

إذا كنت تبحث عن مصطلح يمكن أن يعبر عن التكنولوجيا في القرن الحادي والعشرين فقد يكون الذكاء الاصطناعي هو المرشح الأمثل .

فالذكاء هو مجال التقدم في علم الحاسوب يهدف إلى تطوير أنظمة يمكنها القيام بمهام معقدة بدون التدخل البشري , فالذكاء ليس مجرد مفهوم نظري بل هو جزء لا يتجزأ من حياتنا اليومية .

أما عن الذكاء الاصطناعي في الطب

على الرغم من أنه لا يزال في مراحله الأولى إلا أن الذكاء الاصطناعي يستعد لإحداث تحول في ممارسة الطب وتقديم الرعاية الصحية. حيث حقق تقدماً ملحوظاً في السنوات الماضية في العديد من التخصصات مدعوماً بخوارزميات التعلم الآلي حيث يمكن للرؤية الحاسوبية أن تظهر خصائص وسمات من الصور من الصعب علينا رؤيتها ومعرفتها وهذا هو مبدأ عمل الشبكات العصبونية الالتفافية (CNN(Convolutional Neural Network))

و لعل من الاستخدامات المهمة للذكاء الاصطناعي في مجال الطب الكشف المبكر عن سرطان الرئة للحد من انتشاره حيث يعد سرطان الرئة من أخطر الأمراض في العالم اليوم وهو السبب الرئيسي للوفيات في السنوات السابقة وفق دراسة أجرتها منظمة الصحة العالمية ومن هنا جاءت الحاجة لإيجاد طريقة لاكتشاف السرطان بشكل مبكر لتتم معالجته.

في هذا المشروع قمنا ببناء شبكة عصبونية التلافية باستخدام مبدأ التعلم العميق تقوم هذه الشبكة بتصنيف أنواع سرطان الرئة بالاعتماد على صور شعاعية للرئتين .

Abstract

If you are looking for a term that can express the technology of the 21st century, artificial intelligence may be the perfect candidate.

Artificial intelligence is a field of computer science that aims to develop systems that can perform complex tasks without human intervention. Artificial intelligence is not just a theoretical concept but is an integral part of our daily lives.

As for artificial intelligence in medicine

Although it is still in its early stages, artificial intelligence is poised to revolutionize the practice of medicine and healthcare delivery. It has made significant progress in recent years in many specialties, supported by machine learning algorithms. Computer vision can reveal characteristics and features from images that are difficult for us to see and recognize. This is the principle on which convolutional neural networks work.

Perhaps one of the most important uses of artificial intelligence in medicine is the early detection of lung cancer to reduce its spread. Lung cancer is one of the most serious diseases in the world today and is the leading cause of death in recent years, according to a study by the World Health Organization. Hence the need to find a way to detect cancer early so that it can be treated.

In this project, we built a convolutional neural network using the principle of deep learning. This network classifies types of lung cancer based on X-ray images of the lungs.

1 الفصل الأول :

1	مقدمة
2	1-1- لمحة عن سرطان الرئة
3	1-2- الدراسات المرجعية
6	1-3- تعلم الآلة
6	1-3-1- أنواع تعلم الآلة
7	1-3-2- خطوات تعلم الآلة
8	1-4- التعلم العميق
9	1-4-1- مميزات التعلم العميق
10	1-4-2- معمارية التعلم العميق
13	1-5- البيئة والأدوات المستخدمة
14	1-6- منهجية التطور

2 الفصل الثاني :

15	2-1- التعامل مع البيانات
16	2-2- الشبكة العصبية الالتفافية
17	2-2-1- طبقات الشبكة العصبية الالتفافية

3 الفصل الثالث :

21 3-1 استيراد المكتبات

21 3-2 استيراد الداتا

21 3-3 قراءة الداتا

22 3-4 تحليل الداتا

23 3-5 تجهيز الداتا للتدريب

24 3-6 بناء الشبكة العصبية

26 3-7 تدريب الداتا

27 3-8 النتائج

28 3-9 الصعوبات

28 3-10 التوصيات

29 المراجع

قائمة الأشكال

- الشكل 1 - 1 ارتباط التعلم العميق بتعلم الآلة والنكاء الاصطناعي. 8
- الشكل 1 - 2 معمارية التعلم العميق 12
- الشكل 1 - 3 صور أشعة سينية 13
- الشكل 2 - 1 طبقات الشبكة العصبية الالتفافية. 16
- الشكل 2 - 2 تطبيق فلتر على الصورة 17
- الشكل 2 - 3 تطبيق عملية التجميع. 19
- الشكل 3 - 1 عدد عناصر الداتا. 22
- الشكل 3 - 2 الشبكة العصبية. 25
- الشكل 3 - 3 مراحل التدريب. 26
- الشكل 3 - 4 مخطط نتائج التدريب. 27

قائمة المصطلحات

المصطلح	المختصر الموافق
Adenocarcinoma	ACA
Artificial Neural Network	ANN
Convolutional Neural Network	CNN
Neuroendocrine	N
Squamous Cell Carcinoma	SCC

تعد الرؤية الحاسوبية تقنية تغير مفهومنا للتفاعل مع العالم , ستستمر في فتح آفاق جديدة وتقدم حلولاً مبتكرة لمختلف التحديات . فتخيل عالماً مليئاً بتطبيقات ذكية تحدث ثورة في مختلف المجالات , هذا هو وعد الرؤية الحاسوبية فهي تُمكن الحواسيب من معالجة الصور والفيديوهات، استخلاص المعلومات منها، وفهم محتواها. تشبه عملية الرؤية الحاسوبية إلى حد كبير عملية الإدراك البصري لدى البشر، حيث يتم التقاط الصورة أو الفيديو بواسطة كاميرا، ثم معالجتها باستخدام خوارزميات ذكية لفهم مكوناتها وتحديد خصائصها.

تعتمد الرؤية الحاسوبية على مزيج من التقنيات، تشمل:

معالجة الصور: تحليل خصائص الصورة مثل الألوان، الشكل، والملمس.

التعلم الآلي: تدريب خوارزميات على كميات هائلة من البيانات المرئية لتعلم التعرف على الأنماط والتصنيفات.

الذكاء الاصطناعي: تمكين الحواسيب من فهم المشهد المرئي واستخلاص المعنى منه.

أيضاً لها عدة تطبيقات تشمل:

التعرف على الوجه: المستخدمة في أنظمة المراقبة، التحكم في الوصول، وشبكات التواصل الاجتماعي.

التعرف على الأشياء: المستخدمة في أنظمة الفرز والتخزين الآلي، والبحث عن الصور، والمساعدة في التسوق.

القيادة الذاتية: تمكين المركبات من التنقل بأمان على الطرق دون تدخل بشري.

الرعاية الصحية: تحليل الصور الطبية لتشخيص الأمراض، وتوجيه العمليات الجراحية، ومتابعة مسار العلاج.

الواقع المعزز: دمج العالم الرقمي مع العالم الحقيقي لخلق تجارب تفاعلية غامرة.

مجال الرؤية يتطور بوتيرة متسارعة ف مستقبلاً قد يظهر تقنيات جديدة تتيح تطبيقات أكثر ذكاء وكفاءة.

1-1 - لمحة عن سرطان الرئة :

يحدث سرطان الرئة عندما تطرأ تغيرات على الحمض النووي (DNA) للخلايا الموجودة في الرئتين. ويحمل الحمض النووي للخلية التعليمات التي تخبرها بما يجب أن تفعله. في الخلايا السليمة، يعطي الحمض النووي تعليمات للنمو والتكاثر بمعدل محدد، كما يعطي التعليمات للخلايا بالموت في وقت محدد. أما في الخلايا السرطانية، فإن التغيرات التي تطرأ على الحمض النووي تعطي تعليمات مختلفة. تحدث هذه التغيرات الخلايا السرطانية على إنتاج المزيد من الخلايا بسرعة كبيرة. ويمكن للخلايا السرطانية أن تستمر في العيش بينما تموت الخلايا السليمة. وهذا يؤدي إلى تراكم الكثير من الخلايا.

قد تشكل الخلايا السرطانية كتلة تسمى الورم. ويمكن أن ينمو الورم ليغزو الأنسجة السليمة في الجسم ويتلفها. ومع مرور الوقت، يمكن أن تتفصل الخلايا السرطانية وتنتشر إلى أجزاء أخرى من الجسم. وعندما ينتشر السرطان، يسمى سرطاناً نقائلياً (سرطان خبيث ينتشر من مكان نشأته إلى أماكن أخرى بعيدة عنه).

النوعان الرئيسيان لسرطان الرئة هما: سرطان الرئة ذو الخلايا غير الصغيرة (NSCLC) وسرطان الرئة ذو الخلايا الصغيرة (SCLC).

حوالي 80% إلى 85% من حالات سرطان الرئة هي من نوع NSCLC.

هناك أنواع فرعية لسرطان الرئة ذو الخلايا غير الصغيرة نذكر منها :

السرطان الغدي (Adenocarcinoma).

سرطان الخلايا الحرشفية (Squamous Cell Carcinoma).

أورام الغدد الصماء العصبية (Neuroendocrine tumors). [1]

1-2- الدراسات المرجعية:

العديد من الدراسات قامت على الصور الشعاعية للرئتين وتشخيص الأمراض .

أولاً :

قامت journal of soft computing exploration بنشر مقال عام 2023 بعنوان :

Lung Cancer Classification Using Convolutional Neural Network and Dense Net

في هذا البحث القائم على التعلم العميق تم اقتراح اساليب الشبكة العصبية التلافيفية وشبكة Dense Net (التي تعتمد على فكرة الاتصال الكثيف بين الطبقات أي تتصل كل طبقة في هذه الشبكة بكل الطبقات التي تأتي قبلها مما يزيد من كفاءة التعلم ويسمح بتدفق موثوق ومتربط للمعلومات) للمساعدة في اكتشاف سرطان الرئة وقد أثبتت فعاليتها من خلال دقة تبلغ % 97.48 .

تم استخدام معايير منها ال BATCH_SIZE لإدارة كمية البيانات التي سيتم إدراجها في النموذج وال IMAGE_SIZE لتحديد حجم كل منها وال EPOCHS لمعرفة عدد التكرارات التي يجب تنفيذها من أجل تحقيق نتائج دقيقة , تم استخدام 2073 صورة تم تصنيفها الى ثلاث مجموعات: حالات عادية وحميدة وخبيثة , تم تغيير حجم الصورة (Rescaling) ليصبح في القيمة الدنيا 0 وأعلى قيمة 1 . وتم تقسيم البيانات بشكل عشوائي لتسهيل معالجة النموذج .

في هذا التدريب تم استخدام بنيات CNN و Dense Net للحصول على نتائج أفضل من الأبحاث السابقة نستخدم عدة طبقات في بنية CNN بما في ذلك Conv2D مع فلتر 32 وحجم (3*3) و MAXPOOLING بحجم (2*2) وتم ايضاً إضافة 5 طبقات Conv2D تحتوي 64 فلتر وطبقة مسطحة كثيفة تحتوي 128 فلتر وعملياتي تنشيط RELU و SOFTMAX .

بعدها يتم تحويل الإخراج من الطبقات السابقة باستخدام الطبقة المسطحة الى متجه أحادي الابعاد يمكن استخدامه كطبقة إدخال كثيفة. [2]

ثانياً :

الدورة الدولية للتقدم في مجال الذكاء الصناعي أصدرت مقال عام 2021 بعنوان :

Lung cancer medical images classification using hybrid cnn-svm.

يقدم هذا البحث طريقة هجينة تجمع بين الشبكات العصبية التلافيفية ودعم المتجهات لدراسة تصنيف صور الرئة المقطعية إلى سرطان الغدد الدرقية، وسرطان الخلايا الكبيرة، أو طبيعية، أو سرطان الخلايا الحشوية. كان الهدف هو تحقيق مستوى أعلى من الدقة، وهو الهدف الأساسي لأي نظام للكشف بمساعدة الحاسوب. تم تطبيق الطريقة على مجموعة بيانات صور الرئة المقطعية، وهي مجموعة قياسية ومتاحة للجمهور من صور مقطعية. تم استخدام ما مجموعه 5103 صورة لاختبار الطريقة، وتم تحقيق معدل دقة تصنيف بنسبة 96.71% وهذا يدل على تفوق الطريقة الهجينة المقترحة القائمة على الشبكات العصبية التلافيفية ودعم المتجهات وقدراتها في التطبيقات. علاوة على ذلك، فإن الطريقة المقترحة توفر أيضًا الحساسية، والتخصص، والدقة، ومنحنى التشغيل المميز بنسبة 97.90%، و 99.32%، و 97.96%، و 1.000 على التوالي. بالنسبة للأعمال المستقبلية، يهدف الباحثون إلى اختبار الطريقة مع مجموعات بيانات مختلفة. يمكن تحسين مستوى الدقة من خلال زيادة عدد الصور المستخدمة في الإجراء. بالإضافة إلى ذلك، يمكن تفسير صور الأشعة السينية والأشعة المقطعية بالإصدار البوزيتروني باستخدام هذه الطريقة. يجب أن يكون من الممكن إجراء فحص لجميع هذه الصور. من خلال دراسة وتحليل نتائج التنبؤ لأنواع مختلفة من الصور، سيكون بإمكان الطاقم الطبي استخدام الصور الأنسب لاكتشاف الأورام الرئوية.[3]

ثالثاً:

قامت Scientific Reports بنشر مقال بعنوان :

Deep learning classification of lung cancer histology using CT images.

في هذه الدراسة، تم استخدام تقنية "الراديو ميكس" (تحليل البيانات الكمية من الصور الطبية) للتنبؤ بنوع ورم الرئة غير ذو الخلايا الصغيرة (NSCLC) من خلال صور التصوير المقطعي المحوسب (CT) الروتينية غير الغازية. قمنا بتدريب واختبار شبكات عصبية تلافيفية (CNNs) على مجموعة بيانات تضم 311 مريضاً في المراحل المبكرة من سرطان الرئة غير ذو الخلايا الصغيرة خضعوا لجراحة في مستشفى ماساتشوستس العام (MGH)، مع التركيز على نوعي الورم الأكثر شيوعاً: الغدي (ADC) والخلايا الحرشفية (SCC). تمكنت الشبكات العصبية التلافيفية من التنبؤ بنوع الورم بدقة بلغت 0.89 ($p = 0.018$). كما وجدنا أن استخدام تقنيات التعلم الآلي مثل "أقرب الجيران" (KNN) و"آلة الدعم المتجه" (SVM) على الميزات الكمية المستخرجة من الشبكات العصبية التلافيفية حققت أداءً تمييزياً مماثلاً، حيث بلغت الدقة 0.89 ($p = 0.017$). عملت أفضل شبكة عصبية تلافيفية لدينا كتصنيف احتمالي قوي في مجموعات الاختبار المختلفة، مع توضيحات مرئية قابلة للتفسير لتوقعاتها. يمكن للراديو ميكس القائم على التعلم العميق تحديد الأنواع المختلفة لورم الرئة. ويمكن أن يدعم الطرق الحالية ويعمل كأداة مساعدة للأطباء التشخيصيين. [4]

1-3- تعلم الآلة (Machine Learning):

تعلم الآلة هو فرع من فروع الذكاء الاصطناعي يركز على تطوير خوارزميات ونماذج تمكن الأنظمة من التعلم من البيانات وتحسين أدائها بمرور الوقت دون الحاجة إلى برمجة صريحة. يعتمد تعلم الآلة على فكرة أن الأنظمة يمكن أن تتعلم من التجارب السابقة وتستخدم هذه المعرفة لاتخاذ قرارات أو إجراء تنبؤات.

1-3-1 أنواع تعلم الآلة:

1. التعلم الخاضع للإشراف (Supervised Learning):

- في هذا النوع، يتم تدريب النموذج على مجموعة بيانات تحتوي على أمثلة مدخلة (Features) ومخرجات معروفة (Labels).

- الهدف هو تعلم العلاقة بين المدخلات والمخرجات حتى يتمكن النموذج من التنبؤ بالمخرجات الجديدة.

2. التعلم غير الخاضع للإشراف (Unsupervised Learning):

- هنا، يتم تدريب النموذج على مجموعة بيانات لا تحتوي على مخرجات معروفة.

- الهدف هو اكتشاف الأنماط أو الهياكل في البيانات.

3. التعلم المعزز (Reinforcement Learning):

- يعتمد هذا النوع على مفهوم المكافآت والعقوبات.

- يتعلم النموذج كيفية اتخاذ القرارات عن طريق التجربة والخطأ، حيث يحصل على مكافآت عند اتخاذ قرارات صحيحة وعقوبات عند اتخاذ قرارات خاطئة.

- أمثلة: الألعاب (مثل شطرنج أو Go)، الروبوتات.

1-3-2 خطوات عملية تعلم الآلة :

1. جمع البيانات: جمع بيانات كافية ومتنوعة لتدريب النموذج.
 2. تحضير البيانات: تنظيف البيانات ومعالجتها، مثل إزالة القيم المفقودة أو تحويل البيانات إلى تنسيق مناسب.
 3. اختيار النموذج: اختيار الخوارزمية المناسبة بناءً على نوع المشكلة والبيانات المتاحة.
 4. تدريب النموذج: استخدام مجموعة التدريب لتعليم النموذج كيفية التنبؤ بالمخرجات.
 5. تقييم النموذج: اختبار النموذج باستخدام مجموعة اختبار لتقييم أدائه وقياس دقته.
 6. تحسين النموذج: ضبط المعلمات أو استخدام تقنيات أخرى لتحسين الأداء.
 7. نشر النموذج: استخدام النموذج المدرب في التطبيقات العملية.
- تعلم الآلة هو مجال متطور ومؤثر في التكنولوجيا الحديثة، حيث يلعب دورًا رئيسيًا في العديد من التطبيقات التي نستخدمها يوميًا. يساعد هذا المجال في تحسين الكفاءة، اتخاذ القرارات، وتقديم تجارب شخصية للمستخدمين. [5]

1-4- التعلم العميق:

التعلم العميق Deep Learning هو أحد فروع تعلم الآلة Machine Learning والذي يهتم بشكل أساسي بتطوير خوارزميات تمكّن الحاسب الآلي من تعلم أداء المهام الصعبة التي تتطلب فهماً عميقاً للبيانات وطبيعة عملها .

ويعتمد بشكل أساسي على الشبكات العصبية الاصطناعية ANN .

الشكل التالي يوضح ارتباط التعلم العميق بتعلم الآلة والذكاء الاصطناعي :



الشكل 1 - 1 ارتباط التعلم العميق بتعلم الآلة والذكاء الاصطناعي.

تعتمد أنظمة التعلم العميق على شبكات عصبية اصطناعية مُتعددة الطبقات، تُسمى شبكات عصبية عميقة. تتكون هذه الشبكات من طبقات مترابطة من "العقد" تُشبه الخلايا العصبية، حيث تُعالج البيانات وتُحلَّل من خلالها.

1_4_1 مميزات التعلم العميق:

التعلم من البيانات الضخمة: يُمكن للتعلم العميق استيعاب كميات هائلة من البيانات المُعقدة، مثل الصور والفيديوهات والنصوص، واستخراج المعرفة منها دون الحاجة إلى برمجة محددة.

الدقة العالية: أثبتت أنظمة التعلم العميق قدرتها على تحقيق نتائج عالية الدقة في مجالات متنوعة، مثل التعرف على الصور والترجمة الآلية ومعالجة اللغة الطبيعية.

التعلم المستمر: تتميز أنظمة التعلم العميق بقدرتها على التعلم والتطور بمرور الوقت، مع تحسين أدائها باستمرار من خلال التعرض لبيانات جديدة.

وله عدة تطبيقات :

الرؤية الحاسوبية: التعرف على الوجه، وتصنيف الصور، والكشف عن الأجسام، والقيادة الذاتية.

معالجة اللغة الطبيعية: الترجمة الآلية، وتلخيص النصوص، والرد على الأسئلة، وإنشاء النصوص.

التعرف على الصوت: التعرف على المتحدث، والتحكم الصوتي، وترجمة الكلام.

الروبوتات: التحكم في الروبوتات، والتنقل الذكي، والتفاعل مع البيئة.

الطب: تشخيص الأمراض، وتحليل الصور الطبية، واكتشاف الأدوية.

يُتوقع أن يشهد التعلم العميق ثورة هائلة في مختلف مجالات الحياة، من خلال تحسين قدرات الذكاء الاصطناعي وتوسيع نطاق تطبيقاته.

1-4-2 معمارية التعلم العميق:

تتشكل خوارزميات التعلم العميق من مجموعة من العمليات وما يميز معمارية هذه الخوارزميات هو أنها مستوحاة من الشبكات العصبية في دماغ الإنسان وعليه يطلق على معماريات التعلم العميق مسمى الشبكات العصبية الاصطناعية أو الشبكات العصبونية .

فهي تشير إلى الهيكل والتنظيم الخاص بالشبكات العصبية الاصطناعية التي تُستخدم في تطبيقات التعلم العميق. هذه الممارية تتضمن عدة مكونات رئيسية، وكل منها يلعب دورًا مهمًا في عملية التعلم .

المكونات :

• الطبقات (Layers):

تتكون الشبكات العصبية من مجموعة من الطبقات، وكل طبقة تحتوي على عدد معين من الخلايا العصبية (Neurons). هناك ثلاثة أنواع رئيسية من الطبقات:

- الطبقة المدخلة (Input Layer):

هي الطبقة الأولى في الشبكة، حيث يتم إدخال البيانات إليها.

- الطبقات المخفية (Hidden Layers):

هذه الطبقات تقع بين الطبقة المدخلة وطبقة المخرجات. يمكن أن تحتوي الشبكة على طبقة واحدة أو أكثر من الطبقات المخفية. كل طبقة مخفية تتكون من خلايا عصبية تقوم بمعالجة البيانات المستلمة من الطبقة السابقة وتطبيق دوال التنشيط.

- الطبقة المخرجة (Output Layer):

هي الطبقة النهائية التي تُعطي النتائج. عدد الخلايا العصبية في هذه الطبقة يعتمد على عدد الفئات أو القيم المتوقعة.

- الأوزان (Weights) والانحياز (Biases) :

الأوزان: كل اتصال بين خلايا الأعصاب له وزن، وهو يمثل مدى تأثير الخلية العصبية السابقة على الخلية

العصبية التالية. يتم تحديث هذه الأوزان خلال عملية التدريب لتحسين أداء النموذج.

الانحياز: كل خلية عصبية تحتوي أيضًا على قيمة انحياز تضيف إلى ناتج العملية الحسابية. يساعد

الانحياز في تحسين قدرة النموذج على التعلم.

- دوال التنشيط (Activation Functions) :

تستخدم دوال التنشيط لتحديد ما إذا كانت الخلية العصبية ستُفعّل أم لا.

- التدريب (Training):

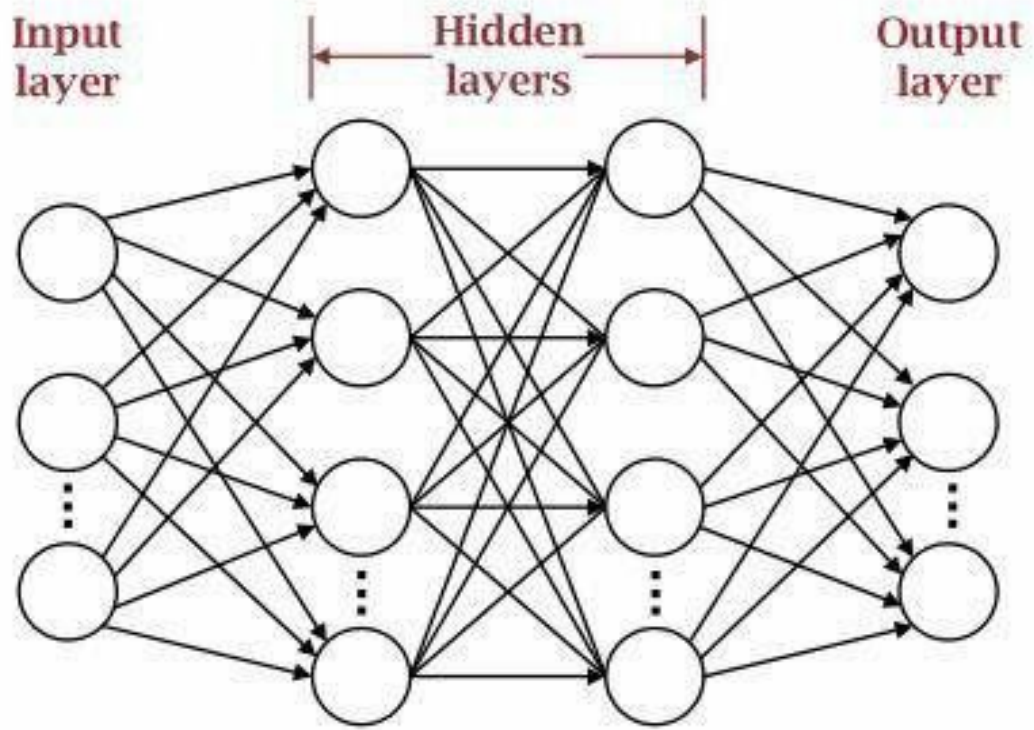
عملية التدريب تتضمن:

- الانتشار الأمامي (Forward Propagation)

- حساب الخطأ (Loss Calculation)

- الانتشار العكسي (Backpropagation). [6]

الشكل التالي يوضح معمارية التعلم العميق ذات طبقتين مخفيتين :



الشكل 1 - 2 معمارية التعلم العميق .

1-5- البيئة والأدوات المستخدمة:

استخدمنا في هذا المشروع للتدريب لغة بايثون

مع العديد من المكتبات منها Keras , TensorFlow , NumPy , Matplotlib .

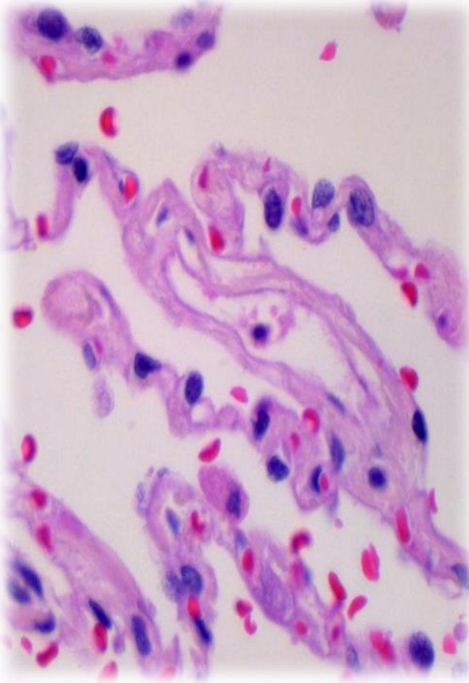
كما أن البيانات عبارة عن 12000 صورة شعاعية مقسمة الى :

Train , Test , Validation .

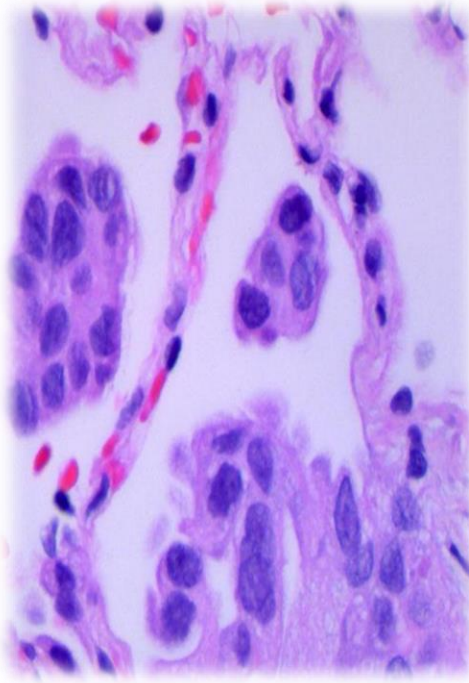
كما حصلنا على الداتا من : [7]

هذه الصور تشير إلى أنواع سرطان الرئة .

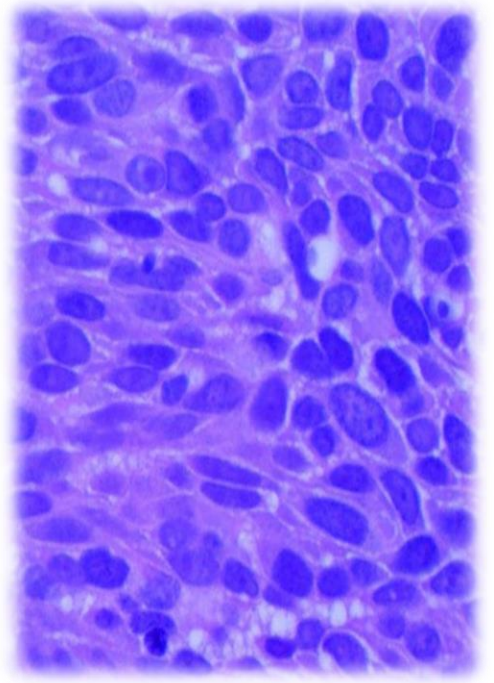
يوضح هذا الشكل صور لرئة مصابة بالسرطان :



Neuroendocrine



Adenocarcinoma



**Squamous cell
carcinoma**

الشكل 1 - 3 صور أشعة سينية .

1-6- منهجية التطور:

سنقدم في هذا البحث شرح كاف عن عملية الشبكات العصبونية الالتفافية وتطبيقها عمليا على بيانات معتمدة ومقارنة النتائج, كما سيتم الشرح المفصل عن أليات عمل طبقات الشبكة العصبونية الالتفافية وامثلة عن كل منها.

سوف نستخدم الدقة (Accuracy) وهي مقياس أساسي لتقييم أداء النموذج في التعلم الآلي. تحدد الدقة النسبة المئوية للتنبؤات الصحيحة التي قام بها النموذج مقارنة بجميع التنبؤات التي قام بها. بعبارة أخرى، إنها تعبر عن مدى قدرة النموذج على تصنيف أو التنبؤ بالقيم الصحيحة بشكل صحيح.

2-1- التعامل مع البيانات والشبكة العصبية الالتفافية:

زيادة البيانات (data augmentation)

سنقوم بتطبيق مفهوم زيادة البيانات وهو عبارة عن تطبيق بعض العمليات على الصور حيث يتم تغيير أبعاد أو شكل الصورة من دون التغيير على محتوى الصورة, نذكر في ما يلي بعض تقنيات زيادة البيانات:

1- التحجيم (resize & rescale):

Resize: هو تغيير أبعاد الصورة (العرض والارتفاع) مع الحفاظ على المحتوى الأصلي, الهدف منها توحيد أبعاد الصور في مجموعة البيانات لتناسب متطلبات النموذج (حجم المدخلات في الشبكة).
Rescale: هو عملية تغيير قيم البكسلات في الصورة ضمن نطاق معين, الهدف تحسين توزيع البيانات وتسريع عملية التعلم.

2 - الدوران (rotation) :

يجب على الشبكة أن تدرك السمات المرادة في الصورة في أي اتجاه معين , أي السمة التي ستظهر لنا عن نوع السرطان في الصورة هل ستبقى ذاتها اذا تم تدوير الصورة في زاوية معينة , أي نستطيع تطبيق التدوير على الصورة وتوليد صور جديدة نضيفها الى الداتا هذا يؤدي الى تحسين الدقة وتقليل الخطأ ومنع ال (overfitting) .

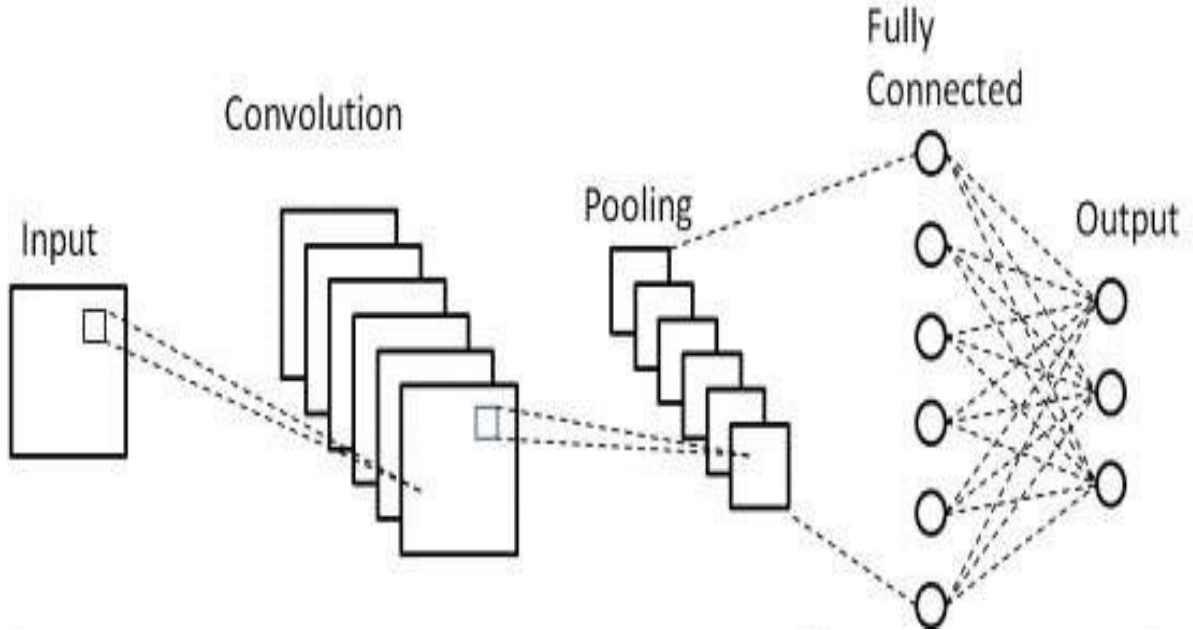
2-2- الشبكة العصبية الالتفافية:

نوع من الشبكات العصبية الاصطناعية وهي أحد تقنيات التعلم العميق , تستخدم بشكل رئيسي في معالجة الصور والفيديو تم تصميمها لتحليل واستخراج المعلومات بشكل فعال وهي تستخدم في تقنيات عديدة مثل تصنيف الصور والكشف عن الأشياء وغيرها , تتميز بوجود طبقة خفية أو أكثر وتتكون من عدة طبقات متتالية .

مواصفات الدخل:

إن الصورة بالنسبة للحاسوب عبارة عن مصفوفة ثلاثية الأبعاد (عرض*ارتفاع*عمق) من قيم تتراوح بين 0-255 .

الشكل التالي يوضح طبقات الشبكة العصبية الالتفافية



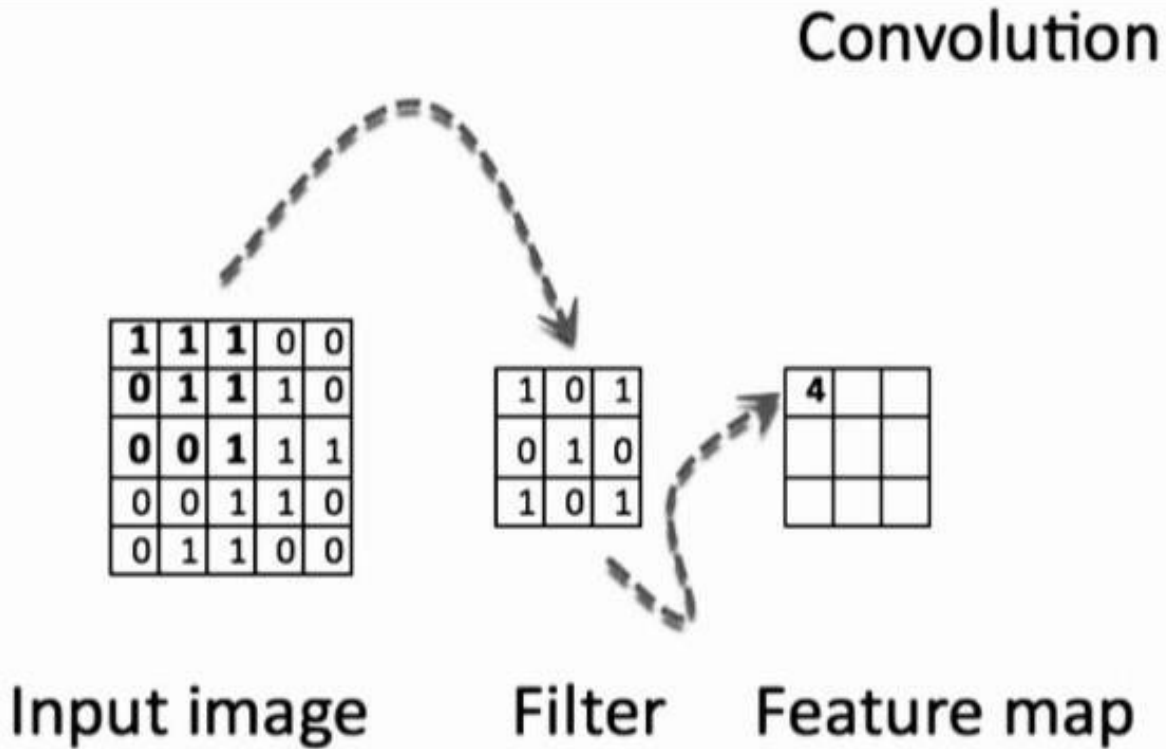
الشكل 2 - 1 طبقات الشبكة العصبية الالتفافية.

2-2-1- طبقات الشبكة العصبية الالتفافية :

1_ الطبقة الالتفافية (convolutional layer):

تقوم هذه الطبقة بعملية تحويل على الصورة لاستخراج الميزات المهمة أو تحديد أنماط معينة منها من خلال تطبيق الفلاتر والفلتر هو عبارة عن مصفوفة ثنائية الأبعاد يطبق على الصورة بشكل متكرر من خلال عمليات حسابية مناسبة لإنتاج قيمة جديدة .. يطبق هذا الفلتر لاكتشاف أنماط محددة في الصورة مثل الحواف , الزوايا, الحدة وغيرها .

يوضح الشكل التالي عملية تطبيق الفلتر على بكسلات الصورة :



الشكل 2 - 2 تطبيق فلتر على الصورة .

لدينا مفهوم يدعى stride يستخدم لتحديد كيفية تحريك الفلاتر عبر الصورة أثناء عملية التحويل وهو يمثل المسافة التي يتم تحريك الفلتر بها في كل خطوة .. له تأثيرات على عملية ال conv وهي :

يؤدي الى تقليل عدد العمليات الحسابية التي يجب إجراؤها .

فقدان المعلومات بحيث يمكن أن يؤدي تحرك الفلتر مقدار كبير الى تخطي بعض المناطق المهمة .

أيضاً لدينا مفهوم ال padding يستخدم لإضافة صفوف وأعمدة ذات قيم صفرية حول الحواف الخارجية قبل تطبيق الفلتر عليها هذا يساعد في الحفاظ على المعلومات في الحواف والحفاظ على دقة البيانات وأيضاً يساهم في تحسين أداء الشبكة CNN .

له أنواع مختلفة :

الأول لا يتم إضافة شيء يسمى valid padding .

الثاني يتم إضافة سطور وأعمدة قيمها صفرية بحيث يكون حجم الإخراج متساوياً تقريباً مع حجم الصورة الأصلية ويسمى same padding .

2- طبقة التجميع (pooling) :

الهدف من عملية ال pooling هو تقليل حجم البيانات وتقليل التكلفة الحسابية وزيادة سرعة التدريب ودقة

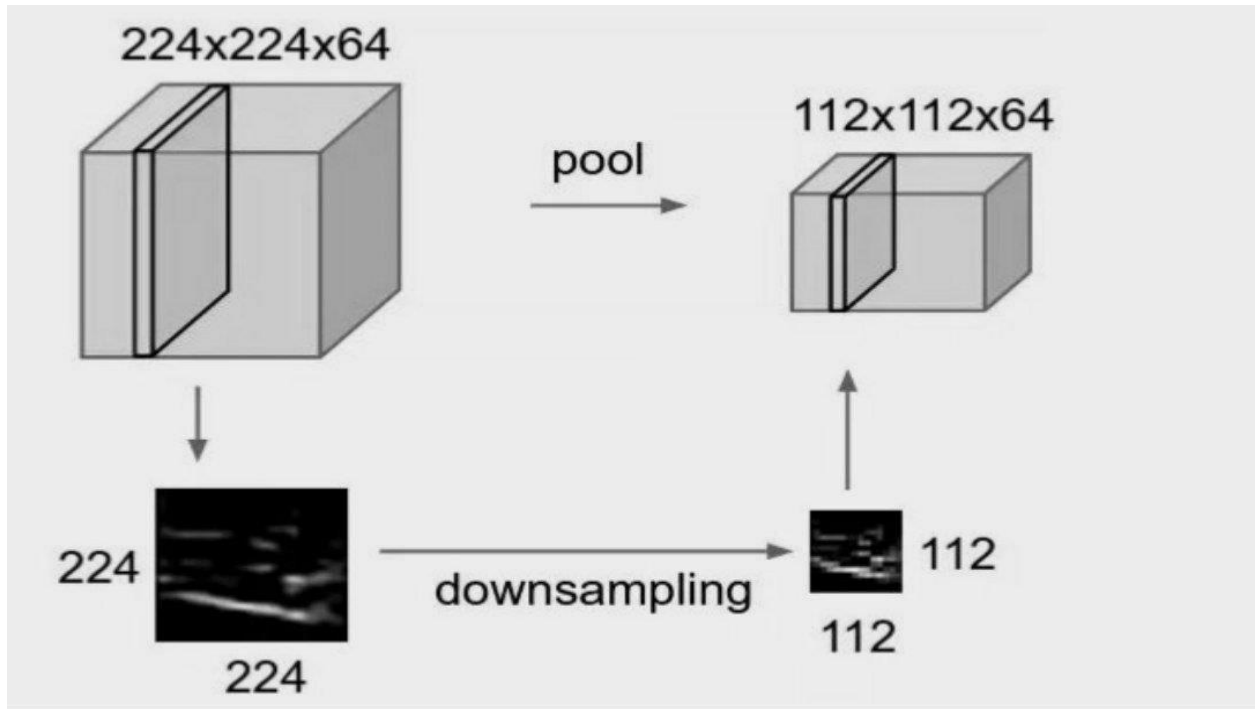
التصنيف ولها نوعان :

- الأول: يقوم بتقليل حجم البيانات بحيث يختار القيمة الكبرى من مجموعة معينة من البيانات MAX . POOLING

- الثاني: يقوم بحساب المتوسط الحسابي لمجموعة البيانات . POOLING AVERAGE

أيضاً تساهم في تحسين قدرة الشبكة على التعامل مع ال overfitting .

يوضح الشكل التالي عملية ال Pooling :



الشكل 2 - 3 تطبيق عملية التجميع.

3_ الطبقة كاملة الارتباط (fully connected layer) :

تعمل هذه الطبقة على دمج الخصائص المستخرجة من الطبقات السابقة لتحديد النتائج النهائية وكل عصبون في هذه الطبقة يتصل بجميع العصبونات في الطبقة السابقة وعند استخدام هذه الطبقة يجب تحويل البيانات الى vector ليتمكن من التصنيف او التوقع .

ويمكن أن تكون مكلفة حسابياً بشكل كبير خاصة عند استخدام شبكات عميقة لذلك يتم استخدام تقنية ال dropout عندما تكون النماذج معقدة وتحتوي على عدد كبير من الطبقات فعندها النموذج يتعلم تفاصيل صغيرة جداً من البيانات التدريبية ويفشل في التعميم على بيانات جديدة . فهذه التقنية تعمل على تجاهل عشوائي لبعض الوحدات في الطبقات الخفية اثناء عملية التدريب .

بشكل اساسي يتم اختيار وحدات عشوائية ليتم اسقاطها خلال كل دورة تدريب مما يجبر النموذج على تعلم ميزات مستقلة بشكل افضل هذا يساعد في تقليل ال overfitting في التعلم وزيادة قدرة النموذج على التعميم على بيانات جديدة .

4_ طبقة الخرج (output layer) :

هي الطبقة النهائية التي تُستخدم لإنتاج المخرجات النهائية للنموذج. تعتمد هذه الطبقة على المعلومات المستخلصة من الطبقات السابقة , الوظائف الأساسية لطبقة الخرج:

1. تحديد المخرجات: تعتمد طبقة الخرج على عدد الفئات أو القيم التي نريد التنبؤ بها .
2. استخدام دالة تفعيل مناسبة.
3. تجميع المعلومات: تقوم طبقة الخرج بتجميع المعلومات المستخرجة من الطبقات السابقة وتحويلها إلى تنسيق يمكن أن يُستخدم للإجابة على السؤال المطروح (مثل تصنيف الصورة أو تحديد الكائنات).
4. التدريب والتحديث: خلال عملية التدريب، يتم تحديث أوزان طبقة الخرج بناءً على الخطأ الناتج عن التنبؤات باستخدام خوارزمية مثل "Backpropagation" . [8]

3-1- استيراد المكتبات:

في البداية سنقوم باستيراد المكتبات المطلوبة وهي:

1. TensorFlow لبناء المودل وإضافة الطبقات عليه والقيام بعملية التدريب.
2. matplotlib.pyplot لعرض النتائج على شكل جداول ومخططات.
3. os للتعامل مع النظام والملفات.
4. NumPy للتعامل مع المصفوفات.

3-2- استيراد الداتا:

سنقوم باستيراد الداتا الموجودة لدينا على غوغل درايف والتي ستكون مقسمة لـ:

- Train set
- Test set.
- Validation set.

3-3- قراءة الداتا:

سنقوم بتعريف التابع `get_ds_size` والذي دخله عبارة عن مجلد ويعيد هذا التابع اسماء المجلدات الموجودة داخله وعدد العناصر في كل مجلد.

حيث سيتم تنفيذ التابع `get_ds_size` على كل من المجلدات :

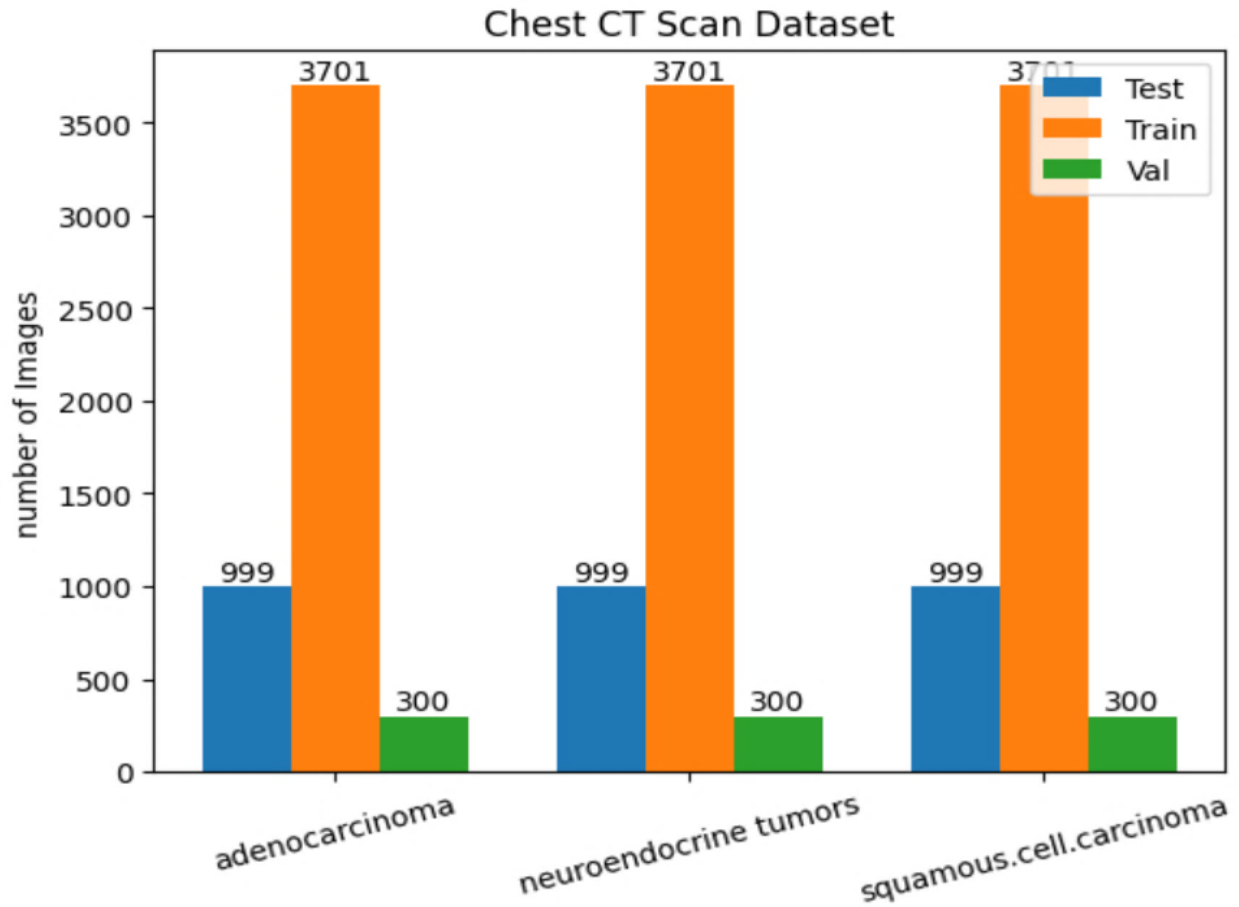
- `train_folder`: الذي يحتوي على الـ `train_set`.
- `test_folder`: الذي يحتوي على الـ `test_set`.
- `validation_folder`: الذي يحتوي على الـ `validation set`.

3-4- تحليل الداتا:

وسيتم ذلك عن طريق مكتبة matplotlib.pyplot حيث سيكون الخرج عبارة عن:

مخطط بياني يمثل عدد عناصر الداتا في كل مجلد مقسمة إلى ثلاثة أقسام حيث كل قسم يكون عبارة عن أنواع أمراض السرطان التي سيقوم البرنامج بتصنيفها وهي:

1. adenocarcinoma: وهو السرطان الغدي.
2. neuroendocrine: وهو سرطان الاورام العصبية الصماوية.
3. squamous cell carcinoma: سرطان الخلايا الحرشفية.



الشكل 3 - 1 عدد عناصر الداتا

3-5- تجهيز الداتا للتدريب:

في البداية سنقوم باستخدام تابع من مكتبة TensorFlow يقوم بتنسيق الداتا بشكل مناسب لتدريب النماذج العصبية حيث يحوي هذا التابع على مجموعة من البارمتارات وهي:

1. المجلدات التي تحتوي على الصور.
2. shuffle=true: حيث يأخذ الصور بشكل عشوائي لتجنب ال over fitting.
3. image size: يقوم بتحويل الصور إلى الحجم المحدد قبل تنسيقها.
4. batch_size: حجم الدفعة فيأخذ الصور على شكل دفعات بدلا من واحد تلوم الاخرى.

ثم بعدها سنقوم بعرض عينة عشوائية من الداتا باستخدام مكتبة matplotlib.pyplot.

سنقوم بتخزين مجموعة البيانات في الذاكرة لتسريع عملية الوصول إليها باستخدام تابع cache.

كما نقوم بعملية shuffle لتقديم عينات عشوائية من البيانات.

ونستخدم التعليمة prefetch حيث تقوم هذه التعليمة بتحميل البيانات التالية في الخلفية مما يقلل من وقت الانتظار ويزيد من كفاءة التدريب.

3-6- بناء الشبكة العصبية:

1. بناء طبقتي resize و rescale:

- حيث تقوم طبقة resize بتغيير حجم الصور إلى قيمة محددة.
- اما طبقة rescale تقوم هذه الطبقة بجعل قيم كل بكسل في الصورة بين 0 و 1 وذلك من اجل تحسين أداء النموذج.

2. بناء طبقتي Random Flip و Random Rotation:

- حيث تقوم طبقة Random Flip بتطبيق انقلاب افقي وعمودي على الصور بشكل عشوائي مما يساعد في زيادة تنوع البيانات وتقليل فرص الـ overfitting.
- كما تقوم طبقة Random Rotation بتدوير الصور بشكل عشوائي بزاوية محددة مع او عكس عقارب الساعة مما يساعد النموذج على تعلم ميزات أكثر عمومية.

3. بناء المودل:

ستكون أول اربع طبقات في المودل هي طبقات:

resize, rescale, Random Flip, Random Rotation.

ثم سنقوم ببناء 8 طبقات حيث تكون عبارة عن:

- اربع طبقات Conv2D حيث كل طبقة تتكون من:
 - i. عدد العصبونات في كل طبقة.
 - ii. حجم الفلتر وهو (3,3).
 - iii. تابع التنشيط المستخدم وهو تابع الـ relu.
- وأربع طبقات MaxPooling2D حيث كل طبقة تحتوي على حجم الفلتر وهو (2,2) وتكون طبقات الـ Conv2D و الـ MaxPooling2D متناوبة فيما بينها بداية بالـ Conv2D.
- طبقة Flatten حيث تقوم هذه الطبقة بتحويل الصور إلى متجه طويل.

- طبقتي Dense حيث تتكون الطبقة الاولى من :

i. 64 عصبون.

ii. تابع التفعيل .relu.

- أما الطبقة الثانية تتكون من:

i. 3 عصبونات وهي عدد الصفوف التي سنصف الداتا لها.

ii. تابع التفعيل softmax.

Model: "sequential_2"

Layer (type)	Output Shape	Param #
sequential (Sequential)	(32, 256, 256, 3)	0
sequential_1 (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)	18,496
max_pooling2d_1 (MaxPooling2D)	(32, 62, 62, 64)	0
conv2d_2 (Conv2D)	(32, 60, 60, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(32, 30, 30, 128)	0
conv2d_3 (Conv2D)	(32, 28, 28, 256)	295,168
max_pooling2d_3 (MaxPooling2D)	(32, 14, 14, 256)	0
flatten (Flatten)	(32, 50176)	0
dense (Dense)	(32, 64)	3,211,328
dense_1 (Dense)	(32, 3)	195

Total params: 3,599,939 (13.73 MB)

Trainable params: 3,599,939 (13.73 MB)

Non-trainable params: 0 (0.00 B)

الشكل 3 - 2 الشبكة العصبية

3_7_ تدريب الداتا:

في البداية سنقوم بإعداد النموذج لعملية التدريب باستخدام التابع compile حيث يتكون هذا التابع من:

- i. optimizer: وهي الخوارزمية التحسين التي يستعمل بها تحديث الأوزان وهي adam.
- ii. loss: يحدد دالة الخسارة وهو مدى اختلاف النتائج المتوقعة عن الفعلية.
- iii. metrics: يحدد مقياس الاداء المستخدم لتقييم النموذج.

ثم نقوم بتدريب الداتا عن طريق التعليمة fit التي تتضمن:

- i. داتا التدريب.
- ii. epochs: يحدد عدد مرات تدريب الداتا.
- iii. batch_size: يحدد حجم الدفعة التي سيتم معالجتها في كل خطوة.
- iv. verbose: لتحديد مستوى التفاصيل التي سيتم عرضها أثناء التدريب.
- v. validation data: داتا التحقق التي سوف تستخدم لتقييم الأداء.

وأخيرا نقوم بحفظ المودل بملف خارجي لاعادة استعادته لتنفيذ العمليات عليه دون التدريب من جديد.

```
Epoch 1/20
347/347 [=====] - ETA: 0s - loss: 0.3679 - accuracy: 0.8404
Epoch 1: accuracy improved from -inf to 0.84040, saving model to //content//drive//MyDrive//models/model-01-0.88.keras
347/347 [=====] - 272s 604ms/step - loss: 0.3679 - accuracy: 0.8404 - val_loss: 0.3240 - val_accuracy: 0.8767
Epoch 2/20
347/347 [=====] - ETA: 0s - loss: 0.2376 - accuracy: 0.9045
Epoch 2: accuracy improved from 0.84040 to 0.90453, saving model to //content//drive//MyDrive//models/model-02-0.92.keras
347/347 [=====] - 199s 574ms/step - loss: 0.2376 - accuracy: 0.9045 - val_loss: 0.2019 - val_accuracy: 0.9200
Epoch 3/20
347/347 [=====] - ETA: 0s - loss: 0.1972 - accuracy: 0.9260
Epoch 3: accuracy improved from 0.90453 to 0.92597, saving model to //content//drive//MyDrive//models/model-03-0.93.keras
347/347 [=====] - 200s 575ms/step - loss: 0.1972 - accuracy: 0.9260 - val_loss: 0.1800 - val_accuracy: 0.9333
Epoch 4/20
347/347 [=====] - ETA: 0s - loss: 0.1761 - accuracy: 0.9300
Epoch 4: accuracy improved from 0.92597 to 0.93002, saving model to //content//drive//MyDrive//models/model-04-0.94.keras
347/347 [=====] - 199s 575ms/step - loss: 0.1761 - accuracy: 0.9300 - val_loss: 0.1543 - val_accuracy: 0.9356
Epoch 5/20
347/347 [=====] - ETA: 0s - loss: 0.1713 - accuracy: 0.9330
Epoch 5: accuracy improved from 0.93002 to 0.93299, saving model to //content//drive//MyDrive//models/model-05-0.92.keras
347/347 [=====] - 200s 575ms/step - loss: 0.1713 - accuracy: 0.9330 - val_loss: 0.1858 - val_accuracy: 0.9244
Epoch 6/20
347/347 [=====] - ETA: 0s - loss: 0.1590 - accuracy: 0.9381
Epoch 6: accuracy improved from 0.93299 to 0.93812, saving model to //content//drive//MyDrive//models/model-06-0.94.keras
347/347 [=====] - 199s 575ms/step - loss: 0.1590 - accuracy: 0.9381 - val_loss: 0.1554 - val_accuracy: 0.9389
Epoch 7/20
347/347 [=====] - ETA: 0s - loss: 0.1503 - accuracy: 0.9367
Epoch 7: accuracy did not improve from 0.93812
347/347 [=====] - 198s 571ms/step - loss: 0.1503 - accuracy: 0.9367 - val_loss: 0.1610 - val_accuracy: 0.9356
Epoch 8/20
347/347 [=====] - ETA: 0s - loss: 0.1317 - accuracy: 0.9456
Epoch 8: accuracy improved from 0.93812 to 0.94560, saving model to //content//drive//MyDrive//models/model-08-0.95.keras
347/347 [=====] - 200s 577ms/step - loss: 0.1317 - accuracy: 0.9456 - val_loss: 0.1177 - val_accuracy: 0.9511
Epoch 9/20
347/347 [=====] - ETA: 0s - loss: 0.1262 - accuracy: 0.9471
Epoch 9: accuracy improved from 0.94560 to 0.94713, saving model to //content//drive//MyDrive//models/model-09-0.96.keras
347/347 [=====] - 199s 572ms/step - loss: 0.1262 - accuracy: 0.9471 - val_loss: 0.1427 - val_accuracy: 0.9411
Epoch 10/20
347/347 [=====] - ETA: 0s - loss: 0.1158 - accuracy: 0.9519
Epoch 10: accuracy improved from 0.94713 to 0.95190, saving model to //content//drive//MyDrive//models/model-10-0.96.keras
347/347 [=====] - 198s 571ms/step - loss: 0.1158 - accuracy: 0.9519 - val_loss: 0.0999 - val_accuracy: 0.9556

Epoch 11/20
347/347 [=====] - ETA: 0s - loss: 0.1138 - accuracy: 0.9542
Epoch 11: accuracy improved from 0.95190 to 0.95425, saving model to //content//drive//MyDrive//models/model-11-0.96.keras
347/347 [=====] - 197s 569ms/step - loss: 0.1138 - accuracy: 0.9542 - val_loss: 0.1106 - val_accuracy: 0.9589
Epoch 12/20
347/347 [=====] - ETA: 0s - loss: 0.1011 - accuracy: 0.9578
Epoch 12: accuracy improved from 0.95425 to 0.95776, saving model to //content//drive//MyDrive//models/model-12-0.92.keras
347/347 [=====] - 197s 568ms/step - loss: 0.1011 - accuracy: 0.9578 - val_loss: 0.1697 - val_accuracy: 0.9244
Epoch 13/20
347/347 [=====] - ETA: 0s - loss: 0.0986 - accuracy: 0.9636
Epoch 13: accuracy improved from 0.95776 to 0.96361, saving model to //content//drive//MyDrive//models/model-13-0.96.keras
347/347 [=====] - 197s 568ms/step - loss: 0.0986 - accuracy: 0.9636 - val_loss: 0.0836 - val_accuracy: 0.9611
Epoch 14/20
347/347 [=====] - ETA: 0s - loss: 0.0965 - accuracy: 0.9614
Epoch 14: accuracy did not improve from 0.96361
347/347 [=====] - 198s 570ms/step - loss: 0.0965 - accuracy: 0.9614 - val_loss: 0.1098 - val_accuracy: 0.9611
Epoch 15/20
347/347 [=====] - ETA: 0s - loss: 0.0809 - accuracy: 0.9665
Epoch 15: accuracy improved from 0.96361 to 0.96650, saving model to //content//drive//MyDrive//models/model-15-0.96.keras
347/347 [=====] - 198s 569ms/step - loss: 0.0809 - accuracy: 0.9665 - val_loss: 0.0861 - val_accuracy: 0.9622
Epoch 16/20
347/347 [=====] - ETA: 0s - loss: 0.0883 - accuracy: 0.9648
Epoch 16: accuracy did not improve from 0.96650
347/347 [=====] - 197s 569ms/step - loss: 0.0883 - accuracy: 0.9648 - val_loss: 0.0621 - val_accuracy: 0.9767
Epoch 17/20
347/347 [=====] - ETA: 0s - loss: 0.0678 - accuracy: 0.9750
Epoch 17: accuracy improved from 0.96650 to 0.97496, saving model to //content//drive//MyDrive//models/model-17-0.98.keras
347/347 [=====] - 198s 572ms/step - loss: 0.0678 - accuracy: 0.9750 - val_loss: 0.0500 - val_accuracy: 0.9800
Epoch 18/20
347/347 [=====] - ETA: 0s - loss: 0.0693 - accuracy: 0.9733
Epoch 18: accuracy did not improve from 0.97496
347/347 [=====] - 198s 569ms/step - loss: 0.0693 - accuracy: 0.9733 - val_loss: 0.0517 - val_accuracy: 0.9811
Epoch 19/20
347/347 [=====] - ETA: 0s - loss: 0.0529 - accuracy: 0.9816
Epoch 19: accuracy improved from 0.97496 to 0.98163, saving model to //content//drive//MyDrive//models/model-19-0.99.keras
347/347 [=====] - 198s 570ms/step - loss: 0.0529 - accuracy: 0.9816 - val_loss: 0.0385 - val_accuracy: 0.9867
Epoch 20/20
347/347 [=====] - ETA: 0s - loss: 0.0574 - accuracy: 0.9776
Epoch 20: accuracy did not improve from 0.98163
347/347 [=====] - 198s 569ms/step - loss: 0.0574 - accuracy: 0.9776 - val_loss: 0.0659 - val_accuracy: 0.9733
```

الشكل 3 - 3 مراحل التدريب

3-8- النتائج:

الجدول التالي يوضح مقارنة بين الدراسات المرجعية ونموذجنا المدرب :

أولاً	ثانياً	ثالثاً	نموذجنا المدرب	من حيث
CNN, Dense Net	CNN, SVM	CNN, KNN, SVM	CNN	التقنيات المستخدمة
0.97	0.96	0.89	0.97	الدقة



الشكل 3 - 4 مخطط نتائج التدريب

3-9- الصعوبات:

لم يكن هناك صعوبات حقيقية واجهتنا بوجود شبكة الأنترنت التي استخدمناها للحصول على العديد من المراجع.

كان الحصول على الداتا ليس بالأمر السهل وخصوصا أن أغلب المرضى لا يقبلون بسحب صورهم وإضافتها إلى مجموعة البيانات.

علاوة على ذلك تتطلب الشبكة العصبية كميات كبيرة من البيانات الدقيقة للحصول على أداء جيد كما يجب أن تكون البيانات خالية من الضوضاء والتشوهات التي يمكن أن تؤثر سلبا على عملية التعلم.

عدم وجود خبرة كافية بالمجال الطبي والحاجز الموجود بين الطبيب وخبراء الذكاء الصناعي.

كما أن الخطأ في مثل هذه المجالات مرفوض رفضا تاما.

3-10- التوصيات:

- جمع بيانات عالية الجودة
- معالجة البيانات: تنظيف البيانات وإزالة الضوضاء والتشوهات.
- زيادة البيانات: استخدام تقنيات زيادة البيانات مثل الدوران لزيادة حجم البيانات المتاحة.
- تقييم الأداء: استخدام مجموعة اختبار مستقلة لتقييم أداء الشبكة العصبية.
- التعلم المستمر: مواصلة تدريب الشبكة العصبية على بيانات جديدة لتحسين أدائها بمرور الوقت.
- استخدام أطر عمل قوية: استخدام أطر عمل مثل TensorFlow لتسريع عملية تطوير وتدريب الشبكات العصبية.
- البحث المستمر: البقاء على اطلاع بأحدث التطورات في مجال التعلم العميق.

-
- [1] Mayo Clinic, "Mayo Clinic Family Health Book," 30 April 2024.
- [2] N. P. Damayanti, M. N. D. Ananda and F. W. Nugraha, "Lung cancer classification using convolutional neural network and DenseNet," *Journal of Soft Computing*, pp. 133-141, September 2023.
- [3] A. Y. Saleh, C. k. Chin, V. Penshie and H. Al-Absi, "Lung cancer medical images classification using hybrid CNN-SVM," *International Journal of Advances in Intelligent Informatics*, pp. 151-162, July 2021.
- [4] T. L. Chaunzwa, A. Hosny, Y. Xu, A. Shafer, N. Diao, M. Lanuti, D. C. Christiani, R. H. Mak and H. J. W. L. Aerts, "Deep learning classification of lung cancer histology using CT images," *Scientific Reports*, 09 March 2021.
- [5] L. Craig and L. Tucci, "What is machine learning and how does it work? In-depth guide".
- [6] Manika, "8 Deep Learning Architectures Data Scientists Must Master," 29 Mar 2024.
- [7] B. Misra, "lung_cancer_images(12000 images, mostly)," 2021.
- [8] M. Gurucharan, "Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network," 20 June, 2024.