



الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

الجامعة السورية الخاصة
كلية الهندسة المعلوماتية
هندسة الذكاء الصناعي وعلوم البيانات

تصنيف أمراض نبات البطاطا باستخدام تقنيات Deep learning

إعداد

عبدالله خادم الجامع

إشراف

الدكتورة ماجدة بكور

المهندس محمد العقلة

مقرر تطبيقات في

هندسة الذكاء الصناعي وعلوم البيانات

السنة الدراسية

2022-2023

الملخص

تعد لفحة أوراق البطاطا واحدة من أكثر أمراض النبات قدرة على التسبب بخسارة كبيرة في نبات البطاطا على مستوى العالم، ولذلك عملنا على تصميم وبناء تقنية جديدة لاكتشاف اصابة نبات البطاطا بمرض لفحة البطاطا باستخدام الشبكات العصبونية التلافيفية العميقة وهي احدى تقنيات التعلم العميق.

يعمل النظام على تصنيف اصابة أوراق نبات البطاطا الى احدى ثلاث فئات وهي ورقة سليمة، او ورقة مصابة بمرض لفحة البطاطا المبكرة، او ورقة مصابة بمرض لفحة البطاطا المتأخرة.

كانت عدد البيانات المجمعة هو 2456 والمأخوذة من موقع Kaggle.com, عملنا على تقسيمها لثلاث مجموعة تضم المجموعة الاولى بيانات التدريب وعددها 1984 صورة لأوراق البطاطا بحالة سليمة او مصابة بأحد المرضين المذكورين، اما المجموعة الثانية فهي مجموعة الاختبار وتضم 256 صورة. اما المجموعة الثالثة فهي مجموعة التحقق وتضم 216 صورة.

تم بناء شبكة CNN من عدة طبقات وعددها 16 طبقة ولقد استخدمنا توابع تفعيل بين الطبقات وهي على نوعين (relu) و (softmax) ولقد استغرقت عملية التدريب زمن قدره 45 د وكانت عملية التحقق ناجحة ومن ثم اختبرنا على عينات جديدة حيث بلغ معدل التصنيف للشبكة المصممة 99.2% وهو معدل عالي.

ان نظام التصنيف المصمم يمكن استخدامه للتأكد من المرض واستخدام العلاج المناسب للنبات وبالتالي تفادي خسارات كبيرة للمزارع وبلدنا

قائمة المصطلحات

الرقم	المصطلح	الاختصار	التعريف
1	Artificial Intelligence	AI	هو علم الهندسة وصناعة الآلات الذكية بما فيها البرمجيات الذكية، حيث تعمل برامج الذكاء الاصطناعي على محاكاة الوظائف المعرفية المتعلقة بقدرات الإنسان، كالتعلم، والإدراك وحل المشاكل، حتى إنها قادرة على التعلم بأقل المدخلات البشرية، وذلك عن طريق تحليل كميات ضخمة من البيانات.
2	Artificial Neural Network	ANN	مجموعة مترابطة من عصبونات افتراضية تنشئها برامج حاسوبية لتتأثر بعمل العصبون البيولوجي أو بنى إلكترونية تستخدم نموذج رياضي معين لمعالجة المعلومات بناء على خوارزميات معينة
3	Application Programming Interface	API	واجهات برمجة التطبيقات هي آليات تُمكن مكوّنين من برمجيات معينة الاتصال ببعضهما وفقاً لمجموعة من الخوارزميات الموضوعية
4	Convolutional Neural Network	CNN	الشبكة العصبية التلافيفية (CNN / ConvNet) هي فئة من الشبكات العصبية العميقة (deep neural network)، يتم تطبيقها بشكل شائع لتحليل كمية ضخمة من الصور.
5	Deep Learning	DL	التعلم العميق هو أحد فروع تعلم الآلة، إذ تكون الخوارزميات مستوحاة من بنية ووظيفة الدماغ والتي يُطلق عليها تسمية الشبكات العصبية الاصطناعية Artificial Neural Networks، وترتكز طريقة التعلم هذه على الشبكات العصبية الاصطناعية، والتي يمكن أن تكون خاضعة للإشراف أو غير مُشرف عليها.
6	Java Script	JS	لغة برمجة عالية المستوى، تستخدم أساساً في متصفحات الويب لإنشاء صفحات أكثر تفاعلية وتطوير صفحات ويب التفاعلية، وتطبيقات الويب،
7	Machine Learning	ML	تعلم الآلة هي إحدى الطرق التي تُدرّب فيها البرمجية على التعلم والتطور تلقائياً عن طريق التجربة، إذ تُستخدم الخوارزميات والنماذج الإحصائية للتعلم دون أي تدخل بشري.

جدول المحتويات Table Of Contents

2..... الملخص	
3..... قائمة المصطلحات	
4..... جدول المحتويات	
6..... الفصل الأول _ منهجية المشروع	1
6..... المقدمة	1.1
6..... تحديد نطاق المشكلة Problem Scoping	1.2
6..... تعريف مشكلة البحث Problem Identifying	1.2.1
7..... الهدف من البحث Problem Solve Goals	1.2.2
7..... النتائج المتوقعة Expected Results	1.2.3
7..... بنية البحث:	1.3
8..... الفصل الثاني _ التعرف على اصابة محصول البطاطا بالأمراض	2
8..... المقدمة	2.1
8..... الدراسة المرجعية	2.2
10..... الدراسة التحليلية للنظام المقترح:	2.3
10..... مرحلة تحضير البيانات Data Preparation	2.3.1
10..... مرحلة جمع البيانات Data Collection	2.3.1.1
10..... تعريف نوع البيانات Data Type	2.3.1.2
10..... أدوات اكتساب البيانات Data Acquisition Tools	2.3.1.3
11..... مرحلة تهيئة بنية البيانات Data Structuring	2.3.2
11..... مرحلة تنظيف البيانات Data Cleaning	2.3.2.1
11..... مرحلة تعزيز البيانات Data Enrichment	2.3.2.2
11..... مرحلة ما قبل المعالجة Data Preprocessing	2.3.3
11..... تغيير حجم الصورة Resize	2.3.3.1
11..... تصغير حجم الصورة Rescale	2.3.3.2
12..... مرحلة معالجة البيانات Data Processing	2.3.4
13..... الدراسة التصميمية والتنفيذية للنظام المقترح	2.4
13..... مرحلة بناء النموذج وتدريبه Model Building & Fit	2.4.1
17..... انشاء خادم باستخدام Fast API	2.4.2
17..... انشاء موقع الكتروني باستخدام React JS	2.4.3
18..... الفصل الثالث _ التطبيق العملي لنظام المقترح	3

18.....الموقع الالكتروني	3.1
20.....التقنيات المستخدمة في انشاء التطبيق:	3.2
20Technologies	
22.....الفصل الرابع _ النتائج والأعمال المستقبلية	4
Error! Bookmark not defined.: النتائج	4.1
22.....الأعمال المستقبلية:	4.2
23.....الخلاصة:	
24.....الفصل الخامس _ المراجع	5

1 الفصل الأول _ منهجية المشروع

1.1 المقدمة

تعتبر البطاطا محصول غذائي معروف عالمياً حيث يُزرع في كل أنحاء العالم، وقد تم تصنيفه في المركز الرابع عالمياً من حيث الأهمية الغذائية بعد محصول القمح والأرز والذرة.

ولأهمية هذا المحصول يعتبر الغذاء الشعبي في الكثير من الدول، وبما ان محصول البطاطا يتميز بكونه مادة سهلة الهضم لاحتوائه على الياف ملينة للمعدة والتي تعتبر خير واق من سرطان الأمعاء. الا انه من جهة اخرى يحتاج محصول البطاطا إلى رعاية خاصة عند عملية تخزينه في المستودعات، حيث يجب أن تخزن البطاطا في مستودعات باردة لتجنب فسادها.

بلغ الإنتاج العالمي لمحصول البطاطا نحو 310 مليون طن متري سنوياً. وتتصدر الصين قائمة الدول المنتجة للبطاطا، وتليها روسيا ثم الهند فالولايات المتحدة الأمريكية

1.2 تحديد نطاق المشكلة Problem Scoping

1.2.1 تعريف مشكلة البحث Problem Identifying

يعمل المزارع على انتاج محصول البطاطا لتوفيره في السوق المحلية خلال مدة 4 أشهر وبعد كل هذا التعب قد يتفاجأ المزارع بمردود قليل والسبب إصابة المحصول بأحد الامراض التي قد تؤدي لموت الزهرة او إصابة الثمرة وبالتالي الكثير من الخسائر المادية الى جانب الجهد والتعب والكثير من الوقت في انتظار المحصول.

ومن هنا كان لابد من العمل على حل هذه المشكلة لتفادي الخسارة الغير المتوقعة منه والتي تعتمد على الظروف الجوية والبيئية.

1.2.2 الهدف من البحث Problem Solve Goals

هناك العديد من الامراض التي يمكن ان تصيب نبات البطاطا ولذلك تم العمل عليها ووضع تقنية قادرة على اكتشاف الاصابة بها وبالتالي يمكن تلخيص الاهداف من البحث في:

- 1- تحليل امراض محصول البطاطا والتركيز على نوعين من المرض
- 2- وضع نموذج لاكتشاف الامراض المدروسة
- 3- تصميم وبناء نظام ذكي للتنبؤ بالإصابة بالمرض وتحديد اسم المرض

1.2.3 النتائج المتوقعة Expected Results:

لحل المشكلة السابقة نعمل على انشاء تطبيق ويب يمكن ان يستفيد منه المهندسين الزراعيين والفلاحين والدول التي تعتمد في انتاجها على محصول البطاطا وكل من يهتم بهذا الاتجاه حيث يقوم المستخدم بأخذ صورة لورقة النبات وتحميلها على التطبيق ويقوم التطبيق بتحليل الاصابة وتحديد فيما إذا كان نبات البطاطا سليم ام مصاب بأحد هذه الأمراض ام لا.

وبالتالي يمكن تلخيص النتائج المتوقعة فيما يلي:

- 1- وضع خوارزمية لتصنيف امراض النبات
- 2- استخدام تقنيات التعلم العميق في المجال الزراعي
- 3- تجنب الكوارث الطبيعية من خلال التنبؤ المبكر بالمرض ومعالجته

1.3 بنية البحث:

يعرض الفصل الأول منهجية المشروع والهدف منه وتحديد نطاق المشكلة والنتائج المتوقعة لحل المشكلة , أما الفصل الثاني فيتحدث عن التعرف على امراض محصول البطاطا حيث يتضمن الدراسة المرجعية والتحليلية والتصميمية والتنفيذية للنظام المقترح , أما الفصل الثالث فيعرض التطبيق العملي للنظام ونظرة عامة لموقع الالكتروني والتقنيات المستخدمة في انشاء التطبيق , أما الفصل الرابع فيتحدث عن النتائج الخاصة بالبحث والتوصيات الممكنة في المستقبل , وفي النهاية الفصل السادس يحتوي على بعض المراجع الخاصة بالبحث.

2 الفصل الثاني _ التعرف على اصابة محصول البطاطا بالأمراض

2.1 المقدمة

هنالك نوعان من الأمراض الشائعة المعروفة باسم اللفحة المبكرة واللفحة المتأخرة، وإذا تمكن المزارع من اكتشاف هذه الأمراض مبكراً وتطبيق العلاج المناسب فسيتمكن من تقليل الخسار المادية , تختلف طرق العلاج الخاصة باللفحة المبكرة واللفحة المتأخرة ولذلك يجب تحديد نوع المرض الموجود بدقة قبل تطبيق العلاج.

2.2 الدراسة المرجعية

اتجه العديد من المهتمين للعمل على الكشف عن الامراض التي تصيب النبات وخاصة مرض البطاطا ومن هذه الدراسات:

- في ورقة بحثية بعنوان: **“Krishi Mitra: Using Machine Learning to Identify Diseases in plants”** استخدام **TensorFlow** بواسطة منهجية نموذج **CNN** لتنفيذ مشروعهم. كانت ميزة هذا النموذج أنه تم حساب مساحة الورقة فقط ويمكن التعرف على المكان المسبب للأمراض في قصب السكر. كان عيب في هذا الحل أن التعقيد الحسابي عالي جداً وهو شيء مطلوب لتنفيذه [1].

- في ورقة بحثية بعنوان: **“Severity Identification of Potato Late Blight Disease from Crop Images Captured under Uncontrolled Environment”**.

استخدم الباحثون في تطبيق النموذج **clustering mean-c Fuzzy** بحيث ان الشبكة العصبونية لا يحتاج إلى تدريب خاص للمزارعين حيث أن مجموعة البيانات لها صور بزوايا مختلفة كانت الميزة الرئيسية للنموذج. كان العيب الوحيد هو أن الصور تم التقاطها بواسطة مزارعين غير مدربين لم تكن جيدة وتحتوي على مجموعة من الأوراق ذات خلفية مشوشة غير واضحة في عدة شرائح [2].

- وفي ورقة بحثية بعنوان **"Potato Disease Detection Using Machine Learning"** كانت معالجة الصور هي التقنية المستخدمة هنا. اكتسب استخدام نموذج **CNN** دقة تحقق تصل إلى 90% وكان الأمن النماذج الأعلى دقة في هذا المشروع. كان النموذج التدريبي الكبير المطلوب هو العيب الرئيسي لهذا النموذج. [3]

- وفي ورقة بحثية بعنوان **"detecting the infectious area along with disease using deep learning in tomato plant leaves"**، كانت المنهجية المقترحة هي **Mask CNN-R** التعلم العميق. كانت ميزة ذلك أن استخدام قناع **CNN-R** يمكن أن يحسن ويسرع عملية الكشف عن العوامل الممرضة في أوراق النبات. كان العيب هو أن الأمر يستغرق وقتاً أطول للمعالجة [4].

- وفي ورقة بحثية بعنوان **"POTATO LEAF DISEASE DETECTION AND CLASSIFICATION USING CNN"** كانت معالجة الصور هي التقنية المستخدمة هنا. اكتسب استخدام نموذج **CNN** دقة تحقق تصل إلى 91.4% وكان الأكثر دقة في هذا المشروع. [5]

2.3 الدراسة التحليلية للنظام المقترح:

2.3.1 مرحلة تحضير البيانات Data Preparation:

للبدء بالعمل على التعرف على إصابة محصول البطاطا بأحد الامراض كان لابد من القيام بجمع البيانات حول هذا النبات وفي حالتنا كانت عملية الجمع عن صور رقمية لهذا النبات

2.3.1.1 مرحلة جمع البيانات Data Collection:

وللحصول على بيانات عن نبات البطاطا كان الأفضل البحث عنها في موقع Kaggle كونه الأكثر ثقة عالمياً

<https://www.kaggle.com/arjuntejaswi/plant-village>

ولدى البحث ضمن موقع Kaggle تم العثور على أكثر من قاعدة بيانات عن امراض نباتات وتم اختيار من قاعدة البيانات الملفات التي تحتوي على صور لأمراض نبات البطاطا حيث تحتوي الملفات على عدد جيد من الصور الرقمية وكان العدد الإجمالي للصور 2456 صورة.

2.3.1.2 تعريف نوع البيانات Data Type:

لدى دراسة نوعية البيانات المتوفرة، تم اختيار بيانات نبات البطاطا التي تحتوي على صور للنبات في حالته سليماً من الامراض وكان العدد الإجمالي للصور هو 456 صورة والحالة الأخرى مصاباً بمرض اللفحة المبكرة وكان العدد الإجمالي للصور هو 1000 صورة والحالة الأخرى مصاباً بمرض اللفحة المتأخرة وكان العدد الإجمالي للصور هو 1000 صورة.

2.3.1.3 أدوات اكتساب البيانات Data Acquisition Tools

ولدى تحليل الطريقة التي سيتم فيها اكتساب البيانات في نظامنا ليتم التعرف على مدى إصابة النبات بمرض لاحظنا انه من الأفضل الاعتماد على الكاميرا الرقمية لالتقاط صورة للنبات

2.3.2 مرحلة تهيئة بنية البيانات Data Structuring:

2.3.2.1 مرحلة تنظيف البيانات Data Cleaning:

البيانات التي تم اختيارها من Kaggle.com هي بيانات نظيفة الى حد ما ولا تحوي قيم خاطئة او فارغة

حيث ان البيانات تحوي على صور ذات حجم 256×256

2.3.2.2 مرحلة تعزيز البيانات Data Enrichment

وهي عملية Augmentation انشاء عدة صور جديدة من الصور الأساسية عن طريق بعض العمليات مثل (تدوير -تقريب -تباين) وهذا يؤدي الى زيادة قوة النموذج بسبب الزيادة في كمية البيانات المستخدمة

2.3.3 مرحلة ما قبل المعالجة Data Preprocessing:

2.3.3.1 تغيير حجم الصورة Resizing

وهي عملية تعديل حجم الصورة الدخل بعملية التكبير او التصغير لحجم محدد تم اعتماده في تصميم التقنية وهو عبارة عن 256×256

2.3.3.2 تقييس الصورة Rescaling

وهي عملية تقييس بيانات الصورة الى مجال جديد وهو $[0, 1]$ من خلال المعادلة التالية :

$$x_{new} = \frac{x_{old}}{255} \dots \dots (2.1)$$

ليكون هناك تقارب بين البيانات وكون تابع التفعيل خرجه بين 0,1

2.3.4 مرحلة معالجة البيانات Data Processing:

يتم في البداية تحميل البيانات من الملف والقيام بعملية الخلط `shuffle` من اجل عدم تدريب النموذج على البيانات بطريقة مرتبة (مما يؤدي الى إعطاء قيم خاطئة) وفق الكود التالي:

```
In [3]: dataset = tf.keras.preprocessing.image_dataset_from_directory(
    "../PlantVillage",
    seed=123,
    shuffle=True,
    image_size=(IMAGE_SIZE, IMAGE_SIZE),
    batch_size=BATCH_SIZE
)
```

Found 2456 files belonging to 3 classes.

ومن ثم نقوم بتقسيم البيانات الى ثلاث اقسام:

(a) بيانات خاصة بالتدريب **Training** وهي تحوي على 80% من البيانات

(b) بيانات خاصة بالتحقق اثناء التدريب بنهاية كل حقبة **Validation** وهي

تحوي على 10% من البيانات

(c) بيانات خاصة بالاختبار بعد الانتهاء من التدريب لحساب دقة النموذج

Testing وهي تحوي على 10% من البيانات

والكود التالي بعرض طريقة التقسيم:

```
In [14]: def get_dataset_partitions_tf(ds, train_split=0.8, val_split=0.1, test_split=0.1, shuffle=True, shuffle_size=10000):
    assert (train_split + test_split + val_split) == 1

    ds_size = len(ds)

    if shuffle:
        ds = ds.shuffle(shuffle_size, seed=12)

    train_size = int(train_split * ds_size)
    val_size = int(val_split * ds_size)

    train_ds = ds.take(train_size)
    val_ds = ds.skip(train_size).take(val_size)
    test_ds = ds.skip(train_size).skip(val_size)

    return train_ds, val_ds, test_ds
```

2.4 الدراسة التصميمية والتنفيذية للنظام المقترح

2.4.1 مرحلة بناء النموذج وتدريبه Model Building & Fit:

عملنا على بناء شبكة عصبونية تلافيفية CNN من اجل تدريب النموذج لأنها من أحدث التقنيات التي يتم العمل بها في الوقت الحالي في مجال تصنيف الصور والتي تستطيع إعطاء نتائج دقيقة

في عملية بناء الشبكة يوجد أربع أنواع اساسية من العمليات وهي:

(a) Convolution: وهي الطبقة التي يتم فيها استخلاص ال feature باستخدام الفلاتر التي يتم استخدامهم عن طريق تطبيق تابع التنشيط activation function (Relu) على جميع عناصر الصورة ويتم تحديد حجم الفلتر باستخدام ال kernel size (3*3) وكما يتم تحديد عدد الفلاتر المراد استخدامها (32 و 64)

(b) Pooling: هناك أنواع مختلفة له حيث يتم العمل على الاحتفاظ بالخصائص الموجودة مع تقليل حجم الصورة مما يؤدي الى تقليل العمليات الحسابية

ويتم إعادة هتان العمليتان 6 مرات بعد اجراء عدة تجارب وحساب الاخطاء

(c) Flatten: وهي عملية تسوية للبيانات من اجل ادخال البيانات الى الطبقة التالية وهي شبكة عصبونية طبيعية ANN

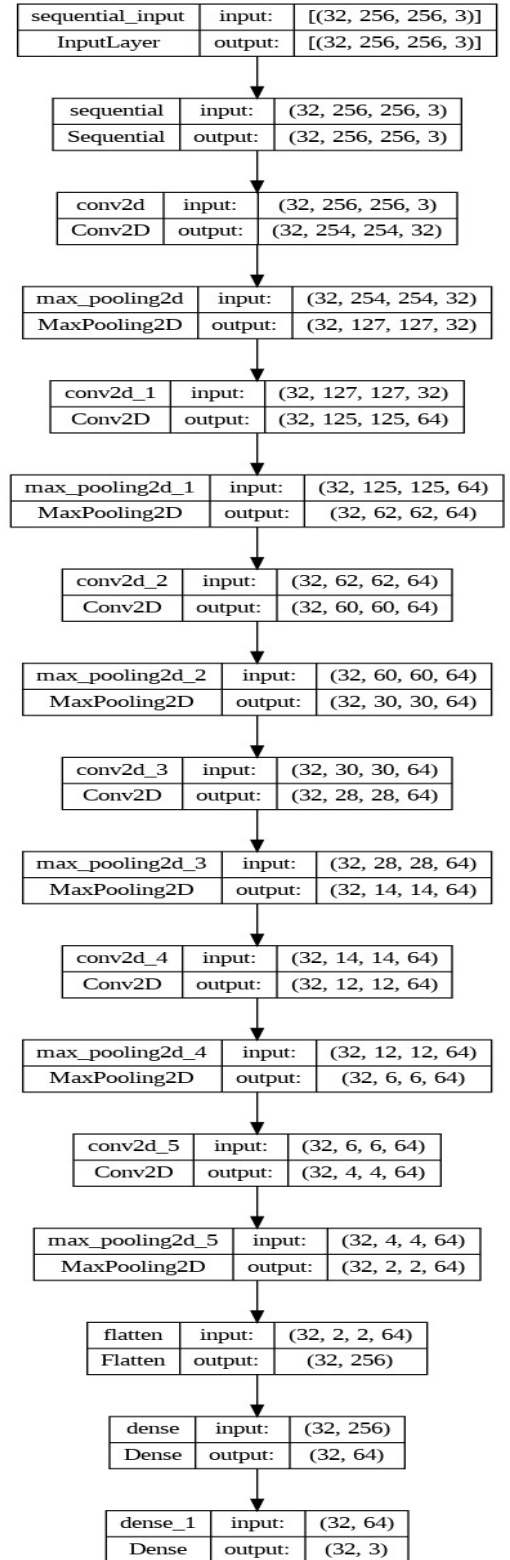
(d) Dense: وبعدها يتم إدخالها الى شبكة من الخلايا العصبونية ومنها الى الطبقة النهائية الخاصة بالخرج النهائي والتي تتكون بشكل أساسي من عدد من الخلايا العصبونية وتكون مساوية لعدد الصفوف المراد تقسيمها (3 صفوف) وتطبيق تابع التنشيط (Softmax) activation function

وفيما يلي تلخيص لهيكلية النموذج الذي تم بناءه:

Model: "sequential_2"

Layer (type)	Output Shape	Param #
=====		
sequential (Sequential)	(32, 256, 256, 3)	0
conv2d (Conv2D)	(32, 254, 254, 32)	896
max_pooling2d (MaxPooling2D)	(32, 127, 127, 32)	0
conv2d_1 (Conv2D)	(32, 125, 125, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(32, 62, 62, 64)	0
conv2d_2 (Conv2D)	(32, 60, 60, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(32, 30, 30, 64)	0
conv2d_3 (Conv2D)	(32, 28, 28, 64)	36928
max_pooling2d_3 (MaxPooling2D)	(32, 14, 14, 64)	0
conv2d_4 (Conv2D)	(32, 12, 12, 64)	36928
max_pooling2d_4 (MaxPooling2D)	(32, 6, 6, 64)	0
conv2d_5 (Conv2D)	(32, 4, 4, 64)	36928
max_pooling2d_5 (MaxPooling2D)	(32, 2, 2, 64)	0
flatten (Flatten)	(32, 256)	0
dense (Dense)	(32, 64)	16448
dense_1 (Dense)	(32, 3)	195
=====		

Total params: 183,747



ومن ثم يتم تحديد توابع التحسين **Optimize (adam)** والخسارة **Loss metrics** والمقاييس **(Sparse_Categorical_Crossentropy)** وهو تابع الدقة لقياس بعملية التدريب **(Accuracy)**

وفي النهاية يتم تدريب النموذج على البيانات ويتم تحديد عدد المرات التي سيتم تدريب النموذج عليها باستخدام **Epoch (50)**

```
Epoch 49/50
54/54 [=====] - 17s 312ms/step - loss: 0.0153 - accuracy: 0.9954 - val_loss: 0.0096 - val_accuracy: 0.9948
Epoch 50/50
54/54 [=====] - 17s 311ms/step - loss: 0.0150 - accuracy: 0.9936 - val_loss: 0.0359 - val_accuracy: 0.9896
```

ويتم في كل مرة بنهاية ال **epoch** قياس توابع الخطأ والدقة بنسبة لبيانات التدريب وبيانات التحقق

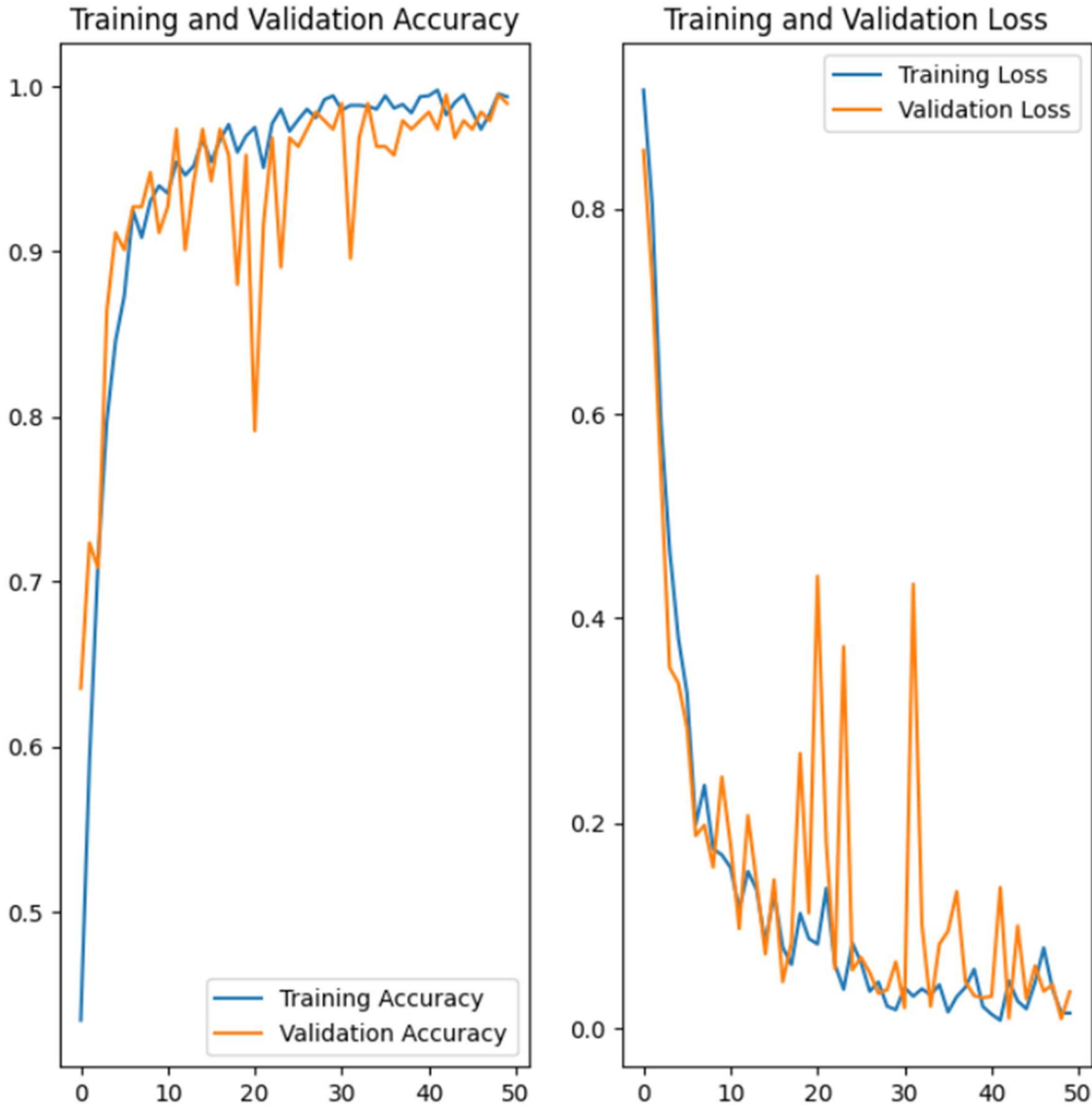
وفي كل دورة نلاحظ زيادة الدقة الخاصة بالنموذج واقتربها من 1 وانخفاض نسبة الخطأ واقتربها من 0 وهذا دليل على قوة النموذج المستخدم

وبعدها يتم حساب الدقة الخاصة بالنموذج كاملا باستخدام البيانات الخاصة بالاختبار

```
In [27]: scores = model.evaluate(test_ds)
```

```
8/8 [=====] - 2s 34ms/step - loss: 0.0423 - accuracy: 0.9922
```

وفيما يلي رسم بياني يوضح تغير توابع الدقة والخطأ مع ال Epoch:



ويعيد النموذج بشكل أساسي مدى صحة الصورة بنسبة ل الصفوف الثلاث على شكل مصفوفة ولذلك نقوم بتحديد المتغير ذو القيمة الأعلى وربطه باسم الصف الخاص فيه وبعدها نقوم بحفظ النموذج مع الاهتمام بالإصدارات الخاصة من اجل ربطه ب Fast API

2.4.2 انشاء خادم باستخدام Fast API:

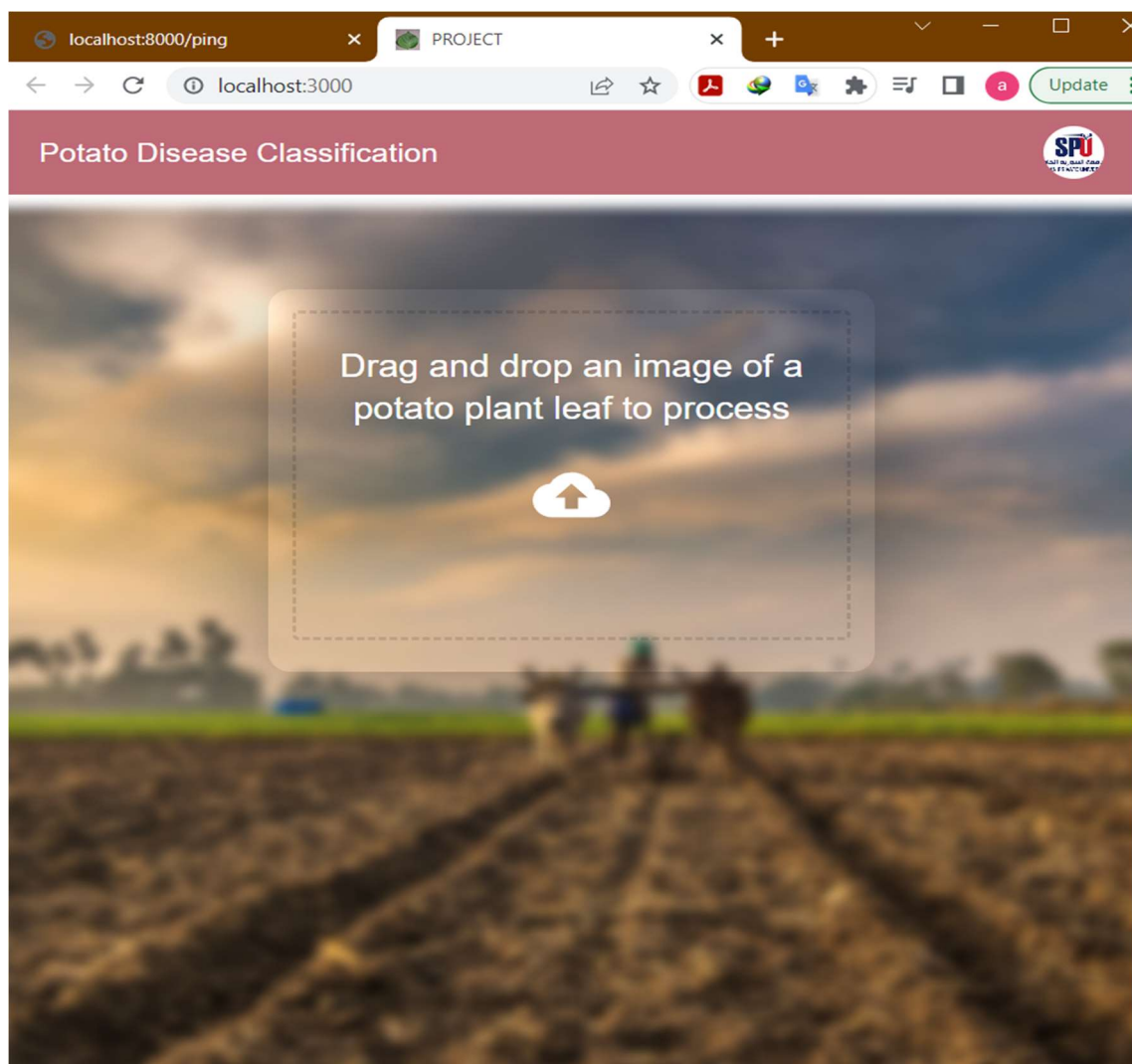
يقوم الخادم بأخذ الملف الخاص من الموقع وتحويله الى مصفوفة رقميه ومن ثم ارساله ليتم معالجته من النموذج ثم إعادة التصنيف مع درجة الثقة في هذا التصنيف حيث يقوم النموذج بإعادة مصفوفة مكونة من ثلاث قيم وهي نسبة انتماء الصورة لكل من الصفوف الثلاث ومن ثم يقوم بأخذ الأكبر وربطه مع الصنف الخاص به وكما يستطيع الخادم من التعامل مع عدد كبير من الطلبات دون حدوث خطأ باستخدام `async & await`

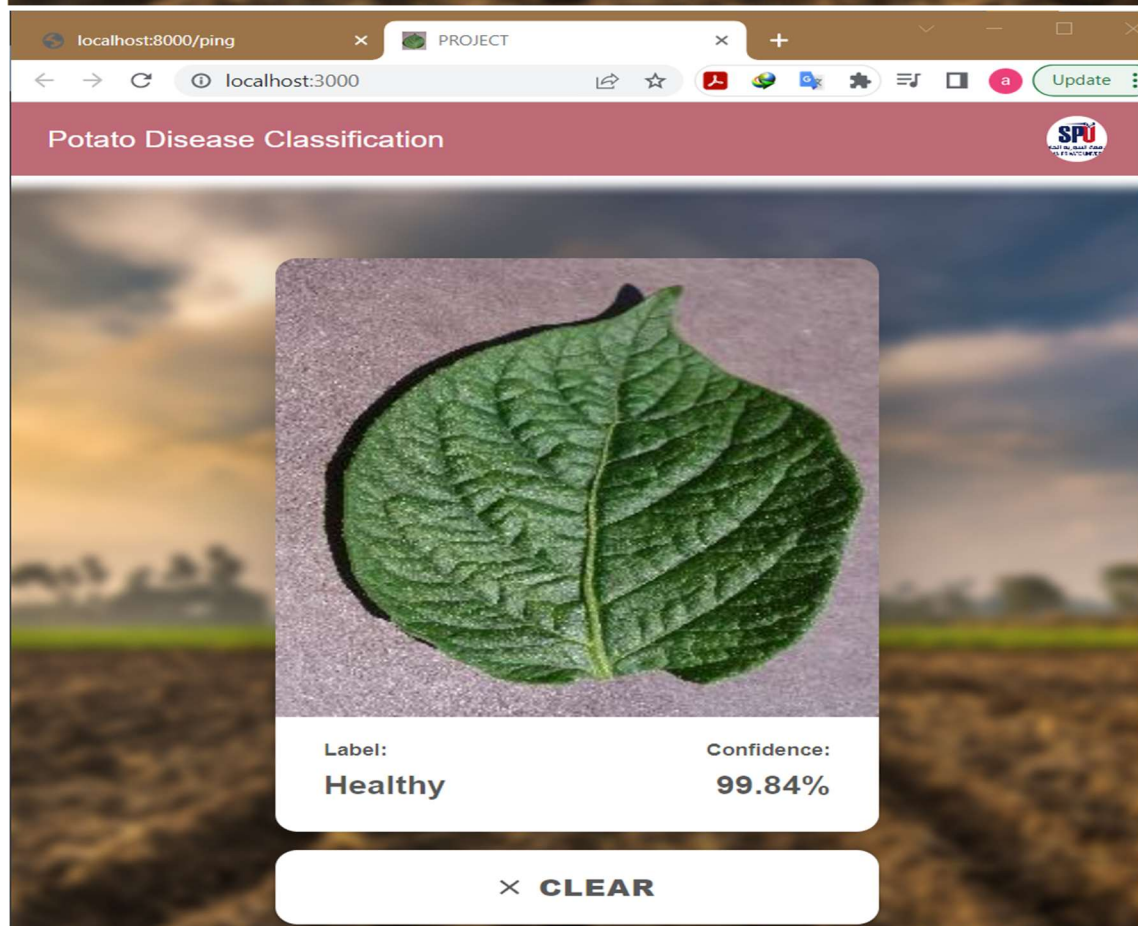
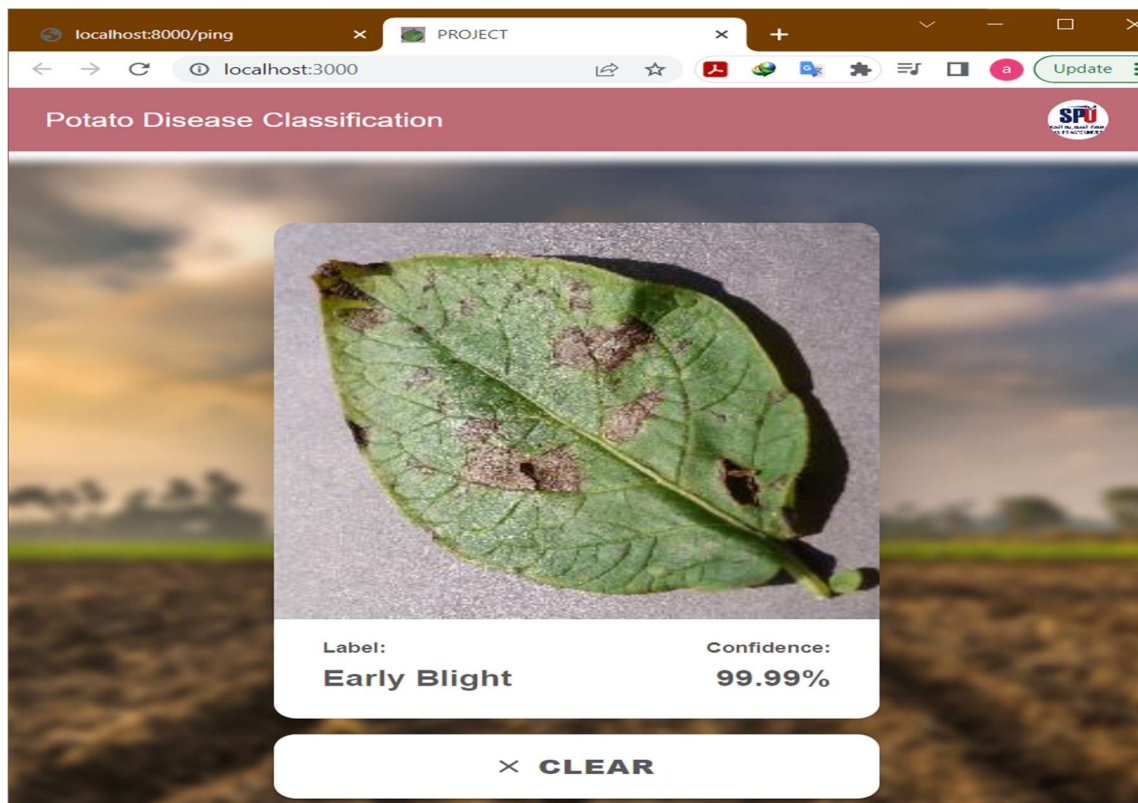
2.4.3 انشاء موقع الكتروني باستخدام React JS:

يقوم الموقع بأخذ الصورة وربطها مع الخادم الخاص بنا لإرجاع القيم الخاصة بالتصنيف كما يتيح لك الموقع إضافة الصورة عن طريق اختيار الصورة او عن طريق القائها `Drag and Drop`

3 الفصل الثالث _ التطبيق العملي لنظام المقترح

3.1 الموقع الالكتروني





3.2 التقنيات المستخدمة في انشاء التطبيق:

Technologies

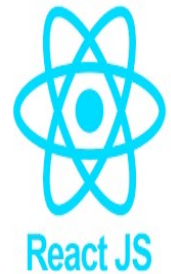
1. Frameworks:

- **React js:**

React is a JavaScript library for building user interfaces.

and used to build single-page applications.

and allows us to create reusable UI components.



- **Fast API:**

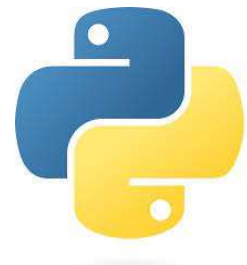
FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints.



2. Programming Languages

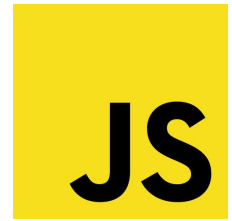
- **Python:**

Python is a high-level, object-oriented, scripting, interpretive, and open source programming language.



- **Java Script:**

JavaScript is a client-side and server-side scripting language inserted into HTML pages and is understood by



3. Web Technologies

- **Hypertext Markup Language (HTML)**

HTML is the standard markup language for creating Web pages. It describes the structure of a Web page. In addition, it consists of a series of elements.



- **CSS**

CSS is the language that is used to style an HTML document and describes how HTML elements should be displayed.



4. Library of AI:

- Tensorflow
- Matplotlib
- Keras
- Numpy



4 الفصل الرابع _ النتائج والأعمال المستقبلية

4.1 النتائج:

يمكن تلخيص النتائج التي حصلنا عليها فيما يلي:

- 1- تم وضع خوارزمية لتحليل امراض محصول البطاطا والتركيز على نوعين من المرض (اللفحة المبكرة واللفحة المتأخرة)
- 2- بناء نموذج لاكتشاف الامراض المدروسة باستخدام CNN
- 3- تدريب النموذج على البيانات المجمعة وإعطاء دقة عالية 99.2% مع خطأ صغير جداً 4.2%
- 4- تصميم وبناء نظام ذكي للتنبؤ بالإصابة بالمرض وتحديد اسم المرض مع نسبة التأكد
- 5- يسمح النظام قدرة على تجنب الكوارث الطبيعية من خلال التنبؤ المبكر بالمرض ومعالجته قبل نهاية الموسم مما يؤدي الى انقاذ المحصول من خسائر ممكنة

4.2 الأعمال المستقبلية:

- العمل على رفع الخادم الخاص على الكلاود باستخدام GCP google cloud او AWS
- استخدام خدمات TF.SERVING من اجل سهولة ربط الإصدارات ودقة العمل
- بناء تطبيق اندرويد باستخدام React native يمكن المزارعين من التقاط صور فورية للنباتات والحصول على النتائج بشكل فوري

الخلاصة:

في الوقت الحاضر ,يقوم المزارعون بكل ما يملكون من اجل حماية عملهم والحفاظ على محاصيلهم , ولكن في بعض الأحيان تصادفهم بعض العوائق التي قد تعود عليهم بخسائر مادية كبيرة , لذلك قمنا بالعمل على التطبيق من اجل حماية العائلات المزارعة الفقيرة التي لا تستطيع دفع أجور الخبراء الزراعيين من دمار محاصيلهم التي هي مصدر دخلهم الوحيد ومنه فإن هدفنا الأساسي هو انشاء تطبيق ويب يتم تقديمه للمزارعين , حيث يقوم المزارعون بأخذ الصورة الخاصة بالنبات وتحميلها على الموقع ويقوم الموقع بإخبارهم إذا كان نبات البطاطا سليم ام مصاب بأحد هذه الأمراض من اجل تطبيق العلاج المناسب في الوقت المناسب.

تم العمل على النظام ليقوم بتصنيف اصابة أوراق نبات البطاطا الى احدى ثلاث فئات وهي ورقة سليمة، او ورقة مصابة بمرض لفحة البطاطا المبكرة، او ورقة مصابة بمرض لفحة البطاطا المتأخرة.

حيث تم العمل على بيانات مجمعة من موقع Kaggle.com عددها 2456

تم بناء شبكة CNN من عدة طبقات وعددها 16 طبقة ولقد استخدمنا توابع تفعيل بين الطبقات وهي على نوعين (relu) و (softmax) ولقد استغرقت عملية التدريب زمن قدره 45 د وكانت عملية التحقق ناجحة ومن ثم اختبرنا على عينات جديدة حيث بلغ معدل التصنيف للشبكة المصممة 99.2% وهو معدل عالي.

وبعد ذلك تم العمل على ربط النموذج باستخدام FAST API ومن ثم تم انشاء موقع الكتروني باستخدام REACT JS وربطه مع الAPI حيث يقوم التطبيق بأخذ الصورة من المزارعين والتصنيف النبات إن كان في حالة صحة أو مريض.

5 الفصل الخامس _ المراجع

1. Parul Sharma, Yash Paul Singh Berwal, Wiqas Ghai “KrishiMitr (Farmer’s Friend): Using Machine Learning to Identify Diseases in Plants” In 2018 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), pp 29-34. IEEE 2018.
2. Sandika Biswas, Bhushan Jagyasi, Bir Pal Singh, Mehi Lal “Severity identification of Potato Late Blight disease from crop images captured under uncontrolled environment“ IEEE 2014
3. Marjanul Islam Tarik, Sadia Akter , Abdullah Al Mamun, Abdus Sattar Potato Disease Detection Using Machine Learning IEEE 2021
4. Piyush Juyal, Sachin Sharma Detecting the Infectious Area Along with Disease Using Deep Learning in Tomato Plant Leaves IEEE 2020
5. Anushka Bangal, Dhiraj Pagar, Hemant Patil, Neha Pande POTATO LEAF DISEASE DETECTION AND CLASSIFICATION USING CNN, International Journal of Research Publication and Reviews 2020
6. <https://www.tensorflow.org/>
7. <https://fastapi.tiangolo.com/lo/>
8. <https://react.dev/>
9. https://www.researchgate.net/publication/347877211_Artificial_Intelligence_in_Potato_Leaf_Disease_Classification_A_Deep_Learning_Approach
10. <https://cipotato.org/blog/smartphone-app-help-farmers-control-potato-sweetpotato-diseases/>
11. <https://www.rtb.cgiar.org/news/ai-for-diagnosing-potato-diseases-will-benefit-millions-of-rural-farmers/#:~:text=The%20AI%2Dpowered%20mobile%20app,manage%20diseases%20in%20their%20field>

Potato Disease Classification Using Deep Learning

Presented By

Abdullah Khadem Aljamee

Supervised By

Dr. Majida Albakoor

Eng. Mohammed Alokleh

Applications Project

Academic Year

2022-2023