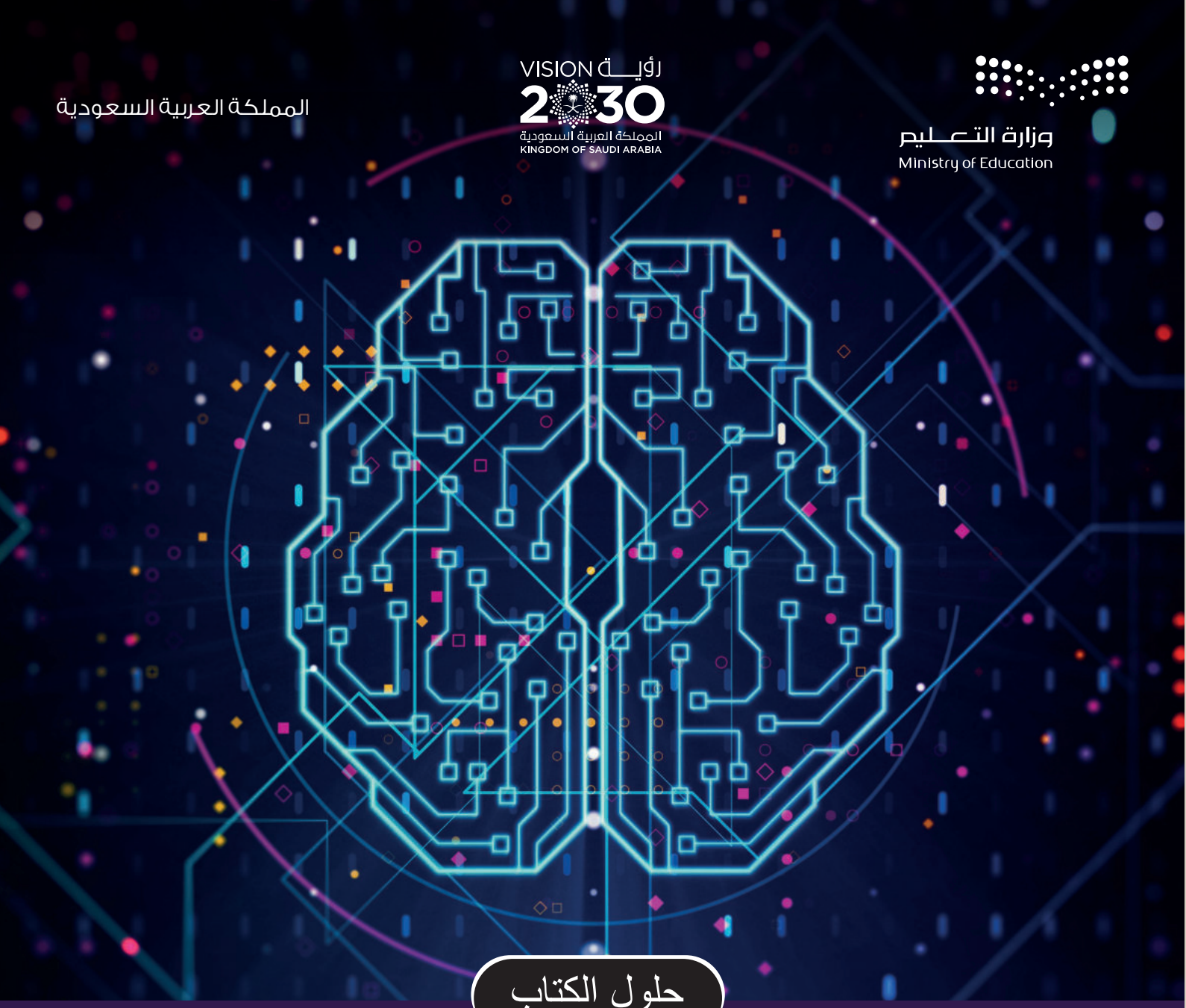
المملكة العربية السعودية

رؤية VISION 2030
المملكة العربية السعودية
KINGDOM OF SAUDI ARABIA

وزارة التعليم
Ministry of Education



حلول الكتاب

الذكاء الاصطناعي 1-2

Artificial Intelligence

وزارة التعليم
Ministry of Education
2023 - 1445

binarylogic

السنة الثالثة
التعليم الثانوي - نظام المسارات

طبعة 2023-1445

4. التعرّف على الصور

سيتعرّف الطالب في هذه الوحدة على التعلُّم الموجَّه وغير الموجَّه، وكيفية توظيفهما للتعرّف على الصور (Image Recognition) عن طريق إنشاء نموذج وتدريبه؛ ليصبح قادرًا على تصنيف صور لرؤوس الحيوانات أو تجميعها. وسيتعرّف أيضًا على توليد الصور (Image Generation) وكيفية تغييرها، أو إكمال الأجزاء الناقصة فيها مع الحفاظ على واقعيتها.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يُعالج الصور معالجة أولية ويستخلص خصائصها.
- يُدرِّب نموذج تعلُّم موجَّه خاص بتصنيف الصور.
- يُعرِّف هيكل الشبكة العصبية.
- يُدرِّب نموذج تعلُّم غير موجَّه خاص بتجميع الصور.
- يولِّد صورًا بناءً على توجيه نصّي.
- يُكمل الأجزاء الناقصة في صورة مُعطاة بطريقة واقعية.

الأدوات

- مفكرة جوبيتر (Jupyter Notebook)
- قوقل كولاب (Google Colab)

وزارة التعليم
Ministry of Education
2023 - 1445

يمكن تقديم إجابات إضافية من قبل الطلبة

تمرينات

1 ما تحدّيات تصنيف البيانات المرئية؟

تلميح: وجّه الطلبة لاستكشاف الإجابة من خلال الصفحة رقم 198 في كتاب الطالب.

2 لديك مصفوفتا قيم Numpy، وهما مصفوفة X_train ومصفوفة Y_train. كل صف في مصفوفة X_train شكله (100،100،3) يمثّل صورة بأبعاد 100x100 وبتنسيق RGB. والصف n في المصفوفة Y_train يمثّل تسمية صورة n في مصفوفة X_train. أكمل المقطع البرمجي التالي، بحيث يُسطّح X_train ثم يُدرّب النموذج MultinomialNB على مجموعة البيانات هذه:

```
from sklearn.naive_bayes import MultinomialNB # imports the Naive Bayes Classifier from sklearn

X_train_flat = np.array ([img.flatten() for img in X_train])

model_MNB = MultinomialNB() # new Naive Bayes model

model_MNB.fit( X_train_flat , y_train ) # fits model on the flat training data
```

3 صِف باختصار طريقة عمل الشبكات العصبية الترشيحية وإحدى مميزاتها الرئيسة.

تلميح: وجّه الطلبة لاستكشاف الإجابة من خلال الصفحات رقم (215-216) في كتاب الطالب.

4 لديك مصفوفتا قيم Numpy، وهما مصفوفة X_train ومصفوفة Y_train. كل صف في مصفوفة X_train شكله (100،100،3) يمثّل صورة بأبعاد 199x100 وبتنسيق RGB. والصف n في المصفوفة Y_train يمثّل تسمية صورة n في مصفوفة X_train. أكمل المقطع البرمجي التالي، بحيث يطبّق تحويلات المخطط التكراري للتدرجات الموجّهة ثم يستخدم البيانات المحوّلة في تدريب نموذج :

```
from skimage.color import ____rgb2gray____ # used to convert a multi-color (rgb) image to grayscale
from sklearn.____preprocessing____ import StandardScaler # used to scale the data
from sklearn.naive_bayes import MultinomialNB # imports the Naive Bayes Classifier from sklearn
X_train_gray = np.array([____rgb2gray____(img) for img in X_train]) # converts training data
X_train_hog = ____np.array([hog(img) for img in X_train_gray])____
scaler = StandardScaler()
X_train_hog_scaled = ____scaler____.fit_transform(X_train_hog)
model_MNB = MultinomialNB()
model_MNB.fit(X_train_flat_scaled, ____y_train____)
```

5 اذكر بعض تحديّات الشبكات العصبية الترشيحية.

تلميح: وجّه الطلبة لاستكشاف الإجابة من خلال الصفحة رقم 216 في كتاب الطالب.

يمكن تقديم إجابات إضافية من قبل الطلبة

تمرينات

1 اذكر الميزة التي تتمتع بها تقنيات التعلُّم غير الموجَّه مقارنة بتقنيات التعلُّم الموجَّه في تحليل الصور.

تلميح: وجّه الطلبة لاستكشاف الإجابة من خلال الصفحة رقم 220 في كتاب الطالب.

2 لديك مصفوفة قيم موحدة X_flat تشمل صورًا مُسطحة، وكل صف في المصفوفة يمثّل صورة مسطحة مُختلفة على هيئة متتالية من الأعداد الصحيحة تتراوح بين 0 و255. أكمل المقطع البرمجي التالي، بحيث يستخدم التجميع التكتلي في تصنيف الصور التي من X_flat إلى خمسة عناقيد مُختلفة:

```
from sklearn.cluster import AgglomerativeClustering # used for agglomerative clustering
AC = AgglomerativeClustering(linkage='ward', n_clusters = 5 )
X_norm = X_flat / 255 # normalizes the data
AC.fit(X_norm) # applies the tool to the data
pred = AC.labels # gets the cluster labels
```

3 عدّد بعض مزايا استخدام التعلُّم العميق التي يمتاز بها على طرائق تجميع الصور التقليدية.

تلميح: وجّه الطلبة لاستكشاف الإجابة من خلال الصفحات رقم (221، 230) في كتاب الطالب.

يمكن تقديم إجابات إضافية من قبل الطلبة

4 لديك مصفوفة قيم موحدة X_flat تشمل صورًا مسطحة، وكل صف في المصفوفة يمثّل صورة مسطحة مُختلفة على هيئة متتالية من الأعداد الصحيحة تتراوح بين 0 و255. أكمل المقطع البرمجي التالي، بحيث يستخدم طريقة وارد (ward) لإنشاء وتصوير رسم شجري للصور في هذه المصفوفة:

```
import scipy.cluster.hierarchy as hierarchy # visualizes and supports hierarchical clustering tasks

import matplotlib.pyplot as plt

X_norm = X_flat / 255 # normalizes the data

plt.figure() # creates a new empty figure

linkage_flat=hierarchy.linkage(X_norm, method='ward')

hierarchy.dendrogram(linkage_flat)

plt.show() #shows the figure
```

5 صِف الطريقة التي يُطبَّق بها التجميع بالشبكات العصبية في تحليل الصور.

تلميح: وجّه الطلبة لاستكشاف الإجابة من خلال الصفحة رقم (221، 230) في كتاب الطالب.

يمكن تقديم إجابات إضافية من قبل الطلبة

تمرينات

1 صِف باختصار عملية رسم صورة بالاسترشاد بنص.

تلميح: وجّه الطلبة لاستكشاف الإجابة من خلال الصفحة رقم 236 في كتاب الطالب.

2 صِف عملية تدريب نماذج الانتشار المستقر.

تلميح: وجّه الطلبة لاستكشاف الإجابة من خلال الصفحة رقم 239 في كتاب الطالب.

يمكن تقديم إجابات إضافية من قبل الطلبة

3 صِف المولِّد والمميِّز في الشبكة التوليدية التنافسية.

تلميح: وجِّه الطلبة لاستكشاف الإجابة من خلال الصفحة رقم 239 في كتاب الطالب.

4 استخدم أداة DiffusionPipeline من مكتبة diffusers لإنشاء صورة لحيوانك المفضل وهو يأكل طعامك المفضل. يمكنك استخدام منصة قوقل كولاب في هذه المُهِمَّة.

تلميح: الإجابة مشابهة للمثال الوارد في الصفحة رقم 240؛ ولكن بدلًا من "صورة أسد أبيض في الغابة." استخدم مثل: "صورة قطة تأكل سمكة".

5 استخدم أداة StableDiffusion2ImagePipleline من مكتبة diffusers لتحويل الحيوان في الصورة المرسومة في التدريب السابق إلى حيوان آخر من اختيارك. يمكنك استخدام منصة قوقل كولاب في هذه المُهِمَّة.

تلميح: الإجابة مشابهه للمثال الوارد في الصفحة رقم 241.

وزارة التعليم
Ministry of Education
2024-1445

5. خوارزميات التحسين واتخاذ القرار

سيتعرّف الطالب في هذه الوحدة على عدة خوارزميات وتقنيات تساعده في إيجاد أكثر الحلول كفاءة لمشكلات التحسين المعقدة، كما سيتعلّم طريقة عمل خوارزميات التحسين، وخوارزميات اتخاذ القرار، وطريقة تطبيقها لحلّ مشكلات متعلقة بالعالم الواقعي ترتبط بتخصيص الموارد والجدولة وتحسين المسارات.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يُصنّف طرائق التحسين لمعالجة مشكلات معقدة.
- يَصِف خوارزميات اتخاذ القرار المُختلفة.
- يَستخدم البايثون لحلّ مشكلات تخصيص الموارد المتعلقة بفِرق العمل.
- يَحلّ مشكلات الجدولة باستخدام خوارزميات التحسين.
- يَستخدم البايثون لحلّ مشكلات الجدولة.
- يَستخدم البرمجة الرياضية لحلّ مشكلات التحسين.
- يُعرّف مشكلة حقيبة الظهر (Knapsack problem).
- يُعرّف مشكلة البائع المُتجوّل (Traveling Salesman problem).

الأدوات

- مفكّرة جوبيتر (Jupyter Notebook)

وزارة التعليم
Ministry of Education
2023 - 1445

يمكن تقديم إجابات إضافية من قبل الطلبة

تمرينات

1 ما مزايا وعيوب استخدام كلٍّ من: خوارزمية القوة المفرطة والخوارزمية الاستدلالية الجشعة في حلّ مشكلات التحسين؟

تلميح: يمكنك توجيه الطلبة للرجوع إلى الصفحة رقم 252 والصفحة رقم 260 من كتاب الطالب للحصول على الإجابة.

2 حلِّلْ طريقة استخدام الخوارزميات الاستدلالية الجشعة لإيجاد الحلول المُثلى في مشكلات التحسين.

تلميح: يمكنك توجيه الطلبة للرجوع إلى الصفحة رقم 253 والصفحة رقم 258 من كتاب الطالب للحصول على الإجابة.

3 أنشئ خوارزمية حلّ جشعة لتحسين مشكلة تكوين أعضاء فريق، من خلال إكمال المقطع البرمجي التالي بحيث تستخدِم خوارزمية الحلّ الاستدلالية الجشعة لتكليف أعضاء الفريق بالمُهِمَّة:

```
def greedy_solver(problem):
    worker_skills=problem['worker_skills'] # worker skills for this problem
    required_skills=problem['required_skills'] # required skills for this problem

    uncovered_required_skills = required_skills.copy() # skills not covered
    best_team=[] # best solution
    uncovered_worker_skills={}
    for worker_id in worker_skills:

        uncovered_worker_skills[worker_id]=worker_skills[worker_id].intersection
(uncovered_required_skills)
    while len(uncovered_required_skills) > 0:

        best_worker_id=None # the best worker to add next
        best_new_coverage=0 # number of uncovered required skills covered by the best worker
        for worker_id in uncovered_worker_skills: # for each worker
            my_uncovered_skills=uncovered_worker_skills[worker_id]
            # if this worker can cover more uncovered required skills than the best worker so far
            if  len(my_uncovered_skills)>best_new_coverage:
                best_worker_id=worker_id # makes this worker the best worker

                best_new_coverage=len(my_uncovered_skills)

        if best_worker_id!=None : # if a best worker was found

            best_team.append(best_worker_id) # adds the worker to the solution
            #removes the best worker's skills from the skills to be covered
            uncovered_required_skills=uncovered_required_skills - uncovered_
worker_skills[best_worker_id]
            # for each worker
            for worker_id in uncovered_worker_skills:

                # remembers only the required uncovered skills that this worker has

                uncovered_worker_skills[worker_id]=uncovered_worker_
skills[worker_id].intersection(uncovered_required_skills)
        else: # no best worker has been found and some required skills are still uncovered
            return None # no solution could be found
    return best_team
```

يمكن تقديم إجابات إضافية من قبل الطلبة

4 اذكر ثلاث مشكلات تحسين مُختلفة من العالم الواقعي، وفي كل مشكلة:

- اضرب مثالًا على دالة موضوعية.
- اضرب مثالين على القيود إن وُجِدَتْ.

تلميح: يمكنك توجيه الطلبة للرجوع إلى الصفحة رقم 250 من كتاب الطالب للحصول على الإجابة.

5 إذا قمتَ بزيادة عدد العمّال في خوارزمية القوة المُفرطة، كيف يؤثر ذلك على المشكلة من حيث عدد الحلول والزمن الحسابي؟

تلميح: يمكنك توجيه الطلبة للرجوع إلى الصفحة رقم 258 من كتاب الطالب للحصول على الإجابة.

وزارة التعليم
Ministry of Education
2022-1445

يمكن تقديم إجابات إضافية من قبل الطلبة

تمرينات

1 صِف استراتيجيتين مُختلفتين (مبادلة، انعكاس، تحويل، إلخ) لأسلوب البحث المحلي لِحلّ مشكلة التباطؤ الموزون للآلة الواحدة.

تُعدُّ المبادلة (Swapping) خطوة أساسية في نهج البحث المحلي، حيث يتم تبديل عنصرين من الحلّ لإنشاء حلّ جديد، وفي سياق مشكلة التباطُؤ الموزون للآلة الواحدة (SMWT) فإن هذا يعني تبديل ترتيب موقعين زارتهما شاحنة واحدة. على سبيل المثال، إذا قامت الشاحنة A بزيارة المواقع 6-5-4-3-2-1 في البداية، فقد تؤدي حركة التبديل إلى المسار الجديد 6-5-3-4-2-1، وتستكشف استراتيجية المبادلة تجميعات مختلفة لإيجاد حلّ أفضل.

تتضمن استراتيجية الانعكاس (Inversion) عكس جزء من مسار الحلّ، وهذا يعني اختيار مجموعة فرعية مستمرة من المواقع التي تزورها الشاحنة وعكس ترتيبها. على سبيل المثال، إذا قامت الشاحنة B في البداية بزيارة المواقع 6-5-4-3-2-1 ، فقد تؤدي الحركة المعكوسة إلى المسار الجديد 6-3-4-5-2-1.

تتضمن استراتيجية التحويل (Shifting) نقل جزء من المسار إلى موضع مختلف داخل مسار الشاحنة نفسه، وهذا يعني تحديد مجموعة فرعية من المواقع المتتالية ونقلها إلى موقع آخر في نفس المسار. على سبيل المثال، إذا قامت الشاحنة C في البداية بزيارة المواقع 6-5-4-3-2-1، فقد تؤدي حركة التحويل إلى المسار الجديد 6-3-2-5-4-1، وتساعد استراتيجية التحويل في استكشاف تجميعات بديلة من المواقع داخل المسار.

2 كم عدد الجداول المُمكنة (الحلول) لنسخة مشكلة التباطؤ الموزون للآلة الواحدة والتي تشتمل على تسع مهام؟

يتم إعطاء عدد التباديل لعدد n من العناصر المميزة بواسطة n! وهو مضروب n، ويُعرف مضروب عدد صحيح موجب n بأنه حاصل ضرب جميع الأعداد الصحيحة الموجبة من 1 إلى n.

بالنسبة إلى n = 9 ، فإن عدد الجداول المُمكنة (الحلول) هو:

$9! = 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 362,880.$

3 أنشئ خوارزمية حلّ بالقوة المُفرطة لمشكلة التباطؤ الموزون للآلة الواحدة، من خلال إكمال المقطع البرمجي التالي بحيث تستخدم الدالةُ القوةَ المُفرطة لإيجاد تبديل الجدولة الأمثل.

```
def brute_force_solver(problem):
    # gets the information for this problem
    durations, weights, deadlines=problem['durations'], problem['weights'],
problem['deadlines']

    job_num = len(durations) # number of jobs
    # generates all possible schedules

    all_schedules = itertools.permutations(range(job_num))
    # initializes the best solution and its total weighted tardiness

    best_schedule = None # initialized to None
    # 'inf' stands for 'infnity'. Python will evaluate all numbers as smaller than this value.

    best_tardiness = float('inf')
    # stores the finish time of each job in the best schedule

    best_finish_times= None # initalized to None

    for schedule in all_schedules: # for every possible schedule
        #evalute the schedule
        tardiness,finish_times=compute_schedule_tardiness(problem, schedule)
        if tardiness<best_tardiness: # this schedule is better than the best so far

            best_tardiness= tardiness

            best_schedule= schedule

            best_finish_times= finish_times

    # return the results as a dictionary
    return {'schedule':best_schedule,
            'tardiness':best_tardiness,
            'finish_times':best_finish_times}
```

4 أنشئ خوارزمية حلّ البحث المحلّي لمشكلة التباطؤ الموزون للآلة الواحدة، من خلال إكمال المقطع البرمجي التالي بحيث تستخدم الدالةُ البحث المحلّي لإيجاد تبديل الجدولة الأمثل.

```
def local_search_solver(problem, greedy_heuristic, swap_selector, max_
iterations):
    # gets the information for this problem
    durations, weights, deadlines=problem['durations'], problem['weights'],
problem['deadlines']

    job_num = len( durations )# gets the number of jobs
    # uses the greedy solver to get a first schedule.
    # this schedule will be then iteratively refined through local search

    greedy_sol = greedy_solver (problem, greedy_heuristic) # remembers the best
schedule so far
    best_schedule, best_tardiness, best_finish_times=greedy_
sol['schedule'],greedy_sol['tardiness'],greedy_sol['finish_times']

    # local search
    for i in range( max_iterations ): # for each of the given iterations
        # chooses which two positions to swap

        pos1,pos2= swap_selector (best_schedule)

        new_schedule = best_schedule. copy ()# creates a copy of the
schedule
        # swaps jobs at positions pos1 and pos2
        new_schedule[pos1], new_schedule[pos2] = best_schedule[pos2], best_
schedule[pos1]
        # computes the new tardiness after the swap
        new_tardiness, new_finish_times = compute_schedule_tardiness(problem,
new_schedule)
        # if the new schedule is better than the best one so far
        if new_tardiness < best_tardiness:
            # the new_schedule becomes the best one

            best_schedule = new_schedule

            best_tardiness = new_tardiness

            best_finish_times= new_finish_times

    # returns the best solution
    return {'schedule':best_schedule,
            'tardiness':best_tardiness,
            'finish_times':best_finish_times}
```

يمكن تقديم إجابات إضافية من قبل الطلبة

5 صِف طريقة عمل البحث المحلّي.

في البحث المحلّي، يُعدَّل الحلّ الذي تم التوصل إليه في البداية بشكلٍ متكرر من خلال فحص الحلول المجاورة التي وُجِدت عن طريق إجراء تعديلات بسيطة على الحلّ الحالي. بالنسبة للعديد من مشكلات التحسين، فهناك طريقة شائعة لتعديل الحلّ تتمثل في تبديل العناصر بشكل متكرر. على سبيل المثال، في مشكلة تكوين الفريق سيحاول أسلوب البحث المحلّي إنشاء فريق أفضل وذلك من خلال تبديل أعضاء الفريق بالعمّال الذين لا يُعدّون حاليًا جزءًا من الفريق.

6 اكتب ملاحظاتك عن نتائج خوارزميات الحلّ الجشعة مقارنة بخوارزميات حلّ البحث المحلّي في مشكلة تشتمل على ثلاثين مُهِمَّة. من وجهة نظرك، لماذا لم تُستخدم خوارزمية حلّ القوة المُفرطة في هذه المشكلة المكوّنة من ثلاثين مُهِمَّة؟

يمكن لخوارزميات حلّ البحث المحلي التعامل مع حالات المشكلات الكبيرة بكفاءة، فهي لا تتطلب فحص جميع الحلول المُمكنة، مما يجعلها مناسبة للمشكلات المعقدة مثل: مشكلة جدولة 30 مُهِمَّة، ومع ذلك فإن حلول البحث المحلية لها قيود مثل: الاعتماد على جودة الحلّ الأولي، وإمكانية التعثر في الحلول دون مستوى الحلّ الأمثل خاصة إذا لم يكن استكشاف الحلول المجاورة شاملًا.

تقوم خوارزميات الحلّ الجشعة باتخاذ أفضل الخيارات محليًا في كل خطوة من خطوات عملية الجدولة، فهي تعطي الأولوية للمكاسب الفورية دون النظر في تأثيرها طويل المدى على الحل الشامل.

تقوم خوارزميات حلّ القوة المُفرطة بتعداد جميع التجميعات الممكنة من الجداول بشكل شامل وتقييم كل منها بناءً على الوظيفة الموضوعية، بينما يضمن هذا النهج إيجاد الحل الأمثل، غير أنه يصبح غير عملي مع زيادة حجم المشكلة.

يمكن تقديم إجابات إضافية من قبل الطلبة

تمرينات

1 اشرح طريقة استخدام البرمجة الرياضية لحلّ مشكلات التحسين المعقدة.

يُمكن للبرمجة الرياضية أن تحلّ العديد من مشكلات التحسين مثل: تخصيص الموارد، وتخطيط الإنتاج، والخدمات اللوجستية والجدولة، وتتميز هذه التقنية بأنها تُوفّر حلًا مثاليًا مضمونًا ويُمكنها التعامل مع المشكلات المعقدة ذات القيود المتعددة. يبدأ حلّ البرمجة الرياضية بصياغة مشكلة التحسين المُعطاة على شكل نموذج رياضي باستخدام المتغيِّرات، حيث تُمثّل هذه المتغيِّرات القيم التي يجب تحسينها، ثم يتم استخدامها لتحديد الدالة الموضوعية والقيود، وهما يصفان المشكلة معًا ويُمكّنان من استخدام خوارزميات البرمجة الرياضية.

2 ما مزايا وعيوب أسلوب برمجة الأعداد الصحيحة المختلطة في حلّ مشكلات التحسين؟

المزايا:

تسمح برمجة الأعداد الصحيحة باستخدام كل من الأعداد الصحيحة أو القيم المنفصلة (Discrete Values) والمتغيِّرات المستمرة (Continuous Variables) أثناء مشكلة التحسين.

توفِّر برمجة الأعداد الصحيحة لغة نمذجة أكثر شمولية من البرمجة الخطية، ويمكن أن تمثِّل أنواع معيَّنة من المشكلات بشكل دقيق.

تستخدِم برمجة الأعداد الصحيحة مجموعة واسعة من تقنيات التحسين التي تتيح حلًا أفضل لكل مشكلة.

العيوب:

تتطلَّب خوارزميات برمجة الأعداد الصحيحة مزيدًا من الوقت والموارد الحاسوبية مثل: طاقة المعالجة والذاكرة.

قد يكون من الصعب فهم خوارزميات برمجة الأعداد الصحيحة للأشخاص غير التقنيِّين.

يمكن أن تستغرق حلول برمجة الأعداد الصحيحة الكثير من الوقت في إنتاج أسوأ السيناريوهات، حيث تستغرق عمليات التنفيذ وقتًا طويلًا جدًا لحلها مقارنة بمشكلة برمجة خطية من نفس الحجم.

يمكن تقديم إجابات إضافية من قبل الطلبة

3 قم بتحليل مشكلتين من مشكلات التحسين يُمكن حلهما باستخدام البرمجة الرياضية، ثم حدّد متغيّرات الحالة ومتغيّرات القرار الخاصة بهما.

	متغيّرات القرار	متغيّرات الحالة
تخطيط الإنتاج	الكمية التي يجب إنتاجها من كل مُنتَج.	تَوفّر المواد الخام، وسعة آلات الإنتاج، وتَوفّر العمالة المطلوبة للإنتاج.
نقل الموارد	عدد السلع التي يجب نقلها من مكان لآخر.	المسافة بين الأماكن التي يجب زيارتها وسعة المركبات.

4 اذكر ثلاث مشكلات تحسين مُختلفة من عائلة مشكلات تحديد المسار.

مشكلة البائع المتجول (Traveling Salesman Problem - TSP): يحتاج فيها البائع إلى زيارة مجموعة من المُدن مرة واحدة دون أن يكرِّر زيارة أيٍّ منها، ثم يعود للمدينة الأصلية، بحيث يقلل المسافة الإجمالية المقطوعة، والهدف هو العثور على أقصر طريق ممكن لزيارة كل مدينة مرة واحدة والعودة إلى مدينة البداية.

مشكلة تحديد مسار المركبات (Vehicle Routing Problem - VRP): تتضمن المشكلة توجيه أسطول من المركبات التي يجب أن تخدم مجموعة من العملاء الذين لديهم مطالب معروفة من مستودع مركزي، والهدف هو العثور على الطُرق المُثلى لكل مركبة لتلبية متطلبات جميع العملاء مع تقليل المسافة الإجمالية المقطوعة أو عدد المركبات المستخدَمة.

مشكلة تحديد مسار المركبات ذات السعة (Capacitated Vehicle Routing Problem – CVRP): تُعدُّ هذه المشكلة امتدادًا لمشكلة تحديد مسار المركبات (VRP)، حيث تتمتع كل مركبة بسعة محدودة ويكون لكل عميل طلب معين، والهدف هو تحديد المسارات المُثلى للمركبات لخدمة جميع العملاء مع احترام قيود السعة لكل مركبة وتقليل المسافة الإجمالية المقطوعة أو عدد المركبات المستخدَمة.

يمكن تقديم إجابات إضافية من قبل الطلبة

5 أنشئ دالة خوارزمية حلّ القوة المُفرطة لمشكلة البائع المُتجوّل، من خلال إكمال المقطع البرمجي التالي بحيث تُظهر الدالة المسار الأفضل والمسافة الإجمالية المُثلى:

```
from itertools import permutations

def brute_force_solver(dist_matrix, location_ids, startstop):
    # excludes the startstop location
    location_ids = location_ids - {startstop}
    # generates all possible routes (location permutations)
    all_routes = permutations(location_ids)
    best_distance = float('inf') # initializes to the highest possible number
    best_route = None # best route so far, initialized to None

    for route in all_routes: # for each route
        distance = 0 # total distance in this route
        curr_loc = startstop # current location

        for next_loc in route:
            distance += dist_matrix[curr_loc, next_loc] # adds the distance of this step
            curr_loc = next_loc # goes the next location
        distance += dist_matrix[curr_loc, startstop] # goes to back to the startstop location
        if distance < best_distance: # if this route has lower distance than the best route
            best_distance = distance
            best_route = route

    # adds the startstop location at the beginning and end of the best route and returns
    return [startstop] + list(best_route) + [startstop], best_distance
```

6 أنشئ خوارزمية حلّ برمجة الأعداد الصحيحة المختلطة لمشكلة البائع المُتجوّل، من خلال إكمال المقطع البرمجي التالي، بحيث تنتقي متغيّرات القرار وقيود الاتصال انتقاءً صحيحًا:

```
def MIP_solver(dist_matrix, location_ids, startstop):

    solver = Model() # creates a solver
    solver.verbose = 0 # setting this to 1 will print info on the progress of the solver
    # creates every transition from every location to every other location

    transitions = list(product(location_ids, location_ids))
    N = len(location_ids) # number of locations
    # creates an empty square matrix full of 'None' values
    x = numpy.full((N, N), None)
    # adds binary decision variables indicating if transition (i->j) is included in the route
    for i, j in transitions:

        x[i, j] = solver.add_var(var_type=BINARY)

    # objective function: minimizes the distance

    solver.objective = minimize(xsum(dist_matrix[i, j] * x[i][j] for
i, j in transitions))

    # Arrive/Depart Constraints
    for i in location_ids:

        solver += xsum(x[i,j] for j in location_ids - {i}) == 1

        solver += xsum(x[j,i] for j in location_ids - {i}) == 1

    # Adds a binary decision variable for each location

    y = [solver.add_var(var_type=INTEGER) for i in
location_ids]

    # Adds connectivity constraints for transitions that do not include the startstop
    for (i, j) in product(location_ids - {startstop}, location_ids -
{startstop}):
        if i != j: # ignores transitions from a location to itself
            solver += y[j] - y[i] >= (N + 1) * x[i, j] - N

    solver.optimize() # solves the problem
```

6. الذكاء الاصطناعي والمجتمع

سيتعرّف الطالب في هذه الوحدة على أخلاقيات الذكاء الاصطناعي وتأثيرها على تطوير أنظمته المتقدمة وتحديد توجهاتها، وسيُقيِّم مدى تأثير أنظمة الذكاء الاصطناعي واسعة النطاق على المجتمعات والبيئة، وكيفية تنظيم مثل هذه الأنظمة للاستخدام الأخلاقي المُستدام، وسيستخدِم بعد ذلك مُحاكي ويبوتس (Webots) لبرمجة طائرة مُسيَّرة على الحركة الذاتية واستكشاف منطقة ما من خلال تحليل الصور.

أهداف التعلُّم

بنهاية هذه الوحدة سيكون الطالب قادرًا على أن:

- يُعرِّف أخلاقيات الذكاء الاصطناعي.
- يُفسِّر مدى تأثير التحيُّز والإنصاف على الاستخدام الأخلاقي لأنظمة الذكاء الاصطناعي.
- يُقيِّم كيفية حل مشكلة الشفافية وقابلية التفسير في الذكاء الاصطناعي.
- يُحلِّل كيفية تأثير أنظمة الذكاء الاصطناعي واسعة النطاق على المجتمع وكيفية وضع قوانين لتنظيمها.
- يُبرمج جهاز الطائرة المُسيَّرة على الحركة الذاتية.
- يُطوِّر نظام تحليل الصور لطائرة مُسيَّرة تُستخدم في استطلاع منطقة معينة .

الأدوات

- ويبوتس (Webots)
- مكتبة أوبن سي في (OpenCV)

وزارة التعليم
Ministry of Education
2023 - 1445

يمكن تقديم إجابات إضافية من قبل الطلبة

تمرينات

1

حَدِّد الجملة الصحيحة والجملة الخاطئة فيما يلي:	صحيحة	خاطئة
1. تهتم أخلاقيات الذكاء الاصطناعي بتطوير أنظمة الذكاء الاصطناعي فقط. تشير أخلاقيات الذكاء الاصطناعي إلى المبادئ والقيم والمعايير الأخلاقية التي تنظم تطوير أنظمة الذكاء الاصطناعي وانتشارها واستخدامها.		✓
2. من المحتمل أن يُؤدي الذكاء الاصطناعي والأتمتة إلى تسريح البشر من الوظائف.	✓	
3. يُمكن أن يُؤدي الافتقار إلى التنوع في فِرق تطوير الذكاء الاصطناعي إلى عدم رؤية التحيُّزات أو عدم معالجتها.	✓	
4. يُمكن أن يساعد دمج المبادئ الأخلاقية في أنظمة الذكاء الاصطناعي في ضمان تطويرها واستخدامها بطريقة مسؤولة.	✓	
5. يتطلب التصميم المعتمد على إشراك الإنسان أن تعمل أنظمة الذكاء الاصطناعي دون أي تدخل بشري. يتطلب التصميم المعتمد على إشراك الإنسان أن تعمل أنظمة الذكاء الاصطناعي مع إشراك العنصر البشري.		✓
6. تدل مشكلة الصندوق الأسود في الذكاء الاصطناعي على صعوبة فهم كيفية وصول خوارزميات الذكاء الاصطناعي إلى قراراتها أو تنبؤاتها.	✓	
7. يُمكن تصميم نماذج الذكاء الاصطناعي لتكييف قراراتها أو نتائجها وفقًا للقيم الأخلاقية الراسخة.	✓	
8. استخدام الذكاء الاصطناعي على نطاق واسع له آثار إيجابية فقط على البيئة. استخدام الذكاء الاصطناعي على نطاق واسع له آثار سلبية وآثار إيجابية.		✓

2

صِف كيف يؤدي الذكاء الاصطناعي والأتمتة إلى تسريح البشر من وظائفهم.

أثارالاستخدام المتزايد للذكاء الاصطناعي والأتمتة (Automation) في مختلف الصناعات المخاوف بشأن تسريح البشر من وظائفهم وتأثيره على سُبل عيش العاملين، فعلى الرغم من أن الذكاء الاصطناعي يُمكنه أن يُؤدي إلى تحسين الكفاءة والإنتاجية، إلا أنه يُمكن أن يُؤدي أيضًا إلى فقدان البشر لوظائفهم وتزايد عدم المساواة في الدخل، مما قد يكون له عواقب اجتماعية واقتصادية سلبية.

يمكن تقديم إجابات إضافية من قبل الطلبة

3 اشرح كيف يمكن أن تساهم بيانات التدريب المُتحيِّزة في تحقيق نتائج ذكاء اصطناعي مُتحيِّزة.

يُعدُّ نظام التوظيف الآلي الذي يستخدم الذكاء الاصطناعي لفحص المرشحين للوظائف من أبرز الأمثلة على الخوارزميّة المُتحيِّزة. افترض أن الخوارزميّة مُدرَّبة على بيانات مُتحيِّزة، مثل أنماط التوظيف التاريخية التي تُفضِّل مجموعات ديموغرافية معيّنة، ففي هذه الحالة قد يعمل الذكاء الاصطناعي على استمرار تلك التحيُّزات ويستبعد المرشّحين المؤهّلين بشكل غير عادل من بين المجموعات متجاهلًا الفئات غير الممثَّلة جيدًا في مجموعة البيانات. على سبيل المثال، افترض أن الخوارزميّة تُفضل المرشحين الذين التحقوا بجامعات النخبة، أو عملوا في شركات مرموقة، ففي هذه الحالة قد يلحق ذلك الضرر بالمرشحين الذين لم يحظوا بتلك الفُرص، أو الذين ينتمون إلى بيئات أقل حظًّا، ويُمكن أن يُؤدي ذلك إلى نقص التنوع في مكان العمل وإلى استمرارية عدم المساواة، ولذلك من المهم تطوير واستخدام خوارزميات توظيف للذكاء الاصطناعي تَستند على معايير عادلة وشفافة، وغير مُتحيِّزة.

4 عرِّف مشكلة الصندوق الأسود في أنظمة الذكاء الاصطناعي.

تكمن مشكلة الصندوق الأسود في الذكاء الاصطناعي في التحدي المُتمثِّل في فهم كيفية عمل نظام قائم على الذكاء الاصطناعي (AI-Based System) باتخاذ القرارات أو إنتاج المُخرَجات؛ مما قد يُصعِّب الوثوق بالنظام أو تفسيره أو تحسينه، وربما يؤثر الافتقار إلى الانفتاح وإلى قابلية التفسير على ثقة الناس في النموذج.

5 قارن بين الآثار الإيجابية والسلبية لأنظمة الذكاء الاصطناعي على البيئة.

الآثار الإيجابية المحتملة:

يُمكن للذكاء الاصطناعي أن يساعد في فهم التحديات البيئية والتعامل معها بشكل أفضل مثل: تغير المناخ، والتلوث، وفقدان التنوع البيولوجي، ويُمكنه أن يساعد في تحليل كميات هائلة من البيانات والتنبؤ بتأثير الأنشطة البشرية المختلفة على البيئة، ويُمكنه كذلك أن يساعد في تصميم أنظمة أكثر كفاءة واستدامة، مثل أنظمة: شبكات الطاقة، والزراعة، والنقل، والمباني.

الآثار السلبية المُحتملة:

هناك مخاوف من تأثير الذكاء الاصطناعي نفسه على البيئة؛ إذ يتطلب تطوير أنظمة الذكاء الاصطناعي واستخدامها قدرًا كبيرًا من الطاقة والموارد؛ مما قد يُسهِّم في انبعاث غازات تُفاقم من مشكلة الاحتباس الحراري وغيرها من الآثار البيئية. على سبيل المثال، قد يتطلب تدريب نموذج واحد للذكاء الاصطناعي قدرًا من الطاقة يعادل ما تستهلكه العديد من السيارات طوال حياتها. بالإضافة إلى ذلك، يمكن أن يساهم إنتاج المُكوِّنات الإلكترونية المُستخدَمة في تصنيع أنظمة الذكاء الاصطناعي في تلوث البيئة مثل: استخدام المواد الكيميائية السامة وتوليد النفايات الإلكترونية. علاوة على ذلك، يُمكن أن يغير الذكاء الاصطناعي علاقتنا بالبيئة بطرائق ليست إيجابية دائمًا، فقد يُؤدي استخدام الذكاء الاصطناعي في الزراعة إلى ممارسات زراعية مكثَّفة ومركَّزة على الصناعة؛ مما يؤثر سلبًا على صحة التربة والتنوع البيولوجي. بالمثل، ربما يُؤدي استخدام الذكاء الاصطناعي في النقل إلى زيادة الاعتماد على السيارات وأساليب النقل الأخرى؛ مما يُسهِّم في تلوث الهواء وتدمير البيئات الطبيعية التي تسكنها الكائنات الحية.

وزارة التعليم
Ministry of Education
2023-1445

يمكن تقديم إجابات إضافية من قبل الطلبة

تمرينات

1 حلّل الدالة move_to_target( ) واشرح كيفية قيام الطائرة المُسيَّرة بحساب موضعها التالي في قائمة نقاط الطريق. كيف يمكن تحسين مسار الطائرة المُسيَّرة لتقليل زمن الطيران بين نقاط الطريق؟

تعمل دالة **move_to_target ( )** على تحريك الطائرة المُسيَّرة إلى موضع معين عن طريق حساب الزاوية وضبط الانحدار، كما أنها تُحدِّث موضع الهدف والفهرس، ولتحسين مسار الطائرة المُسيَّرة يمكن استخدام خوارزميات تخطيط المسار وتقنيات تعلُّم الآلة.

2 قيّم عيوب خوارزميّة التحكُّم الحالية في الطائرة المُسيَّرة عند مواجهة عوامل خارجية مثل: الرياح أو العوائق أو عدم دقة نظام تحديد المواقع العالمي، ثم اقترح وناقش التحسينات التي يمكن القيام بها في خوارزميّة التحكم لجعل الطائرة المُسيَّرة أكثر صمودًا في وجه هذه التحديات.

تلميح: يُعدُّ هذا السؤال من نوع الأسئلة المفتوحة، حيث يمكن للطلبة البحث عن مصطلحات عبر الإنترنت مثل:

- آثار الرياح العاتية على الطائرة المُسيَّرة.
- عدم دقة نظام تحديد المواقع في الطائرة المُسيَّرة.

يمكن تقديم إجابات إضافية من قبل الطلبة

3 استكشف الآثار الأخلاقية للطائرات المُسيَّرة الهوائية في التطبيقات الواقعية مثل: المراقبة وتوصيل الطرود وعمليات البحث والإنقاذ، ثم اكتب عن المخاوف المحتملة الخاصة بالخصوصية، وقضايا السلامة، واحتمالات إساءة استخدام هذه التقنية.

تلميح: يُعدُّ هذا السؤال من نوع الأسئلة المفتوحة، حيث يمكن للطلبة البحث عن مصطلحات عبر الإنترنت مثل:

- أخلاقيات مراقبة الطائرات المُسيَّرة.
- أخلاقيات الاستخدام المحلي للطائرات المُسيَّرة.
- أخلاقيات شحن الطرود باستخدام الطائرات المُسيَّرة.
- أخلاقيات مهام البحث والإنقاذ بالطائرات المُسيَّرة.
- أخلاقيات روبوتات الإنقاذ بالطائرات المُسيَّرة.

4 أضف خاصية تُسجِّل موضع الطائرة المُسيَّرة وارتفاعها واتجاهها على فترات منتظمة أثناء الطيران، ثم اكتب كل الأنماط التي قد تجدها في بيانات السجل.

استخدم دالة **print ()** بعد قراءة المستشعرات للمقطع البرمجي من أجل طباعة المتغيِّرات:

- x_pos (x position), y_pos (y position)
- altitude (z position/altitude)
- roll (roll)
- pitch (pitch)
- yaw (yaw)

وزارة التعليم
Ministry of Education
2023-1445

يمكن تقديم إجابات إضافية من قبل الطلبة

5 جرِّب استخدام قيم مختلفة لثوابت PID في برنامج المُتحكِّم (K_VERTICAL_P، K_ROLL_P، K_PITCH_P). ولاحظ كيفية تأثير هذا التغيرات على استقرار الطائرة المُسيَّرة واستجابتها، ثم ناقش الموازنات بين الاستقرار والاستجابة.

K_VERTICAL_P: يؤثر هذا الثابت على استقرار الوضع الرأسي للطائرة المُسيَّرة، وستؤدي زيادته إلى زيادة استقرار الطائرة المُسيَّرة، ولكنه سيجعلها أيضًا أقل استجابة للتغييرات في موضع الهدف، وسيؤدي تقليله إلى جعل الطائرة المُسيَّرة أكثر استجابة، ولكنه سيجعلها أيضًا أقل استقرارًا.

K_ROLL_P و **K_PITCH_P**: يؤثر هذان الثابتان على استقرار واستجابة زاويتَي انحدار والتفاف الطائرة المُسيَّرة، حيث ستؤدي زيادتهما إلى زيادة استقرار الطائرة المُسيَّرة ، ولكنهما سيجعلانها أيضًا أقل استجابة للتغييرات في موضع الهدف، وسيؤدي تقليلهما إلى جعل الطائرة المُسيَّرة أكثر استجابة ، ولكنهما سيجعلانها أيضًا أقل استقرارًا.

تُعدُّ المفاضلة بين الاستقرار والاستجابة مشكلة شائعة في أنظمة التحكم، وغالبًا ما تأتي زيادة الاستقرار على حساب انخفاض الاستجابة، في حين أن زيادة الاستجابة غالبًا ما تأتي على حساب انخفاض الاستقرار، ولذلك من المهم إيجاد التوازن الصحيح بين الاستقرار والاستجابة لضمان قدرة الطائرة المُسيَّرة على الطيران بسلاسة ودقة.

يمكن تقديم إجابات إضافية من قبل الطلبة

تمرينات

1 عدّل برنامج المُتحكّم الخاص بك بحيث لا يتحقق من وجود المجلد بالفعل في المسار. هل يتسبب ذلك في أية تعقيدات في تنفيذ المُحاكاة؟

المقطع البرمجي الذي يتحقق من وجود المجلد هو:

```
if not os.path.exists(folder_path):
    # creates the folder if it doesn't exist already
        os.makedirs(folder_path)
        print(f"Folder \"detected\" created!")
    else:
        print(f"Folder \"detected\" already exists!")
```

عن طريق إزالة سطر المقطع البرمجي الخارجي "if not os.path.exists(folder_path): else:" سيتم تشغيل السطر الداخلي "os.makedirs (folder_path)" في المرة الأولى وإنشاء المجلد.

بإزالة التحقق لن يتمكن البرنامج من العمل مرة أخرى؛ لأن المجلد سيكون موجودًا بالفعل.

2 عدّل برنامج المُتحكّم بحيث يقوم بالاكتشاف كل 10 ثوانٍ. هل تلاحظ أي فَرق في تكرار ما تطبعه وحدة التحكم وفي الصور المحفوظة؟

```
class Mavic(Robot):
    # Constants and variables omitted

    def __init__(self):
        # Initialization code omitted for brevity
        self.last_detection_time = datetime.now()

    def run(self):
        # Main loop code omitted for brevity
        while self.step(self.time_step) != -1:
            # Detection code
            current_time = datetime.now()
            time_since_last_detection = (current_time - self.last_detection_time).
total_seconds()
            if time_since_last_detection >= 10:
                self.detect_human()
                self.last_detection_time = current_time
```

يمكن تقديم إجابات إضافية من قبل الطلبة

3 ماذا سيحدث لمُخرَجات الصورة إذا قمت بدمج أبعاد الألوان حسب التسلسل المعتاد بدلًا من التسلسل المعكوس؟ دوّن ملاحظاتك وفقًا لذلك.

إذا تم دمج قنوات الألوان **RGB** بدلًا من **BGR**، سيتم التبديل بين قنوات اللون الأحمر والأزرق ولا يمكن لمكتبة أوبن سي في **(OpenCV)** أحيانًا التعرف على البشر في الصور الملتقطة، كما يمكن ملاحظة تبديل الألوان هذا في الصور التي تتم معالجتها وحِفظها في المجلد.

4 أجرِ تجارب على المُعامِلين الرابع والخامس في الدالة rectangle(). دوّن ملاحظاتك وفقًا لذلك.

يحدِّد المُعامِل الرابع لون المستطيل في **BGR**، ويحدِّد المعامل الخامس مدى سماكة خطوط المستطيل. على سبيل المثال: باستخدام **(0، 255، 0)** و**0** للمُعامِلين الرابع والخامس، سيكون المستطيل أخضر وسيكون عرض خطوطه رفيعًا جدًّا.

5 عدِّل برنامج المُتحكِّم الخاص بك بحيث يطبع قيم الالتفاف والانحدار والانعراج للطائرة المُسيَّرة عند اكتشاف أي شخص.

```
def detect_human(self):
        # Detection code omitted for brevity
        roll, pitch, yaw = self.imu.getRollPitchYaw()
        current_time = datetime.now()
        print(current_time)
        print("Found a person in coordinates [{:.2f}, {:.2f}] with roll {:.2f}, pitch
{:.2f}, and yaw {:.2f}"
              .format(x_pos, y_pos, roll, pitch, yaw))
        # Saves annotated image to file with timestamp
        current_time = current_time.strftime("%Y-%m-%d_%H-%M-%S")
        filename = f"detected/IMAGE_{current_time}.png"
        cv2.imwrite(filename, img)
```

وزارة التعليم
Ministry of Education
2023 - 1445