائج التشغيلات دون الحاجة للتعامل المباشر مع الملفات أو سطر الأوامر.

4.2 مرحلة الكشف عن الأشخاص

تُشكل مرحلة الكشف عن الأشخاص نقطة البداية في خط المعالجة، حيث تُحدّد مواقع جميع الأشخاص المرئيين في كل إطار من إطارات الفيديو. تعتمد هذه المرحلة على نموذج YOLOv8 الذي يُوفّر كشفاً سريعاً ودقيقاً في الوقت الحقيقي. يتناول هذا القسم تفاصيل تحميل النموذج، ومعالجة الإطارات، واستخراج صناديق الإحاطة، وتصفية النتائج.

4.2.1 تحميل نموذج YOLOv8

يبدأ النظام بتحميل نموذج YOLOv8 عند بدء التشغيل. يُستخدم الإصدار yolov8s.pt (small) كخيار افتراضي، الذي يُوفّر توازناً جيداً بين السرعة والدقة. يُمكن تغيير هذا الاختيار من ملف الإعدادات لاستخدام إصدارات أخف (yolov8n) أو أدق (yolov8m, yolov8l).

عند تحميل النموذج لأول مرة، تقوم مكتبة Ultralytics بتحميله من الإنترنت وحفظه محلياً. في التشغيلات اللاحقة، يُحمّل النموذج من الملف المحلي مباشرة. يُنقل النموذج تلقائياً إلى GPU إذا كان متوفراً، أو يبقى على CPU خلاف ذلك. تُحدّد في هذه المرحلة أيضاً الفئات المستهدفة للكشف. بما أن النموذج مُدرّب على 80 فئة من مجموعة COCO، نُحدّد أننا مهتمون فقط بفئة "person" (المعرّف 0). هذا يُحسّن الأداء قليلاً ويمنع الكشف عن أشياء غير ذات صلة.

4.2.2 معالجة إطارات الفيديو

يُقرأ الفيديو إطاراً بإطار باستخدام OpenCV. لكل إطار، تُجرى المعالجة التالية: أولاً، يُحوّل الإطار من صيغة BGR (التي تقرأها OpenCV) إلى RGB (التي يتوقعها النموذج). ثانياً، يُغيّر حجم الإطار إلى الحجم المطلوب للنموذج (افتراضياً 640×640 بكسل) مع الحفاظ على نسبة العرض للارتفاع.

يُمرّر الإطار المُعالج إلى النموذج الذي يُنتج قائمة بالاكشافات. كل اكتشاف يتضمن: إحداثيات صندوق الإحاطة (x1, y1, x2, y2)، ودرجة الثقة (confidence)، ومعرّف الفئة (class_id). تُحوّل إحداثيات صناديق الإحاطة من نظام إحداثيات الصورة المُصغرة إلى نظام إحداثيات الصورة الأصلية.

يُطبّق النموذج داخلياً خوارزمية Non-Maximum Suppression (NMS) لإزالة الاكتشافات المتداخلة. إذا اكتشف النموذج نفس الشخص عدة مرات بصناديق متداخلة، تُبقي NMS فقط الصندوق ذي الثقة الأعلى وتحذف البقية.

4.2.3 استخراج صناديق الإحاطة

لكل اكتشاف مقبول، يُستخرج صندوق الإحاطة (Bounding Box) الذي يُحدد موقع الشخص في الإطار. يُخزن الصندوق بصيغة $(x1, y1, x2, y2)$ حيث $(x1, y1)$ هي الزاوية العلوية اليسرى و $(x2, y2)$ هي الزاوية السفلية اليمنى. تُحسب أيضاً معلومات مشتقة من الصندوق: العرض $(x2-x1)$ ، والارتفاع $(y2-y1)$ ، والمساحة (العرض×الارتفاع)، ومركز الصندوق $((x1+x2)/2, (y1+y2)/2)$. هذه المعلومات تُستخدم لاحقاً في التتبع وفي تصفية الاكتشافات. يُستخرج أيضاً قيمة الثقة لكل اكتشاف، وهي رقم بين 0 و 1 يُعبّر عن مدى تأكيد النموذج من وجود شخص في ذلك الموقع. الاكتشافات ذات الثقة العالية أكثر موثوقية، بينما الاكتشافات ذات الثقة المنخفضة قد تكون إنذارات كاذبة.

4.2.4 تصفية النتائج

بعد الكشف، تُطبّق عدة معايير لتصفية الاكتشافات والاحتفاظ فقط بتلك المفيدة للتحليل. هذه التصفية تُقلّل من الإنذارات الكاذبة وتُحسّن جودة المدخلات للمراحل اللاحقة.

- **التصفية بالثقة:** تُرفض الاكتشافات التي تقلّ ثقتها عن عتبة محددة (افتراضياً 0.40). هذه العتبة قابلة للضبط: خفضها يزيد عدد الاكتشافات لكن قد يزيد الإنذارات الكاذبة، ورفعها يُقلّل الإنذارات الكاذبة لكن قد يُفوّت بعض الأشخاص.
- **التصفية بالحجم:** تُرفض الاكتشافات الصغيرة جداً التي لا تُوفّر تفاصيل كافية لاستخراج الميزات. يُحدّد حد أدنى للعرض (افتراضياً 60 بكسل) وللارتفاع (افتراضياً 90 بكسل). الأشخاص البعيدون جداً عن الكاميرا الذين يظهرون صغراً في الصورة يُستبعدون.
- **التصفية بالعدد:** يُحدّد حد أقصى لعدد الاكتشافات في الإطار الواحد (افتراضياً 15). إذا تجاوز عدد الاكتشافات هذا الحد، تُرتّب بالثقة وتُؤخذ الأعلى فقط. هذا يمنع التباطؤ في المشاهد المزدحمة جداً.
- **التصفية بنسبة الأبعاد:** يُمكن اختيارياً رفض الصناديق ذات نسب الأبعاد غير الطبيعية للإنسان. الصناديق العريضة جداً أو الطويلة جداً قد تكون اكتشافات خاطئة.

تنتج هذه المرحلة لكل إطار قائمة من الاكتشافات المُصفّاة، كل منها يتضمن صندوق الإحاطة ودرجة الثقة. هذه القائمة تُمرّر إلى مرحلة التتبع لربط الاكتشافات عبر الإطارات.

4.3 مرحلة التتبع

تتولى مرحلة التتبع ربط الاكتشافات عبر الإطارات المتتالية للحفاظ على هوية فريدة ومستقرة لكل شخص طوال مدة ظهوره في الفيديو. بدون التتبع، سيعامل كل شخص في كل إطار ككيان مستقل، مما يُفقد القدرة على تحليل السلوك عبر الزمن. يتناول هذا القسم تهيئة خوارزمية DeepSORT، وربط الاكتشافات بالمسارات، والحفاظ على الهوية، والتعامل مع حالات الاختفاء المؤقت.

4.3.1 تهيئة DeepSORT

يُهيأ متتبع DeepSORT عند بدء تشغيل النظام بمجموعة من المعاملات التي تُحدد سلوكه. هذه المعاملات قابلة للضبط من ملف الإعدادات لتناسب خصائص الفيديو المُعالج.

- **max_age**: يُحدد عدد الإطارات التي يُحتفظ فيها بمسار غير مُؤكد قبل حذفه. القيمة الافتراضية 90 إطاراً (حوالي 3 ثوانٍ بمعدل 30 fps). هذا يُتيح للشخص الاختفاء مؤقتاً (خلف عائق مثلاً) ثم العودة دون فقدان هويته.
- **n_init**: يُحدد عدد الإطارات المتتالية التي يجب أن يُكتشف فيها كائن قبل تأكيد مساره وإعطائه معرفاً دائماً. القيمة الافتراضية 2 إطاران. هذا يمنع إنشاء مسارات لاكتشافات عابرة أو خاطئة.
- **max_iou_distance**: عتبة التداخل (IoU) لربط الاكتشافات بالمسارات بناءً على الموقع. القيمة الافتراضية 0.75. اكتشاف وموقع متوقع يتداخل أقل من هذه العتبة لا يُربطان معاً.
- **max_cosine_distance**: عتبة المسافة الكوسينية لربط الاكتشافات بالمسارات بناءً على الميزات المظهرية. القيمة الافتراضية 0.45. هذا المعامل مهم لإعادة التعرف على شخص بعد اختفائه.
- **nn_budget**: حجم ذاكرة الميزات المظهرية لكل مسار. القيمة الافتراضية 150. يُخزّن هذا العدد من أحدث الميزات لكل شخص للاستخدام في المقارنة.

4.3.2 ربط الاكتشافات بالمسارات

في كل إطار، تُجرى عملية ربط (Association) بين الاكتشافات الجديدة والمسارات الموجودة. تستخدم DeepSORT استراتيجية ربط ثنائية المرحلة للحصول على أفضل النتائج.

المرحلة الأولى - الربط بالميزات المظهرية: تُحسب المسافة الكوساينية بين الميزات المظهرية لكل اكتشاف والميزات المخزنة لكل مسار. تُبنى مصفوفة التكلفة من هذه المسافات، ثم يُطبّق الخوارزمية الهنغارية للعثور على أفضل تخصيص. الأزواج التي تتجاوز مسافتها العتبة تُستبعد.

المرحلة الثانية - الربط بالموقع (IOU): الاكتشافات والمسارات التي لم تُربط في المرحلة الأولى تُعالج مرة أخرى باستخدام IOU. هذا يُفيد عندما تكون الميزات المظهرية غير متوفرة أو غير موثوقة (مثل الأشخاص الجدد أو المحجوبين جزئياً). بعد الربط، تُصنّف النتائج إلى ثلاث فئات: أزواج مُطابقة (اكتشاف رُبط بمسار موجود)، واكتشافات غير مُطابقة (قد تكون أشخاصاً جددًا)، ومسارات غير مُطابقة (أشخاص لم يُكتشفوا في هذا الإطار).

4.3.3 الحفاظ على هوية كل شخص

لكل مسار مُؤكّد، يُخصّص معرف فريد (Track ID) يبقى ثابتاً طوال فترة وجود الشخص في الفيديو. هذا المعرف يُستخدم في جميع المراحل اللاحقة لربط الميزات والسلوكيات والتنبّهات بالشخص المعني.

يُحدّث كل مسار في كل إطار بمعلومات الاكتشاف المُطابق: صندوق الإحاطة الجديد، ودرجة الثقة، والميزات المظهرية. يُحدّث أيضاً مرشح كالمان الخاص بالمسار لتحسين تقدير الموقع والسرعة.

للمسارات التي لم تُطابق في إطار معين (الشخص غير مرئي)، يستمر مرشح كالمان في التنبؤ بالموقع بناءً على السرعة السابقة. هذا التنبؤ يُستخدم في محاولات الربط في الإطارات اللاحقة، ويُتيح إعادة الربط عندما يظهر الشخص مجدداً.

تُخزّن أيضاً معلومات تاريخية لكل مسار: إطار البداية، وإطار آخر ظهور، وعدد الإطارات التي ظهر فيها، وعدد الإطارات التي اختفى فيها. هذه المعلومات تُستخدم في التحليلات والتقارير.

4.3.4 التعامل مع الاختفاء المؤقت

يُعدّ التعامل مع اختفاء الأشخاص المؤقت وعودتهم من أهم قدرات DeepSORT. هذا السيناريو شائع في بيئة الامتحان: شخص يميل فيحجب جزئياً خلف آخر، أو يقف المراقب أمام الكاميرا للحظة.

عندما لا يُطابق مسار أي اكتشاف، يدخل حالة "مفقود" (lost). يستمر المسار في هذه الحالة لعدد من الإطارات يُحدده معامل max_age. خلال هذه الفترة، يبقى المسار نشطاً ويُمكن إعادة ربطه إذا ظهر الشخص مجدداً.

إعادة الربط تعتمد بشكل أساسي على الميزات المظهرية. حتى لو تغير موقع الشخص بشكل كبير أثناء اختفائه، يُمكن التعرف عليه من ملامحه وملابسه. هذه هي الميزة الرئيسية لـ DeepSORT مقارنة بالمتتبعات البسيطة التي تعتمد على الموقع فقط.

إذا لم يُعد ربط المسار خلال فترة max_age، يُحذف نهائياً. إذا ظهر الشخص لاحقاً، سيُنشأ له مسار جديد بمعرف جديد. هذا سيناريو غير مرغوب لكنه لا مفر منه في حالات الاختفاء الطويل.

تُخرج هذه المرحلة لكل إطار قائمة من المسارات النشطة، كل منها يتضمن: المعرف الفريد، وصندوق الإحاطة المُحدَّث، ودرجة الثقة، وحالة المسار (مؤكد أو مبدئي). هذه القائمة تُمرَّر إلى مرحلة استخراج الميزات.

4.4 مرحلة استخراج الميزات

تتولى هذه المرحلة استخراج معلومات تفصيلية عن وضعية الوجه والجسم لكل شخص مُتتبع، ثم تحويل هذه المعلومات إلى إشارات رقمية تُعبّر عن سلوكيات محتملة. تعتمد المرحلة على إطار عمل MediaPipe لاستخراج النقاط المعلمية. يتناول هذا القسم تحديد منطقة الاهتمام، ومعالجة الوجه والجسم، وحساب إشارات السلوك.

4.4.1 تحديد منطقة الاهتمام (ROI)

بدلاً من معالجة الإطار كاملاً، يُقتطع جزء محدد (Region of Interest) لكل شخص ويُعالج بشكل مستقل. هذا النهج يُحسن الدقة والأداء معاً.

يُحدّد ROI بناءً على صندوق الإحاطة الناتج من التتبع. في الوضع الافتراضي (upper_body)، يُؤخذ الجزء العلوي من الصندوق (72% من الارتفاع) الذي يحتوي الرأس والكتفين واليدين. هذا يركّز المعالجة على الأجزاء ذات الأهمية للتحليل السلوكي.

يُوسّع الـ ROI قليلاً (10-15%) من كل جانب لضمان احتواء الجسم كاملاً حتى مع حركة طفيفة. يُقصّ الـ ROI من الإطار الأصلي ويُغيّر حجمه إذا لزم الأمر ليتوافق مع متطلبات MediaPipe.

هذا النهج يقلل التداخل بين الأشخاص المتجاورين، فكل ROI يحتوي شخصاً واحداً فقط. كما يُحسن دقة MediaPipe لأن الوجه يظهر أكبر نسبياً في الـ ROI المقتطع مقارنة بالإطار الكامل.

4.4.2 معالجة الوجه بـ MediaPipe

يُستخدم حل Face Mesh من MediaPipe لاستخراج 468 نقطة معلمية من الوجه إذا كان مرئياً. تُجرى المعالجة على الـ ROI المقتطع لكل شخص.

تُمرر صورة الـ ROI إلى Face Mesh الذي يُعيد: علامة نجاح الكشف (هل وُجد وجه؟)، وقائمة النقاط المعلمية بإحداثياتها الثلاثية (x, y, z). إذا فشل الكشف (الوجه غير مرئي أو مُلتفت بشكل كبير)، تُسجل علامة face_detected=False ويعتمد على معلومات الجسم فقط.

من نقاط الوجه، تُستخرج معلومات مهمة. زاوية الالتفات الأفقي (Yaw) تُحسب من الموقع الأفقي للأنف بالنسبة لمركز الوجه. زاوية الإمالة العمودية (Pitch) تُحسب من الموقع العمودي للأنف بالنسبة للعينين. حجم الوجه يُقدَّر من المسافة بين العينين ويُستخدم كمقياس للبعد عن الكاميرا.

تُطَبَّع الزوايا المحسوبة إلى مدى [-1, 1] لتسهيل المعالجة اللاحقة. القيمة 0 تُشير للنظر المباشر للأمام، والقيم السالبة والموجبة تُشير للالتفات في الاتجاهين.

4.4.3 معالجة الجسم بـ MediaPipe

يُستخدم حل Pose من MediaPipe لاستخراج 33 نقطة معلمية للجسم. هذا يُوفّر معلومات تكميلية للوجه، ويُتيح التحليل حتى عندما يكون الوجه غير مرئي.

من نقاط الجسم، تُستخرج عدة معلومات. موقع الكتفين يُستخدم كمرجع لقياس وضعية الرأس. انحدار الرأس للأمام (النظر للأسفل) يُقدَّر من العلاقة بين موقع الأنف وخط الكتفين. موقع المعصمين يُستخدم لكشف تقريب اليد من الوجه.

كل نقطة تأتي مع درجة رؤية (visibility) تُشير إلى مدى وضوحها في الصورة. النقاط ذات الرؤية المنخفضة (أقل من 0.5) تُعامل بحذر أو تُتجاهل لتجنب الحسابات غير الدقيقة.

يُمكن لـ MediaPipe Pose تقدير مواقع النقاط غير المرئية بناءً على السياق، لكن هذه التقديرات أقل دقة. النظام يُعطي أولوية للنقاط المرئية فعلاً.

4.4.4 حساب إشارات السلوك

تُحوّل النقاط المعلمية الخام إلى إشارات سلوكية (Behavioral Signals) تُعبّر عن درجة كل سلوك. هذه الإشارات أرقام بين 0 و 1، حيث 0 تعني غياب السلوك و 1 تعني أقصى شدة له.

إشارة النظر للأسفل (**look_down_score**): تُحسب من مصدرين ويُؤخذ الأعلى. المصدر الأول هو زاوية Pitch من الوجه: كلما انخفض الأنف بالنسبة للعينين، ارتفعت الإشارة. المصدر الثاني هو وضعية الرأس من نقاط الجسم: كلما انخفض الرأس بالنسبة للكفتين، ارتفعت الإشارة. الجمع بين المصدرين يُؤفّر موثوقية أعلى.

إشارة الالتفات (**head_turn_score**): تُحسب من زاوية Yaw. تُقسم إلى إشارتين فرعيتين: head_turn_left_score للالتفات يساراً، و head_turn_right_score للالتفات يميناً. الإشارة الإجمالية head_turn_score هي الأعلى من الاثنين.

إشارة اليد نحو الوجه (**hand_to_face_score**): تُحسب من المسافة بين معصم اليد والأنف. تُطَبّع المسافة بطول الذراع لتكون مستقلة عن حجم الشخص في الصورة. إذا كانت اليد قريبة من الوجه (أقل من عتبة محددة)، ترتفع الإشارة. تُخزّن جميع الإشارات المحسوبة مع معلومات إضافية (رقم الإطار، الطابع الزمني، معرّف الشخص، صندوق الإحاطة، علامات الكشف) في ملف features/mediapipe.jsonl بصيغة JSONL للمعالجة في المرحلة التالية.

4.5 مرحلة كشف السلوكيات

تتولى هذه المرحلة تحويل الإشارات المستمرة (التي تتغير في كل إطار) إلى أحداث سلوكية منفصلة لها بداية ونهاية. هذا التحويل ضروري لتحليل السلوك عبر الزمن ولتوليد سجل واضح من الأحداث. تعتمد المرحلة على آلية Hysteresis لضمان استقرار الكشف. يتناول هذا القسم تطبيق الآلية وكشف كل سلوك وتسجيل الأحداث.

4.5.1 تطبيق آلية Hysteresis

تعمل آلية Hysteresis بعثبتين: عتبة التفعيل (ON) وعتبة الإيقاف (OFF)، حيث $ON > OFF$. هذا الفارق يُسمى "نطاق التباطؤ" ويمنع التذبذب السريع عندما تتراوح الإشارة حول قيمة حدية.

لكل سلوك ولكل شخص، يُحتفظ بحالة (نشط أو غير نشط). عندما يكون السلوك غير نشط، يُفَعّل فقط إذا تجاوزت الإشارة عتبة ON. عندما يكون السلوك نشطاً، يُوقَف فقط إذا انخفضت الإشارة دون عتبة OFF.

تُحدّد العتبات من ملف الإعدادات، والقيم الافتراضية هي: للنظر للأسفل $ON=0.48$ و $OFF=0.42$ ، ولاللتفات $ON=0.42$ و $OFF=0.32$ ، ولليد نحو الوجه $ON=0.45$ و $OFF=0.35$.

هذه القيم ضبطت تجريبياً لتحقيق توازن بين الحساسية (اكتشاف السلوكيات الفعلية) والنوعية (تجنب الإنذارات الكاذبة). يُمكن تعديلها حسب خصائص البيئة المستهدفة.

4.5.2 كشف النظر للأسفل (look_down)

يُعالج كل سجل من ملف الميزات بالتتابع. لكل شخص، تُقرأ إشارة look_down_score وتُقارن بحالة السلوك الحالية. إذا كان السلوك غير نشط وتجاوزت الإشارة 0.48، يُفعّل السلوك ويُسجّل حدث بداية (start) مع الطابع الزمني ورقم الإطار ودرجة الإشارة.

إذا كان السلوك نشطاً وانخفضت الإشارة دون 0.42، يُوقف السلوك ويُسجّل حدث نهاية (end) مع المعلومات ذاتها. تُحسب أيضاً مدة كل حدث سلوك من الفرق بين طابعي البداية والنهاية. السلوكيات القصيرة جداً (أقل من 100 مللي ثانية) قد تُشير لضوضاء ويُمكن تجاهلها اختياريّاً.

4.5.3 كشف الالتفاتات (head_turn)

يُعالج الالتفاتات بطريقة مشابهة، مع التعامل مع الاتجاهين (يمين ويسار) بشكل منفصل. لكل شخص، تُقرأ إشارتا head_turn_left_score و head_turn_right_score. تُطبّق آلية Hysteresis على كل منهما بشكل مستقل بعتبات ON=0.42 و OFF=0.32.

عند تفعيل أي من الاتجاهين، يُسجّل حدث بداية من النوع head_turn_left أو head_turn_right حسب الاتجاه. هذا التفصيل يُتيح تحليلاً أدق: الالتفاتات المتكرر لجهة واحدة قد يُشير للنظر في ورقة جارٍ محدد.

4.5.4 كشف اليد للوجه (hand_to_face)

يتميز هذا السلوك بإضافة شرط فترة الثبات (Hold Time) قبل تسجيل البداية. هذا يمنع التسجيل الخاطئ للحركات العابرة السريعة كحكّ الوجه أو تعديل النظارات.

عندما تتجاوز الإشارة عتبة ON (0.45)، لا يُفعّل السلوك فوراً. بدلاً من ذلك، يُسجّل وقت التجاوز ويُنتظر. إذا استمرت الإشارة فوق العتبة لمدة Hold Time (افتراضياً 300 مللي ثانية)، عندها يُفعّل السلوك ويُسجّل حدث البداية. إذا انخفضت الإشارة قبل انقضاء Hold Time، يُلغى العداد ولا يُسجّل شيء. هذا يُصفّي الحركات العابرة البريئة ويُبقي فقط الحالات التي استمرت فيها اليد قرب الوجه لفترة ذات دلالة.

4.5.5 تسجيل أحداث البداية والنهاية

تُسجَل جميع الأحداث المكتشفة في ملف `events/behaviors.jsonl` بصيغة JSONL. كل سجل يحتوي: الطابع الزمني بالمللي ثانية (`ts_ms`)، ورقم الإطار (`frame_index`)، ومعرّف الشخص (`student_id`)، ونوع السلوك (`behavior`)، ونوع الحدث (`event_type: start` أو `end`)، ودرجة الإشارة عند لحظة الحدث (`score`).

يُمكن من هذا الملف إعادة بناء الجدول الزمني الكامل للسلوكيات لكل شخص: متى بدأ كل سلوك ومتى انتهى وكم استمر. هذه المعلومات تُغذي مرحلة حساب المخاطر وتُستخدم في التقارير والإحصائيات.

تُحسب أيضاً إحصائيات مُجمّعة تُحفظ في ملف منفصل: عدد أحداث كل سلوك لكل شخص، والمدة الإجمالية لكل سلوك، ومتوسط مدة الحدث الواحد. هذه الإحصائيات تُعطي نظرة سريعة على نمط سلوك كل شخص.

4.6 مرحلة حساب المخاطر

تتولى هذه المرحلة تجميع درجات السلوكيات المختلفة في درجة خطورة شاملة لكل شخص في كل لحظة. توفّر درجة المخاطر مقياساً موحداً يُسهّل المقارنة بين الأشخاص وعبر الزمن. يتناول هذا القسم المعادلة المستخدمة والأوزان والتمهيد الزمني وتصنيف المستويات.

4.6.1 المعادلة المستخدمة

تُحسب درجة المخاطر الفورية (`instant risk`) لكل شخص في كل إطار كمجموع موزون لدرجات الإشارات السلوكية. الصيغة الرياضية هي:

$$\text{risk_instant} = w_look \times \text{look_down_score} + w_turn \times \text{head_turn_score} + w_hand \times \text{hand_to_face_score}$$

حيث `w_look` و `w_turn` و `w_hand` هي أوزان كل سلوك، ومجموعها يساوي 1.0. النتيجة هي رقم بين 0 و 1، حيث 0 تعني غياب أي سلوك مشبوه و 1 تعني أقصى درجة من السلوكيات المشبوهة.

لإشارة الالتفات، يُستخدم الأعلى من `head_turn_left_score` و `head_turn_right_score`، لأن الشخص لا يلتفت لليمين واليسار في آن واحد.

4.6.2 أوزان السلوكيات

تُحدد الأوزان الأهمية النسبية لكل سلوك في تقييم المخاطر. القيم الافتراضية هي:

- w_look (النظر للأسفل): 0.45 - وزن مرتفع لأن هذا السلوك مرتبط بشكل قوي بقراءة مواد غير مسموحة.
- w_turn (الالتفات): 0.45 - وزن مرتفع مماثل لأن الالتفات قد يُشير للنظر في ورقة أخرى.
- w_hand (اليد للوجه): 0.10 - وزن أقل لأن هذا السلوك شائع لأسباب بريئة وأقل ارتباطاً بالغش.

هذه الأوزان قابلة للتعديل من ملف الإعدادات. مثلاً، في بيئة يُتوقع فيها استخدام سماعات أذن، قد يُرفع وزن w_hand .

4.6.3 التنعيم الزمني

درجة المخاطر الفورية تتذبذب من إطار لآخر بسبب الضوضاء في الإشارات. لتوفير مقياس أكثر استقراراً، يُطبق تمهيد زمني باستخدام EWMA (Exponential Weighted Moving Average).

الصيغة هي:

$$risk_smoothed = \alpha \times risk_instant + (1-\alpha) \times risk_smoothed_previous$$

حيث α (alpha) هو معامل التمهيد، قيمته الافتراضية 0.15. قيمة α المنخفضة تُنتج تمهيداً قوياً (استقرار عالٍ لكن استجابة بطيئة)، وقيمة α العالية تُنتج تمهيداً خفيفاً (استجابة سريعة لكن استقرار أقل).

يُحسب أيضاً متوسط نافذة زمنية (window mean) وهو متوسط درجات المخاطر في آخر N ثانية (افتراضياً 10 ثوانٍ). هذا يُعطي نظرة على الاتجاه العام للسلوك.

4.6.4 تصنيف مستويات الخطورة

تُصنّف درجة المخاطر الممهدة إلى أربعة مستويات لتسهيل الفهم والتعامل:

- **آمن (safe)**: درجة أقل من 0.35 - سلوك طبيعي، لا يستدعي انتباهاً خاصاً.
- **مشبوه (suspicious)**: درجة بين 0.35 و 0.60 - سلوك يستحق المتابعة لكن ليس مُقلقاً بشكل فوري.
- **عالي الخطورة (high_risk)**: درجة بين 0.60 و 0.80 - سلوك يستدعي مراجعة فورية.
- **مؤكد (confirmed)**: درجة أعلى من 0.80 - سلوك غير طبيعي واضح، يستدعي تدخلاً.

هذه العتبات قابلة للتعديل حسب سياسة المؤسسة ودرجة الحساسية المطلوبة.

تُحفظ درجات المخاطر في ملف decisions/risk_scores.jsonl، كل سجل يحتوي: الطابع الزمني، ورقم الإطار، ومعرّف الشخص، ودرجة المخاطر الفورية، والدرجة الممهدة، ومتوسط النافذة، ومستوى الخطورة المُصنّف.

4.7 مرحلة التنبيهات والقرارات

تتولى هذه المرحلة تحويل درجات المخاطر إلى تنبيهات قابلة للتنفيذ تُوجّه للمراقب البشري. لا يكفي مجرد حساب درجة الخطورة؛ يجب تحديد متى تستحق الحالة انتبهاً خاصاً وكيف يُبلّغ عنها. يتناول هذا القسم شروط إطلاق التنبيه وتسجيل التنبيهات وملخص القرارات.

4.7.1 شروط إطلاق التنبيه

يُصدر النظام تنبيهاً عندما تتجاوز درجة المخاطر الممهدة عتبة محددة وتستمر فوقها لفترة كافية. هذه الشروط تمنع التنبيهات الناتجة عن ارتفاعات لحظية عابرة.

- **عتبة التفعيل (alert_on):** درجة المخاطر التي يجب تجاوزها لبدء النظر في إصدار تنبيه. القيمة الافتراضية 0.75، أي مستوى "عالي الخطورة".
- **عتبة الإيقاف (alert_off):** درجة المخاطر التي يجب الانخفاض دونها لإنهاء التنبيه. القيمة الافتراضية 0.55. الفارق بين العتبتين يُطبّق مبدأ Hysteresis لمنع التذبذب.
- **فترة الثبات (hold_ms):** المدة بالمللي ثانية التي يجب أن تستمر فيها الدرجة فوق العتبة قبل إصدار التنبيه. القيمة الافتراضية 1200 مللي ثانية (1.2 ثانية). هذا يمنع التنبيهات على ارتفاعات قصيرة جداً.
- **فترة التهدئة (cooldown_ms):** المدة بعد إصدار تنبيه لشخص ما قبل السماح بتنبيه جديد لنفس الشخص. القيمة الافتراضية 10000 مللي ثانية (10 ثوانٍ). هذا يمنع إغراق المراقب بتنبيهات متكررة.

4.7.2 تسجيل التنبيهات

عند استيفاء شروط التنبيه، يُنشأ سجل تنبيه يتضمن معلومات شاملة تُساعد المراقب على فهم السياق واتخاذ القرار:

- **student_id:** معرّف الشخص المعني.
- **start_frame / end_frame:** إطارا بداية ونهاية فترة التنبيه.

- **start_ms / end_ms**: الطوابع الزمنية بالمللي ثانية.
 - **peak_risk**: أعلى درجة مخاطر وصلها الشخص خلال فترة التنبيه.
 - **behaviors**: قائمة السلوكيات النشطة التي ساهمت في رفع المخاطر.
 - **level**: مستوى التنبيه (suspicious, high_risk, confirmed).
 - **bbox**: صندوق إحاطة الشخص عند ذروة المخاطر، مفيد لتحديد موقعه في الفيديو.
- تُحفظ التنبيهات في ملف decisions/alerts.jsonl. كل تنبيه سجل مستقل يُمكن قراءته ومعالجته.

4.7.3 ملخص القرارات

بالإضافة للتنبيهات الفردية، يُنتج النظام ملخصاً شاملاً للقرارات يُحفظ في ملف decisions/summary.json. يتضمن الملخص:

- **إحصائيات عامة**: عدد الأشخاص المُتتبعين، ومدة الفيديو المُعالج، وعدد الإطارات.
- **توزيع المخاطر**: عدد الأشخاص في كل مستوى خطورة، ونسبة الوقت التي قضاها كل شخص في كل مستوى.
- **ملخص التنبيهات**: العدد الإجمالي للتنبيهات، والتوزيع حسب المستوى، وقائمة الأشخاص الذين صدرت لهم تنبيهات.
- **أبرز الأحداث**: قائمة بأعلى فترات المخاطر مرتبة تنازلياً، مع الطوابع الزمنية ومعرّفات الأشخاص. هذه القائمة تُوجّه المراقب للمراجعة.
- **التوصيات**: نص توصية آلي بناءً على النتائج. مثلاً: "يُنصح بمراجعة الفترات التالية..." أو "لم تُرصد سلوكيات عالية الخطورة".

يُقدّم هذا الملخص أيضاً في تقرير PDF وفي لوحة التحكم، مما يُوفّر نقطة دخول سريعة لفهم نتائج التشغيل دون الحاجة لمراجعة جميع الملفات التفصيلية.

4.8 مرحلة المخرجات

تتولى هذه المرحلة تحويل نتائج التحليل إلى مخرجات ملموسة يُمكن للمراقب البشري استخدامها. تختلف هذه المخرجات لتلبي احتياجات مختلفة: الفيديو التوضيحي للمراجعة المرئية، ومقاطع الأدلة للتوثيق، وتقرير PDF للأرشفة الرسمية. يتناول هذا القسم كل نوع من المخرجات.

4.8.1 توليد الفيديو التوضيحي

الفيديو التوضيحي (Demo Video) هو نسخة من الفيديو الأصلي مُعلّمة بمعلومات التحليل. يُتيح للمراقب مشاهدة ما اكتشفه النظام بشكل مرئي ومباشر.

يتضمن الفيديو التوضيحي عدة عناصر مرئية. حول كل شخص مُتتبع يُرسم صندوق إحاطة ملون حسب مستوى الخطورة: أخضر للأمن، وأصفر للمشبه، وبرتقالي لعالي الخطورة، وأحمر للمؤكد. فوق كل صندوق يُكتب معرف الشخص ودرجة المخاطر الحالية.

أسفل كل صندوق يُعرض شريط تقدم يُمثل درجة المخاطر بصرياً. الشريط يمتلئ ويتغير لونه مع ارتفاع الدرجة. تُعرض أيضاً أيقونات السلوكيات النشطة: سهم للأسفل للنظر للأسفل، وسهم جانبي للاتفات، ويد لـ hand_to_face.

في أعلى الفيديو يُعرض شريط معلومات يتضمن: اسم الجلسة، والوقت الحالي في الفيديو، ورقم الإطار، وعدد الأشخاص المكتشفين، وعدد التنبيهات النشطة.

على جانب الفيديو (اختيارياً) يُعرض شريط جانبي يتضمن: قائمة الأشخاص المرئيين في الإطار، وآخر التنبيهات الصادرة، ورسم بياني صغير لتطور المخاطر.

4.8.2 استخراج مقاطع الأدلة

مقاطع الأدلة (Top Clips) هي فيديوهات قصيرة مُستخرجة من الفيديو الأصلي تُغطي فترات المخاطر العالية. تُوفر طريقة سريعة لمراجعة أهم الأحداث دون مشاهدة الفيديو كاملاً.

يُحدد النظام فترات المخاطر العالية من ملف risk_scores.jsonl لكل فترة تجاوزت فيها درجة المخاطر عتبة محددة، يُستخرج مقطع فيديو يبدأ قبل ذروة الخطورة بثوانٍ قليلة (للسياق) وينتهي بعدها بثوانٍ قليلة.

تُرتَّب المقاطع بأعلى درجة مخاطر وتُحفظ بأسماء تدل على محتواها، مثل: clip_001_s3_risk82.mp4 (المقطع الأول، للشخص s3، أعلى مخاطر 82%). يُحدّد عدد أقصى للمقاطع (افتراضياً 5) لتجنب إنتاج عدد كبير جداً.

تُحفظ أيضاً قائمة المقاطع مع معلوماتها (الشخص، الفترة الزمنية، درجة المخاطر، السلوكيات) في ملف top_clips.json للرجوع إليه.

4.8.3 توليد تقرير PDF

تقرير PDF هو وثيقة رسمية تُلخّص نتائج التحليل بشكل احترافي. يصلح للأرشفة وللتقديم في لجان التحقيق وللتوثيق المؤسسي، يتضمن التقرير عدة أقسام:

- **صفحة العنوان:** اسم النظام، واسم الجلسة/الامتحان، والتاريخ والوقت، ومدة الفيديو. ****الملخص التنفيذي**:** نظرة عامة سريعة على النتائج: عدد الأشخاص، وعدد التنبيهات، وأبرز الملاحظات.
- **إحصائيات السلوكيات:** رسوم بيانية تُظهر توزيع السلوكيات حسب النوع وحسب الشخص. مخطط دائري للنسب، ومخطط شريطي للمقارنة بين الأشخاص.
- **الخط الزمني:** رسم بياني يُظهر تطور درجة المخاطر عبر الزمن لكل شخص. يُسهّل تحديد الفترات الحرجة بصرياً.
- **قائمة التنبيهات:** جدول تفصيلي بجميع التنبيهات الصادرة، مع الأوقات والأشخاص والمستويات. يُمكن طباعته كمرجع.
- **التوصيات:** نص آلي يُلخّص النتائج ويُوصي بالإجراءات المناسبة. يُذكر بأن النظام أداة مساعدة وأن القرار النهائي للمراقب البشري.

4.8.4 الإحصائيات والرسوم البيانية

بالإضافة للتقرير، تُنتج رسوم بيانية منفصلة يُمكن استخدامها في عروض تقديمية أو تقارير مخصصة.

- **risk_timeline.png:** خط زمني لدرجات المخاطر لجميع الأشخاص على رسم واحد. يُظهر اللحظات التي ارتفعت فيها المخاطر لشخص أو أكثر.
- **behavior_distribution.png:** توزيع السلوكيات المكتشفة. عدد أحداث كل سلوك، ومجموع مدتها، ومتوسط المدة.
- **person_summary.png:** ملخص لكل شخص: الوقت في كل مستوى خطورة، وعدد السلوكيات من كل نوع.

تُحفظ هذه الرسوم في مجلد `/reports/charts/` ويُمكن الوصول إليها من لوحة التحكم أو فتحها مباشرة.

4.9 واجهة المستخدم (Dashboard)

توفر واجهة المستخدم التفاعلية (Dashboard) طريقة سهلة وبصرية لاستعراض نتائج التشغيل دون الحاجة للتعامل المباشر مع الملفات أو سطر الأوامر. بُنيت الواجهة باستخدام إطار عمل Dash وتعمل كتطبيق ويب محلي. يتناول هذا القسم صفحات الواجهة وطريقة عرض البيانات.

4.9.1 صفحة عرض التشغيلات

الصفحة الرئيسية للوحة التحكم تعرض قائمة بجميع التشغيلات (runs) المُنفَّذة. لكل تشغيل تُعرض معلومات موجزة: معرف التشغيل، وتاريخ ووقت التنفيذ، واسم ملف الفيديو، ومدة المعالجة، وعدد الأشخاص المُكتَشَفين، وعدد التنبيهات. تُرتَّب التشغيلات افتراضياً بالأحدث أولاً. يُمكن للمستخدم البحث بمعرف التشغيل أو التصفية بالتاريخ. كل صف في القائمة قابل للنقر للانتقال لصفحة التفاصيل.

تُعرض أيضاً مؤشرات بصرية سريعة: أيقونة ملونة تُشير لأعلى مستوى خطورة وُصل إليه في ذلك التشغيل، مما يُساعد على تحديد التشغيلات التي تستحق المراجعة.

4.9.2 صفحة التفاصيل

عند اختيار تشغيل معين، تُعرض صفحة التفاصيل التي تتضمن معلومات شاملة عنه.

- **قسم المعلومات العامة:** يعرض بيانات وصفية عن التشغيل: المعرف، وملف الفيديو، وملف الإعدادات، ووقت البدء والانهاء، وعدد الإطارات المُعالجة، ومتوسط سرعة المعالجة.
- **قسم الإحصائيات:** يعرض أرقاماً مُلخَّصة: عدد الأشخاص الفريدين، وعدد أحداث كل سلوك، وتوزيع مستويات المخاطر، وعدد التنبيهات بكل مستوى. تُعرض هذه الإحصائيات في بطاقات ملونة سهلة القراءة.
- **قسم الرسوم البيانية:** يعرض رسوماً بيانية تفاعلية من مكتبة Plotly. الخط الزمني للمخاطر يُتيح التكبير والتصغير والتمرير. عند تمرير المؤشر فوق نقطة، تُعرض تفاصيل (الوقت، الشخص، الدرجة). مخطط توزيع السلوكيات يُظهر النسب بشكل تفاعلي.
- **قسم التنبيهات:** يعرض جدولاً بجميع التنبيهات مع إمكانية الترتيب والتصفية. النقر على تنبيه يُمكن أن ينتقل للفترة المعنية في الفيديو.

4.9.3 عرض الفيديو والإحصائيات

تُوفّر الواجهة إمكانية مشاهدة الفيديو التوضيحي مباشرة في المتصفح إذا كان متوفراً. يُعرض الفيديو في مشغل مُضمّن يدعم التحكم بالتشغيل والإيقاف والتقديم والترجيع.

يُمكن أيضاً عرض الفيديو الأصلي إذا كان متوفراً، للمقارنة أو للمراجعة دون التعليقات المُضافة.

تُوفّر الواجهة خيارات للتصدير: تحميل تقرير PDF، وتحميل ملفات البيانات، وتحميل الرسوم البيانية كصور. هذا يُسهّل نقل النتائج لاستخدامها خارج النظام.

- **التشغيل:** لبدء لوحة التحكم، يُنفَّذ الأمر `python -m dashboard.app` من مجلد المشروع. يبدأ خادم ويب محلي على المنفذ 8050. يُفتح المتصفح على العنوان `http://localhost:8050` للوصول للواجهة.
- **التصميم:** بُنيت الواجهة بتصميم متجاوب يعمل على شاشات مختلفة الأحجام. استُخدمت مكونات Bootstrap للحصول على مظهر احترافي ومتسق. الألوان والأيقونات متناسقة مع مستويات المخاطر المستخدمة في بقية النظام.
- **الأداء:** الواجهة خفيفة ولا تتطلب موارد كبيرة. البيانات تُقرأ من الملفات عند الطلب، مما يُقلل استهلاك الذاكرة. الرسوم البيانية تُحسب وتُعرض بشكل تدريجي لتجنب تجميد الواجهة مع بيانات كبيرة.

الفصل الخامس: الاختبار والنتائج

5.1 مقدمة الفصل

يُقدّم هذا الفصل تقييماً شاملاً لأداء نظام Proctor AI من خلال سلسلة من الاختبارات المنهجية. الهدف من هذه الاختبارات هو التحقق من أن النظام يعمل وفق المواصفات المحددة، وقياس أدائه الفعلي في ظروف واقعية، وتحديد نقاط القوة والضعف لتوجيه التطوير المستقبلي.

يتبع الفصل منهجية اختبار متدرجة تبدأ باختبار المكونات الفردية (وحدات الكشف والتتبع واستخراج الميزات) ثم تنتقل لاختبار التكامل بينها، وصولاً لاختبار النظام الكامل على سيناريوهات واقعية. هذا التدرج يُسهّل تحديد مصدر أي مشكلة ويضمن تغطية شاملة.

يبدأ الفصل بوصف بيانات الاختبار المستخدمة: الفيديوها ومواصفاتها والسيناريوهات التي تُعطىها. ثم يتناول اختبار كل وحدة رئيسية: دقة كشف الأشخاص، واستقرار التتبع، ودقة كشف السلوكيات المختلفة. يُقيّم أيضاً الأداء الحاسوبي: سرعة المعالجة واستهلاك الموارد.

يعرض الفصل نتائج تشغيل كامل على فيديو اختباري، مع ملخص للسلوكيات المكتشفة والتنبيهات الصادرة وعينات من المخرجات. يُختتم بتحليل شامل للنتائج يُحدد نقاط القوة ونقاط الضعف والعوامل المؤثرة على الدقة.

من المهم التنويه أن التقييم الكامل لنظام كهذا يتطلب بيانات مُعلّمة (ground truth) واسعة النطاق، وهو ما يتجاوز نطاق هذا المشروع. الاختبارات المقدّمة هنا تُوفّر مؤشرات قيمة على أداء النظام، لكنها لا تُمثّل تقييماً إحصائياً شاملاً بالمعنى الأكاديمي الدقيق.

5.2 بيانات الاختبار

يعتمد تقييم النظام على فيديوها اختبارية تُحاكي بيئة قاعة الامتحان. يتناول هذا القسم وصف هذه الفيديوها ومواصفاتها التقنية والسيناريوهات التي تُعطىها.

5.2.1 وصف الفيديوهات المستخدمة

استُخدمت للاختبار فيديوهات تمثيلية تُظهر أشخاصاً يؤدون سلوكيات متنوعة في وضعية جلوس تُحاكي وضعية الامتحان. تتضمن السلوكيات المُمثَّلة: النظر للأمام بشكل طبيعي (الوضع الأساسي)، والنظر للأسفل باتجاه الحُصن، والالتفات يميناً ويساراً بدرجات مختلفة، ووضع اليد على الوجه قرب الفم أو الأذن.

صُوِّرت الفيديوهات بكاميرا ثابتة من زاوية أمامية مرتفعة قليلاً، مُحاكاةً لموقع كاميرا CCTV نموذجي في قاعة. الإضاءة مستقرة ومنظمة. عدد الأشخاص في المشهد يتراوح بين شخص واحد وعدة أشخاص لاختبار قدرة النظام على التعامل مع مشاهد متعددة.

5.2.2 مدة الفيديو وعدد الإطارات

الفيديو الاختباري الرئيسي مدته حوالي 5 دقائق، بمعدل 25 إطاراً في الثانية، أي ما يُعادل 7500 إطار تقريباً. هذه المدة كافية لاختبار استقرار التتبع وتنوع السلوكيات مع البقاء ضمن وقت معالجة معقول.

استُخدمت أيضاً مقاطع أقصر (30-60 ثانية) لاختبارات محددة: اختبار دقة الكشف في ظروف مختلفة، واختبار استجابة النظام لسلوكيات محددة، واختبار الأداء مع أعداد مختلفة من الأشخاص.

5.2.3 عدد الأشخاص في المشهد

تتضمن الاختبارات سيناريوهات بأعداد مختلفة من الأشخاص: شخص واحد لاختبار دقة استخراج الميزات بدون تداخل، وثلاثة أشخاص لاختبار التتبع المتعدد الأساسي، وحتى ستة أشخاص لاختبار قدرة النظام على التوسع.

يتضمن بعض السيناريوهات حالات تحدّي تقاطع مسارات الأشخاص، وحجب جزئي لشخص خلف آخر، ودخول وخروج أشخاص من المشهد. هذه الحالات تختبر متانة خوارزمية التتبع.

5.3 اختبار وحدة الكشف

تتولى وحدة الكشف تحديد مواقع الأشخاص في كل إطار. يُقيّم هذا القسم أداء الوحدة من حيث دقة الكشف والتعامل مع الحالات الصعبة.

5.3.1 دقة كشف الأشخاص

أظهرت وحدة الكشف المعتمدة على YOLOv8s أداءً ممتازاً في كشف الأشخاص الظاهرين بوضوح في الإطار. في الظروف الطبيعية (إضاءة جيدة، الشخص مواجه للكاميرا أو بزاوية معتدلة)، بلغت نسبة الكشف الناجح أكثر من 95%.

عتبة الثقة المستخدمة (0.40) أثبتت ملاءمتها: منخفضة بما يكفي لكشف الأشخاص البعيدين نسبياً، ومرتفعة بما يكفي لتجنب معظم الإنذارات الكاذبة. في حالات نادرة، كُشفت أجزاء من الخلفية كأشخاص (إنذارات كاذبة)، لكن هذه الحالات صُفِّيت غالباً بواسطة معايير الحجم الأدنى.

5.3.2 التعامل مع الحجب الجزئي

في حالات الحجب الجزئي (جزء من الجسم مخفي خلف عائق أو شخص آخر)، استمر النظام في كشف الشخص طالما بقي جزء كافٍ منه مرئياً. الحجب حتى 30-40% من الجسم لم يؤثر بشكل كبير على الكشف.

مع الحجب الشديد (أكثر من 50%)، بدأت نسبة الكشف بالانخفاض. في حالة الحجب شبه الكامل، فُقد الكشف كلياً. هذا سلوك متوقع ولا يُمثل مشكلة في الاستخدام الطبيعي حيث يكون معظم الأشخاص مرئيين.

5.3.3 ملاحظات الأداء

سرعة الكشف على GPU (RTX 3060) بلغت حوالي 45-50 إطاراً في الثانية للإطارات بدقة p1080، أي أسرع من الوقت الحقيقي بمرتين تقريباً. على CPU فقط، انخفضت السرعة إلى 3-5 إطارات في الثانية، وهو مقبول للمعالجة اللاحقة. استهلاك ذاكرة GPU بلغ حوالي 2 جيجابايت للنموذج والمعالجة، مما يترك مجالاً كافياً على بطاقات +4 جيجابايت.

5.4 اختبار وحدة التتبع

تتولى وحدة التتبع الحفاظ على هوية كل شخص عبر الإطارات المتتالية. يُقيّم هذا القسم استقرار الهوية والتعامل مع الحالات الصعبة.

5.4.1 استقرار هوية الأشخاص

أظهرت خوارزمية DeepSORT استقراراً جيداً في الحفاظ على هوية الأشخاص عبر الفيديو. في السيناريوهات البسيطة (أشخاص ثابتون أو بحركة محدودة)، احتفظ كل شخص بمعرّفه طوال مدة ظهوره دون أي تغيير.

في السيناريوهات الأكثر تعقيداً (تقاطع المسارات، حركة نشطة)، حافظ النظام على الهوية في معظم الحالات بفضل الميزات المظهرية التي تُميز كل شخص. ملابس وملامح كل شخص تُشكّل بصمة مظهرية يستخدمها النظام لإعادة التعرف.

5.4.2 عدد تبديلات الهوية (ID Switches)

تبديل الهوية يحدث عندما يُخصّص معرّف شخص لشخص آخر خطأً. هذا مقياس مهم لجودة التتبع.

في الاختبارات المُجرّاة على فيديو 5 دقائق مع 3-4 أشخاص، بلغ عدد تبديلات الهوية صفراً في معظم التشغيلات. في سيناريوهات التقاطع المتعمد، حدثت تبديلات نادرة (1-2 حالة) عندما تقاطع شخصان بملابس متشابهة جداً.

هذه النتائج تؤكد فعالية DeepSORT في البيئة المستهدفة، حيث يجلس الأشخاص في مقاعد ثابتة نسبياً ولا يتقاطعون كثيراً.

5.4.3 التعامل مع الاختفاء والظهور

اختُبرت قدرة النظام على إعادة التعرف على شخص بعد اختفائه المؤقت. في سيناريو الاختفاء لثانية أو ثانيتين (مروراً خلف عائق)، أُعيد ربط الشخص بمعرّفه الأصلي بنجاح في أكثر من 90% من الحالات.

مع الاختفاء الأطول (أكثر من 3 ثوانٍ بإعدادات max_age=90)، بدأت نسبة إعادة الربط الناجح بالانخفاض. بعد تجاوز فترة max_age، يُحذف المسار ويُنشأ معرّف جديد عند عودة الشخص.

5.5 اختبار كشف السلوكيات

يُقيّم هذا القسم دقة كشف كل سلوك من السلوكيات المستهدفة ومعدل الإنذارات الكاذبة.

5.5.1 دقة كشف النظر للأسفل

اختُبر كشف النظر للأسفل بسيناريوهات متنوعة: شخص ينظر لحضنه بشكل واضح، وشخص يميل قليلاً للقراءة، وشخص ينظر لورقة على الطاولة (وضع طبيعي).

أظهر النظام قدرة جيدة على التمييز بين النظر للأسفل الواضح (تُفعّل الإشارة) والإمالة الخفيفة الطبيعية (لا تُفعّل). العتبات المضبوطة (ON=0.48, OFF=0.42) حققت توازناً مناسباً.

في بعض الحالات، أدى انحناء الشخص للأمام للتركيز إلى تفعيل الإشارة رغم كونه سلوكاً طبيعياً. هذا يُعدّ من الإنذارات الكاذبة المتوقعة التي يُفترض أن يُقيّمها المراقب البشري في السياق.

5.5.2 دقة كشف الالتفات

اختُبر كشف الالتفات بدرجات مختلفة: التفات خفيف (15-20 درجة)، ومتوسط (30-45 درجة)، وكامل (60+ درجة). الالتفات الكامل كُشف بموثوقية عالية (>95%). الالتفات المتوسط كُشف بنسبة جيدة (<80%). الالتفات الخفيف غالباً لم يُفعّل الإشارة، وهو السلوك المطلوب لتجنب الإنذارات على الحركات الطبيعية. التمييز بين الالتفات يميناً ويساراً عمل بشكل صحيح، مع تسجيل الاتجاه الصحيح في جميع الحالات المختبرة.

5.5.3 دقة كشف اليد للوجه

اختُبر كشف اليد للوجه بسيناريوهات: وضع اليد على الفم/الخد، ووضع اليد على الأذن، وحكّ الوجه السريع، واليد بالقرب من الوجه دون لمسه.

فترة الثبات (ms300) نجحت في تصفية معظم حالات الحكّ السريع العابرة. الحالات التي استمرت فيها اليد قرب الوجه لفترة أطول سجّلت كأحداث.

معدل الإنذارات الكاذبة لهذا السلوك كان الأعلى مقارنة بالسلوكيات الأخرى، لأن وضع اليد على الوجه شائع لأسباب بريئة (تفكير، تعب، تعديل النظارات). لهذا أُعطي هذا السلوك وزناً أقل في حساب المخاطر.

5.5.4 الإنذارات الكاذبة

الإنذارات الكاذبة (False Positives) هي السلوكيات الطبيعية التي تُكتشف كمشبوّهة. تحليل الاختبارات أظهر:

- النظر للأسفل: إنذارات كاذبة نادرة، غالباً مع انحناء شديد للتركيز.

- الالتفات: إنذارات كاذبة قليلة، غالباً عند التفات لسؤال المراقب.

- اليد للوجه: إنذارات كاذبة متوسطة، بسبب شيوع السلوك لأسباب بريئة.

المعدل الإجمالي للإنذارات الكاذبة ظل ضمن حدود مقبولة، مع الأخذ بعين الاعتبار أن النظام مُصمّم كأداة مساعدة والقرار النهائي للمراقب البشري.

5.6 اختبار الأداء

يُقيّم هذا القسم الأداء الحاسوبي للنظام: سرعة المعالجة واستهلاك الموارد.

5.6.1 سرعة المعالجة (FPS)

قيست سرعة المعالجة على تكوينين: مع GPU (NVIDIA RTX 3060) وبدون GPU (CPU فقط - Intel i7-10700).

- مع GPU: بلغ متوسط سرعة المعالجة 25-30 إطاراً في الثانية للفيديو p1080 مع 3-4 أشخاص. هذا يعني أن النظام قادر على المعالجة في الوقت الحقيقي. مع زيادة عدد الأشخاص إلى 6-8، انخفضت السرعة إلى 18-22 fps بسبب زيادة عمليات استخراج الميزات.
- بدون GPU: انخفضت السرعة بشكل كبير إلى 2-4 إطارات في الثانية. هذا يعني أن فيديو 5 دقائق يحتاج 30-60 دقيقة للمعالجة. رغم البطء، هذا مقبول للمعالجة اللاحقة حيث لا يُشترط الوقت الحقيقي.

5.6.2 استهلاك الذاكرة

- ذاكرة RAM: استهلك النظام 4-6 جيجابايت من ذاكرة النظام أثناء التشغيل. هذا يشمل: تحميل النماذج، وتخزين إطارات الفيديو المؤقتة، وحالات المتتبع، والمخرجات المجمعة.

- ذاكرة (VRAM) GPU: استهلك النظام حوالي 2-3 جيجابايت من ذاكرة البطاقة الرسومية. النموذج yolov8s يستهلك القسم الأكبر، مع استهلاك إضافي لنموذج الميزات المظهرية في DeepSORT.

هذه المتطلبات تتوافق مع المواصفات المذكورة في فصل بيئة العمل، وتؤكد إمكانية تشغيل النظام على أجهزة متوسطة المواصفات.

5.6.3 زمن المعالجة الكلي

قيس زمن المعالجة الكلي لسيناريوهات مختلفة (شاملاً جميع المراحل من الكشف حتى توليد التقرير):

- فيديو 1 دقيقة، شخص واحد: ~1.5 دقيقة (مع GPU)

- فيديو 5 دقائق، 3 أشخاص: ~8 دقائق (مع GPU)

- فيديو 5 دقائق، 3 أشخاص: 45~ دقيقة (بدون GPU)

توليد الفيديو التوضيحي يُضيف وقتاً إضافياً يعتمد على طول الفيديو الأصلي، عادةً 1-2 ضعف مدة الفيديو.

5.7 نتائج التشغيل الكامل

يعرض هذا القسم نتائج تشغيل النظام الكامل على الفيديو الاختباري الرئيسي، كمثال توضيحي على مخرجات النظام.

5.7.1 ملخص السلوكيات المكتشفة

في تشغيل على فيديو مدته 5 دقائق يتضمن 3 أشخاص يؤدون سلوكيات متنوعة، كانت النتائج كالتالي:

الشخص الأول (s1): كُشفت 4 أحداث نظر للأسفل (مجموع المدة: 45 ثانية)، وحدثان التقات يمين (مجموع المدة: 12 ثانية)، ولا أحداث يد للوجه.

الشخص الثاني (s2): كُشف حدث واحد نظر للأسفل (8 ثوانٍ)، و3 أحداث التقات يسار (22 ثانية)، وحدث واحد يد للوجه (5 ثوانٍ).

الشخص الثالث (s3): كُشف حدثان نظر للأسفل (18 ثانية)، ولا أحداث التقات، وحدثان يد للوجه (11 ثانية).

5.7.2 التنبيهات الصادرة

بناءً على درجات المخاطر المحسوبة والعتبات المُعدّة، صدرت التنبيهات التالية:

- تنبيه مستوى "عالي الخطورة" للشخص s1 عند الدقيقة 2:15، بسبب تزامن النظر للأسفل مع الالتفات. أعلى درجة مخاطر: 0.78.

- تنبيه مستوى "مشبوه" للشخص s2 عند الدقيقة 3:40، بسبب التقات متكرر. أعلى درجة مخاطر: 0.62.

لم تصدر تنبيهات للشخص s3 رغم وجود بعض السلوكيات، لأن درجة المخاطر الممهدة لم تتجاوز العتبة لفترة كافية.

5.7.3 عينات من المخرجات

أُنْتُجَت المخرجات التالية:

- **الفيديو التوضيحي:** فيديو مدته 5 دقائق يُظهر صناديق الإحاطة الملونة حول كل شخص مع أشرطة المخاطر وأيقونات السلوكيات. لحظات التنبيهات مُمَيَّزة بصرياً.
- **مقاطع الأدلة:** استُخْرِجَ مقطعان من أعلى فترات المخاطر، كل منهما 10-15 ثانية، يُغطيان التنبيهات الصادرة.
- **تقرير PDF:** تقرير من 5 صفحات يتضمن ملخصاً تنفيذياً، ورسوماً بيانية للسلوكيات والمخاطر، وجدول التنبيهات، وتوصيات المراجعة.
- **ملفات البيانات:** ملفات JSONL تحتوي الميزات المستخرجة (7500 سجل)، وأحداث السلوكيات (12 حدث)، ودرجات المخاطر (7500 سجل)، والتنبيهات (2 تنبيه).

5.8 تحليل النتائج

يُلَخَّص هذا القسم نتائج الاختبارات ويُحدد نقاط القوة والضعف والعوامل المؤثرة على أداء النظام.

5.8.1 نقاط القوة

أظهرت الاختبارات عدة نقاط قوة للنظام:

- **دقة كشف عالية:** نموذج YOLOv8s يكشف الأشخاص بدقة ممتازة في الظروف الطبيعية، مع قدرة جيدة على التعامل مع الحجب الجزئي.
- **استقرار التتبع:** خوارزمية DeepSORT تُحافظ على هوية الأشخاص بشكل موثوق، مع قدرة على إعادة التعرف بعد الاختفاء المؤقت.
- **توازن في كشف السلوكيات:** العتبات المضبوطة وآلية Hysteresis تُحقق توازناً معقولاً بين الحساسية وتجنب الإنذارات الكاذبة.
- **أداء مقبول:** سرعة المعالجة كافية للوقت الحقيقي مع GPU، ومقبولة للمعالجة اللاحقة بدون GPU.
- **مخرجات شاملة ومفيدة:** تنوع المخرجات (فيديو، مقاطع، تقرير، بيانات) يلبي احتياجات مختلفة.

5.8.2 نقاط الضعف

حُددت أيضاً نقاط ضعف تستحق المعالجة:

- حساسية للإضاءة: الأداء ينخفض بشكل ملحوظ في ظروف الإضاءة السيئة أو المتقلبة.
- إنذارات كاذبة لليد للوجه: هذا السلوك يُنتج إنذارات كاذبة أكثر من غيره بسبب شيوعه لأسباب بصرية.
- اعتماد على رؤية الوجه: عندما يكون الوجه غير مرئي (التفات شديد)، تقل دقة بعض الإشارات.
- عدم فهم السياق: النظام لا يُميز بين الالتفات لسؤال المراقب والالتفات للنظر في ورقة الجار.

5.8.3 العوامل المؤثرة على الدقة

حُددت عدة عوامل تؤثر بشكل كبير على دقة النظام:

- جودة الفيديو: الدقة العالية والإضاءة الجيدة تحسّن جميع مراحل المعالجة.
 - زاوية الكاميرا: الزاوية الأمامية المرتفعة قليلاً مثالية؛ الزوايا الجانبية أو الخلفية تقلل الدقة.
 - المسافة عن الكاميرا: الأشخاص البعيدون جداً يظهرون صغاراً ويصعب استخراج ميزاتهم بدقة.
 - تشابه الأشخاص: الأشخاص بملابس متشابهة جداً قد يُسببون أخطاء في التتبع.
 - ضبط العتبات: العتبات تحتاج ضبطاً حسب البيئة؛ قيم غير مناسبة تؤثر سلباً على التوازن.
- هذه العوامل يجب مراعاتها عند نشر النظام، ويُنصح بإجراء ضبط أولي على فيديوهات من البيئة الفعلية قبل الاستخدام الإنتاجي.

الفصل السادس: الخاتمة والتوصيات

6.1 ملخص المشروع

قدّم هذا المشروع نظام Proctor AI، وهو نظام ذكي لدعم قرار مراقبة الامتحانات يعتمد على تقنيات الرؤية الحاسوبية والتعلم العميق. يهدف النظام إلى مساعدة المراقبين البشريين من خلال الكشف التلقائي عن السلوكيات التي قد تُشير إلى محاولات غش، وتقديم هذه المعلومات بشكل منظم وموثّق لدعم اتخاذ القرار.

انطلق المشروع من إدراك التحديات التي تواجه المراقبة التقليدية للامتحانات: محدودية قدرات المراقب البشري على التغطية الشاملة والمستمرة، وصعوبة التوثيق الموضوعي للسلوكيات، والإرهاق الذي يُضعف الأداء مع الوقت. في المقابل، تُوفّر التقنيات الحديثة في الرؤية الحاسوبية والتعلم العميق أدوات قوية يُمكن توظيفها لتعزيز قدرات المراقبة.

يعمل النظام على تحليل فيديوهات قاعات الامتحان عبر خط معالجة متعدد المراحل. تبدأ العملية بكشف الأشخاص في كل إطار باستخدام نموذج YOLOv8، ثم تتابعهم عبر الإطارات باستخدام خوارزمية DeepSORT. تُستخرج بعد ذلك ميزات الوجه والجسم باستخدام MediaPipe، وتُحوّل إلى إشارات سلوكية تُعبّر عن النظر للأسفل والالتفات ووضع اليد على الوجه. تُحوّل الإشارات المستمرة إلى أحداث سلوكية منفصلة باستخدام آلية Hysteresis التي تمنع التذبذب. تُجمّع درجات السلوكيات في درجة خطورة شاملة تُمهدّ زمنياً باستخدام EWMA. يُصدر النظام تنبيهات عند تجاوز العتبات المحددة، ويُنتج مخرجات متنوعة: فيديو توضيحي، ومقاطع أدلة، وتقرير PDF، وملفات بيانات.

الترّم المشروع بفلسفة واضحة: النظام أداة لدعم القرار لا لاتخاذ. لا يُصدر النظام حكماً بالغش، بل يُشير إلى سلوكيات تستحق المراجعة ويترك القرار النهائي للمراقب البشري الذي يستطيع فهم السياق الكامل. هذا النهج يُراعي الاعتبارات الأخلاقية ويضمن العدالة في التعامل مع الطلاب.

6.2 ملخص الإنجازات

يُلخّص هذا القسم الأهداف التي تحققت والمساهمة العلمية للمشروع.

6.2.1 الأهداف المحققة

تمت مراجعة الأهداف الموضوعية في بداية المشروع ونُقيّم مدى تحقق كل منها:

- الهدف الأول - وحدة الكشف والتتبع: تحقق بالكامل. بُنيت وحدة تكشف الأشخاص بدقة عالية وتتبعهم بهوية مستقرة عبر الفيديو.
- الهدف الثاني - وحدة استخراج الميزات: تحقق بالكامل. بُنيت وحدة تستخرج نقاط الوجه والجسم وتحسب إشارات السلوك منها.
- الهدف الثالث - وحدة كشف السلوكيات: تحقق بالكامل. بُنيت وحدة تُحوّل الإشارات إلى أحداث باستخدام Hysteresis.
- الهدف الرابع - نظام تقييم المخاطر: تحقق بالكامل. بُني نظام يحسب درجة خطورة موزونة مع تمهيد EWMA وتصنيف المستويات.
- الهدف الخامس - نظام التنبيهات: تحقق بالكامل. بُني نظام يُصدر تنبيهات بشروط محددة مع Cooldown و Hold Time.
- الهدف السادس - المخرجات المتعددة: تحقق بالكامل. طُوّرت أدوات لتوليد فيديو توضيحي ومقاطع أدلة وتقرير PDF وملفات بيانات.
- الهدف السابع - لوحة التحكم: تحقق بالكامل. بُنيت واجهة ويب تفاعلية لاستعراض النتائج.
- الهدف الثامن - التوثيق: تحقق بالكامل. أُعد توثيق شامل للمشروع.

6.2.2 المساهمة العلمية

يُقدّم المشروع عدة مساهمات:

- نموذج تطبيقي متكامل: يُوفّر المشروع مثلاً عملياً على بناء نظام دعم قرار يجمع عدة تقنيات (YOLO, DeepSORT, MediaPipe, Hysteresis, EWMA) في خط معالجة موحد.
- منهجية لكشف السلوك: يُقدّم منهجية لتحويل الإشارات المستمرة إلى أحداث سلوكية باستخدام Hysteresis، قابلة للتطبيق في سياقات أخرى.
- توازن بين الأتمتة والحكم البشري: يُقدّم نموذجاً للتعامل مع التحدي الأخلاقي للأتمتة في سياقات الحكم على السلوك.

6.3 التحديات والحلول

واجه المشروع عدة تحديات تقنية ومنهجية، وطُبِّقَت حلول للتغلب عليها.

6.3.1 التحديات التقنية

- **تحدي التذبذب في الإشارات:** الإشارات المستخرجة من MediaPipe تتذبذب بشكل طبيعي بسبب الضوضاء وعدم الاستقرار في الكشف. هذا كان يُسبب تسجيل أحداث بداية ونهاية متكررة للسلوك الواحد.
- **تحدي التوازن بين الحساسية والنوعية:** ضبط عتبات الكشف يتطلب توازناً دقيقاً. عتبات منخفضة جداً تُنتج إنذارات كاذبة كثيرة، وعتبات عالية جداً تُفوت سلوكيات فعلية.
- **تحدي الأداء:** تشغيل عدة نماذج تعلم عميق (YOLO, DeepSORT features, MediaPipe) في الوقت الحقيقي يتطلب إدارة دقيقة للموارد.
- **تحدي تنوع المدخلات:** الفيديوهات قد تأتي بدقات ومعدلات إطارات مختلفة، ويجب أن يتعامل النظام معها جميعاً.

6.3.2 الحلول المطبقة

- **حل التذبذب - آلية Hysteresis:** استخدام عتبتين (ON و OFF) بدلاً من عتبة واحدة يمنع التذبذب عند حدود القيم. هذا الحل بسيط وفعال ولا يتطلب موارد إضافية.
- **حل التوازن - ضبط تجريبي مع قابلية التعديل:** ضُبِطَت العتبات تجريبياً للحصول على توازن معقول، مع جعلها قابلة للتعديل من ملفات الإعدادات. هذا يُتيح ضبطاً خاصاً لكل بيئة.
- **حل الأداء - معالجة محسنة:** استخدام Dynamic ROI يُقلل حجم البيانات المُعالجة لكل شخص. استغلال GPU يُسرّع المعالجة بشكل كبير. المعالجة التدفقية (إطار بإطار) تُقلل استهلاك الذاكرة.
- **حل تنوع المدخلات - تطبيع وتكيف:** يُقرأ معدل الإطارات من الفيديو ويُستخدم لحساب الطوابع الزمنية الصحيحة. الأحجام تُطَبَّع عند المعالجة. الإعدادات تسمح بتحديد معدل إطارات مفترض إذا لم يكن متوفراً.

حلول إضافية:

- فترة الثبات (Hold Time) لسلوك اليد للوجه تُصَفِّي الحركات العابرة.
- التمهيد بـ EWMA يُوفّر استقراراً لدرجات المخاطر.

- فترة التهدئة (Cooldown) تمنع تكرار التنبيهات المزعج.

6.4 العمل المستقبلي

يفتح المشروع الحالي آفاقاً عديدة للتطوير والتوسع المستقبلي.

6.4.1 تحسينات ممكنة

- تحسين دقة كشف السلوكيات: يُمكن تدريب نماذج مخصصة للتعرف على أنماط السلوك المشبوه بدلاً من الاعتماد على قواعد ثابتة. نماذج التعلم العميق للتصنيف الزمني (مثل LSTM أو Transformer) قد تلتقط أنماطاً أكثر تعقيداً.
- تحسين الأداء: تحسين الشيفرة للاستفادة من التوازي بشكل أفضل. استكشاف نماذج أخف وزناً (مثل yolov8n) مع ضبط دقيق للأجهزة محدودة الموارد. تنفيذ معالجة الدفعات (batch processing) للفيديوهات المتعددة.
- تحسين واجهة المستخدم: إضافة إمكانية المراجعة التفاعلية مع تعليقات المراقب. دعم المعالجة المباشرة (live) مع عرض النتائج آنياً. تطبيق جوال للمراقبة عن بُعد.

6.4.2 ميزات إضافية مقترحة

- كشف الأجهزة الإلكترونية: تدريب نموذج للكشف عن الهواتف المحمولة والسماعات الصغيرة، وهو تحدّي تقني لكنه قابل للتحقيق مع بيانات تدريب مناسبة.
- تحليل التفاعل بين الأشخاص: كشف التواصل المحتمل بين طلاب متجاورين (نظرات متبادلة، إيماءات).
- تحليل الصوت: دمج تحليل الصوت للكشف عن الهمس أو الحديث غير المسموح.

6.4.3 توسيع نطاق المشروع

- دعم كاميرات متعددة: معالجة فيديوهات من عدة كاميرات تغطي قاعة واحدة، مع دمج النتائج.
- دعم الامتحانات الإلكترونية: تكييف النظام للعمل مع كاميرا الويب لمراقبة الامتحانات عن بُعد.
- التكامل مع أنظمة المؤسسة: واجهات برمجية (API) للتكامل مع أنظمة إدارة الامتحانات الموجودة.

- النشر السحابي: نسخة تعمل على السحابة لتسهيل الاستخدام دون الحاجة لتثبيت محلي.

6.5 التوصيات

بناءً على الخبرة المكتسبة من تطوير واختبار النظام، تُقدّم التوصيات التالية للمستخدمين المحتملين والمطورين والباحثين.

توصيات للمستخدمين:

أولاً، يجب فهم طبيعة النظام كأداة مساعدة وليس حاكماً. النظام يُشير إلى سلوكيات تستحق المراجعة، لكن القرار النهائي يبقى للمراقب البشري الذي يستطيع فهم السياق الكامل.

ثانياً، يُنصح بإجراء ضبط أولي للعتبات على فيديوهات من البيئة الفعلية قبل الاستخدام الإنتاجي. القيم الافتراضية قد تحتاج تعديلاً حسب خصائص القاعة والكاميرا والإضاءة.

ثالثاً، يجب ضمان جودة كافية للفيديو: إضاءة جيدة ومستقرة، وكاميرا ثابتة بزاوية مناسبة، ودقة لا تقل عن 720p.

رابعاً، يُنصح بإعلام الطلاب بوجود نظام المراقبة، التزاماً بمبادئ الشفافية والأخلاقيات.

توصيات للمطورين:

أولاً، البنية المعيارية للمشروع تُسهّل التعديل والتوسيع. يُمكن استبدال أي مكون (الكاشف، المتتبع، مستخرج الميزات) بتعديلات محدودة.

ثانياً، ملفات الإعدادات YAML تُتيح تجريب معاملات مختلفة دون تعديل الشيفرة. يُنصح بالاستفادة من هذه المرونة.

ثالثاً، ملفات JSONL المُخرجة تُوفّر بيانات غنية للتحليل الإضافي. يُمكن بناء أدوات تحليل مخصصة تستفيد منها.

توصيات للباحثين:

أولاً، المشروع يُوفّر نقطة انطلاق للبحث في تحليل السلوك وكشف الغش. يُمكن البناء عليه لاختبار خوارزميات أو نماذج جديدة.

ثانياً، مجال كشف الغش يحتاج مجموعات بيانات معيارية (benchmark datasets) للتقييم الموضوعي. إنشاء مثل هذه المجموعات مساهمة قيّمة.

ثالثاً، الجانب الأخلاقي لأنظمة المراقبة الآلية يستحق البحث والنقاش. التوازن بين الفعالية والخصوصية والعدالة موضوع مهم.

References