



الجمهورية العربية السورية  
الجامعة الافتراضية السورية  
برنامج الإجازة في الهندسة المعلوماتية

مشروع تخرج

أعد لنيل درجة الإجازة في الهندسة المعلوماتية

عنوان المشروع

تحديد الأورام الدماغية في صور الرنين المغناطيسي  
باستخدام التعلم العميق والشبكات العصبونية

Brain tumor segmentation in magnetic resonance imaging  
using deep learning and neural networks

إعداد الطلاب

حسن بلال

إبراهيم فارس

إشراف

د. ياسر خضرا

مقدم في 07/10/2021

## الفهرس

2	الفهرس
4	الاختصارات
5	المصطلحات
7	مقدمة
8	صور الرنين المغناطيسي واستخدامها لاكتشاف الأورام الدماغية
9	1.1- صور الرنين المغناطيسي
11	2.1- التجزئة (segmentation)
12	التعريف بمجموعة البيانات وتهيئتها
13	1.2- مجموعة البيانات
14	2.2- تهيئة مجموعة البيانات
14	2.2.1- تقسيم مجموعة البيانات
16	2.2.2- معالجة البيانات (image preprocessing)
17	تحديد الأورام باستخدام التعلم العميق
18	1.3- التعلم العميق
20	2.3- الشبكات العصبونية التلافيفية Convolutional Neural Networks
20	1.2.3- طبقة التلاف Convolutional Layer
21	2.2.3- طبقة التجميع Pooling Layer
22	3.2.3- طبقة استنظام الحزمة BatchNormalization Layer
23	3.3- الشبكة العصبونية ذات المعمارية "U" U-NET Architecture

23	1.3.3- قسم التقليلص encoder path
24	2.3.3- قسم البناء decoder path
24	3.3.3- الطبقة المتسلسلة Concatenate Layer
25	التنفيذ البرمجي
26	1.4- بيئة العمل
27	2.4- بناء النموذج
31	3.4- تدريب النموذج
32	الاختبارات والنتائج
33	1.5- مقاييس الأداء
35	2.5- دالة الخسارة loss function
36	3.5 اختبار النموذج
37	بنية توثيق المشروع ودليل الاستخدام
38	1.6- بنية توثيق المشروع ومحتوياته
39	2.6- متطلبات التنفيذ
40	3.6- تنفيذ البرنامج
41	الخاتمة والآفاق المستقبلية
43	المراجع

## الاختصارات

MRI: Magnetic Resonance Imaging

ROI: Region Of Interest

CNN: Convolutional Neural Networks

MICCAI: Medical Image Computing and Computer Aided Interventions

MSD: Medical Segmentation Decathlon Dataset

STD: Standard Deviation

RELU: Rectified Linear Unit

DSC: Dice Similarity Coefficient

## المصطلحات

- Segmentation: عملية تحديد مكان الورم في الدماغ
- Glioblastoma: نوع سرطاني عدواني يبدأ في خلايا تسمى الخلايا النجمية
- Sequences: سلاسل صور الرنين المغناطيسي
- Lower-grade glioma: نوع من الأورام التي تأتي من خلايا تسمى الخلايا الدبقية
- Label: هو خرج الشبكة العصبونية قد يكون رقم أو صنف أو صورة
- Dataset: مجموعة البيانات
- Training set: مجموعة التدريب
- Validation set: مجموعة التحقق
- Test set: مجموعة الاختبار
- Overfitting: التلييق الزائد
- Underfitting: عدم التلييق
- Hyperparameters: المعاملات الفائقة
- Generalization: التعميم
- Normalization: استنظام
- Standardization: التوحيد
- Loss function: تابع الخسارة
- Optimizer: المحسن
- Feedback: تغذية استرجاعية
- Backpropagation: هي الخوارزمية التي تبحث عن أفضل قيم أوزان الشبكة العصبونية
- Convolution: عملية التلاف
- Padding: الحشو
- Pooling layer: طبقة التجميع
- Voxel: بكسل ثلاثي البعد
- Encoder path: قسم التقليس
- Decoder path: قسم البناء
- transposed convolutional layer: طبقة التلاف الناقلة
- Concatenate layer: الطبقة المتسلسلة
- Activation function: تابع التنشيط

## المصطلحات

Parameters: المعاملات

Trainable parameters: معاملات التدريب

Epoch: دورة

Batch: دفعة

Weights: أوزان

Bias: انزياح

## مقدمة

الأورام الدبقية هي من أكثر الأورام الدماغية شيوعاً عند البالغين، وغالباً تنشأ من الخلايا الدبقية وتتسلل إلى الأنسجة المحيطة [1] وعلى الرغم من تطور الأبحاث بشكل كبير في مجال الأورام الدبقية، لكن لازال هناك ضعف وفقر في تشخيصها عند المرضى، وتعد هذه الأورام خطيرة جداً على المريض حيث أن معدل استمرار حياة المريض بهذه الأورام سنتين أو أقل [2]. إن علاج هذه الأورام يمكن أن يزيد من معدل عمر المريض، ولكننا نحتاج إلى تقييم تطور المرض وذلك لتحديد استراتيجية العلاج المختارة ويتم القيام بهذه المراقبة والتقييم باستخدام عدة طرق أحدها مراقبة الأنسجة الدماغية المتضررة باستخدام صور الرنين المغناطيسي، يمكن القيام بعملية المراقبة يدوياً ولكنها متعبة وغير دقيقة لذلك لا بد من إيجاد طرق أوتوماتيكية للقيام بهذه العملية.

إن القيام بعملية تحديد الورم أوتوماتيكياً تشكّل بحد ذاتها تحدياً وتقوم عليها العديد من الأبحاث حالياً ولذلك اخترنا أن نقوم بهذا المشروع لمحاولة إثراء وإضافة جزء صغير في سبيل تطوير هذا المجال.

صور الرنين المغناطيسي واستخدامها لاكتشاف الأورام الدماغية

## الفصل الأول

صور الرنين المغناطيسي واستخدامها لاكتشاف الأورام الدماغية



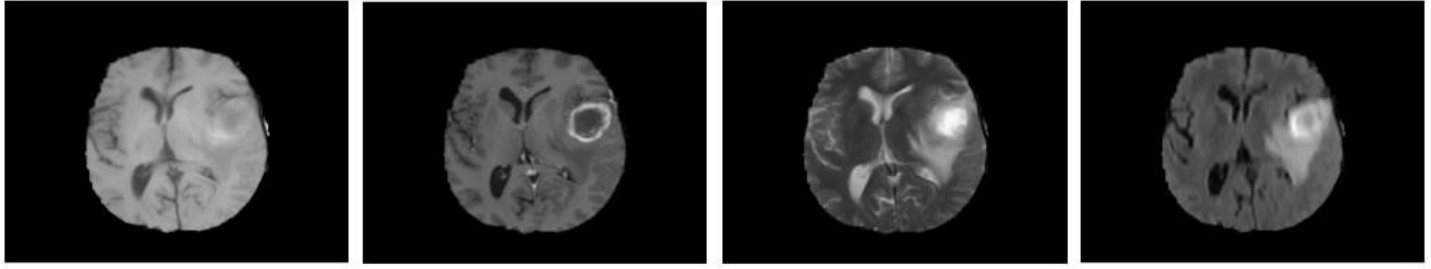
## 1.1- صور الرنين المغناطيسي



"I have the MRI scan of your brain. The right hemisphere is clogged with computer passwords."

مع تقدم التقنيات المستخدمة في الطب تطورت الأساليب المستخدمة في تشخيص الأمراض وتتبع تقدم العلاج وأصبح بالإمكان تصوير أي جزء من جسم الإنسان بعدة وسائل وأحد هذه التقنيات هي صور الرنين المغناطيسي (MRI) فما هي هذه التقنية؟ MRI: هي صور طبية تستخدم في الطب الشعاعي لإظهار صورة تشريحية ثلاثية الأبعاد لأحد أعضاء الجسم باستخدام حقل مغناطيسي قوي وهي صور ثلاثية الأبعاد حيث أبعادها هي (طول، عرض، عمق). [Wikipedia]

ويتم إنشاء هذه الصور عن طريق إطلاق طاقة (مجال مغناطيسي) لأنسجة الدماغ وذلك لإثارة دوران البروتونات ضمن الأنسجة وهذه المرحلة تسمى radio frequency energy ثم يتم إيقاف مصدر الطاقة ويتم بعدها تسجيل الطاقة التي تصدرها الأنسجة عندما تعود البروتونات لمكانها وهذه المرحلة تسمى rate of relaxation وحسب اختلاف كمية الطاقة التي يصدرها كل نسيج يكون التباين وتظهر صورة ال MRI. صور ال MRI مفيدة جداً في التشخيص المبكر للأورام وخاصة في الدماغ حيث يقوم الأطباء باستخدامها باكتشاف وجود الورم وتحديد مكانه وحجمه.



الشكل 1 السلاسل المختلفة لصور الرنين المغناطيسي وهي من اليسار إلى اليمين (T1, T1-Gd, T2, FLAIR)

يوجد عدة سلاسل sequences لصور ال MRI في الدماغ تختلف حسب اختلاف الاليات التي تم تصويرها بها وهي T1 وتستخدم بشكل أساسي لتمييز الأنسجة السليمة ولإظهار التفاصيل التشريحية، و T2 التي تستخدم لتحديد منطقة الوذمة التي تنتج إشارة ساطعة في الصورة، ويوجد العديد من السلاسل sequences الأخرى مثل (FLAIR, T1-Gd, ....) وتختلف استخداماتها حسب ميزاتها ومع استخدام هذه السلاسل sequences يصبح أبعاد صورة ال MRI (طول، عرض، عمق، سلسلة) [3].

## 2.1- التجزئة (segmentation)

كما قلنا سابقاً صور الـ MRI تستخدم في اكتشاف الأورام وتحديد مكانها (segmentation) وهي عملية ضرورية لتحديد حجم الأنسجة المتضررة ويساهم بشكل كبير في رسم خطة العلاج ولكن بما أنها صور ثلاثية الأبعاد فإن عملية تحديد مكان وحجم الورم هي عملية صعبة ومملة للطبيب ومعرضة بشكل كبير للخطأ وذلك لأنه يجب أن يفحص أكثر من مئة صورة ثنائية البعد ليحدد مكان الورم في كل منها، لذلك يتم محاولة تنفيذ هذه المهمة بشكل مؤتمت.

ويمكن استخدام التكنولوجيا لتنفيذ هذه المهمة بأكثر من طريقة نذكر منها:

- **بشكل نصف أوتوماتيكي:** حيث تحتاج هذه الطريقة إلى تدخل العنصر البشري في مهام محددة فقط (مثل تجهيز الصورة، إعطاء تقييم للنتائج، تحسين النتائج، ....) ويمكن أن يعطي تقدير لمكان الورم ROI وبعدها تأتي مهمة الخوارزمية الأوتوماتيكية لتقوم بتحديد منطقة الورم حيث تقوم بتنفيذ مهمتها ثم تستقبل مراجعة من الخبير وبناء عليها تقوم بتعديل النتيجة [3]. ومن أمثلة هذه الطريقة [Hamamci A, et al]، في هذه الورقة البحثية تم تقديم طريقة للقيام بالـ segmentation حيث تتطلب أولاً من خبير تهيئة الصورة برسم أكبر قطر ممكن للورم ثم تأتي العملية الأوتوماتيكية على مرحلتين الأولى تحديد احتمالية موقع الورم والثاني لاحتمال موقع الأنسجة العادية [8].

- **بشكل أوتوماتيكي كامل:** في هذه الطرق لا يوجد تدخل لأي عنصر بشري حيث يتم جمع المعارف السابقة مع الذكاء الاصطناعي لتنفيذ هذه المهمة. في هذه الطريقة يوجد أكثر من آلية حيث أولاً كان يستخدم خوارزميات تعلم الآلة التقليدية مثل (KNN, Decision Tree, ...) ولكن هذه الخوارزميات كانت محدودة ولم تصل نتائجها للطموح المطلوب [4].

وبسبب سلبية الطرق السابقة ومحدوديتها تم القيام بأبحاث ودراسات لتنفيذ الـ segmentation في صور الرنين المغناطيسي الدماغية باستخدام تقنيات التعلم العميق والشبكات العصبونية حيث باستخدام هذه التقنيات يمكن تجنب المحدودية عن طريق قدرتها على التعلم بنفسها لاستخراج ميزات من الصورة. [4] وفي هذا المشروع سنبحث في استخدام هذه التقنيات وفعاليتها في الـ segmentation.

التعريف بمجموعة البيانات وتهيئتها

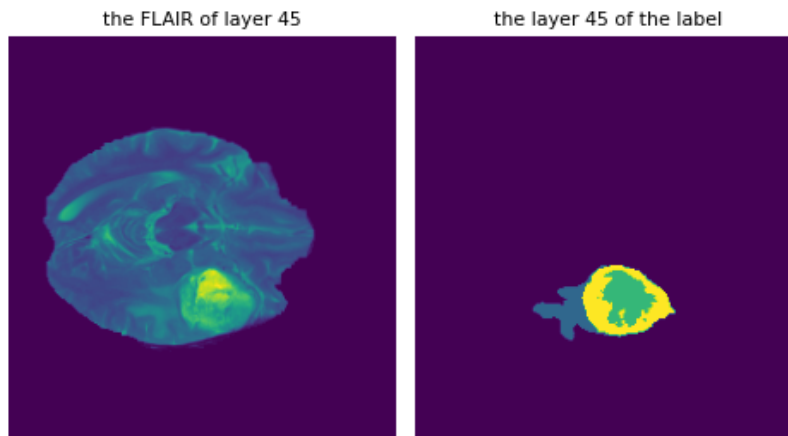
## الفصل الثاني

التعريف بمجموعة البيانات وتهيئتها

## 1.2- مجموعة البيانات

إن أي خوارزمية تعلم عميق تعتمد بشكل أساسي على كمية البيانات المتوفرة التي سيتم استخدامها لتدريب النموذج. في المجال الطبي يعاني الباحثين من قلة البيانات المتوفرة والذي يعود لعدة أسباب ومن أهمها ان المؤسسات الطبية لا تشارك بيانات مرضاها بسبب اعتبارات الخصوصية والخوف من العواقب القانونية والغرامات الكبيرة. كما ان البيانات التي نستخدمها لتدريب الشبكات العصبونية يجب ان تكون معالجة ولكل مثال فيها label يقابله ولذلك حتى عندما يكون من المسموح مشاركة بعض البيانات سنحتاج خبراء لمعالجتها وتجهيزها.

يوجد العديد من مجموعات البيانات المفتوحة المصدر والمعالجة التي من المسموح استخدامها لأغراض بحثية ودراسية، وسنقوم في مشروعنا باستخدام بيانات (MSD) [5] والتي تم تقديمها من قبل مؤتمر (MICCAI) في غرناطة اسبانيا عام 2018 [6]. تحتوي مجموعة البيانات هذه على مجموعات بيانات فرعية لخدمة عشر تحديات [5]، سنستخدم نحن مجموعة البيانات المستخدمة في تحدي (Brain Tumor) وفي هذه المجموعة تم استخدام بيانات من تحدي (BRATS) لعامي 2015 و2016 وهي مؤلفة من 485 عينة (MRI) متعددة السلاسل sequences من مرضى مشخصين إما (glioblastoma) أو (lower-grade glioma) وكل عينة MRI مؤلفة من السلاسل sequences التالية: (T1, T1-Gd, T2, FLAIR)، حيث تم الحصول على هذه البيانات من فحوصات الرنين المغناطيسي ضمن 19 مؤسسة طبية مختلفة، ثم تم تجميعها ومعالجتها و تجهيزها وتصنيفها إلى {edema, enhancing, and non-enhancing tumor} لإنشاء مجموعة بيانات متاحة للقيام بأبحاث لتحديد أورام الدماغ [5,6].



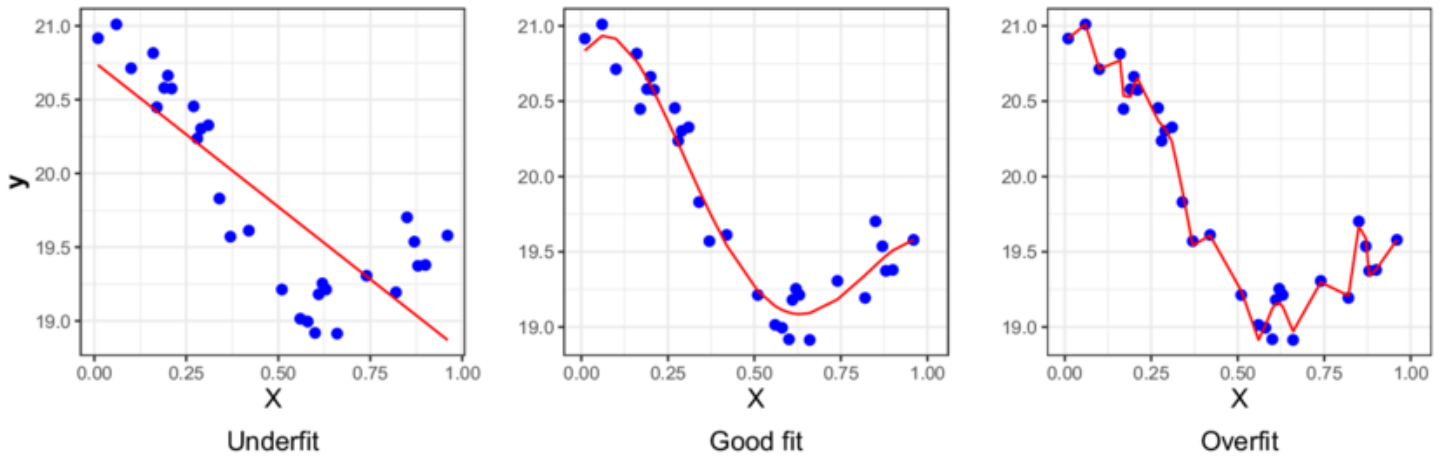
الشكل 2 مثال لأحد عينات مجموعة البيانات

ملاحظة: إن الصور الموجودة في مجموعة البيانات صيغتها DICOM ولفتحها نحتاج إلى استخدام مكتبة nibabel في لغة البرمجة Python حيث يتم تحويلها إلى مصفوفة متعددة الأبعاد عناصرها هي قيم بكسلات الصورة وبعدها يتم عرضها باستخدام مكتبة matplotlib.

## 2.2- تهيئة مجموعة البيانات

قبل إدخال البيانات إلى الشبكة العصبونية يجب تقسيمها وتجهيزها لتناسب مع المطلوب، ويتم تهيئة مجموعة البيانات على أكثر من مرحلة:

### 2.2.1- تقسيم مجموعة البيانات



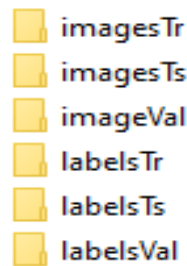
الشكل 3 حالة التليق الزائد overfitting [مصدر الشكل]

إن الهدف الأساسي من تعلم الآلة هو بناء نموذج قادر على التعامل مع البيانات لم يراها من قبل، أي من غير الصحيح تقييم النموذج باستخدام نفس البيانات التي استخدمناها في مرحلة التدريب، والسبب هو أن النتيجة التي سنحصل عليها لن تعرب عن الأداء الفعلي للنموذج، ويعود ذلك لعدة أسباب من أهمها احتمال حدوث تليق زائد overfitting أثناء تدريب النموذج كما هو واضح من الجزء الأيمن من الشكل 2، ففي حالة التليق الزائد إذا قمنا باختبار النموذج على نفس بيانات مجموعة التدريب سنحصل على دقة كبيرة غير واقعية.

في الجزء الأيسر من الشكل 2 لدينا حالة عدم التليق حيث نجد أن المعادلة الخطية المستخدمة لا تغطي النقاط بشكل جيد، وفي هذه الحالة ستكون نسبة الخطأ في النموذج كبيرة، أما في الجزء الأيمن لدينا حالة التليق الزائد وفي هذه الحالة يبالغ النموذج في التعلم على عينات التدريب بحيث أنه يعطي نتائج ممتازة على هذه العينات ولكنه غير قادر على التعميم generalization على بيانات لم يراها من قبل، أما في الحالة التي في الوسط فيعطي النموذج نسبة خطأ صغيرة ويمكن له التعامل مع بيانات جديدة لم يراها من قبل وهذا ما نريده.

تم تقسيم البيانات إلى ثلاث مجموعات بشكل قياسي حسب الآتي:

- مجموعة التدريب training set: تستخدم البيانات في هذه المجموعة لضبط أوزان الشبكة العصبونية في مرحلة تدريب النموذج، وتم تخصيص 400 صورة من عينات مجموعة البيانات لهذا القسم، وتم تحديد هذا العدد من البيانات لأنه يجب أن لا يكون صغير حتى يتمكن النموذج من التعلم على عدد كافٍ من البيانات ونحصل على تنبؤ جيد، كما يتبقى عدد كافي من العينات من أجل اختبار النموذج.
  - مجموعة التحقق validation set: نحتاج في مرحلة التدريب إلى اختيار أفضل بنية ممكنة للنموذج مع تجنب حدوث تلبيق زائد، لذا نستخدم هذه المجموعة بشكل مستمر في مرحلة التدريب بهدف تقييم أداء النموذج عند اختيار القيم المختلفة للمعاملات المفرطة hyperparameters مثل عدد الطبقات وعدد الخلايا في كل طبقة ونوع تابع التنشيط وغيرها، أي أننا نختار قيم المعاملات المفرطة التي تعطي أعلى دقة لمجموعة التحقق أثناء التدريب، ورغم عدم استخدام مجموعة التحقق في ضبط الأوزان إلا أننا لا نستخدمها في تقييم النموذج لأنه تم استخدامها في ضبط المعاملات المفرطة، وتم تخصيص 50 صورة من عينات مجموعة البيانات لهذا القسم.
  - مجموعة الاختبار test set: وهي البيانات التي لم يعتمد عليها النموذج في أي قرار تم اتخاذه أثناء عملية التدريب، أي لم نستخدمها لا في ضبط الأوزان ولا في ضبط المعاملات المفرطة hyperparameters، لذا يمكن استخدام هذه المجموعة بكل ثقة لتقييم أداء النموذج المدرب، وتم تخصيص 35 صورة من عينات مجموعة التدريب لهذا القسم.
- في نهاية هذه المرحلة قمنا بتقسيم مجموعة البيانات إلى مجموعة التدريب (400 صورة) ومجموعة التحقق (50 صورة) ومجموعة الاختبار (35 صورة) وأصبحت لدينا مجلدات البيانات كالاتي:



الشكل 4 مجلدات البيانات

- |   |  |
|---|--|
| imagesTr: صور الدخول لمجموعة بيانات التدريب.  | labelsTr: صور الورم الموافقة لبيانات التدريب.  |
| imagesTs: صور الدخول لمجموعة بيانات الاختبار. | labelsTs: صور الورم الموافقة لبيانات الاختبار. |
| imageVal: صور الدخول لمجموعة بيانات التحقق.   | labelsVal: صور الورم الموافقة لبيانات التحقق.  |

## 2.2.2- معالجة البيانات (image preprocessing)

تحدثنا في الفقرة 2.2.1 عن أفضل طريقة لتقسيم البيانات للحصول على أعلى تقييم ممكن من النموذج، ومن أجل هذا الهدف يجب معالجة دخل الشبكة العصبونية وال label بأفضل الطرق. وتهدف معالجة البيانات إلى جعل البيانات أكثر قابلية للتعدل داخل الشبكة العصبونية [7]، وقد قمنا بعمليات معالجة البيانات التالية:

- **تغيير أبعاد صور الدخل:** عند دراسة الصور من مجموعة البيانات التي لدينا نجد أن السلسلة T1 هي أقل السلاسل sequences احتواءً للمعلومات التي تفيد النموذج في تحديد الورم، لذلك أول عملية معالجة قمنا بها هي التخلص من هذه السلسلة لتصبح أبعاد كل صورة من صور مجموعة البيانات على الشكل التالي (240,240,150,3). ونلاحظ في القسم الأيمن من الشكل 2 أن الجزء الذي لا يحوي على بكسلات تنتمي للورم يعد كبيراً نسبياً، لذلك تخلصنا من عدة بكسلات من أطراف صور الدخل ومن الصور الناتجة من النموذج، وذلك لتسريع تنفيذ النموذج أثناء التدريب وأثناء تحديد الورم، حيث أصبحت أبعاد صور مجموعة البيانات (128,128,128,3) وأبعاد ال label (128,128,128).
- **الاستنظام normalization:** بعد تحويل أحد صور مجموعة البيانات لمصفوفة لدراسة بكسلاتها وجدنا أن قيم هذه البكسلات تتراوح بين 0 وال 2239، وهذا المدى بين البكسلات كبير، كما أنه من الخطير أن تكون قيم البكسلات أكبر من قيم الأوزان الافتراضية للشبكة العصبونية لأنها قد تسبب عدم تحسن هذه الأوزان وبالتالي عدم قدرة النموذج على تحقيق أداء عالٍ، لذلك قمنا بجعل قيم البكسلات بين ال 0 وال 1 وذلك عن طريق قسمة كل بيكسل على قيمة أكبر بيكسل في الصورة.
- **التوحيد standardization:** بعد تصغير قيم البكسلات لأبد من جعل متجانسة وذلك لجعل عملية التعلم أسهل على الشبكة العصبونية، ولذلك في عملية التوحيد جعلنا المتوسط الحسابي لقيم بكسلات كل صورة يساوي الصفر والانحراف المعياري يساوي الواحد من خلال تنفيذ العمليتين الحسابيتين التاليتين على كل بيكسل في الصورة:

$$\text{value} = \text{value} - \text{mean}$$

$$\text{value} = \text{value} / \text{std}$$



تحديد الأورام باستخدام التعلم العميق

### الفصل الثالث

تحديد الأورام باستخدام التعلم العميق

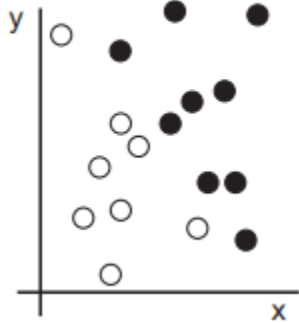
### 1.3- التعلم العميق

هو جزء من أساليب تعلم الآلة الواسعة ويعتمد على الشبكات العصبونية الصناعية مع التعلم التمثيلي، والتعلم يمكن أن يكون تحت الاشراف أو شبه خاضع للإشراف أو بدون إشراف. [Wikipedia]

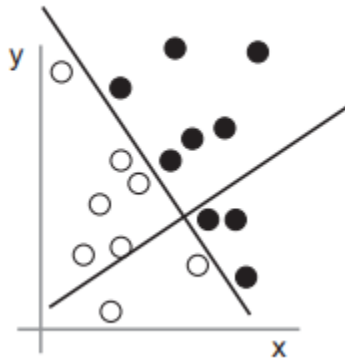
يعتبر التعلم العميق من أحدث المفاهيم في مجال تعلم الآلة، حيث نقوم فيه ببناء شبكة عصبونية تتألف من عدة طبقات مخفية، أي أن حجر الأساس في التعلم العميق هو الشبكات العصبونية، وقد حقق التعلم العميق تقدماً كبيراً في الكثير من المجالات مثل التعرف على الصور والترجمة الآلية والتعرف على الكلام ومعالجة النصوص حيث تفوق بذلك على معظم الأساليب التي كانت تستخدم سابقاً في هذه المجالات، وأصبح من أكثر تقنيات الذكاء الصناعي انتشاراً في مختلف مجالات الحياة.

ولنفهم جيداً كيف يعمل التعلم العميق يجب فهم كيف يعمل تعلم الآلة، حيث في تعلم الآلة تقوم الخوارزمية باكتشاف القواعد لتنفيذ مهمة محددة من البيانات التي لديها، وببساطة فإن خوارزميات تعلم الآلة تقوم بالبحث عن تمثيلات مفيدة لبيانات الدخل ضمن مساحة محددة من الاحتمالات ليصبح بإمكاننا تصنيف هذه البيانات وتنفيذ مهمتنا باستخدامها.

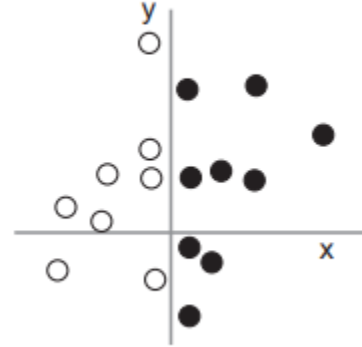
1: Raw data



2: Coordinate change



3: Better representation



الشكل 5 طريقة عمل خوارزمية التعلم الآلي للبحث عن أفضل تمثيل للبيانات

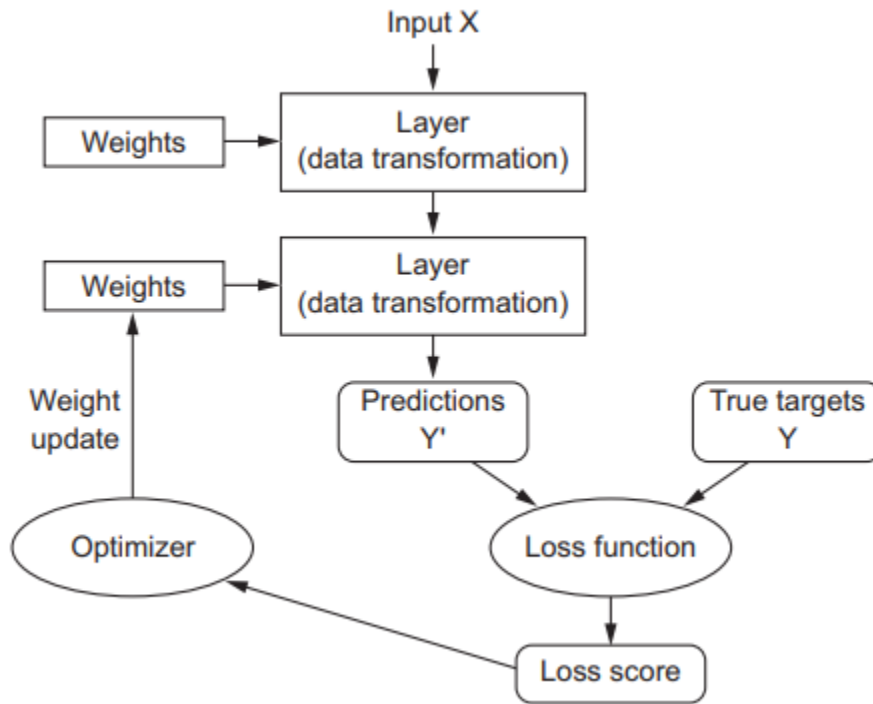
أما التعلم العميق فهو حقل فرعي محدد من التعلم الآلي ولا يعتمد على طريقة تمثيل البيانات بل يقوم بالتعلم من خلال طبقات بشكل متزايد. وكلمة عميق في مصطلح التعلم العميق تشير إلى الطبقات وعددها.

يقوم التعلم العميق بربط المدخلات إلى أهدافها labels عبر سلسلة عميقة من عمليات تحويل البيانات البسيطة (الطبقات) وعمليات التحويل هذه تتعلم من البيانات التي تنطبق عليها. ولنكون محددين فإن ما تفعله كل طبقة للبيانات المدخلة لها يُخزن في أوزان الطبقة والتي هي مجموعة من الأرقام.

وبذلك يكون التعلم هو إيجاد مجموعة من القيم لأوزان كل الطبقات في الشبكة العصبونية بحيث أن هذه الشبكة ستقوم بربط بيانات الدخل بالنتائج المستهدفة بشكل صحيح.

ولتنفيذ هذه العمليات يجب أن نكون قادرين على قياس مدى قرب خرج الشبكة العصبونية قريب من النتيجة المتوقعة، وهذه مهمة دالة الخسارة loss function.

وبعد أن نقيس مدى قرب الخرج من الهدف نقوم بعمل تغذية استرجاعية feedback لتعديل قيم الأوزان في طبقات الشبكة العصبونية، وهذه تكون مهمة المحسن (optimizer) الذي يستخدم خوارزمية Backpropagation وهي الخوارزمية الأساسية في التعلم العميق [7].



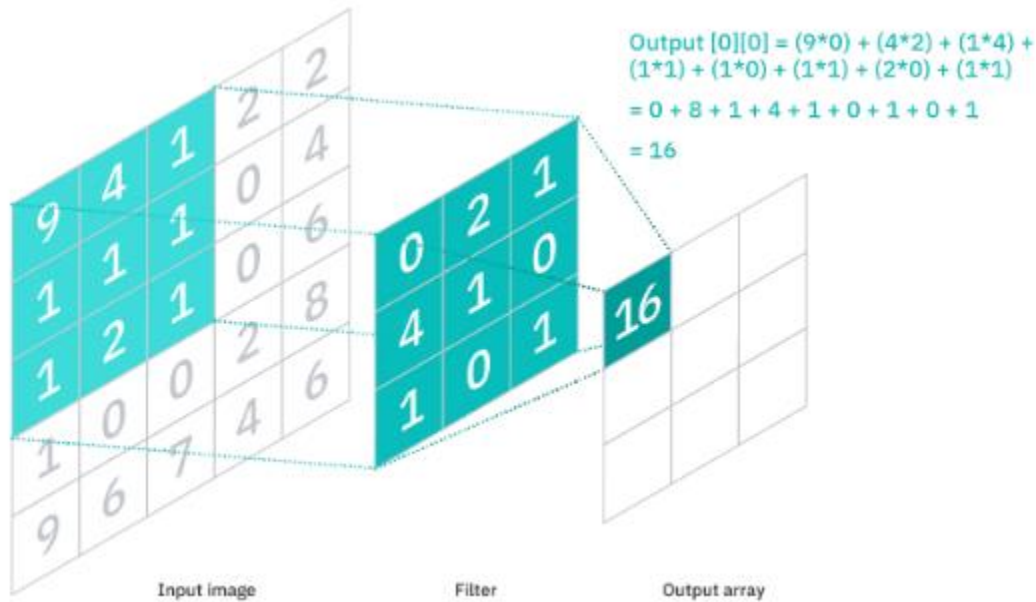
الشكل 6 طريقة عمل التعلم العميق

## 2.3- الشبكات العصبونية التلافيفية Convolutional Neural Networks

الشبكات العصبونية التلافيفية CNN هي نوع خاص من الشبكات العصبونية لمعالجة البيانات التي لها طوبولوجيا معروفة مثل الشبكة، ومن الأمثلة على ذلك الصور والتي يمكن اعتبارها شبكة ثنائية الأبعاد من البكسلات، وأيضاً الصور ثلاثية البعد كصور الرنين المغناطيسي MRI التي سنستخدمها في مشرونا، وقد حققت الشبكات العصبونية التلافيفية نجاحات كبيرة في التطبيقات العملية، وسميت الشبكات العصبونية التلافيفية بهذا الاسم للإشارة إلى أنها تستخدم العملية الرياضية التي تدعى جداء التلاف convolution وهي نوع خاص من العمليات الخطية، لذا يتم تعريف شبكات التلاف على أنها شبكات عصبونية تستخدم جداء التلاف بدلاً من جداء المصفوفات في واحدة من طبقاتها على الأقل [9].

### 1.2.3- طبقة التلاف Convolutional Layer

يتم في طبقة التلاف استخدام مجموعة مرشحات filters تنفذ الطبقة من خلالها عملية جداء تلاف أثناء مرورها على الدخل حسب أبعاده، والمعاملات المفرطة hyperparameters في هذه الطبقة هي أبعاد المرشح وعدد المرشحات ومقدار الخطوة والحشو padding، حيث يؤثر عدد المرشحات على عمق صورة الخرج فمثلاً لو استخدمنا 4 مرشحات أبعادها 3x3 على صورة ثنائية البعد أبعادها 5x5 سيكون أبعاد الخرج 3x3x4، أما مقدار الخطوة فهو المسافة أو عدد البكسلات التي يتحرك بها مركز المرشح على صورة الدخل، والحشو هو يستخدم عادة لنحصل على صورة خرج أبعادها مساوية لأبعاد صورة الدخل من خلال إضافة بكسلات على أطراف صورة الدخل [10].



الشكل 7 مثال عن عملية التلاف مصدر الشكل [10]

نلاحظ في الشكل 7 صورة ثنائية البعد أبعادها 5x5 موجودة على دخل طبقة التلاف، حيث يمثل الجزء الملون بلون أزرق فاتح فيها البكسلات التي سيتم تطبيق عملية التلاف بينها وبين المرشح filter، ونلاحظ وجود مرشح أبعاده 3x3، ولنفرض أن طول الخطوة واحد على البعدين، عندها سيتحرك المرشح filter 9 مرات ليمسح كل الصورة، وعند الانتهاء من كل عمليات التلاف نحصل على مصفوفة جديدة تعتبر الخرج أبعادها 3x3.

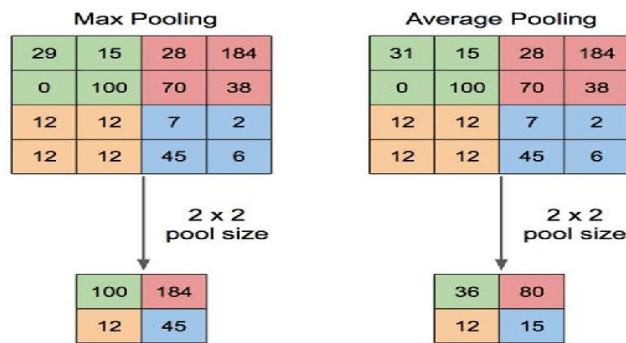
مع كل طبقة تلاف convolutional layer يزداد تعقيد الشبكة العصبونية التلافية CNN، مما يؤدي إلى ملاحظة ميزات أكبر من الصورة. حيث في الطبقات الأولى من الشبكة يتم اكتشاف بسيطة مثل الألوان أو الحواف العمودية والأفقية. ومع تقدم الطبقات ضمن الشبكة نبدأ باكتشاف وتحديد عناصر أكبر ضمن الصورة حتى نتعرف على الغرض المطلوب تحديده في النهاية [10].

### 2.2.3- طبقة التجميع Pooling Layer

تسمى طبقة التجميع بطبقة الاختزال أيضاً، حيث تقوم بتقليل الأبعاد، مما يؤدي لتقليل عدد الأوزان المدربة، وهي مشابهة في طريقة عملها لطبقة التلاف convolutional layer حيث تقوم بتمرير مرشح filter على صورة الدخل، ولكن مع عدة اختلاف أولها أن المرشح filter لا يحوي أوزان، ولا تقوم بعملية جداء التلاف convolution بل تقوم بتطبيق دالة تجميع على البكسلات التي يغطيها الفلتر ليتم تجميع النتيجة في بكسل صورة الخرج [10].

يوجد نوعين لعمليات التجميع pooling:

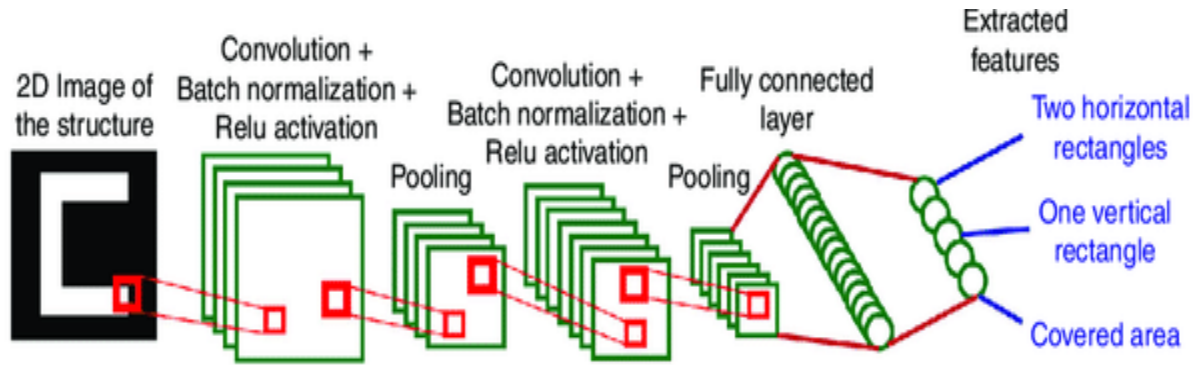
- تجميع الأعلى max pooling: أثناء مرور المرشح filter على صورة الدخل يقوم بإرسال القيمة الأعلى في كل حقل يقوم بتغطيته إلى صورة الخرج.
- تجميع المتوسط average pooling: أثناء مرور المرشح filter على صورة الدخل يقوم بإرسال متوسط قيم بكسلات كل حقل يقوم بتغطيته إلى صورة الخرج.



الشكل 8 مثال على عمل طبقة التجميع

### 3.2.3- طبقة استنظام الحزمة Batch Normalization Layer

توضع هذه الطبقة عادة بعد كل طبقة تلاف convolutional layer، حيث تقوم هذه الطبقة بتنفيذ عمليات حسابية لاستنظام normalize خرج الطبقة، فتجعل قيمة المتوسط الحسابي mean للخرج يساوي الصفر، والانحراف المعياري std للخرج يساوي الواحد.

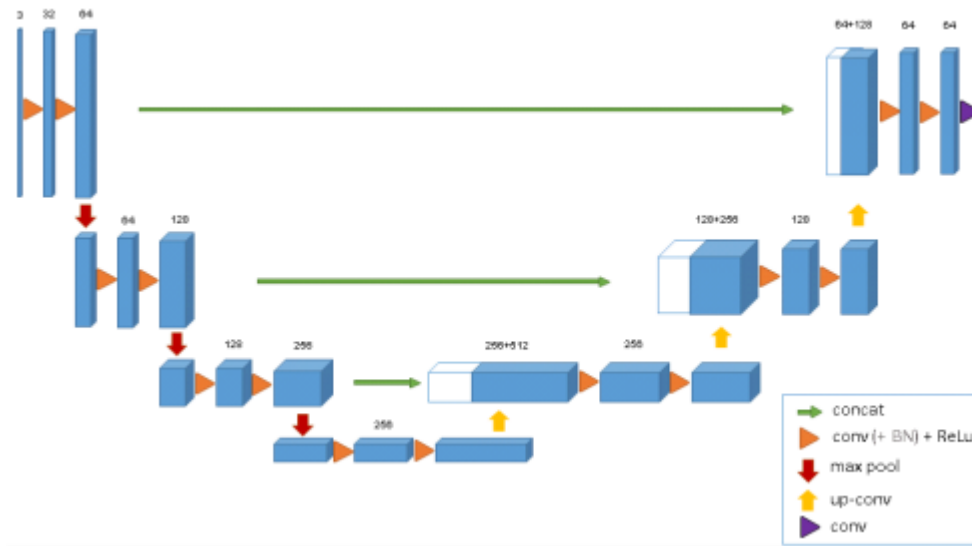


الشكل 9 مثال على شبكة عصبونية تلافيفية بسيطة [مصدر الشكل]

عادة ما يتم استخدام الشبكات العصبونية التلافيفية CNN التقليدية في مهام التصنيف أو المهام التي يكون خرجها label وحيد [12]، لكن في مهام تحديد الأغراض في الصور ثنائية البعد والصور ثلاثية البعد يكون استخدام الشبكات العصبونية التلافيفية CNN التقليدية فيه العديد من السلبيات مثل ضعف الأداء، الحاجة لحجم كبير من مجموعات البيانات (وهذا غير متوفر في البيانات الطبية)، فقدان المعلومات المكانية spatial information [11]. لذلك سنحتاج لاستخدام معمارية جديدة تعتمد على نفس مفاهيم الشبكات العصبونية التلافيفية CNN التقليدية لكن مع ميزات أخرى تتجاوز سلبياتها وهو ما سنراه في معمارية U-Net.

### 3.3- الشبكة العصبونية ذات المعمارية "U" U-NET Architecture

إن عملية تحديد الأغراض segmentation في الصورة يمكن اعتبارها كتصنيف لكل بكسل pixel في الصورة وتحديد الصنف class الذي ينتمي له، وفي حالتنا سنحتاج لتصنيف كل فوكسل voxel، ولكن من الصعب حسابياً القيام بعملية التصنيف لكل فوكسل voxel على حدى وهنا تأتي أهمية الشبكة العصبونية U-Net التي استخدمناها في هذا المشروع، حيث في هذه الشبكة العصبونية تأخذ الصورة كلها كمدخل وتقوم بمعالجتها من خلال قسم التقليل encoder path وتحدد الورم على صورة الخرج باستخدام قسم البناء decoder path، بدون ان نقوم بأي عملية تقسيم للصورة [12,13].



الشكل 10 مثال على شبكة عصبونية U-Net بسيطة مصدر الشكل [13]

#### 1.3.3- قسم التقليل encoder path

يمثل قسم التقليل encoder path الجزء الأيسر من الشكل 10، حيث يستخدم هذا القسم خواص الشبكة العصبونية التليفية CNN [12]، وبكل خطوة في هذا القسم تنقص أبعاد الصورة ولكن يزداد عمقها، حيث يقوم هذا القسم باستخراج الميزات من الصورة ككل.

نلاحظ في الشكل 10 أن هذا القسم يتألف من مجموعة من الكتل blocks وفي كل كتلة لدينا عدد من الطبقات، حيث تقوم الصورة أولاً بالمرور على طبقة تلاف convolutional layer، ثم على طبقة استنظام الحزمة BatchNormalization Layer، وفي الأخير لدينا طبقة التجميع pooling layer وتكرر هذه الطبقات في كل الكتل blocks حتى نستخرج الميزات المهمة لتحديد الورم من الصورة في النهاية [12,13].

### 2.3.3- قسم البناء decoder path

يمثل قسم البناء decoder path الجزء الأيمن من الشكل 10، ويعتبر دخل هذا القسم هو نفسه خرج قسم التقليل encoder path، في كل خطوة في هذا القسم يتم زيادة أبعاد الصورة مع تقليل العمق، ينتج من هذا القسم صورة تم فيها تحديد الفوكسلات voxels التي تنتمي للورم [12].

وكما هو واضح في الشكل 10 يتألف هذا القسم أيضاً من مجموعة من الكتل blocks وفي كل كتلة لدينا عدد من الطبقات وهي نفس الطبقات في الكتل الموجودة في قسم التقليل encoder path، حيث يمر الدخل أولاً على طبقة التلاف الناقل transposed convolutional layer حيث تقوم هذه الطبقة بعكس ما تقوم به طبقة التلاف convolutional layer، حيث تقوم بتحويل شكل خرج الكتلة التي تسبقه إلى دخل أكبر للكتلة التالية مع الحفاظ على نمط الخرج السابق.

### 3.3.3- الطبقة المتسلسلة Concatenate Layer

نلاحظ في الشكل 10 وجود سهم لونه أخضر، يربط كل كتلة block قسم التقليل encoder path إلى الكتلة المقابلة لها في قسم البناء decoder path، يعبر هذا السهم عن الطبقة المتسلسلة concatenate layer التي مهمتها أن تقوم بالحفاظ على المعلومات المكانية spatial information بين الكتل المتقابلة، فبعد تحليل الصورة وتحديد البكسلات التي تنتمي للورم قد يتم فقدان بعض المعلومات ومنها المعلومات المكانية spatial information، لذلك لا بد من إيجاد طريقة للحفاظ على هذه المعلومات عند بناء صورة الخرج وهذا ما تقوم به الطبقة المتسلسلة concatenate layer [11].



## الفصل الرابع

## التنفيذ البرمجي

#### 1.4- بيئة العمل

استخدمنا في هذا العمل لغة بايثون Python وهي اللغة الأكثر استخداماً في مجال تعلم الآلة في السنوات الأخيرة [14] ، ومن أسباب انتشارها الواسع في هذا المجال هو سهولة تعلمها وإمكانية استخدامها لتحقيق البرنامج المطلوب بعدد أقل من الأسطر البرمجية مقارنة بلغات برمجة أخرى مثل C++, Java، بالإضافة إلى أنها تحوي على العديد من المكتبات البرمجية لمعالجة البيانات الكبيرة وعرض الرسوم البيانية بسهولة، كما نجد أن معظم الجامعات الرائدة عالمياً تعتمد هذه اللغة في مقررات تعلم الآلة.

لبناء وتدريب نموذج الشبكة العصبونية قمنا باستخدام منصة Google Colab، وهي منصة برمجية وخدمة سحابية مجانية للباحثين والطلاب في مجال علوم البيانات مقدمة شركة Google، وتقدم العديد من الميزات والفوائد حيث يمكن استخدام موارد أعلى من المتوفر على الحواسيب المحلية، كما توفر إمكانية تدريب الشبكة العصبونية باستخدام وحدة معالجة الرسومات (Graphics Processing Unit (GPU، بالإضافة أننا لا نحتاج إلى إعداد البيئة لأنها تأتي مع حزمة مثبتة سابقاً من المكتبات وأطر العمل البرمجية، مع إمكانية إضافة مكتبات خارجية لاستخدامها، ويوضح الجدول 1 تفاصيل أكثر عن بيئة العمل.

جدول 1 بيئة تدريب واختبار النموذج.

Programming Language	Python 3.7.12
Framework	TensorFlow 2.6.0
Graphics Processing Unit (GPU)	NVIDIA Tesla K80 (12GB)
Memory	13GB

## 2.4- بناء النموذج

لبناء نموذج التعلم العميق قمنا باستخدام الشبكة العصبونية U-Net التي تحدثنا عنها في الفقرة 3.3، يبدأ هذا النموذج بطبقة دخل input layer يتم إدخال البيانات إلى هذه الطبقة أولاً ليتم تجهيزها وتمريرها للنموذج، وقد تم تقسيم مجموعة التدريب إلى دفعات batches حجم كل منها 2 عينة ليكون شكل الدخل (2,128,128,128,3)، ثم بعدها في قسم التقليل encoder path استخدمنا خمسة كتل blocks في كل كتلة تتكون من طبقة تلاف ثلاثية البعد Conv3D تم فيها استخدام مرشح filter أبعاد 3x3x3، وطبقة استنظام الحزمة BatchNormalization layer وبعدها قمنا بتطبيق تابع التنشيط activation function وفي حالتنا استخدمنا التابع Rectified Linear Unit(Relu) الذي يعطى بالعلاقة  $\text{relu}(z) = \max(0, z)$  ثم كررنا هذه الطبقات في نفس الكتلة، ومررنا خرج كل كتلة block على طبقة تجميع pooling layer واستخدمنا نوع max pooling باستثناء الكتلة الخامسة، ويوضح الجدول 2 تفاصيل أكثر عن قسم التقليل encoder path.

جدول 2 تفاصيل الطبقات في قسم التقليل

الكتلة blocks	الطبقة layer	شكل الخرج output shape	عدد معاملات number of parameters
Block1	conv0 (conv3D)	(2,128,128,128,16)	1312
	bn0 (batchNormaliztion)	(2,128,128,128,16)	64
	relu0 (activation layer)	(2,128,128,128,16)	0
	conv1 (conv3D)	(2,128,128,128,16)	6928
	bn1 (batchNormaliztion)	(2,128,128,128,16)	64
	relu1 (activation layer)	(2,128,128,128,16)	0
	maxPool1 (pooling layer)	(2,64,64,64,16)	0
Block2	conv0 (conv3D)	(2,64,64,64,32)	13856
	bn0 (batchNormaliztion)	(2,64,64,64,32)	128
	relu0 (activation layer)	(2,64,64,64,32)	0
	conv1 (conv3D)	(2,64,64,64,32)	27680
	bn1 (batchNormaliztion)	(2,64,64,64,32)	128
	relu1 (activation layer)	(2,64,64,64,32)	0
	maxPool2 (pooling layer)	(2,32,32,32,32)	0

Block3	conv0 (conv3D)	(2,32,32,32,64)	55360
	bn0 (batchNormaliztion)	(2,32,32,32,64)	256
	relu0 (activation layer)	(2,32,32,32,64)	0
	conv1 (conv3D)	(2,32,32,32,64)	110656
	bn1 (batchNormaliztion)	(2,32,32,32,64)	256
	relu1 (activation layer)	(2,32,32,32,64)	0
	maxPool3 (pooling layer)	(2,16,16,16,64)	0
Block4	conv0 (conv3D)	(2,16,16,16,128)	221312
	bn0 (batchNormaliztion)	(2,16,16,16,128)	512
	relu0 (activation layer)	(2,16,16,16,128)	0
	conv1 (conv3D)	(2,16,16,16,128)	442496
	bn1 (batchNormaliztion)	(2,16,16,16,128)	512
	relu1 (activation layer)	(2,16,16,16,128)	0
	maxPool4 (pooling layer)	(2,8,8,8,128)	0
Block5	conv0 (conv3D)	(2,8,8,8,256)	884992
	bn0 (batchNormaliztion)	(2,8,8,8,256)	1024
	relu0 (activation layer)	(2,8,8,8,256)	0
	conv1 (conv3D)	(2,8,8,8,256)	1769728
	bn1 (batchNormaliztion)	(2,8,8,8,256)	1024
	relu1 (activation layer)	(2,8,8,8,256)	0

بعد الانتهاء من قسم التقليل encoder path قمنا ببناء قسم البناء decoder path حيث يتألف من أربع أجزاء كل جزء يتكون من طبقة التلاف الناقل transposed convolutional layer ثم بعدها يمر الخرج على طبقة متسلسلة concatenate layer لينتهي الجزء بكتلة block مشابهة للكتلة التي استخدمناها في قسم التقليل encoder path، ويوضح الجدول 3 تفاصيل أكثر عن قسم البناء decoder path ماعدا الطبقات المتسلسلة concatenate layer سيتم وضعها في جدول لوحدها.

جدول 3 تفاصيل الطبقات في قسم التقليل

الكتلة blocks	الطبقة layer	شكل الخرج output shape	عدد معاملات number of parameters
-	transpose5 (Conv3DTranspose)	(2,16,16,16,128)	262272
Decode3	conv0 (conv3D)	(2,16,16,16,128)	884864
	bn0 (batchNormaliztion)	(2,16,16,16,128)	512
	relu0 (activation layer)	(2,16,16,16,128)	0
	conv1 (conv3D)	(2,16,16,16,128)	442496
	bn1 (batchNormaliztion)	(2,16,16,16,128)	512
	relu1 (activation layer)	(2,16,16,16,128)	0
-	transpose3 (Conv3DTranspose)	(2,32,32,32,64)	65600
Decode2	conv0 (conv3D)	(2,32,32,32,64)	221248
	bn0 (batchNormaliztion)	(2,32,32,32,64)	256
	relu0 (activation layer)	(2,32,32,32,64)	0
	conv1 (conv3D)	(2,32,32,32,64)	110656
	bn1 (batchNormaliztion)	(2,32,32,32,64)	256
	relu1 (activation layer)	(2,32,32,32,64)	0
-	transpose2 (Conv3DTranspose)	(2,64,64,64,32)	16416
Decode1	conv0 (conv3D)	(2,64,64,64,32)	55328
	bn0 (batchNormaliztion)	(2,64,64,64,32)	128
	relu0 (activation layer)	(2,64,64,64,32)	0
	conv1 (conv3D)	(2,64,64,64,32)	27680
	bn1 (batchNormaliztion)	(2,64,64,64,32)	128
	relu1 (activation layer)	(2,64,64,64,32)	0
-	transpose1 (Conv3DTranspose)	(2,128,128,128,16)	4112

outConve	conv0 (conv3D)	(2,128,128,128,16)	13840
	bn0 (batchNormaliztion)	(2,128,128,128,16)	64
	relu0 (activation layer)	(2,128,128,128,16)	0
	conv1 (conv3D)	(2,128,128,128,16)	6928
	bn1 (batchNormaliztion)	(2,128,128,128,16)	64
	relu1 (activation layer)	(2,128,128,128,16)	0

وبعد الانتهاء من بناء طرفي الشبكة العصبونية U-Net قمنا بإضافة الطبقات المتسلسلة concatenate layers، حيث تم وصل هذه الطبقات بين كتل blocks من قسم التقليل encoder path مع كتل blocks من قسم البناء decoder path، ويوضح الجدول 4 تفاصيل أكثر عن هذه الطبقات.

جدول 4 تفاصيل الطبقات المتسلسلة concatenate layers

الاسم (النوع) name (type)	البداية start	النهاية end
concat4 (concatenate)	transpose5	Block4
concat3 (concatenate)	transpose3	Block3
concat2 (concatenate)	transpose2	Block2
concat1 (concatenate)	transpose1	Block1

وفي نهاية النموذج نضيف طبقة تلاف convolutional layer مع أربع مرشحات filters وتحوي تابع تنشيط activation function من نوع "softmax" الذي يعمل على تحويل خرج الشبكة إلى شعاع من الاحتمال مجموع قيمه يساوي 1 وطوله يساوي عدد الصفوف classes التي نريد التعرف عليها وهي 4 صفوف في حالتنا.

بعد الانتهاء من بناء نموذج الشبكة العصبونية U-Net وجدنا أن:

- عدد المعاملات فيه "5,651,716"

- عدد معاملات التدريب هو "5,648,772"

- عدد المعاملات التي لن تتدرب هو "2,944"

### 3.4- تدريب النموذج

قمنا بتدريب النموذج على عدد دورات epochs يساوي 40 دورة، وقد تم تقسيم مجموعة التدريب إلى دفعات batches حجم كل منها 2 عينة.

إن شكل الدخل هو مصفوفة لها خمسة أبعاد (batch\_size, height, width, depth, sequences)، كما أن خرج كل طبقة هي مصفوفة لها خمسة أبعاد أيضاً وهي نفس أبعاد الدخل، حيث يبقى البعد الأول (حجم الدفعة) ثابتاً، في حين أنه من الممكن لبقية الأبعاد أن تتغير حسب مواصفات الدفعة، أما بالنسبة للمعاملات القابلة للتعلم trainable parameters في طبقات التلاف convolutional layers فهي عبارة عن القيم التي يتكون منها كل مرشح filter (وتمثل الأوزان weights) بالإضافة إلى قيمة لكل مرشح تسمى الانزياح bias والتي تماثل الحد الثابت b في التابع الخطي  $y = ax + b$  حيث نعلم أن هذا الحد يسمى بانزياح المستقيم من أجل الحصول على تمثيل جيد للبيانات، لكن بدوننا نجد أن المستقيم دائماً يمر من المبدأ (0,0) مما يجعل الحصول على تنبؤ جيد أمراً صعباً.

بعد تدريب الشبكة العصبونية تم الحصول على 40 نموذج، وتم اختيار النموذج ذي الدقة الأعلى وكان عند الدورة 33، سنتحدث في الفصل القادم عن كيفية حساب دقة النموذج.

## الفصل الخامس

## الاختبارات والنتائج

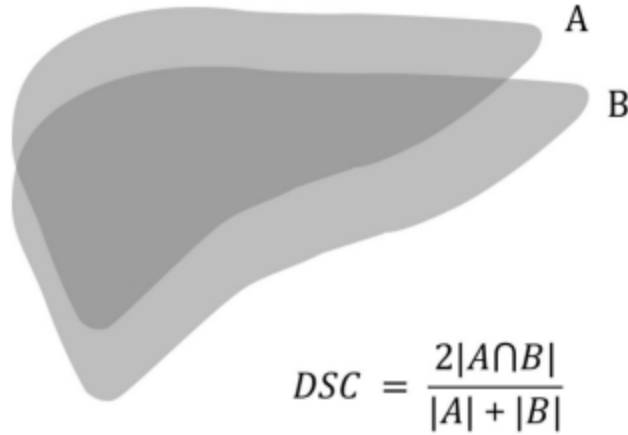


### 1.5- مقاييس الأداء

تعد مقاييس الأداء جزءاً أساسياً في كل مشروع تعلم عميق، حيث يخبروننا بشكل رقمي إذا ما كنا نحرز تقدماً خلال تدريب نموذج الشبكة العصبونية.

في مهام ومشاريع التعلم العميق التي تقوم بتحديد غرض object في الصورة، لا نستطيع استخدام مقاييس الأداء التقليدية بل نحتاج إلى خوارزميات قياس أداء مخصصة، وخاصةً مع وجود عدم توازن في صفوف الخرج، حيث قلنا سابقاً أن كمية البكسلات pixels التي تعبر عن الخلفية أكبر بكثير من باقي الصفوف.

لذلك قمنا باستخدام خوارزمية dice coefficient ويوضح الشكل 11 كيفية عمل هذه الخوارزمية:



الشكل 11 Dice Similarity Coefficient

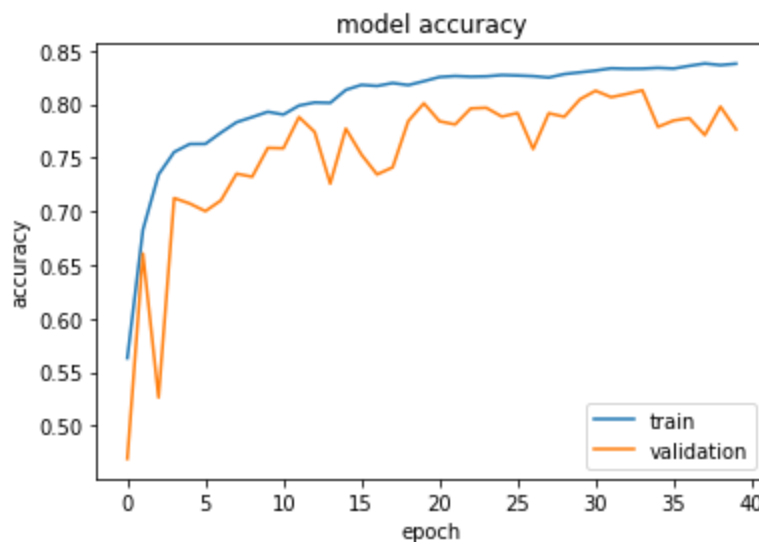
حيث تقوم هذه الخوارزمية DSC بحساب درجة تشابه خرج الشبكة العصبونية مع الصورة الأصلية للخرج label [16].

$$DC(f, x, y) = \frac{1}{N} \sum_{c=1}^C (DC_c(f, x, y))$$

الشكل 12 علاقة DSC لأكثر من صف class مصدر الشكل [11]

يوضح الشكل 10 كيفية حساب معامل DSC لأكثر من صف class، حيث يتم حساب مجموع المعامل DSC لكل الصفوف ثم يتم القسمة على عددها.

وفي النموذج المدرب لدينا تم الحصول على قيمة DSC مساوية لـ 81.27% وهي نسبة جيدة نسبياً في مجالنا.



الشكل 13 تغير قيمة DSC أثناء تدريب النموذج

يوضح الشكل 12 مقدار تغير معامل DSC أثناء تدريب نموذج الشبكة العصبونية U-Net، ونلاحظ أن قيمة المعامل لبيانات التدريب في الدورة 33 epoch ليس أكبر قيمة ولكن هي أكبر قيمة لبيانات التحقق لذلك اخترنا النموذج عند الدورة 33 ليكون هو النموذج الذي سنستخدمه في نظامنا.

يوجد في مجال الرعاية الطبية healthcare أيضاً بعض مقاييس الأداء التي تعمل على عينة واحدة من عينات مجموعة البيانات مثل الحساسية sensitivity والخصوصية specificity.

تعتبر الحساسية sensitivity عن احتمالية أن يعطينا خرج النموذج بكسل pixel ينتمي للورم مع العلم أن هذا البكسل ينتمي للورم.

أما الخصوصية specificity فتعبر عن احتمالية أن يعطينا خرج النموذج بكسل pixel لا ينتمي للورم مع العلم أنه لا ينتمي للورم.

تم حساب الحساسية sensitivity والخصوصية specificity على عينة عشوائية من عينات مجموعة البيانات فأعطت النتائج التالية الموضحة في الجدول 5.

جدول 5 الحساسية sensitivity والخصوصية specificity لعينة عشوائية

class type نوع الصف	Background	Edma	Non-enhancing	Enhancing
sensitivity الحساسية	0.99	0.80	0.85	0.73
specificity الخصوصية	0.92	0.99	0.99	0.99

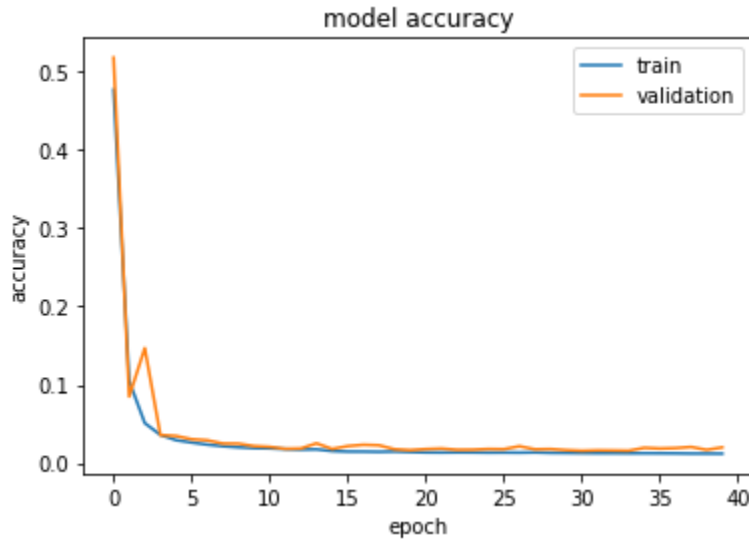
## 2.5- دالة الخسارة loss function

هي أهم جزء في عملية تدريب نموذج الشبكة العصبونية، حيث نسعى خلال التدريب إلى خفض قيمة خرج هذه الدالة أكثر ما يمكن عن طريق إعطاء قيم جديدة للأوزان weights اعتماداً على خوارزمية backpropagation. في النموذج الذي قمنا بتدريبه اعتمدنا على دالة الخسارة لمعامل DSC وتعطي قيمة بين الصفر والواحد حيث تحسب بالعلاقة التالية:

$$L_{Dice}(p, q) = 1 - \frac{2 \times \sum_{i,j} p_{ij} q_{ij} + \epsilon}{\left( \sum_{i,j} p_{ij}^2 \right) + \left( \sum_{i,j} q_{ij}^2 \right) + \epsilon}$$

الشكل 14 Dice Loss مصدر الشكل [11]

وقد أعطى النموذج المختار قيمة 0.012، والشكل 15 يوضح تغير قيمة دالة الخسارة LDice أثناء تدريب النموذج.



الشكل 15 تغير قيمة دالة الخسارة LDice أثناء تدريب النموذج

ونلاحظ أن قيمة تابع الخسارة لمجموعة التدريب training set مساوية تقريباً لقيمة تابع الخسارة لبيانات التحقق validate set بعد الدورة الخامسة وهذا دليل على عدم وجود تليبق زائد overfitting.

### 3.5 اختبار النموذج

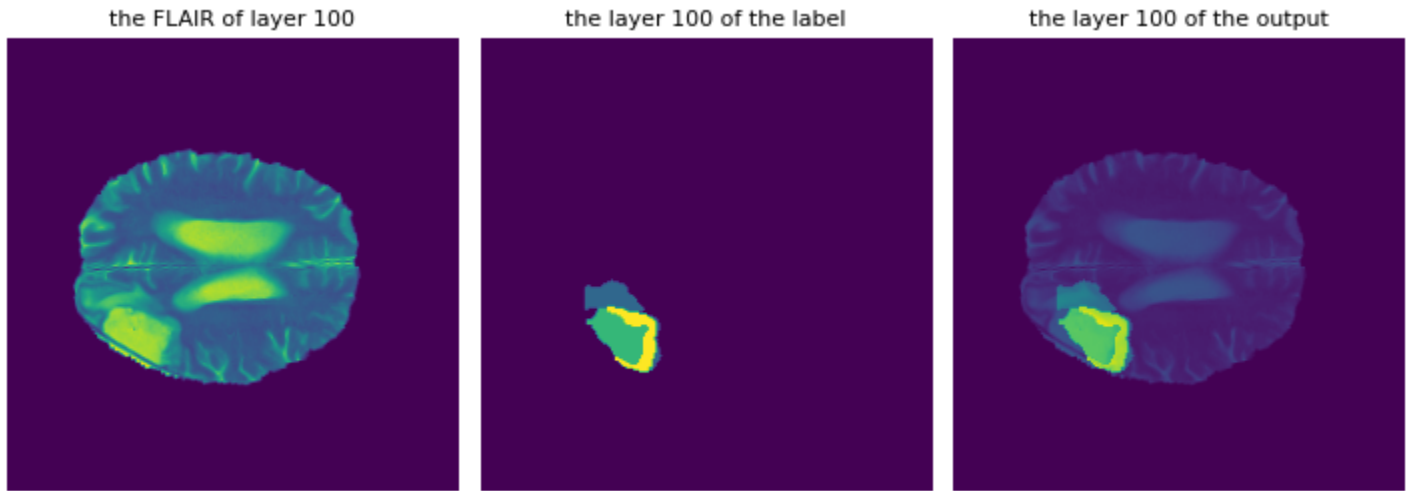
بعد الانتهاء من تدريب الشبكة العصبونية وتقييم أدائها واختبار النموذج الذي سنستخدمه، لابد من اختبار النموذج على مجموعة بيانات لم يشاهدها أثناء تدريبه وهنا تأتي مهمة مجموعة الاختبار test set، حيث تم تجهيزها وإدخالها إلى النموذج وحساب معاملات تقييم الأداء عليها.

تم الحصول على دقة 81.3% بالنسبة لمعامل DSC، أما تابع الخسارة LDice فقيمته 0.013.

جدول 6 نتائج بعض المشاريع المشابهة مصدر الشكل [3]

author	الطريقة method	Dice Score
Human Rater	يدوياً (خبرة وتدريب طبي)	%88
Pereira et al.	شبكة عصبونية تلافيفية CNN مع مرشحات 3x3 filters	%88
Dvorak and Menze	استخدام CNN مع خوارزمية K-means	%83
Hamamci et al	نموذج توليدي (generative model) يستخدم الأتمتة الخلوية	%72

يوضح الشكل 16 الخرج النهائي لأحد الأمثلة مقارنةً مع دخل النموذج.



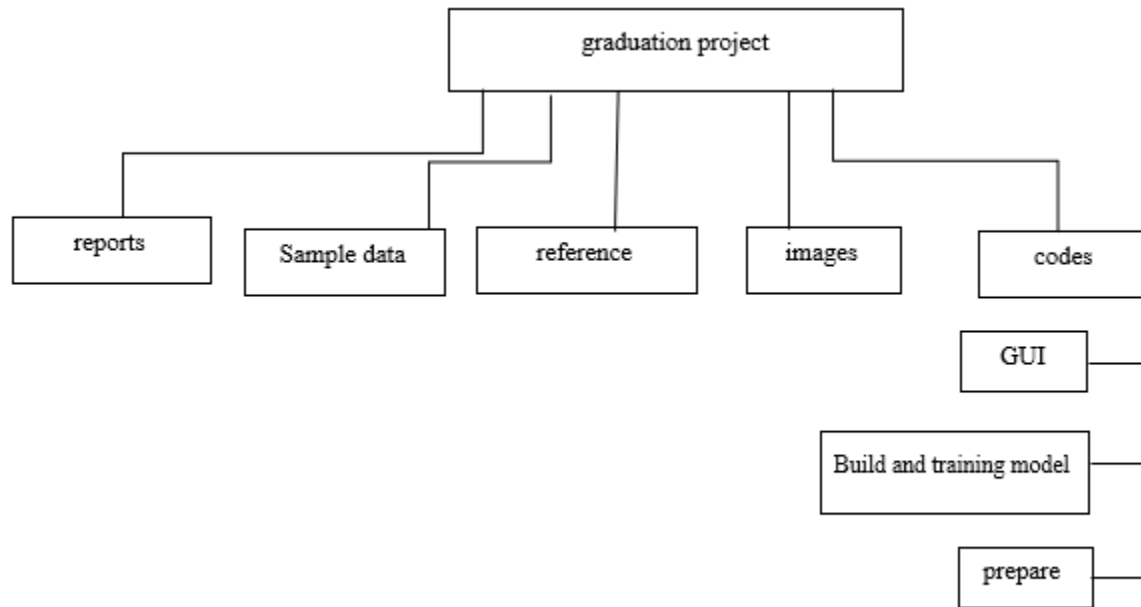
الشكل 15 مثال للخرج النهائي

## الفصل السادس

### بنية توثيق المشروع ودليل الاستخدام

## 1.6- بنية توثيق المشروع ومحتوياته

يوجد في علوم الحاسوب مصطلح يدعى فصل الاهتمامات (SoC) Separation of Concerns وهو مبدأ تصميم يتم فيه تقسيم البرنامج إلى أجزاء منفصلة قدر الإمكان، وبما أننا في هذا العمل نسعى إلى إنشاء نواة نظام قابلة للتطوير المستقبلي، فقد حاولنا جعل بنية توثيق المشروع واضحة من خلال تقسيمه إلى عدة أجزاء لكل منها مهمة منفصلة. في بداية فترة العمل على المشروع، بالاطلاع على مشاريع عديدة في مجال تعلم الآلة والشبكات العصبونية على الإنترنت وتسجيل بعض الملاحظات حول الأفكار التي يمكن الاستفادة منها في إنشاء البنية المناسبة لهذا المشروع. نتيجة لذلك قمنا بتقسيم المشروع إلى الأجزاء الموضحة في الشكل 16، وسنقوم في هذه الفقرة بشرح دول كل جزء بشكل مختصر.



الشكل 16 عرض بنية توثيق المشروع على شكل مخطط شجري

يحتوي المجلد graduation project على جميع ملفات ومجلدات المشروع، وفيما يلي سنستعرض بعض التفاصيل عن المجلدات والملفات الفرعية:

- المجلد codes:

يحتوي هذا المجلد على جميع الأكواد البرمجية للمشروع ويحتوي أيضاً على مجلد GUI الذي يحتوي واجهة المستخدم، أما الملف build\_and\_trainning\_model فهذا الملف موجود بنوعين، ملف بلاهة ipynb والذي يمكن تشغيله باستخدام

jupyter notebook أو باستخدام google colab والنوع الثاني بلاهة py والذي يمكن فتحه والاطلاع عليه بأي محرر يدعم لغة البرمجة python، أما الملف prepare.py فهو يحوي الرموز البرمجية المستخدمة لتقسيم البيانات إلى (training, validation, testing).

- المجلد images:

يحتوي هذا المجلد على الصور التي تم استخراجها أثناء تطوير المشروع، مثل صور الدقة وصور تابع الخسارة وصور الأمثلة وغيرها.

- المجلد reference:

يحتوي هذا المجلد على بعض الأوراق البحثية التي تم الاعتماد عليها في هذا المشروع.

- المجلد sample data:

هذا المجلد فيه بعض الأمثلة على البيانات المستخدمة في تدريب واختبار نموذج الشبكة العصبونية.

- المجلد reports:

يحتوي هذا المجلد على تقرير المشروع بالإضافة إلى العرض التقديمي والمحاضرة التعريفية بالمشروع، ويحتوي أيضاً على ملف logs.csv الذي يوصف تغير قيم معايير التقييم والأداء أثناء تدريب النموذج.

## 2.6- متطلبات التنفيذ

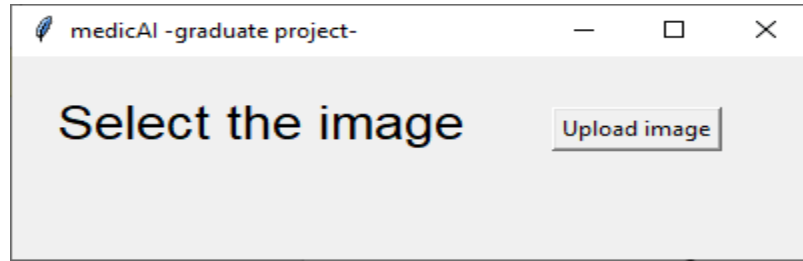
تعد بايثون Python واحدة من لغات البرمجة المدعومة بشكل كبير من الطورين حول العالم، مما يساعد على إطلاق إصدار جديد منها كل بضعة أشهر، في حين أن المكتبات والأدوات تحتاج وقتاً أطول لتحديثها لتتوافق مع الإصدارات الجديدة من بايثون، لذلك قمنا باستخدام إصدار Python 3.7.9، ولذلك يجب استخدام هذا الإصدار عند تنفيذ البرنامج والتأكد من حذف الإصدار الموجود مسبقاً على الجهاز وذلك لضمان عدم حدوث تضارب بينهما.

كذلك نحتاج إلى تثبيت المكتبات والأدوات الضرورية لعمل البرنامج، ولتسهيل عملية التثبيت يجب الدخول إلى مجلد المشروع، وبعدها بكل بساطة يتم كتابة الأمر التالي في محرر الأوامر "pip install -r requirments.txt" حيث يحوي الملف requirments.txt كل المكتبات والأدوات المستخدمة في المشروع، ولكن يجب تشغيل virtual private network (VPN) أثناء التثبيت لضمان عدم حدوث أي مشاكل

ملاحظة: تم استخدام الإصدار 2.6.0 من إطار العمل TensorFlow وهذا الإصدار لا يعمل على معالجات الحواسيب القديمة، لذلك يجب اختبار النظام على حاسوب حديث نسبياً لكي يعمل.

### 3.6- تنفيذ البرنامج

إن الهدف من بناء البرنامج البسيط هو اختبار أداء النموذج في تحديد الأورام في الدماغ بشكل مباشر بدلاً من الاكتفاء بالنتيجة التي حصلنا عليها من مجموعة الاختبار test set. استخدمنا tkinter لتصميم الواجهة بلغة بايثون، ولتشغيل واجهة المستخدم يجب فتح محرر الأوامر cmd في مسار المجلد GUI ثم يجب تنفيذ الأمر التالي في محرر الأوامر cmd "python main.py"، لتظهر الواجهة التالية:

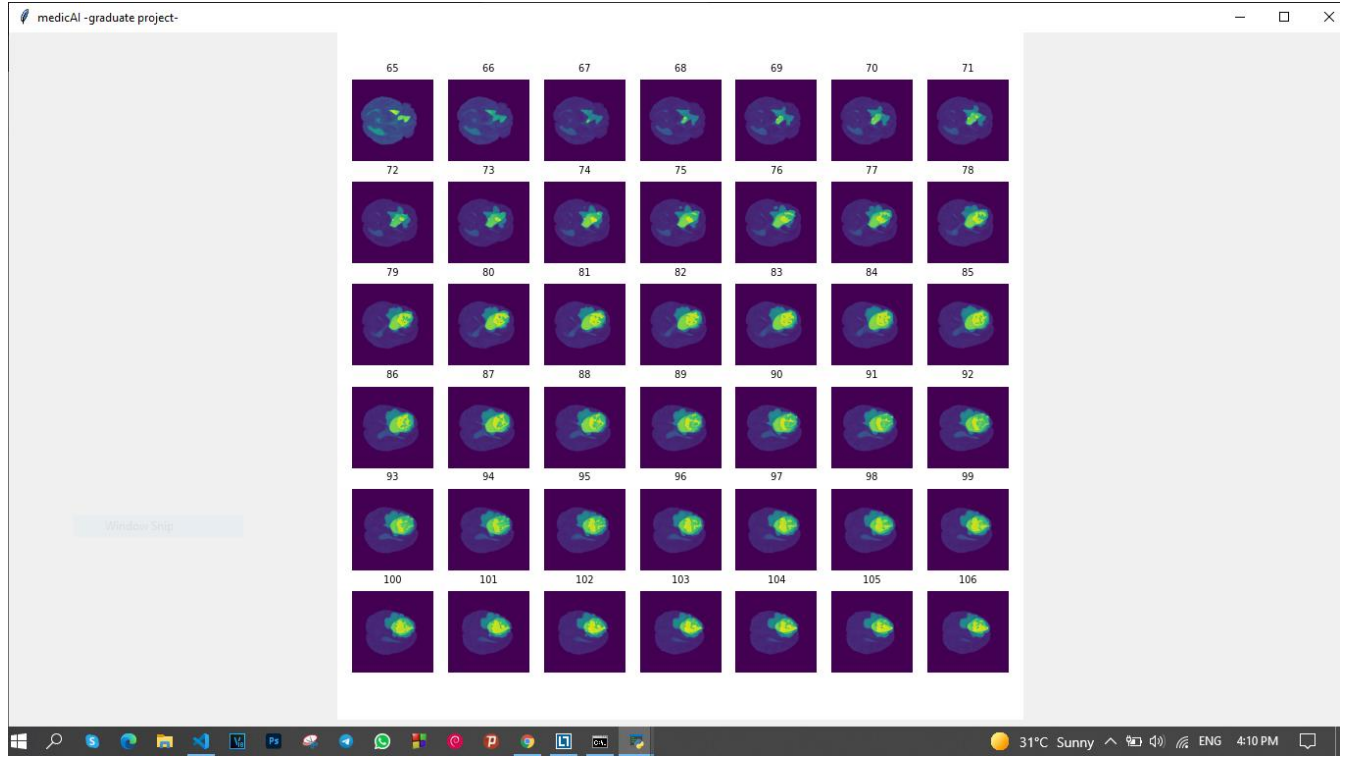


الشكل 17 واجهة اختيار الصورة

ثم بعد اختيار الصورة يقوم النموذج بتحديد الورم فيها ونقوم بإظهار الطبقات (65 to 106) حيث وجدنا أن هذه الطبقات هي أكثر الطبقات احتواءً على الورم، ولم نقم بعرض كل طبقات الصور لتسريع عمل البرنامج.



## الخاتمة والآفاق المستقبلية



الشكل 18 خرج البرنامج النهائي

## الخاتمة والآفاق المستقبلية

قمنا في هذا المشروع ببناء نظام لتحديد الورم في الدماغ باستخدام صور الرنين المغناطيسي MRI، وذلك من خلال تطبيق أحدث خوارزميات وتقنيات التعلم العميق والشبكات العصبونية، حيث تم بناء وتدريب نموذج شبكة عصبونية باستخدام مجموعة بيانات مجهزة لهذه المهمة، وتم اختبار النموذج المدرب باستخدام الطرق المحددة لهذا النوع من المشاريع (DSC, Sensitivity, Specificity)، وتم بعدها بناء واجهة مستخدم لاختبار النموذج بكل سهولة.

إن بعض الصعوبات التي واجهتنا في هذا المشروع هي عدم وجود مقرر حول الشبكات العصبونية ضمن برنامج الإجازة في هندسة نظم المعلومات ISE، لذلك قمنا بالتعلم الذاتي لتلك المقررات من عدة مصادر أكاديمية متاحة على الانترنت، بالإضافة لبعض المشاكل في الإمكانيات وخاصةً مع ظروف الكهرباء وغيرها من المشاكل الحالية التي يواجهها الطالب.

من الآفاق المستقبلية لهذا المشروع محاولة تطوير النتائج التي حصلنا عليها عن طريق تعديل نموذج الشبكة العصبونية، أو استخدام طرق واستراتيجيات أخرى، والمشاركة في المسابقات العالمية التي تعنى بمثل هذه المشاريع.

### الخاتمة والأفاق المستقبلية

كما سنسعى لبناء موقع الكتروني لاستخدام النموذج بسهولة ويسر، وجعله نقطة انطلاق لربط مجال الذكاء الاصطناعي بمجال الرعاية الصحية في بلدنا.

## المراجع

- [1] E. C. Holland, (2001) “Progenitor cells and glioma formation”.
- [2] H. Ohgaki and P. Kleihues, (2005) “Population-based studies on incidence, survival rates, and genetic alterations in astrocytic and oligodendroglial gliomas”.
- [3] Ali Isin, Cem Direkoglu, Melike Sah (2016) “Review of MRI-based brain tumor image segmentation using deep learning methods”.
- [4] Zeynettin Akkus, Alfia Galimzianova, Assaf Hoogi, Daniel L. Rubin, Bradley J. Erickson, (2017) “Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions” 2017
- [5] <http://medicaldecathlon.com/>
- [6] Amber L. Simpson, and others (2019) “A large annotated medical image dataset for the development and evaluation of segmentation algorithms”.
- [7] François Chollet, (2017) “Deep Learning with Python Book”
- [8] Hamamci A, et al. (2011) “Tumor-Cut: segmentation of brain tumors on contrast enhanced MR images for radiosurgery applications”.
- [9] “Deep Learning.” [Online]. Available: <https://www.deeplearningbook.org/>.
- [10] “Convolutional Neural Networks” [Online]. Available: [ibm.com/cloud/learn/convolutional-neural-networks](https://ibm.com/cloud/learn/convolutional-neural-networks).
- [11] “ai-for-medical-diagnosis | Coursera” [Online]. Available: [coursera.org/learn/ai-for-medical-diagnosis](https://coursera.org/learn/ai-for-medical-diagnosis).
- [12] Philipp Fischer, Thomas Brox, (2015) “U-Net: Convolutional Networks for Biomedical Image Segmentation”.
- [13] Özgün Çiçek, (2016) “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation”.
- [14] S. Raschka, J. Patterson, and C. Nolet, (2020) “Machine Learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence.”
- [15] “deep learning specialization | Coursera” [Online]. Available: [coursera.org/specializations/deep-learning](https://coursera.org/specializations/deep-learning)
- [16] Reuben R Shamir, (2018) “Continuous Dice Coefficient: a Method for Evaluating Probabilistic Segmentations”