مؤسسة النهضة العلمية

مراقبة تحفيظ القرآن الكريم بناء على التعرف على الكلام وتقنيات البرمجة اللغوية العصبية

عمر صالح شكلاوون[1]، علي سالم شفتر[1]، مصطفى علي أبوزريدة*[1]، أكرم محمد زكي[2]، زينب عطار باشي[2]

[1] قسم علوم الحاسب الآلي، كلية تكنولوجيا المعلومات، جامعة مصراتة. ليبيا

[2] كلية تكنولوجيا المعلومات والاتصالات، الجامعة الإسلامية الدولية في ماليزيا، كوالالمبور، ماليزيا

*abuzaraida@it.misuratau.edu.ly

الخلاصة: تسارع في العقود الاخيرة تطوير تقنيات الذكاء الاصطناعي، مثل تقنية التعرف على الكلام، وتسارعت أيضًا التطبيقات التي تعتمد على تقنية التعرف على الكلام حيث تقلل هذه التطبيقات وقت وجهد إنجاز المهام المرغوبة من المستخدم. تعتمد هذه التقنيات في عملها على معالجة اللغة الطبيعية واستخدام تقنيات تعليم الالة مثل الشبكات العصبية. في هذه الدراسة، تم تصميم نظام للمساعدة في تذكر وحفظ القران الكريم باستخدام نموذج ماركوف الخفي (HMM) ، وهو أحد أشهر النماذج المستخدمة في التعرف على الكلام وتمييز الاصوات. في هذا النظام، يمكن للمستخدم تلاوة القرآن على النظام المقترح الذي يساعد أي شخص يريد مراجعة حفظ القرآن. سيعطي النظام تنبيهًا في حالة ارتكاب أي خطأ في ترتيب الآيات، ويساعد على معرفة الآية التالية عن طريق إظهار رسالة بذلك. أظهرت النتائج أن النظام حقق معدل دقة 96٪.

الكلمات الجوهرية: القرآن الكريم، الذكاء الاصطناعي، تعليم الالة، التعرف على الاصوات، نموذج ماركوف المخفي.

1 مقدمة:

انتشر استخدام التطبيقات المساعدة على الهواتف الذكية وأجهزة الكمبيوتر، وأصبح من المعتاد بالنسبة لنا إرسال طلباتنا عن طريق الصوت لرؤية الاستجابة النصية جاهزة على أجهزتنا.

تطبيق المساعدة هو مزيج من تقنيات التعلم العميق والبرمجة اللغوية العصبية (NLP)، لتحليل استعلامنا الصوتي، واسترداد الإجابة وتقديمها في نص. بعد ذلك، تقوم بإجراء العملية المستهدفة بناءً على هذا النص. هنا، إذا كان الاستعلام سؤالًا، فإن النظام يبحث عنه في المتصفح لإظهار الإجابة الصحيحة. بينما إذا كان أمرًا تنفيذيًا، فإنه ينفذه على الفور، مثل إجراء مكالمة إلى جهة اتصال معينة أو إرسال رسالة أو بريد إلكتروني [1].

القرآن الكريم هو الكتاب المقدس الرئيسي للإسلام. ويتكون من 30 جزءًا. عدد آيات القرآن 6236، مقسمة إلى 114 سورة. تنطق آيات القرآن في التلاوة العربية المسماة "التجويد". كما أن حكم ترتيب الآيات في القرآن مبني على الوحي، ولم يتدخل أحد في الترتيب. عند التلاوة بشكل صحيح يجب مراعاة عدة عوامل، بما في ذلك فهم قواعد نطق الحروف العربية بشكل صحيح. وكذلك يشترط عند التلاوة حفظ ترتيب الآيات في السورة.

لذلك، فكل من يريد حفظ القرآن الكريم يحتاج إلى أن يتحقق من حفظه وصحة ترتيب الآيات عند التلاوة. هذه النقاط تعتبر من المشاكل التي يواجهها العديد من الأفراد الراغبين في حفظ القرآن الكريم وفي مراجعة تحفيظهم والتأكد من ترتيب الآيات في السور. تهدف هذه الدراسة إلى تصميم وبناء نظام ذكي لمعالجة المشاكل المذكورة، وسيتم استخدام تقنيات التعرف على الكلام لتطوير نظام لتلاوة القرآن، ومراجعة التحفيظ والتنبيه في حالة وجود أخطاء في ترتيب الآيات، وبيان موقع الخطأ. يحصل هذا النظام على تلاوة القارئ من خلال ميكروفون ثم يحولها إلى نص ثم يتحقق من صحة هذه الآية أثناء التلاوة ويضمن ترتيبها، من خلال قاعدة بيانات يتم فيها حفظ النص القرآني بالترتيب.

في هذا البحث، سيتم تنفيذ تطبيق عبر الإنترنت لتحويل الصوت المسجل من قبل التطبيق من القارئ إلى نص من خلال تقنيات التعرف على الكلام ثم مقارنة هذا النص وعرضه على واجهة التطبيق. ويطلب من القارئ تكرار التلاوة لتصحيحها، وإذا لم يتمكن من العثور على الآية الصحيحة، فسيعطي النظام الآية الصحيحة للقارئ. وستغطي هذه الدراسة تقنية تحويل الصوت إلى نص، ثم التحقق من النص في قاعدة البيانات، ومعرفة ترتيبها في السور، وانتظار الآية التالية.

في هذا البحث، سيتم استخدام تقنية التعرف على الكلام التي توفرها جوجل والتي تحول الصوت إلى نص. ومع ذلك، لن يتعامل النظام مع الأخطاء التي قد تحدث بناءً على (حركة) الحروف (تشكيل) كحدود لهذه الدراسة.

2 المنهجية:

يقوم النظام بأربع عمليات رئيسية، وهي مرحلة إدخال التلاوة، ومرحلة تحويل التلاوة إلى نص، ومرحلة معالجة النص، ومرحلة مقارنة النص بقاعدة البيانات. في هذا القسم، سنشرح هذه المراحل بالتفصيل.

1.2 مراحل النظام:

ويشمل المراحل الرئيسية التي يتكون منها النظام. يوضح الشكل 1 تسلسل هذه المراحل، بدءًا من مرحلة إدخال القراءة إلى مرحلة إخراج العرض.

أ. مرحلة القراءة:

في هذه المرحلة، يتلقى النظام القراءة من خلال الميكروفون. في هذه المرحلة، تكون متطلبات الأجهزة عبارة عن ميكروفون. يحول الميكروفون الاهتزازات الصوتية في الهواء إلى إشارات إلكترونية ليتم تسجيلها كإشارات رقمية. هناك نوعان من الميكروفونات مستقلة أو متكاملة مع الأجهزة الإلكترونية. في هذا النظام، يتم استخدام النوع الثاني للحصول على صوت القارئ. في هذه المرحلة، يقوم النظام أيضًا بتحويل الإشارات الصوتية إلى إشارات إلكترونية. تحتوي القراءة المكتسبة على خطوتين:

- **الخطوة الأولى (ما قبل القراءة):**

في هذه المرحلة، يأخذ النظام عينة من الضوضاء المحيطة (غير المرغوب فيها)، مثل صوت الرياح أو صوت السيارات، لفترة زمنية محددة (ثانية أو أقل) ، فيتجاهل النظام هذا الصوت أثناء الاستماع إلى القراءة.

- **الخطوة الثانية (اكتساب القراءة):**

في هذه المرحلة، يستمع النظام إلى صوت القراءة ويسجله من أجل تحويل إهتزازات الصوت إلى إشارات إلكترونية. يبدأ النظام التسجيل في بداية القراءة، متجاهلاً مدى الضوضاء التي استغرقها في المرحلة الأولى، وينهي النظام التسجيل عند توقف القراءة.

ب. تحويل التلاوة إلى نص:

تتضمن عملية التعرف على الكلام عدة مكونات: خطوات النمذجة الصوتية (البرمجة اللغوية العصبية).

تتضمن النمذجة الصوتية تقسيم الكلمات المنطوقة إلى أصوات فردية (صوتيات) وتحويل الصوتيات إلى أحرف مقابلة لها ثم استخدام خوارزمية (HMM) [2،3] لتحديد الكلمات التي تم نطقها.

نظرًا للطبيعة المتغيرة ببطء لإشارات الكلام، فإن التحليل الطيفي قصير المدى هو الطريقة الأكثر شيوعًا لوصف إشارات الكلام. عند فحصها على مدى فترة زمنية قصيرة بما فيه الكفاية (بين 10 و 25 مللي ثانية)، تكون خصائصها ثابتة إلى حد ما. كما تجد أن متجهات الميزة المستخرجة أكثر من 10 إلى 25 مللي ثانية تتوافق مع الصوتيات الفردية. يوضح الشكل 2 تمثيل الحروف في شكل رموز.

Arabic consonants	IPA symbol selection
ا	aː
ب	b
ت	t
ث	θ
ج	ʤ
ح	ħ
خ	x
د	d
ذ	ð
ر	r
ز	z
س	s
ش	ʃ

ص	sˤ
ض	dˤ
ط	tˤ
ظ	ðˤ
ع	ʕ
غ	ɣ
ف	f
ق	q
ك	k
ل	l
م	m
ن	n
ه	h
و	w
ي	j
ء	ʔ

الشكل 2: تمثيل الحروف في شكل رموز

للتعرف على الكلام في (عمَّ) [2،3]، نخصص لكل وحدة أساسية (صوت) فريدة من نوعها كما هو موضح في الشكل 3.

الشكل 3: فونيمات الكلمة (عمَّ)

علاوة على ذلك، تتوافق الحالات مع متجهات الميزة. لفهم ذلك، خذ مثالاً على أداة التعرف على الكلمات المعزولة، كل كلمة في المفردات لها فونيمات متميزة. كما هو موضح في إخراج المخطط الكتلي للنموذج الصوتي هو تسلسل الصوت كما هو موضح في الشكل 4. من خلال النظر إلى القاموس بطريقة عكسية (صوتي – كلمة) يمكننا العثور على الكلمات المقابلة.

Word	Pausal mapping - with case ending
عَمَّ	/ʕamma/
يَتَسَاءَلُونَ	/jatasaːʔaluːna/
عَنِ	/ʕani/
النَّبَإِ	/ʔannabaʔi/
الْعَظِيمِ	/ʔalʕaðˤijmi/

الشكل 4: قاموس الكلمات

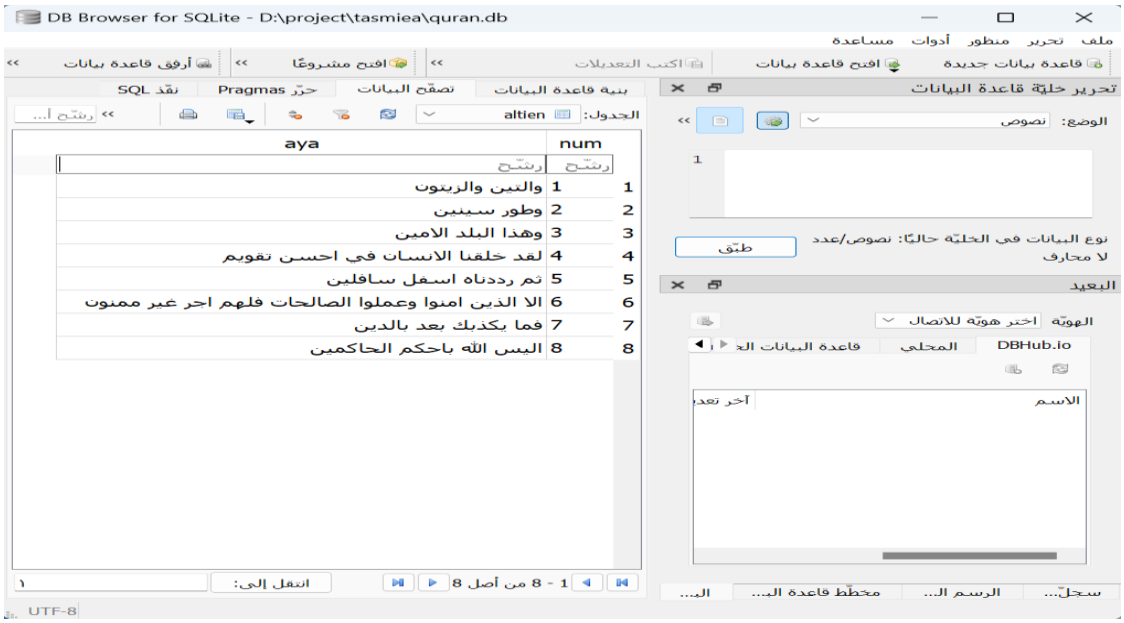
ج. معالجة النص:

تعمل البرمجة اللغوية العصبية على تسهيل وتسريع عملية التعرف على الكلام [4]. تُستخدم البرمجة اللغوية العصبية في أنظمة التعرف على الكلام لتفسير ما يقوله الشخص وتعزيز النص، استنادًا إلى فهم اللغة الطبيعية (NLU)، وتوليد اللغة الطبيعية (NLG)، ودلالات اللغة الطبيعية، وبراغماتية اللغة الطبيعية، للحصول على الجزء الأكثر أهمية من الكلمات والجمل. لقد قمنا بتحسين تقنيات (البرمجة اللغوية العصبية) لكي تتماشى مع تلاوة القرآن، لأن تلاوة القرآن لا تتسامح مع الأخطاء، حيث أن هذه التقنية تحسن النص ليكون مثل التداول أو ليكون صحيحًا نحويًا، لذلك يغير بعض الحروف أو الكلمات بما يتماشى مع الجملة.

يتضمن التحسين تطوير خوارزمية تستخدم (البرمجة اللغوية العصبية). تركز هذه الخوارزمية على تصحيح الحروف المحددة التي يخطئ القارئ في نطقها دون تغيير بنية الجملة أو معناها بشكل عام. من خلال الاستفادة من تقنية البرمجة اللغوية العصبية، تربط الخوارزمية الكلمات من مرحلة خوارزمية HMM بأشكالها الأصلية، دون أي تحسين فيما يتعلق بالدلالات أو البراغماتية. وهذا يسمح للخوارزمية بتمثيل كل جزء من الجملة بدقة دون إجراء أي تعديلات.

د. مقارنة النص بقاعدة البيانات:

يستخدم النظام قاعدة بيانات؛ حيث تم حفظ كل سورة من سور القرآن الكريم في شكل جدول. يحتوي جدول كل سورة على آيات السورة مع ترتيبها في السورة. يوضح الشكل (5) جدول سورة التين.



الشكل 5: قاعدة بيانات سورة التين

في هذه المرحلة، يقارن النظام النص الناتج عن مرحلة التحويل إلى نص مع الآيات الموجودة في قاعدة بيانات النظام. يبحث النظام عن الآية التي قرأها القارئ وعن طريق استعلامات نظام إدارة قواعد البيانات (DBMS). ثم تتحقق مما إذا كانت الآية موجودة أم لا في جدول السورة.

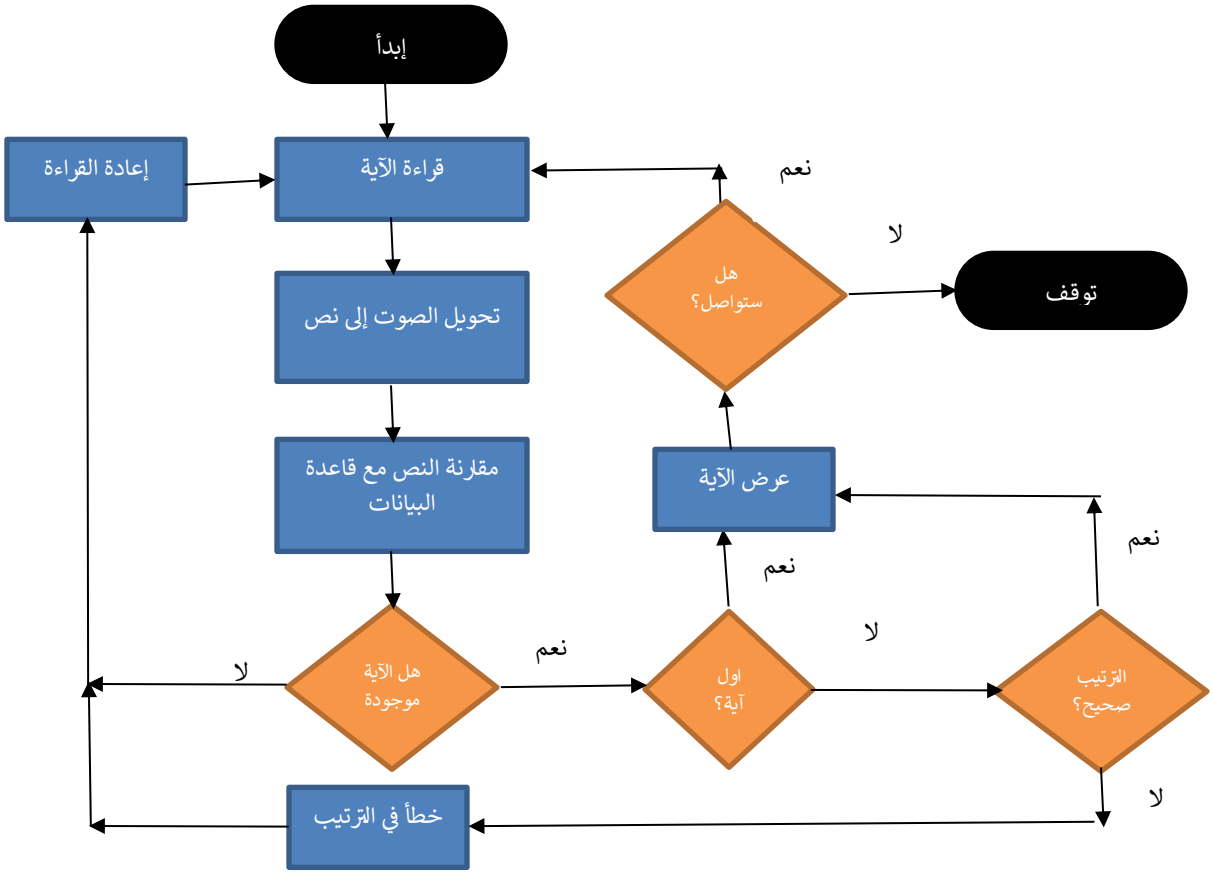
هـ. مخرجات النظام:

بعد مرحلة مقارنة النص مع قاعدة البيانات، يقوم النظام ببعض العمليات، وهو ما إذا كانت الآية موجودة في قاعدة البيانات أم لا. إذا لم يتم العثور على الآية في قاعدة البيانات، يعرض النظام أنه لم يتم العثور على الآية، ويجب على المستخدم إعادة قراءة الآية مرة أخرى.

إذا كانت الآية هي الأولى والآية التي يقرأها القارئ، فإن النظام يعرض الآية ورقمها على القارئ، ثم يسأل النظام القارئ عما إذا كان يريد إكمالها أم لا. إذا أراد القارئ أن يستمر، فإنه يستمع إلى الآية التالية، وإذا كان القارئ لا يريد أن يكمل، ينتهي النظام. في حالة أن الآية التي يقرأها القارئ ليست الأولى التي يقرأها، يتحقق النظام من أن الآية هي التي تتبع الآية التي قبلها. يظهر النظام أن هناك خطأ في ترتيب الآيات، ويجب على القارئ إعادة قراءتها، كما هو موضح في الشكل 6.

2.2 عمليات النظام:

ويشمل جميع العمليات التي يقوم بها النظام. يوضح الشكل 6 آلية عمل النظام.



الشكل 6: عمليات النظام

3. التنفيذ ومناقشة النتائج:

1.3 متطلبات تنفيذ النظام:

أ. لغة البرمجة:

في حين يمكن استخدام أي لغة برمجة لتطوير النظام المقترح، يمكن لبعض اللغات أن تجعل العمل أسرع وأسهل. قمنا باستخدام لغة بايثون [5،6] لتنفيذ النظام المقترح. لغة بايثون هي لغة برمجة مترجمة عالية المستوى وموجهة للكائنات مع دلالات ديناميكية. إن هياكل البيانات فيها عالية المستوى المدمجة، والكتابة الديناميكية، والربط الديناميكي تجعلها مثالية للتطوير السريع للتطبيقات وكلغة برمجة نصية أو وسيلة لتوصيل المكونات الحالية. يؤكد بناء جملة بايثون البسيط والسهل التعلم على سهولة القراءة، مما يقلل من تكاليف صيانة البرنامج.

ب. بيئة تطوير بايثون المتكاملة (IDE):

بيئة التطوير المتكاملة (IDE) هي برنامج مصمم لتطوير البرمجيات. عادةً ما تدمج IDES العديد من الأدوات التي تم إنشاؤها خصيصًا لتطوير البرامج، بما في ذلك:

- محرر تعليمات برمجية يتعامل مع إبراز بناء جملة التعليمات البرمجية والإكمال التلقائي.

- البناء والتنفيذ والتصحيح.
- شكل من أشكال التحكم في المصدر.

غالبًا ما يكون محرر التعليمات البرمجية المخصص أصغر وأسرع، ولكنه أقل ثراءً بالميزات.
سيستخدم النظام المقترح بايثون IDE PyCharm.

ج. الحزم والمكتبات المطلوبة:

مكتبة التعرف على الكلام

تتيح مكتبة التعرف على الكلام التعرف على الكلام باستخدام محركات وواجهات برمجة تطبيقات متعددة [7].
متطلبات التعرف على الكلام:

- بايثون 2.6 أو 2.7 أو 3.3+ (مطلوب).
- PyAudio 0.2.11+ (مطلوب فقط لإدخال الميكروفون).

➢ **PyAudio**

PyAudio مطلوب فقط إذا كانت هناك حاجة إلى إدخال الميكروفون (الميكروفون). إصدار PyAudio 0.2.11+ مطلوب، حيث عرفت الإصدارات السابقة أخطاء في إدارة الذاكرة عند التسجيل من الميكروفونات في مواقف معينة.
إذا لم يتم تثبيت PyAudio، فستظل المكتبة تعمل، ولكن سيتم رفع خطأ السمة عند محاولة إنشاء كائن ميكروفون.

➢ **FLET**

FLET هو إطار عمل قوي لواجهة المستخدم (UI) يتيح الإنشاء السريع لتطبيقات الويب التفاعلية وسطح المكتب والجوال في بايثون دون معرفة مسبقة بتقنيات الويب مثل HTTP أو HTML أو CSS أو JavaScript. يمكنك إنشاء واجهة مستخدم مع عناصر تحكم تستند إلى أدوات فلاتر للتأكد من أن برامجك تبدو احترافية وحديثة.

➢ **متطلبات FLET**:

بايثون 3.7 أو أحدث على ويندوز أو لينكس أو ماك أو إس.

➢ **SQLite:**

SQLite هو محرك قاعدة بيانات SQL عالي الموثوقية وكامل الميزات ومكتفي ذاتيًا وصغير وسريع يتم تنفيذه باستخدام مكتبة اللغة C.
SQLite هو محرك قاعدة البيانات الأكثر استخدامًا على مستوى العالم. وهو مدمج في معظم أجهزة الكمبيوتر والهواتف المحمولة ويأتي مع العديد من التطبيقات الأخرى التي يستخدمها الناس يوميًا.

4. التنفيذ والمناقشة:

في هذا القسم، سنقدم المنهجية المستخدمة للتنفيذ ونناقش النتائج المحققة.

1.4 التنفيذ:

- **الواجهة الخلفية:**

تم استدعاء المكتبات التي يتطلبها تنفيذ النظام في الشكل 7، وتم استدعاء وظيفة "المتعرف" من مكتبة التعرف على الكلام للاستماع إلى الصوت.

```
import speech_recognition as sr
import sqlite3 as sq
from flet import *

 recognizer = sr.Recognizer()
```

الشكل 7: المكتبات المطلوبة لتنفيذ النظام

في الشكل 8، تم توصيل الميكروفون وتشغيله، وتم تسجيل ثانية من الضوضاء المحيطة ليتم تجاهلها أثناء الاستماع إلى الصوت. كما تم تسجيل التلاوة في شكل بيانات صوتية بتنسيق موجة.

```
with sr.Microphone() as source:
    print("Adjusting noise ")
    recognizer.adjust_for_ambient_noise(source, duration=1)
    print("<------>  Recording for aya ... ", end=" ")
    recorded_audio = recognizer.listen(source)
    print("  Done recording     <------>")
```

الشكل 8: توصيل الميكروفون وتشغيله

في الشكل 9، تسمى الوظيفة ويتم تمرير التسجيل الصوتي لها واللغة المستخدمة للتحويل الصوتي، وتعيد الآية المحولة إلى نص دون تحسينها.

```
print("Recognizing the aya")
aya_ = recognizer.recognize_google(
    recorded_audio,
    language="ar-SA",
    show_all=True,
    with_confidence=True
)
```

الشكل 9: تحويل الآية إلى نص دون تحسينها

- **الواجهة الامامية:**

تم إنشاء واجهة في بايثون باستخدام مكتبة فليت لتوفير تجربة مريحة للمستخدم أثناء التلاوة. يعرض الشكل 10 الواجهة الرئيسية للنظام، بينما لا يتم عرض الواجهة قبل أن يبدأ القارئ بالقراءة في هذا السياق.

تحتوي الواجهة الرئيسية في الشكل 10 على أيقونتين، واحدة لبدء التسجيل الصوتي وواحدة لعرض الآية الصحيحة في حالة نسيان الآية التالية. بالإضافة إلى ذلك، يتضمن مربعين نصيين:

1. مربع نص بلا حدود يعرض جميع الآيات التي تم تلاوتها حتى الآن.
2. مربع نص محدد يعرض آخر آية تم تلاوتها. إذا كان صحيحًا، يعرض مربع النص الآية، بينما إذا كان هناك خطأ، يعرض مربع النص نوع الخطأ الذي ارتكبه القارئ.

الشكل 10: واجهة النظام

2.4 آلية عمل النظام:

يفتح النظام واجهة التلاوة تلقائيًا. إذا أراد القارئ بدء التلاوة، فإنه يضغط على زر الميكروفون، الموضح في الشكل السابق.

الحالات التي تتخلف عن التلاوة:

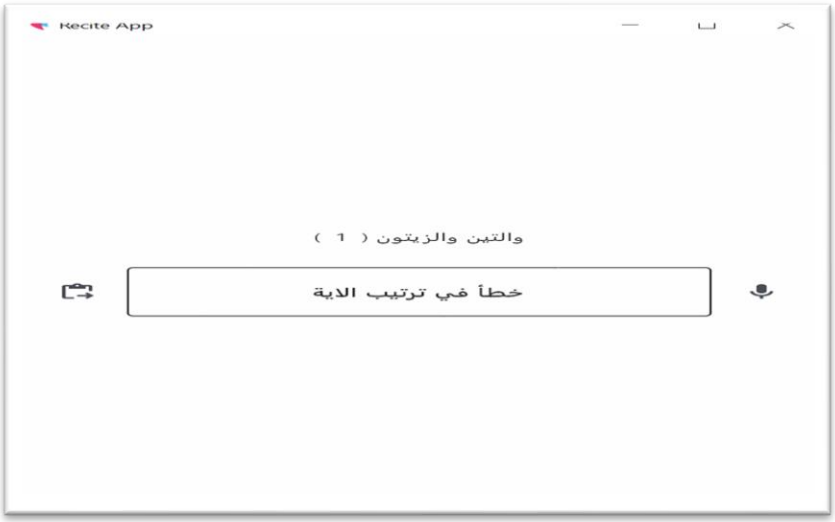
- **تلاوة خالية من الأخطاء**

يوضح الشكل 11 أن القارئ قرأ سورة التين دون أخطاء.

الشكل 11: قام المستخدم بتلاوة سورة التين دون أخطاء

- **التلاوة والأخطاء في ترتيب الآية.**

في الشكل 12، قرأ القارئ الآية الأولى من سورة التين، لكنه نسي الآية التالية. ونتيجة لذلك، اكتشف النظام الخطأ ونبه المستخدم في مربع النص المحدد.



الشكل 12: خطأ في ترتيب الآيات

- **سماع أي شيء آخر ليس من "الآيات".**

يوضح الشكل 13 سيناريو قام فيه القارئ بتلاوة الآية الأولى من سورة التين ثم قرأ شيء لم يكن من سورة التين. ونتيجة لذلك، اكتشف النظام الخطأ ونبه المستخدم في مربع النص المحدد. يوضح هذا قدرة النظام على تحديد الآيات غير الصحيحة بدقة وتقديم ملاحظات في الوقت الفعلي للقارئ.

الشكل 13: القارئ يخطئ في قراءة الآية

- **يكتشف النظام "الآية" التالية.**

يوضح الشكل 14 أن النظام قد اكتشف الآية الثانية من سورة التين بعد أن طلب منه القارئ القيام بذلك باستخدام الزر الموجود على يسار المربع، بحيث يعرض النظام الآية الثانية في المربع.

الشكل 14: طلب القارئ الآية التالية

- **مشكلة الصوت**

يوضح الشكل 15 أن القارئ قرأ الآية الأولى من سورة التين، ثم قرأ الآية الثانية، لكنها لم تكن واضحة، لذلك طلب النظام تكرار الآية وتنبيهه في الصندوق.

الشكل 15: قراءة الآية غير واضحة

- **مشكلة في الحروف أو الكلمات**

يوضح الشكل 16 أن القارئ تلى الآية الأولى من سورة التين، ثم تلى الآية الثانية، لكن القارئ أخطأ في أحد الحروف أو إحدى الكلمات، لذلك طلب النظام تكرار الآية وتنبيهه في الصندوق.

الشكل 16: خطأ في قراءة أحد الحروف

5. مناقشة النتائج:

في هذا القسم، سنناقش اختبار النظام وتقييم الأداء. اختبرنا النظام من خلال قيام عشرة قراء بتلاوة سورة التين من الذاكرة باستخدام النظام، بينما قمنا بمراقبة أداء النظام وحساب دقته باستخدام استبيان.

أثناء اختبار الأداء، ارتكب القراء أخطاء متعمدة لاختبار دقة النظام. تمت معايرة النظام مع سورة الإخلاص، وتم إجراء الاختبار على سورة التين. تم اختبار النظام على عشرة مستخدمين يتلون سورة التين في بيئات مختلفة، بما في ذلك غرفة مغلقة وفي سيارة وفي الهواء الطلق. كان المستخدمون من الرجال وكان العمر من 16 إلى 32 بمتوسط 24.6.
كانت نتائج اختبار النظام وتقييم الأداء على النحو التالي:

- حدد النظام بشكل صحيح 90 ٪ من الآيات.
- حدد النظام بنجاح الآية التالية الصحيحة وعرضها على المستخدم بدقة 100 ٪.
- حدد النظام بشكل صحيح الآيات غير الصحيحة بالترتيب في كل مرة (دقة 100 ٪).
- حدد النظام أخطاء الكلمات أو الحروف في الآية بدقة 95 ٪.

6. النتيجة:

في هذا المشروع، قمنا بتصميم وتنفيذ نظام قادر على الاستماع إلى تلاوة القرآن الكريم من القارئ وتحديد ما إذا كانت التلاوة صحيحة أم غير صحيحة. كما أن النظام قادر على تصحيح الأخطاء وعرض الآية التالية في حالة نسيانها.

قمنا بمعايرة النظام باستخدام سورة الإخلاص واختباره على سورة التين. أظهرت نتائج الاختبار أن الدقة الإجمالية للنظام كانت 96 ٪، حيث حدد النظام بشكل صحيح 90 ٪ من الآيات، وعرض بنجاح الآية التالية 100 ٪ من الوقت، وتحديد الآيات غير الصحيحة بدقة بالترتيب في كل مرة (دقة 100 ٪)، وتحديد أخطاء الكلمات أو الحروف بشكل صحيح في الآية بدقة 95 ٪. يمكن أن يكون هذا النظام أداة مفيدة للأفراد الذين يرغبون في تحسين تلاوتهم للقرآن الكريم، وخاصة أولئك الذين يتعلمون قراءة القرآن. كما يمكن استخدامه للكشف عن الأخطاء التي تحدث أثناء التلاوة وتصحيحها، مما يحسن دقة التلاوة وجودتها.

7. التوصيات:

تم تصميم هذا المشروع واختباره في عينة من السور واقتصر على اختباره من قبل مجموعة صغيرة من المستخدمين بسبب ضغط الوقت ومحدودية الموارد. نحن نتطلع إلى تحسين هذا النظام من خلال بناء نموذجنا الخاص لتحويل الصوت إلى نص يمكن أن يعزز إخراج قابلية استخدام النظام. نوصي أيضًا بالنقاط التالية:

- استخدام تقنيات الذكاء الاصطناعي في تصحيح "أحكام" تلاوة القرآن الكريم.
- توفير نظام يصحح التلاوة بالصوت لمساعدة ذوي الاحتياجات الخاصة.
- استخدام البرمجة اللغوية العصبية وربطها بتفسير القرآن وفهمه.
- توفير إمكانيات الأجهزة ومجموعات البيانات، لخلق نموذج خاص لخدمة تحفيظ القرآن الكريم.
- إمكانية إضافة التعرف على الكلمات.
- تضمين "تشكيل" الحروف في العمل المستقبلي من خلال تدريب النموذج على النموذج المصمم للتعامل مع تلاوات القرآن.

8. المراجع:

1. Jordan, Michael I. "Artificial intelligence—the revolution hasn't happened yet." Harvard Data Science Review 1.1 (2019): 1-9.

2. Nilsson, Mikael, and Marcus Ejnarsson. "Speech recognition using hidden markov model." (2002).

3. Schuller, Björn, Gerhard Rigoll, and Manfred Lang. "Hidden Markov model-based speech emotion recognition." 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03).. Vol. 2. Ieee, (2003).

4. Kamath, Uday, John Liu, and James Whitaker. Deep learning for NLP and speech recognition. Vol. 84. Cham, Switzerland: Springer, (2019).

5. McKinney, Wes. Python for data analysis. " O'Reilly Media, Inc.", (2022).

6. Sahoo, Kabita, et al. "Exploratory data analysis using Python." International Journal of Innovative Technology and Exploring Engineering 8.12 (2019): 4727-4735.

7. Viarbitskaya, Tatsiana, and Andrzej Dobrucki. "Audio processing with using Python language science libraries." 2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA). IEEE, 2018.

8. Abushariah, M. A. A. M., Ainon, R. N., Zainuddin, R., Elshafei, M., & Khalifa, O. O. (2012). Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus. Int. Arab J. Inf. Technol., 9(1), 84-93.