الجمهوريـــة الجزائريـــــة الديمقراطيـــة الشعبيـــة
وزارة التعليم العالي والبحث العلمي

جامعـــة حمـــة لخضـــر الـــــوادي
كلية العلوم الدقيقة
قسم الإعلام الآلي

مذكرة نهاية التخرج
تدخل ضمن متطلبات الحصول على شهادة

ليســانس أكاديمي

الميدان: رياضيات و إعلام آلي
الشعبة: الإعلام الآلي
التخصص : أنظمة معلوماتية

الموضوع

التعرف على اشارات المرور باستخدام الشبكات العصبية

من إنجاز الطلبة:
- دقعة منار
- عرجون العفراء

من اقتراح وتأطير الأستاذ:
بن علي عبد الكامل

نوقشت يوم 2020/10/01 أمام اللجنة المكونة من الأساتذة:

الاسم واللقب	الجامعة	الصفة
لجدل ابراهيم	جامعة الشهيد حمه لخضر – الوادي	رئيسا
بن علي عبد الكامل	جامعة الشهيد حمه لخضر – الوادي	مشرفا ومقررا
الزايز فوزي	جامعة الشهيد حمه لخضر – الوادي	مناقشا

السنة الجامعية: 2020/2019

الإهداء

إلى من لايمكن للكلمات أن توفى حقهما

إلى من لايمكن للأرقام أن تحصى فضائلهما

إلى والدي العزيزين أدامهما الله لي

و ارحم ميتهم "عرجون أحمد "

إلى اخوتي سندي و دعامتي دقعة "العيد و فراس صدر الدين "

و الاخوة عرجون "ياسر و معاذ "

و إلى اختي حبيبتي "دقعة مواهب "

و إلى أختي التي لم تلدها أمي " قرون سوسن حورية "

و إلى الأخوات عرجون "شيماء و نجلاء و جهينة و رحاب "

إلى رفقاء الدرب و الدراسة

إلى كل من يسعى في طلب العلم

إلى كل من سقط من قلمي سهواء

.....أهدي هذا العمل المتواضع

شكر و تقدير

الحمد لله الذي أنار لنا درب العلم و معرفة و أعاننا على أداء هذا الواجب و وفقنا إلى إنجاز هذا العمل .

نتوجه بجزيل الشكر و الإمتنان إلى كل من ساعدنا من قريب أو من بعيد على إنجاز هذا العمل في تضليل ما واجهناه من صعوبات ، و نخص بالذكر الأستاذ المشرف الدكتور بن علي عبد الكامل الذي لم يبخل علينا بتوجيهاته و نصائحه القيمة التي كانت عونا لنا في إتمام هذا البحث .

و لا يفوتنا أن نشكر:

- ☒ كل الأساتذة بقسم الإعلام الآلي لكلية العلوم الدقيقة _جامهة الشهيد حمة لخضر _ الوادي
- ☒ لكل من الطالب س . يونس و س. صلاح الدين
- ☒ و الى كل زملاء الدراسة

ملخص:

منذ ظهور لوحات المرور وانتشارها في جميع الدول أصبح لا بد منها في الطرقات لما تعود به من فوائد كمقوماتها المسبقة عن الطرق وتقليل الأخطار، نظرا لحاجتنا الماسة الملاحظة لوحات المرور والطرقات كذال الحاجة لتطوير السيارات وجعلها سيارات ذكية هذا ما دفعنا إلى إنشاء نموذج شبكات عصيبة إلتفافية في مجال التعلم العميق للتعرف صور لوحات المرور وتصنيفها، وذلك اعتمادا على بيئة البرمجة و عدة من ال Packages هم Pandas و TensorFlow و OpenCv و NumPy و Matplotlibو NumPy و SkLearn ، وجزء صغير من GTSRB Dataset وهذا عن طريق ادخالها في خوارزمية التعلم العميق خوارزمية (CNN).

الكلمات المفتاحية: التعلم العميق ، الشبكة العصبية التلافيفية ، تصنيف إشارات المرور ، التعلم تحت الإشراف.

Summary:

Since the emergence of traffic signs and their spread in all countries, they have become essential in the roads due to the benefits they benefit from, such as their preconditions for roads and reducing risks, due to our urgent need to observe traffic and road signs as well as the need to develop cars and make them smart cars. This is what prompted us to create a hard bypass network model in the field of deep learning To identify and classify traffic billboards, depending on the programming environment, and several of the packages are Pandas, TensorFlow, OpenCv, NumPy, Matplotlib, NumPy, SkLearn, and a small part of (GTSRB Dataset) and this by inserting it into the deep learning algorithm, the (CNN) algorithm.

Keywords: deep learning, convolutional neural network, traffic signs classification, supervised learning.

Résumé:

Depuis l'émergence des panneaux de signalisation et leur diffusion dans tous les pays, ils sont devenus incontournables sur les routes en raison des avantages dont ils bénéficient, tels que leurs conditions préalables pour les routes et la réduction des risques, compte tenu de notre besoin urgent d'observer les panneaux de signalisation et les routes ainsi que la nécessité de développer des voitures et d'en faire des voitures intelligentes. C'est ce qui nous a poussé à créer un modèle de réseaux de contournement dur dans le domaine de l'apprentissage Il est également utilisé pour reconnaître et classer les images des panneaux de signalisation, en fonction de l'environnement de programmation et de leurs différents packages

Pandas, TensorFlow, OpenCv, NumPy, Matplotlib, NumPy, SkLearn, et une petite partie de GTSRB Dataset et ce en les insérant dans l'algorithme de deep learning, l'algorithme CNN.

Mots clés: apprentissage profond, réseau de neurones convolutifs, classification des panneaux de signalisation, apprentissage supervisé.

الفهرس:

ملخص: 4

الفهرس: 6

قائمة الصور: 8

قائمة الاشكال والجداول: 10

مقدمة عامة: 11

I. الفصل الاول: تقديم عام.

مقدمة: 14

1. تعريف الذكاء الاصطناعي : 14
2. تعلم الألة : 15
   - 2.1. تعريف: 15
   - 2.2. مبدأ تعلم الآلة: 17
   - 2.3. العلاقة بين تعلم الآلة والذكاء الاصطناعي وجمع البيانات: 17
   - 2.4. أهمية تعلم الآلة: 18
   - 2.5.استخدامات تعلم الالة : 18
   - 2.6. أنواع مشاكل ومهام تعلم الالة : 19
3. شبكات العصبية الاصطناعية (بالإنجليزية: ARTIFICIAL NEURAL NETWORK ANN): 21
   - 3.1. تعريف: 21
   - 3.2. تاريخ الشبكات العصبية الاصطناعية: 21
   - 3.3. كيفية عمل الشبكات العصبية : 22
   - 3.4. كيفية تعلم الشبكات العصبية : 22
   - 3.5. أنواع الشبكات العصبية: 23

II. الفصل الثاني: المفاهيم الاساسية في بناء نماذج التصنيف.

1. أنواع ال LAYERS التي تتكون منها النماذج: 26
   - 1.1. تعريف ال (Convolution Layer): 26
   - 1.2. آلية تطبيق ال (Filter Of Convolution) على الصورة: 27
   - 1.3. خصائص ال (Convolution): 33
   - 1.4. Relu function : 34
   - 1.5. (Pooling Layer): 36
   - 1.6. (Flattening Layer): 39
   - 1.7. (Fully Connected Layers) أو (Dense Layers): 40
   - 1.8. (Dropout): 43

2. (DATASET): 46
2.1. (Data Normalisation): 47
2.2. (One-Hot Encoding): 49
3. آلية تدريب النماذج : 50
3.1. مفاهيم أساسية لتطبيق عملية ال (Back propagation) 50
3.2. (Back Propagation): 54

III. الفصل الثالث: التصميم و الإنجاز.

1. البايثون (PYTHON): 57
1.1. استخدامات البايثون : 57
1.2. ال (Packages) المتوفرة على (Python) التي تمكن من إنشاء ال (Deep Learning Models): 57
2. مرحلة التطبيق أو التنفيذ (IMPLEMENTATON): 59
خاتمة عامة: 68
قائمة المصادر والمراجع: 69

قائمة الصور:

الصورة رقم 01: رسم توضيحي لعمل الشبكات العصبية الالتفافية CNN [53].....23
الصورة رقم 02: توضح إختلاف أبعاد الفيلتر الخاصة بعملية الإلتفافية كذلك إختلاف خطوات تنقلها. م[4] [1].....27
الصورة رقم 03: توضح أجزاء من تطبيق الإلتفافية على صورة رمادية م[6].....28
الصورة رقم 04: توضح بقية عملية الإلتفافية بعد تطبيق ال PADDING ونتيجتها م[6]. 29
الصورة رقم 05: توضح خطوات عملية الإلتفافية على صورة ملونة [6].....31
الصورة رقم 06: توضح إختلاف نتائج تطبيق الإلتفافية على نفس الصورة حسب إختلاف الفيلتر م[2].....32
الصورة رقم 07: الصورة الأصلية قبل تطبيق عملية الإلتفافية عليها م[5]. 32
الصورة رقم 08: توضح نتائج تطبيق عدة فلترات خاصة بعدة طبقات إلتفافية من مستويات مختلفة فالنموذج م[5]. ..33
الصورة رقم 09: توضح صيغة ومنحنى الدالة RELU م [13].....35
الصورة رقم 10: توضح تطبيق ال RELU على نتائج الإلتفافية م[16]. 35
الصورة رقم 11:توضح تطبيق ال MAX POOLING على مخرجات عملية الإلتفافية [18]. 37
الصورة رقم 12: نتائج تطبيق ال (CONVOLUTION) ثم (RELU) ثم ال (MAX POOLING) على صورة معينة م[10]. .38
الصورة رقم 12: توضح تطبيق عملية ال FLATTENING على FEATURE MAP م [15]. 39
الصورة رقم 13: توضح شبكة عصبية من عدة طبقات [17].....40
الصورة رقم 14:توضح شبكة عصبية من عدة طبقات مرتبطة بالمدخلات [7].....41
الصورة رقم 15: توضح كيفية حساب قيم كل عصب من الشبكة العصبية [18]. 41
الصورة رقم 16: توضح أن قيمة العصبون هي قيمة خطية [11]. 41
الصورة رقم 17: توضح مكونات العصبون وقيم كل من مدخلاته ومخرجاته [9]. 42
الصورة رقم 18: توضح طريقة عمل ال SOFTMAX ACTIVATION FUNCTION م[14]. 42
الصورة رقم 19: توضح توقف بعض الأعصاب في الشبكة بسبب عمل DROPOUT م[19]. 44
الصورة رقم 20: توضح كيفية إضافة ال DROPOUT LAYER برمجيا م[19].....44
الصورة رقم 21: توضح تأثير ال DROPOUT على دقة مجموعة التحقق م[19].....45
الصورة رقم 22: توضح ال (DATASET) على شكل جدول م[23] 46
الصورة رقم 23: توضح ال (DATASET) على شكل صور م[23].....46
الصورة رقم 24: توضح التسمية الفئوية لحيوانين من بين 6 فئات حيوانيةم[26] 50
الصورة رقم 25: توضح صيغة مجموع قيم دالة الخسارة)ِCROSS ENTROPY) لمخرجات ال (NEURAL NETWORK)
بدلالة التسمية الفئوية للهدف (ONE HOT ENCODING LABEL) و (SOFTMAX ACTIVATION FUNCTION) م[40] 51
الصورة رقم 26: توضح مدى تطور ال (WEIGHT) بدلالة ال (LOSS FUCNTION) عند إستخدام قيمة (LEARNING RATE)
صغيرةم[32] 52

الصورة رقم 27: توضح مدى تطور ال (WEIGHT) بدلالة ال (LOSS FUCNTION) عند إستخدام قيمة (LEARNING RATE) كبيرةم[32].....52
الصورة رقم 28: توضح كيفية تغير قيمة ال (WEIGHT) إعتمادا على قيم الميل وإتجاهه في منحنى تغير ال (WEIGHT) بدلالة تغير ال (LOSS FUCNTION) م[34].....53
الصورة رقم 29: توضح كيفية تغير قيمة ال (WEIGHT) إعتمادا على قيم الميل وإتجاهه في منحنى تغير ال (WEIGHT) بدلالة تغير ال (LOSS FUCNTION) م[35].....53
الصورة رقم 30: توضح العمليات الحسابية لخوارزمية ال (BACK PROPAGATION) لحساب قيمة ال (WEIGHT) الجديدة W^+(K,T). م[36] 54
الصورة رقم 31: توضح عمل ال IMPORT لجميع المكاتب اللازمة.....59
الصورة رقم 32: توضح ملفات الداتاسات 60
الصورة رقم 33: توضح لوحات المرور المتوفرة على الداتاسات.....60
الصورة رقم 34: توضح محتوى ملف ال TEST.CSV الخاص بالداتاسات 61
الصورة رقم 35: توضح محتوى ملف ال TRAIN.CSV الخاص بالداتاسات 61
الصورة رقم 36: توضح محتوى مجلد ال TRAIN 62
الصورة رقم 37: توضح محتوى المجلد TEST الخاص بالداتاسات.....62
الصورة رقم 38: توضيح هيكلة النموذج.....63
الصورة رقم 39: توضح دالة تهيئة مجموعة البيانات قبل المعالجة.....63
الصورة رقم 40: توضح تقسيم مجموعة التدريب للحصول على مجموعة تحقق.....64
الصورة رقم 41: توضح تجديد قيم ال HYPERPARAMETRS و دالة ال OPTIMIZER ثم عمل COMPIL للنموذج.....64
الصورة رقم 42: توضح الوظيفة FIT الخاصة بتدريب النموذج و متغيراتها.....64
الصورة رقم 43: توضح بداية عملية تدريب النموذج 65
الصورة رقم 44: توضح الوظيفة EVALUATE الخاصة بتقييم النموذج و متغيراتها 65
الصورة رقم 45: توضح النتائج النهائية للتدريب و تقييم النموذج 65
الصورة رقم 46: توضح وظيفة رسم مخططات التدريب و التحقق.....66
الصورة رقم 47: توضح مخرجات محفوظات النموذج 66
الصورة رقم 48: توضح تزايد منحنيات الدقة لمجموعة التدريب و التحقق.....67
الصورة رقم 49: توضح تناقص منحنيات دالة الخسارة لمجموعة التدريب و التحقق 67

قائمة الاشكال والجداول:

الشكل1: رسم توضيحي لمكونات الأساسية لتعلم الآلة.م[50].....16

الشكل2: مخطط يوضح أنواع و مهام تعلم الآلة. م[51].....20

الشكل03: جدول يوضح دقة (SVM MODEL) و (BPANN MODEL) على البيانات الأصلية و على البيانات بتقنيات (NORMALIZATION) مختلفةم[24].....48

الشكل رقم 04: أعمدة بيانية توضح مقارنة مختلف تقنيات ال (NORMALIZATION) إعتمادا على أداء (SVM MODE) و (BPANN MODEL) م[24] 49

مقدمة عامة:

تشترك جميع الكائنات الحية في وجود منظومات عصبية (SYSTEMS NEURAL) تمكنها من التعامل والتفاعل مع البيئة المحيطة بها كما تساعدها في التحكم في العمليات الحيوية اللازمة لاستمرار الحياة لهذه الكائنات. وتختلف المنظومات العصبية من كائن الى اخر حيث تكون بسيطة التركيب وطبيعية العمل في أولية ذات التركيب الخلوي البسيط ومعقدة التركيب وطبيعية العمل في الكائنات الاكثر علوا مثل الإنسان. وتعتبر المنظومة العصبية للإنسان اعقد المنظومات العصبية على الاطلاق والتي يتركز معظمها في المخ البشري الذي يتميز بطبيعة عمل ادت الى تفوق الإنسان على سائر المخلوقات الأخرى في قدرات التفهم والتعرف على الأشكال والرموز والتعلم والتحدث والتذكر والإدراك والسيطرة الدقيقة على الجهاز الحركي وما الى ذلك من العديد من الصفات والقدرات التي ال يستطيع اي كائن اخر غير الإنسان الى الوصول اليها، حيث حاول العلماء تقليد المنضومة العصبية الانسانية وذلك لفتح المجال للتكنولوجيا لكي تساعد الانسان في حياته اليومية بفعالية، فظهر مصطلح الذكاء الاصطناعي في الساحة التكنولوجية.

الذكاء الاصطناعي هو سلوك وخصائص معينة تتسم بها البرامج الحاسوبية، تجعلها تحاكي القدرات الذهنية البشرية وأنماط عملها. من أهم هذه الخاصيات القدرة على التعلم والاستنتاج ورد الفعل على أوضاع لم تبرمج في الآلة، استخدم الذكاء الاصطناعي في الكثير من المجالات ولاقى ابهارا كبير وأحد هذه المجالات هو التعرف على الصور حيث استخدم العلماء هذا المجال في التعرف على ومعالجة صور الوجه الانساني كما استخدم على التعرف على الاشياء وذلك للأهمية التي يقدمها هذا المجال، وأحد الطرق المستعملة في هذا المجال هي تعليم الالة للقيام بهذه العمليات.

تعلم الآلة هو أحد فروع الذكاء الاصطناعي التي تهتم بتصميم وتطوير خوارزميات وتقنيات تسمح للحواسيب بامتلاك خاصية "التعلم". بشكل عام هناك مستويين من التعلم: الاستقرائي والاستنتاجي. يقوم الاستقرائي باستنتاج قواعد وأحكام عامة من البيانات الضخمة. المهمة الأساسية للتعلم الآلي هو استخراج معلومات قيمة من البيانات، حيث يستخدم في تعليم الذكاء الاصطناعي التعرف على الصور وتحليلها وتصنيفها، يستخدم التعلم الآلي في العديد من المجالات من الهندسة إلى الطب، فمثلا يمكن للحواسيب تحديد مكان الاورام الخبيثة اليوم من خلال الصور المقطعية للمريض حيث تعتبر نقلة نوعية في تشخيص الامراض.

تقوم كبرى الشركات اليوم على غرار Google و Amazon وAppel... ببناء نظام القيادة الذاتية في السيارات، حيث تكون السيارة قادرة على التعرف المسافات بينها وبين السيارات الاخرى والتعرف على الاشارات المرورية باستخدام التعرف على الصور وتحليلها حيث يقومون باستخدام الذكاء

الاصطناعي للقيام بذلك، ويقومون باتجارب وتعليم الالة الاشارات المرورية واختبارها على نطاق واسع، وهذا ماسنحاول القيام به في هذا العمل وهو برنامج يقوم بالتعرف على على الاشارات المرورية، حيث تكمن اهمية هذا العمل في انه من اهم المكونات في نظام القيادة الذاتية.

فكيف سننمذج ذلك تطبيقيا؟ وماهي المراحل للوصول الى النتائج النهائية؟

سنتطرق في هذه المذكرة الى تعلم الالة وكيفية تطبيق ذلك الى الواقع عن طريق برنامج التعرف على اشارات المرور، وذلك في ثلاث فصول محورية:

الفصل الاول: تقديم عام.

الفصل الثاني: المفاهيم الاساسية في بناء نماذج التصنيف.

الفصل الثالث: التصميم و الإنجاز.

الفصل الاول: تقديم عام

I. الفصل الاول: تقديم عام

مقدمة:

يسلط هذا الفصل الضوء على مفهوم الذكاء الصناعي وتعلم الالة ومبدأها، وذالك بتقديم تعرف لكل منهما، وذكر اهم انواعهما مع تقديم العلاقة بينهما وما مدى اهمية ذلك.

يحتوي الفص على النقاط التالية:

تعريف الذكاء الاصطناعي.

تعلم الالة

الشبكات العصبية الاصطناعية (Artificial Neural Network ANN).

1. تعريف الذكاء الاصطناعي :

هو دراسة السلوك الذكي في البشر و الحيوانات و الآلات كما أنه يمثل محاولة لإيجاد السبل التي يمكن بها ادخال مثل هذا الساوك على الآلات الإصطناعية علاوة على ما سبق ، يعد الذكاء الصطناعية من أصعب الموضوعات و أكثرها إثارة للجدل للبشرية بأسرها .

_ و قد لا تتضح صعوبة هذا الموضوع في بادئ الأمر و لكن يتم إدارتها بعد ذلك بشكل كبير، كما يهدف الذكاء الاصطناعي الوصول الى الأنظمة الذكية التي يمكن بها التصرف بالنحو الذي يتصرف به البشر من حيث التعلم و التفهم .

ومن أهم أنواعه :

الذكاء الاصطناعي المحدود أو الضيق: (بالإنجليزيّة: Weak AI or Narrow AI) يعتبرأحد أنواع الذكاء الاصطناعي التي تستطيع القيام بمهام مُحددة وواضحة، كالسيارات ذاتيّة القيادة، أو حتى برامج التعرُف على الكلام أو الصورو.

الذكاء الاصطناعي العام: (بالإنجليزيّة: General AI)، وهو النوع الذي يُمكن أن يَعمل بقدرة تُشابه قدرة الإنسان من حيث التفكير، إذ يُركز على جعل الآلة قادرة على التفكير والتخطيط من تلقاء نفسها وبشكل مُشابه للتفكير البشري.

الذكاء الاصطناعي الفائق: يُعتبر الذكاء الاصطناعي الفائق (بالإنجليزيّة: Super AI) النوع الذي قد يفوق مستوى ذكاء البشر، والذي يستطيع القيام بالمهام بشكل أفضل مما يقوم به الإنسان المُتخصص وذو المعرفة.

الآلات التفاعليّة: يُعتبر الذكاء الاصطناعي الخاص بالآلات التفاعليّة (بالإنجليزيّة: Reactive Machines) أبسط أنواع الذكاء الاصطناعي؛ إذ يفتقر هذا النوع إلى القدرة على التعلُم من الخبرات السابقة أو التجارب الماضيّة لتطوير الأعمال المستقبليّة، فهو يتفاعل مع التجارب الحاليّة لإخراجها بأفضل شكل مُمكن.

الذاكرة المحدودة: يستطيع نوع الذكاء الاصطناعي ذو الذاكرة المحدودة (بالإنجليزيّة: Limited Memory) تخزين بيانات التجارب السابقة لفترة زمنيّة محدودة، ويُعد نظام القيادة الذاتيّة.

نظريّة العقل: (بالإنجليزيّة: Theory of Mind)، يُعنى هذا النوع من الذكاء الاصطناعي بفهم الآلة للمشاعر الإنسانيّة، والتفاعل مع الأشخاص والتواصل معهم.

الإدراك الـذّاتي: يُعتبر نوع الإدراك الذاتي (بالإنجليزيّة: Self-Awareness) من التوقعات المستقبلية التي يصبو إليها علم الذكاء الاصطناعي، بحيث يتكون لدى الآلات وعي ذاتي ومشاعر خاصة. م [52]

2. تعلم الألة :

2.1. تعريف:

هو تصميم برامج للتعلم من الخبرات السابقة،و هو اضافة مقدرة التعلم لأجهزة الحاسوب حيث يتم تصميم وتطوير الخوارزميات والتقنيات التي تمكن الحاسب من امتلاك خاصية "التعلم" ، المهمة الاساسية للتعلم الآلي هو استخراج معلومات قيمة من بيانات تدريبية ثم يستفيد منها في بناء نموذج قادرعلى التنبوء بشكل االبيانات الجديدة .

ويشار إلى أن مصطلح **تعلم الآلة** قد ظهر بإيعازٍ من رائد الذكاء الاصطناعي آرثر صامويل في سنة 1959 ضمن نطاقِ عمل مختبراتIBM ، ومن الجديرِ بالذكر فإن الآلة في هذه الحالة يجب أن تعتمد على تحليل البيانات المدخلة إليها مسبقًا لمواجهة الأوامر والمهام المطلوبةمنها، فيكون دور العنصر البشري ضئيلًا جدًا في نهاية المطاف

كما سيقع على عاتق الآلة مسؤولية اتخاذ القرار عند الحاجة لذلك، وتحديد ما يجب تنفيذه من مهامٍ ومتى وكيف ولماذا دون أي مساعدةٍ بشريةٍ إطلاقًا، إذ سيسهم ذلك حتمًا في إنجاز المهام بأسرع وقتٍ ممكن مقارنةً مع الوقت الذي يستهلكه البشر لإنجاز المهام.

الشكل1: رسم توضيحي لمكونات الأساسية لتعلم الآلة.م[50]

2.2. مبدأ تعلم الآلة:

قد يبدو الأمر معقدًا في البداية حول كيفية تعلم الآلة ومدى إمكانية تحقيق ذلك فعليًا؛ إلا أن ذلك ليس مستحيلًا إطلاقًا! فقد ظهر العديد من التطبيقات العملية للذكاء الاصطناعي القائم على مبدأ تعلم الآلة، ومن أشهرها الروبوت صوفيا. أما فيما يتعلق بمبدأ العمل فإن الخوارزميات هي الأساس في تطبيق التعلم الآلي، حيث تتألف هذه الخوارزميات من سلسلةٍ من الأوامر والتعليمات والإرشادات الضرورية لتوجيه الآلة أو الحاسوب للكيفية التي يجب تنفيذ المهام بها، إذ تؤدي الخوارزميات دور العقل المدبر في الآلة لما تقوم به من استقطابٍ للبيانات وتجميعها وتحليلها والاعتماد أخيرًا على البيانات المُحللة ليتم تحديد الكيفية الواجب تنفيذ المهمة بها.

تعتمد الخوارزميات المستخدمة في تعلم الآلة على مجموعةٍ من النماذج الرسومية وأدوات القرار كشجرة القرار ومعالجة اللغات الطبيعية والشبكات العصبية الاصطناعية للقيام بمهمة أتمتة البيانات المُحللة والمعالجة؛ وبالتالي تحفيز الآلة على اتخاذ القرار والقيام بالمهام الموكولة لها بكل سهولةٍ. ولا بد من الإشارة إلى أن الشبكات العصبية الاصطناعية المستخدمة في تعلم الآلة تؤدي دورًا في غاية الأهمية يضاهي دور الأعصاب وشبكاتها في جسم الإنسان البشري ودماغه، وانطلاقًا من الدور المعقد الذي تقوم به الخوارزميات وأدواتها فقد ظهرت الحاجة الملحة للإتيان بما يُعرف بالتعلم المتعمق (Deep Learning).

2.3. العلاقة بين تعلم الآلة والذكاء الاصطناعي وجمع البيانات:

يمكن رسم أبعاد العلاقة بين الذكاء الاصطناعي وتعلم الآلة وعملية التنقيب عن البيانات على شكلِ 3 مظلاتٍ متفاوتة الحجم؛ حيث يعتبر علم الذكاء الاصطناعي بمثابةِ المظلة الأكبر التي تضم تحتها مباشرةً مظلة تعلم الآلة، بينما تحتضن الأخيرة مظلة التنقيب عن البيانات واستخلاصها، وبذلك فإنه يُستخلص من الحديث أن الذكاء الاصطناعي يتربع على قمة الهرم ويتمثل دوره بالسعي لبرمجة الآلات والحواسيب بالاعتمادِ على عدةِ طرقٍ ليضاهي الذكاء البشري في نهاية المطاف معتمدًا على الأساليب المختلفة في اتخاذ القرار والتفكير. أما Machine Learning فإنه يمثل دور الطبقة التي تلي قمة الهرم، ويتمل دوره في تنفيذ مهمة الأتمتة والبرمجة وتعليم الآلات على استخدام البيانات المتوفرة لديها في اتخاذ القرار، وهنا يظهر دور Data Mining بالبحثِ عن البيانات ذات العلاقة وتوظيفها في أداء المهمة.

2.4. أهمية تعلم الآلة:

تمكن فرع تعلم الآلة من اكتساب أهمية بالغة في العصر الحديث، وتحديدًا بعد التطورات الهائلة التي طرأت على الذكاء الاصطناعي وخوارزمياته، وتتمثل أهميته فيما يلي م [50]:

- توفير كمٍ أكبر من البيانات الضرورية لاتخاذ القرار.
- القدرة على تخزين البيانات بأكبر قدرٍ ممكنٍ.
- معالجة البيانات حاسوبيًا يعتبر أقل تكلفةٍ ماديةٍ من توظيف الأيدي البشرية.
- تحليل أكبر قدر من البيانات بمختلف مستوياتها سواءً كانت بسيطةً أو معقدةً.
- ضمان الحصول على نتائجٍ وقراراتٍ أكثر دقةً وبأسرع وقتٍ.
- تمكين المنشآت والمنظمات من رصد الفرص الملائمة لتحقيق الأرباح وتفادي المخاطر المجهولة لها.
- المساعدة في اختيار القرار الأمثل من بين مجموعةٍ من البدائل المتاحة

2.5.استخدامات تعلم الالة :

من الطبيعي أن يتوارد إلى الأذهان تساؤلاتٍ تتمحور حول الجهات المهتمة في التعلم الآلي، حيث يمكن توظيف ذلك في العديد من مجالات الحياة، ومن أهم الجهات المهتمة بالتعلم الآلي:

- الحكومات
- الخدمات المالية والمصرفية
- قطاع الرعاية الصحية
- التسويق والمبيعات
- التنقيب عن النفط والغاز

2.6. أنواع مشاكل ومهام تعلم الالة :

وعادة ما تُصنّف مهام التعلم الآلي إلى فئتين، اعتمادًا على ما إذا كانت هناك "إشارة" تعليمية أو "تغذية راجعة" من عملية التعلّم، في نظام التعلّم المعني.

التعلّم الخاضع للإشراف: حيث ينشأ النظام عن وظيفة من بيانات التدريب يتم إعطاء الحاسوب أمثلة من المُدخلات والمخرجات المتوقعة منها(لايجاد علاقة تربط بين المدخلات و المخرجات)، والغرض من هذا النوع من التعلّم هو وضع قاعدة عامّة أو خريطة واضحة لكيفية ربط المُدخلات مع مُخرجاتها. ومن حالاته الخاصة، ربط الإشارة المُدخلة مع مُخرَج واحد مُعيّن.

التعلّم الخاضع جزئيًا للإشراف: وفيه يُعطى الحاسوب إشارة تدريب غير مكتملة؛ وهي عبارة عن مجموعة من الأمثلة التدريبية مع العديد من المُخرجات المفقودة.

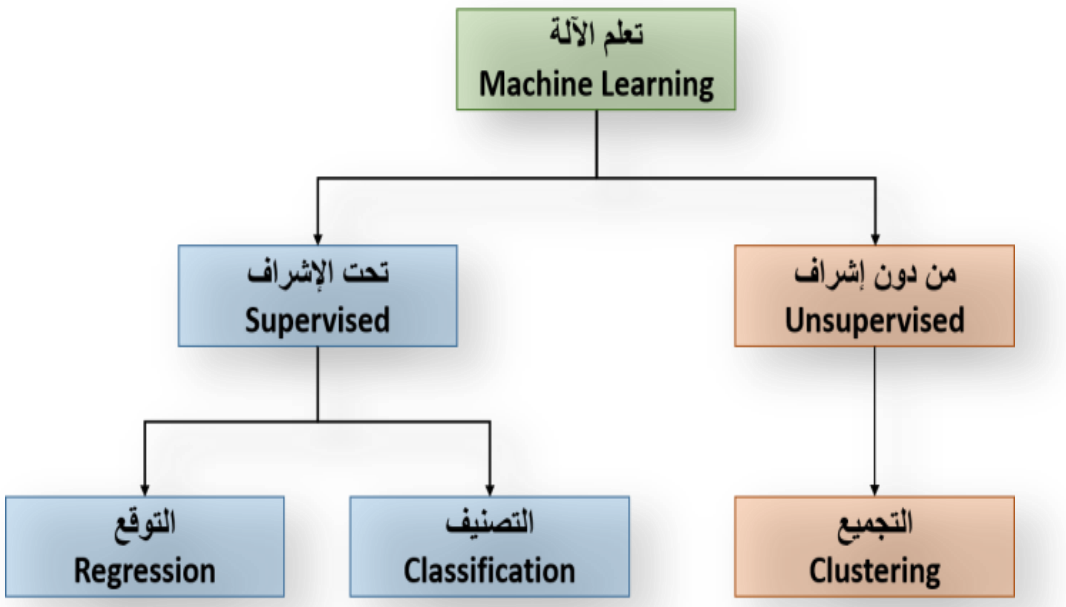
التعلّم دون إشراف: حيث يحاول نظام التعلم استنتاج هيكل البيانات غير المعلمة (ترك الخوارزميات التعلم على نفسها في استكشاف هيكل مدخلاتها ، و ذلك لإكتشاف الأقمار الخفية في البيانات.

وتظهر فئات أخرى لمهام التعلّم الآلي عندما أخذ المُخرجات المرغوبة من نظام تعلم الآلة في عين الاعتبار

عند اتباع :

- ✓ **طريقة التصنيف (Classification):** تُقسّم المُدخلات إلى تصنيفين أو أكثر، ويجب على المتعلّم أن يُنتج نموذجًا لربط المدخلات غير المرئية إلى واحد أو أكثر من هذه التصنيفات. وعادة ما يتم التعامل مع هذا بطريقة خاضعة للإشراف. تعد تصفية الرسائل غير المرغوب فيها مثالًا للتصنيف، حيث تكون المُدخلات عبارة عن رسائل بريد إلكتروني (أو رسائل أخرى) والتي تُصنّف إلى "غير مرغوب فيها" أو "ليست رسائل غير مرغوب فيها" .
- ✓ **طريقة الميل (Regression):** تُعتبر كذلك من ضمن المشاكل التي تخضع للإشراف، تكون المُخرجات فيها ذات طابع مستمر وليس مُنفصل .

- ✓ **طريقة التجميع (clustering):** تُقسّم المُدخلات إلى مجموعات. وعلى عكس التصنيف، فإن المجموعات ليست معروفة مُسبقًا، مما يجعل هذه المهمة من المشكلات غيرالخاضعة للإشراف.
- ✓ **طريقة تقدير الكثافة (Density estimation):** يجد مدى توزيع المدخلات مساحة مُعيّنة.
- ✓ **طريقة تخفيض الأبعاد (Dimensionality Reduction):** تقوم بتبسيط المُدخلات عن طريق ربطها في مساحات أقل أبعادًا. تعد نمذجة الموضوع مشكلة ذات صلة، حيث يتم إعطاء البرنامج قائمة من الوثائق التي تحتوي على لُغة بشرية، وتكون مهمته معرفة الوثائق التي تُغطي مواضيع مشابهة. م[51]



الشكل2: مخطط يوضح أنواع و مهام تعلم الآلة. م[51]

3. شبكات العصبية الاصطناعية (بالإنجليزية: Artificial Neural Network :(ANN

3.1. تعريف:

هي تقنيات حسابية مصممة لمحاكاة الطريقة التي يؤدي بها الدماغ البشري مهمة معينة، وذلك عن طريق معالجة ضخمة موزعة على التوازي، ومكونة من وحدات معالجة بسيطة، هذه الوحدات ما هي إلا عناصر حسابية تسمى عصبونات أو عقد (Nodes، Neurons) والتي لها خاصية عصبية ، من حيث أنها تقوم بتخزين المعرفة العملية والمعلومات التجريبية لتجعلها متاحة للمستخدم وذلك عن طريق ضبط الأوزان.

إذاً الANN تتشابه مع الدماغ البشري في أنها تكتسب المعرفة بالتدريب وتخزن هذه المعرفة باستخدام قوى وصل داخل العصبونات تسمى الأوزان التشابكية. وهناك أيضا تشابه عصبي حيوي مما يعطي الفرصة لعلماء البيولوجيا في الاعتماد على ANN لفهم تطور الظواهر الحيوية.

3.2. تاريخ الشبكات العصبية الاصطناعية:

في حين أن الشبكات العصبية تمثل بالتأكيد تكنولوجيا الكمبيوتر الحديثة القوية، إلّا أن الفكرة تعود إلى عام 1943، مع اثنين من الباحثين في جامعة شيكاغو هما وارن مكولوغ وهو عالم عصبي، ووالتر بيتس عالم الرياضيات.

كانت الشبكات العصبية في الخمسينيات من القرن العشرين منطقة خصبة لأبحاث الشبكات العصبونية على الكمبيوتر، بما في ذلك نظام Perceptron الذي حقق التعرف على النمط المرئي المستند إلى العين المركبة للذبابة.

وفي عام 1975 تم تطوير أول شبكة عصبية متعددة الطبقات، مما مهد الطريق لمزيد من التطوير في الشبكات العصبية ، وهو إنجاز كان البعض يعتقد أنه مستحيل قبل أقل من عقد من الزمن.

تجدد الاهتمام في عام 1982 بشكل كبير في الشبكات العصبية عندما اخترع جون هوبفيلد الأستاذ في جامعة برينستون الشبكة العصبية النقابية، وكان الابتكار هو أن البيانات يمكن أن تنتقل في اتجاه

ثنائي وليس كما كان في السابق بشكل أحادي الاتجاه فقط، وهي معروفة أيضًا باسم مخترعها كشبكة هوبفيلد، تتمتع الشبكات العصبية الاصطناعية في أيامنا هذه بشعبية واسعة ونمو كبير.

3.3. كيفية عمل الشبكات العصبية :

عادةً ما تتضمن الشبكة العصبية عددًا كبيرًا من المعالجات التي تعمل بالتوازي وترتب في مستويات، أي أن الطبقة الأولى التي تتلقى معلومات المدخلات الخام مماثلةً للأعصاب البصرية في المعالجة البصرية البشرية، وتتلقى كل طبقة متتالية المخرجات من الطبقة التي سبقتها، ثم تقوم الطبقة الأخيرة بإخراج منتجات معالجة النظام.

لدى كل عقدة معالجة مجال صغير خاص بها من المعرفة بما في ذلك ما شاهدته أو أي قواعد بيانات تمت برمجتها في الأصل أو تم تطويرها لنفسها بنفسها، كما تكون الطبقات مترابطة بشكلٍ كبير، مما يعني أن كل عقدة في الطبقة n ستكون متصلة بالعديد من العقد في الطبقة n-1.

الشبكات العصبية معروفة بكونها متكيفة، مما يعني أنها تعدل نفسها عندما تتعلم من التدريب الأولي، وبشكلٍ لاحق توفر المزيد من المعلومات حول العالم.

3.4. كيفية تعلم الشبكات العصبية :

بخلاف الخوارزميات الأخرى، لا يمكن برمجة الشبكات العصبية بتعلمها العميق بشكلٍ مباشر للمهمة، وبدلًا من ذلك تشبه الدماغ النامي للطفل، أي أنها تحتاج إلى تلقين المعلومات.

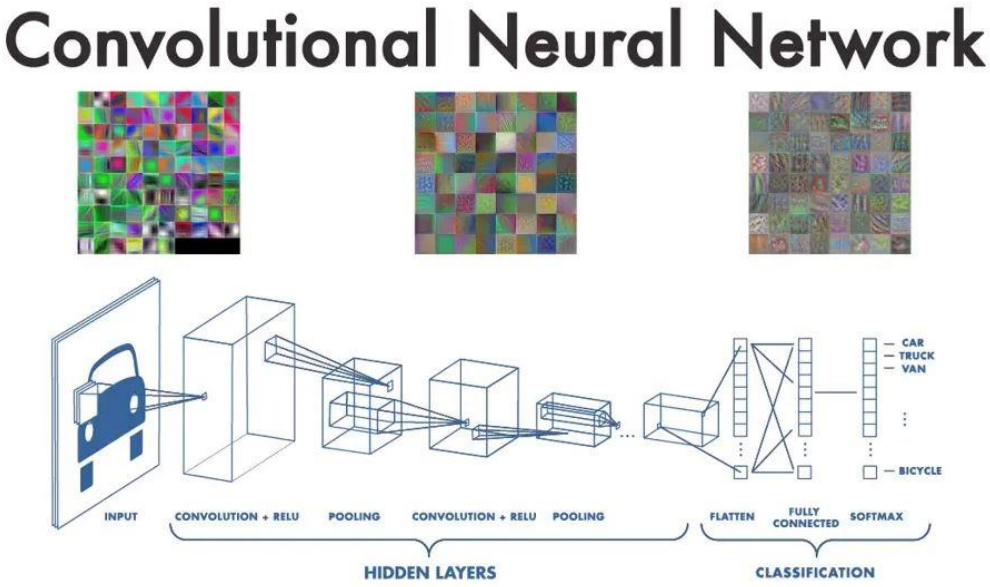
هنالك ثلاث استراتيجيات للتعلم وهي:

- ✓ **التعلم الخاضع للإشراف:** هذه الاستراتيجية التعليمية هي أبسطها حيث توجد مجموعة بيانات موصوفة والتي يمر بها الكمبيوتر، ويتم تعديل الخوارزمية حتى تتمكن من معالجة مجموعة البيانات للحصول على النتيجة المرجوة.
- ✓ **التعلم بدون إشراف:** يتم استخدام هذه الاستراتيجية في الحالات التي لا تتوفر فيها مجموعة بيانات مُعلِّمة للتعلم منها، حيث تقوم الشبكة العصبية بتحليل مجموعة البيانات ثم تقوم دالة التكلفة بإخبار الشبكة العصبية عن المدى البعيد عن الهدف، ثم تقوم الشبكة العصبية بالتكيف لزيادة دقة الخوارزمية.
- ✓ **التعلم المعزز:** في هذه الخوارزمية يتم تعزيز الشبكة العصبية للحصول على نتائج إيجابية وتعاقب على نتيجة سلبية، مما يجبر الشبكة العصبية على التعلم مع مرور الوقت.

3.5. أنواع الشبكات العصبية:

أحيانًا يتم وصف الشبكات العصبية من حيث عمقها، بما في ذلك عدد الطبقات الموجودة بين المدخلات والمخرجات أو ما يسمى بالنماذج المخفية، هذا هو السبب في أن استخدام مصطلح الشبكة العصبية مرادف تقريبًا للتعلم العميق، كما يمكن وصفها أيضًا بعدد العقد المخفية التي يمتلكها النموذج أو من حيث عدد المدخلات والمخرجات لكل عقدة.

- ✓ **الشبكات العصبية التلافيفيةCNN (convolutional Neural Network):** تحظى بشعبية اليوم لا سيما في مجال التعرف على الصور، وقد تم استخدام هذا النوع المحدد من خوارزمية الشبكة العصبية في العديد من التطبيقات الأكثر تطورًا في الذكاء الاصطناعي، بما في ذلك التعرف على الوجه وترقيم النص ومعالجة اللغة الطبيعية، تعتبر الشبكات العصبونية الالتفافية CNN (Convolutional Neural Networks) نوعاً خاصاً وهاماً من أنواع الشبكات العصبونية المستخدمة في التعلم العميق والتي تتميز بأنها تتعلم بشكل مباشر من الصور المستخدمة كدخل في بنيتها
- ✓ وتعتبر حلاً للكثير من مشاكل الرؤية الحاسوبية Computer Vision



الصورة رقم 01: رسم توضيحي لعمل الشبكات العصبية الالتفافية CNN [53]

- ✓ **أبسط أنواع الشبكات العصبية هو ذات التغذية العصبية الأمامية Feedforward Neuralnetworks:** وهذا النوع من خوارزمية الشبكة العصبية الاصطناعية يمرر المعلومات مباشرة من الإدخال إلى عقد المعالجة إلى النتائج، وقد لا تحتوي على طبقات عقدة مخفية، مما يجعل عملها أكثر قابلية للتفسير.

- ✓ **الأكثر تعقيدًا هي الشبكات العصبية المتكررة:** حيث تعمل خوارزميات التعلم العميق هذه على حفظ مخرجات عقد المعالجة وإدخال النتيجة في النموذج، هذه هي الطريقة التي يقال فيها إن النموذج يتعلم.

خاتمة:

في هذا الفصل، وضعنا تعريفا للذكاء الاصطناعي مع ذكر أهم أنواعه ، كما قدمنا تعريفا لتعلم الألة وماهيتها ومبدأها وانواعها واستخداماتها كما تطرقنا الى الشبكة العصبية احد تطبيقات تعلم الالة وكيفية عمله وانواعه.

وفي الختام وبعدما رأينا اهمية تعلم الالة و الشبكات العصبية و مدى اسهامها لكثير من الاعمال وسرعتها في اعطاء النتائج، ماهي تطبيقات هذه التقنية وكيفية عملها على المستوى البرمجي و التطبيقي؟ وهذا ما سنتطرق اليه في الفصل القادم.

الفصل الثاني: المفاهيم الاساسية في بناء نماذج التصنيف

II. الفصل الثاني: المفاهيم الاساسية في بناء نماذج التصنيف

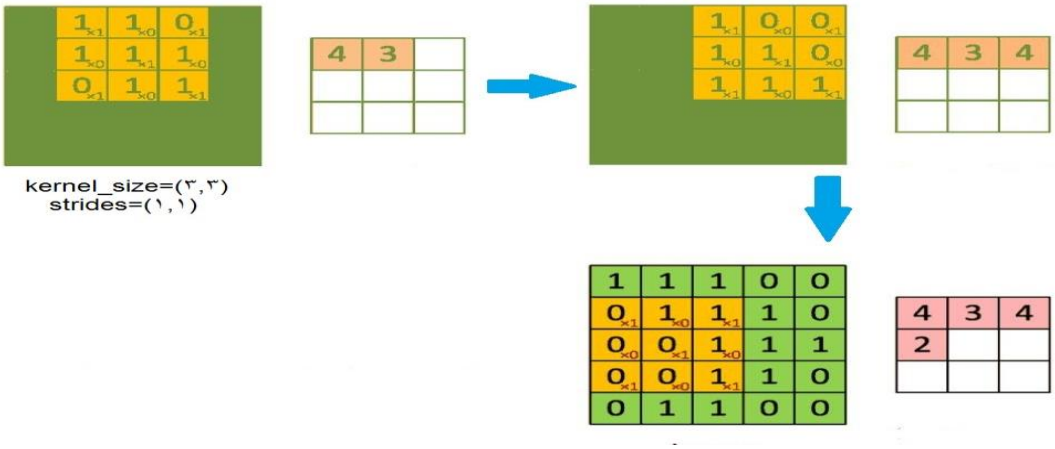
مقدمة:

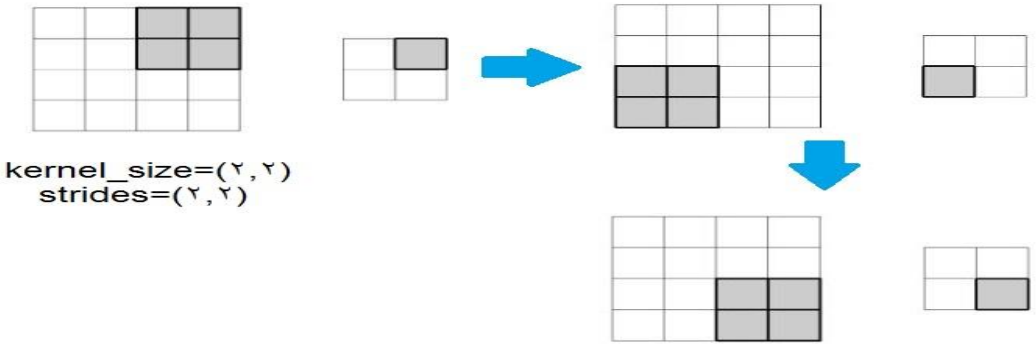
بعد ان عرفنا الذكاء الصناعي وتعلم الالة وعرفنا اهميتها، نتطرق في هذا الفصل الى المفاهيم الاساسية في بناء نماذج التصنيف وما هي مكوناتهم وخصائصهم و وظائفهم.

1. أنواع ال layers التي تتكون منها النماذج:

1.1. تعريف ال (Convolution Layer):

أحد أنواع ال (Layers) المهمة و الأساسية في الشبكات العصبية الإلتفافية (CNNs) يعمل على إستخراج الميزات المهمة (Important Features) من الصور م[3]، عند إحتواء شبكة ال (ConvNet) على عدة (Convolution Layers) هذا يجعلها تستخرج ميزات عالية المستوى م[4]، مبدأ عمله هو تطبيق (Filter) والمعرف بمصفوفة تدعى بال (Kernel) تحتوي هاته المصفوفة على قيم تسمى الـ (Weights) وتكون عشوائية في البداية يكون حجم هاته المصفوفة أصغر من الصورة المراد معالجتها عادة تكون بالأبعاد (3x3) يمكن تحديد قيم أبعادها يدويا بما يسمى (kernel_size) كما يمكن تحديد عدد خطوات تنقلها عبر المحورين (x,y) للصورة المراد معالجتها بمتغير يسمى (strides)، يسمى مركز مصفوفة ال (Kernel) بال (Origin). م[3] [4]





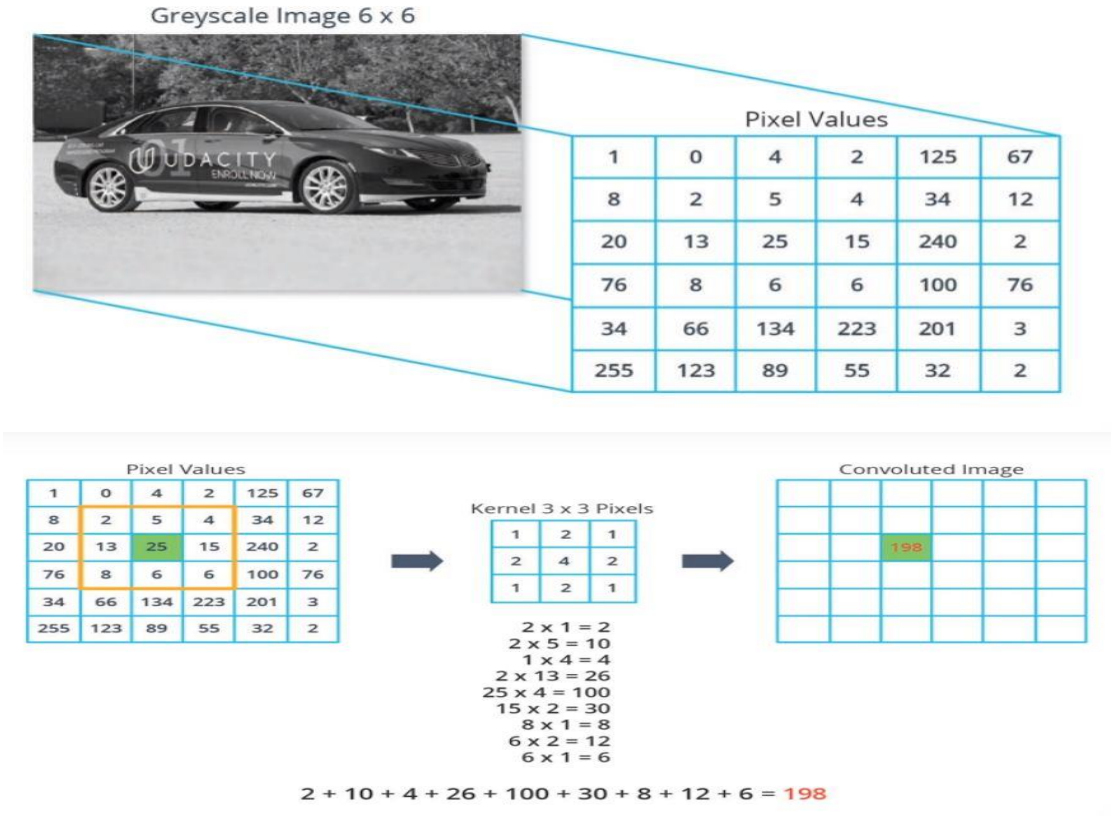
الصورة رقم 02: توضح إختلاف أبعاد الفيلتر الخاصة بعملية الإلتفافية كذلك إختلاف خطوات تنقلها. م[4] [1]

1.2. آلية تطبيق ال (Filter Of Convolution) على الصورة:

تتحرك مصفوفة ال (Kernel) على الصورة كل مرة بقيمة خطوة التي حددناها مسبقا على المحور (y) بدءا من اليسار إلى اليمين وتنحدر تدريجيا كل مرة بقيمة خطوة التي حددت هيا أيضا مسبقا على المحور (x) نزولا بعد إنهاء كل سطر من الأعلى إلى الأسفل، حيث في كل مرة يوضع مركز مصفوفة ال (Kernel) على (Pixel) معين من الصورة ثم تضرب كل قيمة (Weight) من ال (Kernel) بقيمة ال (Pixel) المقابلة لها فالصورة ثم يتم جمع جميع القيم وتضاف لها قيمة صغيرة تسمى ال (Bias) ثم أخيرا الناتج يوضع في (Pixel) آخر في صورة جديدة تسمى بال (Feature Map) بنفس إحداثيات ال (Pixel) السابق فالصورة التي تمت معالجتها، حيث أن ال (Feature Map) التي تحصلنا عنها بعد المعالجة تحتوي على العموم على الميزات المهمة التي تم إستخراجها من الصورة السابقة، كما أنه يمكن تطبيق عدة (Kernels) من طرف ال (Convolution Layer) على نفس الصورة حيث كل (Kernel) يعطي لنا (Feature Map) مختلفة عن ال (Feature Maps) الخاصة بال (Kernels) الآخرين. [3]

أمثلة توضيحية:

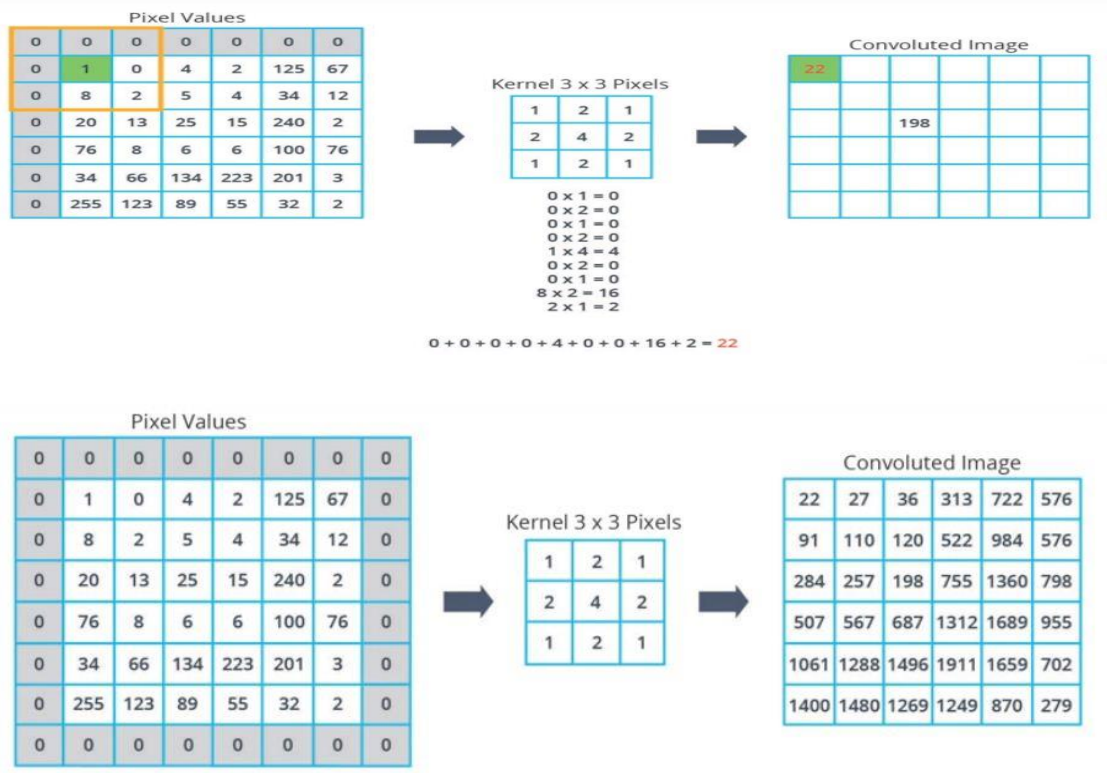
- المثال الأول: التطبيق على صورة رمادية (Grayscale Image) لاتحتوي على (Channels) خاصة بالألوان.



الصورة رقم 03: توضح أجزاء من تطبيق الإلتفافية على صورة رمادية م[6].

- فالصورة الأخيرة أعلاه نلاحظ أنه عند تموضع مركز ال (Kernel) في أحد ال (Pixels) التي على حدود أو حافة الصورة أصبح جزء من ال (Kernel) خارج الصورة لا يقابل أي قيم (Pixels) من الصورة وهذا لا يسمح لنا بتطبيق ال (Filter Of Convolution) على ال (Pixels) التي على حواف الصورة، إذا قمنا بتخطي هاته ال (Pixels) التي على الحافة وقمنا فقط بمعالجة ال (Pixels) التي أدنى من حواف الصورة فهذا يجعلنا نخسر القليل من الميزات التي فالصورة، عندما تكون هاته ال (Pixels) التي على حافة الصورة قليلة الأهمية فإننا نتخطاها ونقوم بمعالجة بقية ال (Pixels) الأخرى التي أدنى من حواف الصورة فنتحصل فالنهاية على (Feature Map) أقل أبعادا من الصورة التي تمت معالجتها

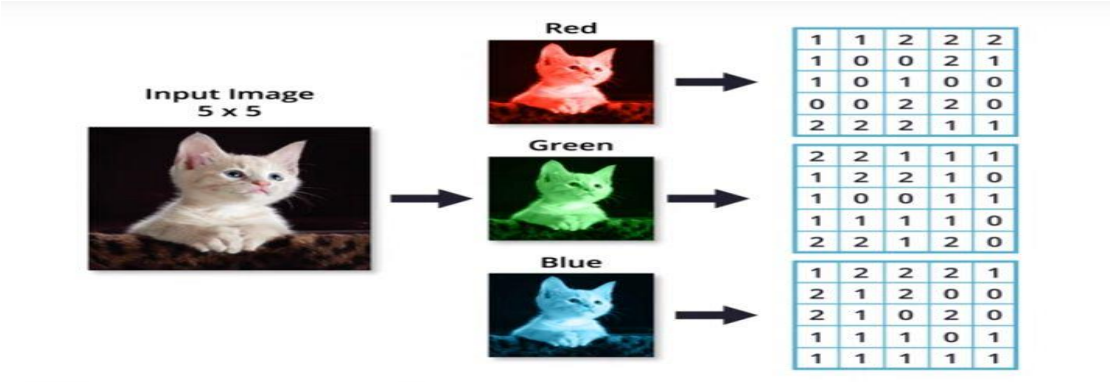
ويسمى هذا الخيار ب (Valid Padding) لأنها لا تحافظ على نفس الأبعاد، أحيانا تكون هاته ال (Pixels) التي على حواف الصورة ذات أهمية عالية لما تحمله من ميزات (Features) فنسعى لعدم خسارتها بخيار آخر يسمى الـ (Zero Padding) وهو إضافة القدر الكافي من الأصفار على الحواف حتى نتمكن من تطبيق ال (Filter Of Convolution) على جميع ال (Pixels) بدون إستثناء ونتفادى خسارة الميزات التي تحملها تلك ال (Pixels) التي على الحواف وتكون أبعاد ال (Feature Map) المتحصل عليها لها نفس أبعاد الصورة ويسمى هذا الخيار بطريقة أخرى ال (Same Padding) لأنه يحافظ على نفس الأبعاد، فالصورة أدناه توضيح لذلك. [4]

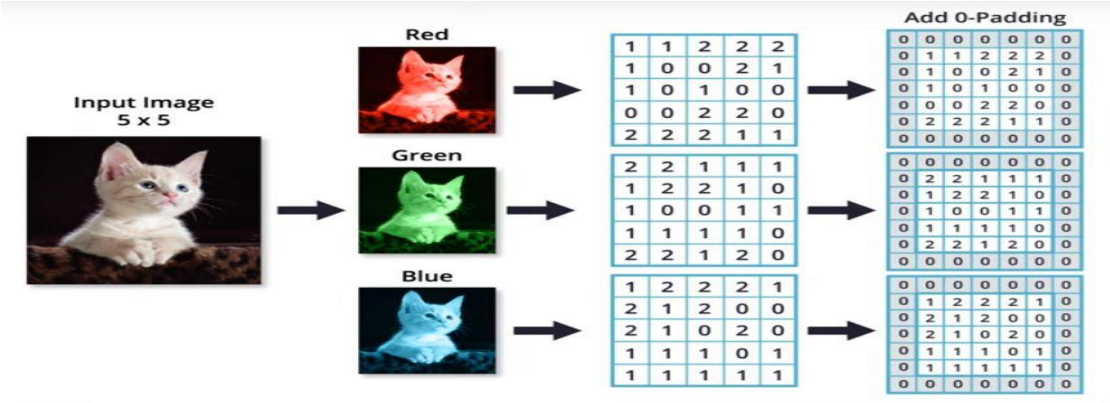


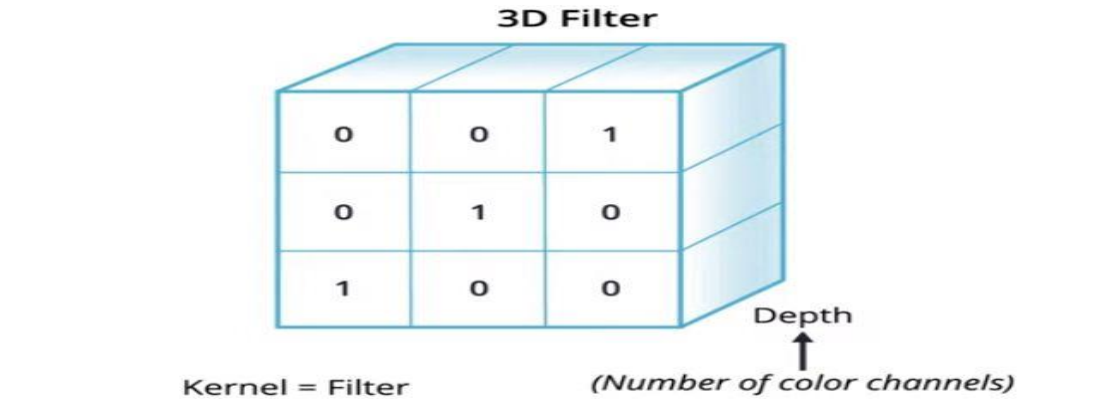
الصورة رقم 04: توضح بقية عملية الإلتفافية بعد تطبيق ال padding ونتيجتها م[6].

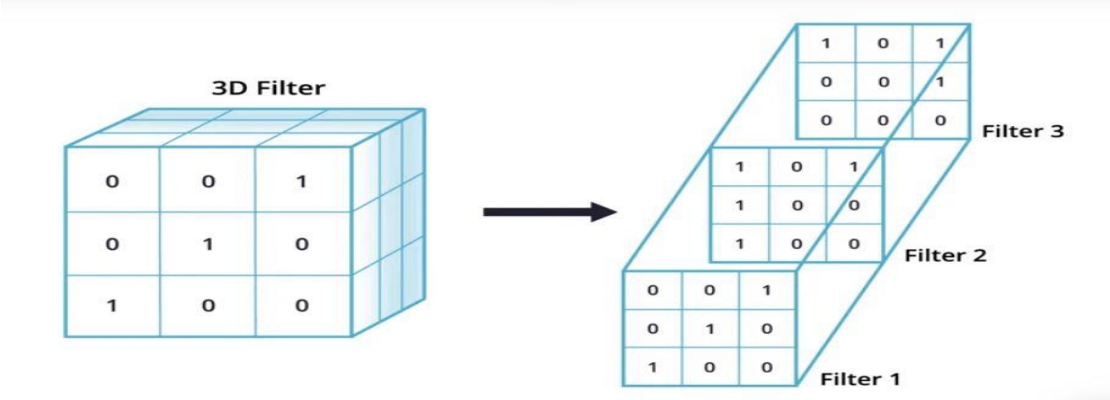
- المثال الثاني: التطبيق على صورة من نوع (RGB Image) تحتوي على 3 (Channels)

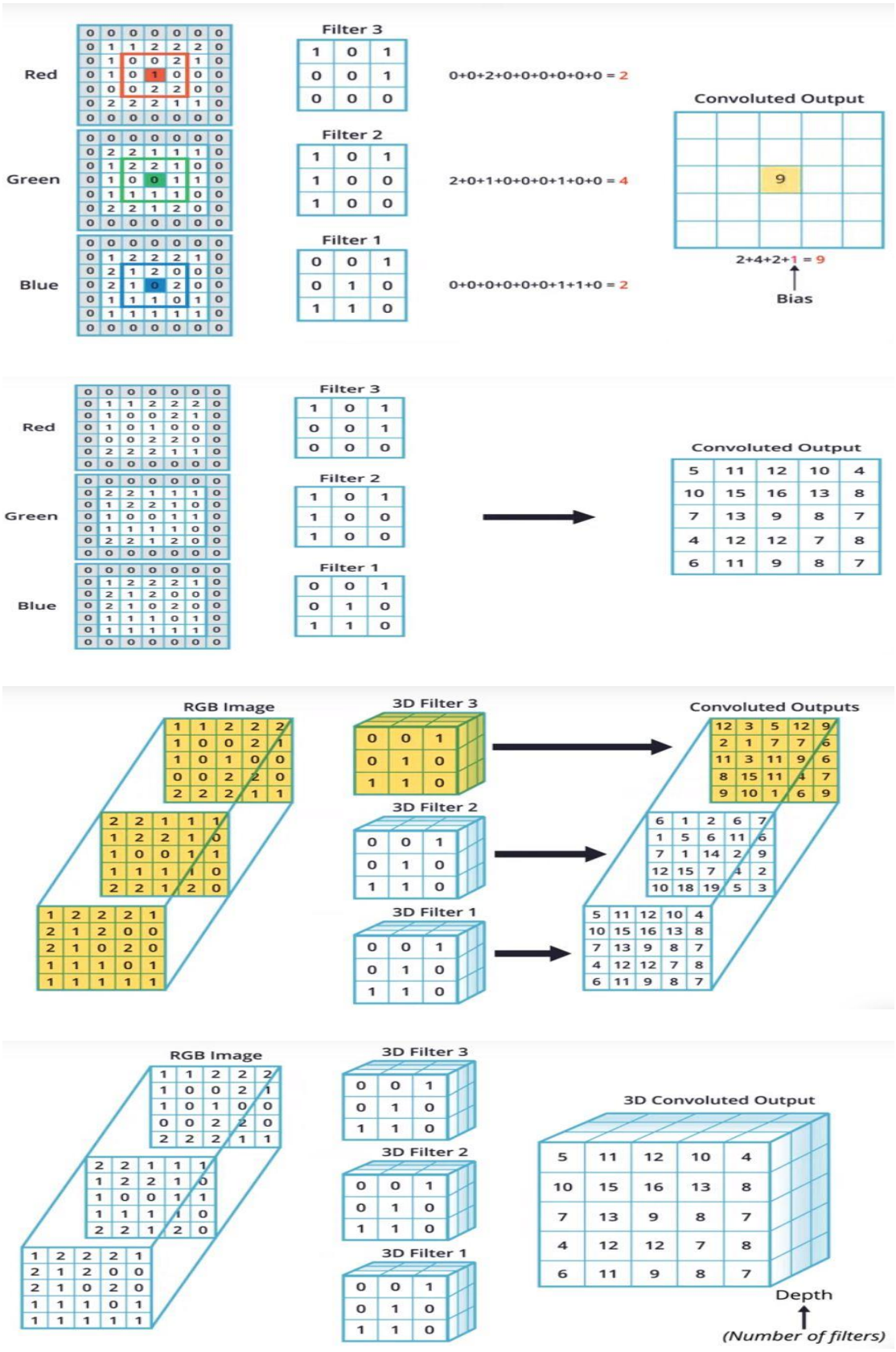
* بما أن الصورة تحتوي على 3 (Channels) فسيتم معالجتها ب (Kernel) بنفس عدد ال (Channels) أي (Depth Of Kernel =3) يعني (3D Kernel) فالصور التالية توضيح لذلك.


Input Image
5 x 5
Red
Green
Blue


Input Image
5 x 5
Red
Green
Blue
Add 0-Padding

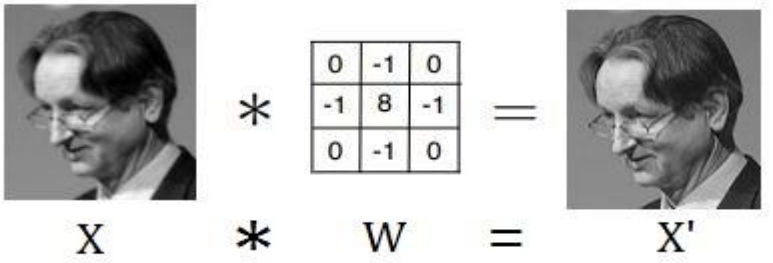

3D Filter
Depth
(Number of color channels)
Kernel = Filter

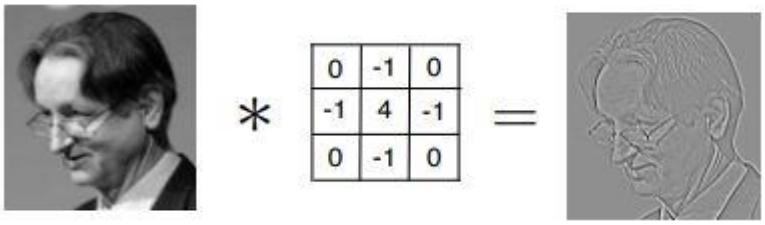

3D Filter
Filter 3
Filter 2
Filter 1



الصورة رقم 05: توضح خطوات عملية الإلتفافية على صورة ملونة [6].

- هنا أمثلة توضح كيف أن ال (Feature Map) تختلف بإختلاف ال (Kernel) المطبقة عليها.

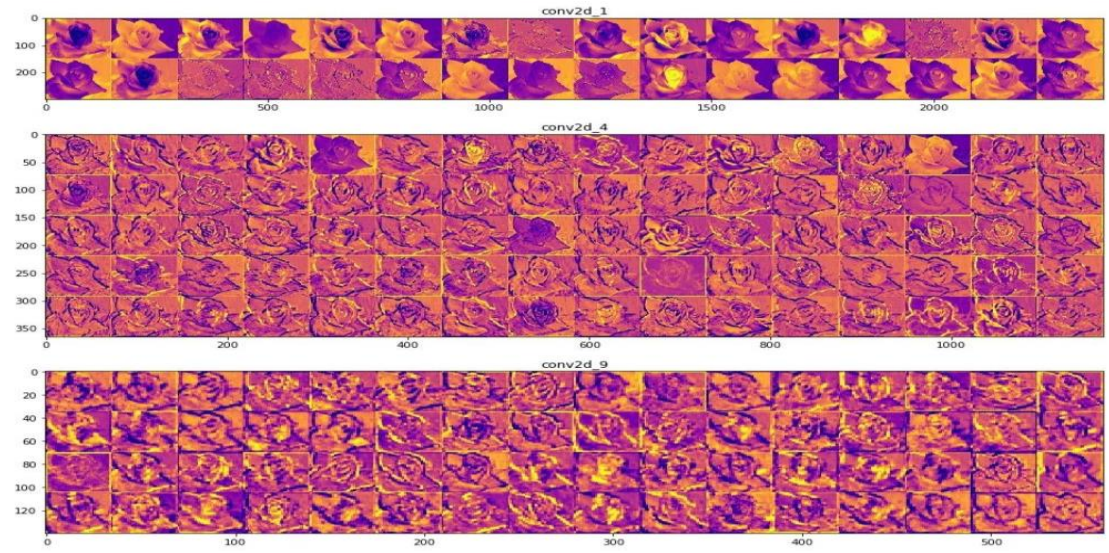




الصورة رقم 06: توضح إختلاف نتائج تطبيق الإلتفافية على نفس الصورة حسب إختلاف الفيلتر م[2]

- كذلك هنا بعد تطبيق ال (Kernels) الخاصة بكل من ال (Convolution Layers1,4And9) من (Model InceptionV3). م[5]

الصورة رقم 07: الصورة الأصلية قبل تطبيق عملية الإلتفافية عليها م[5].



الصورة رقم 08: توضح نتائج تطبيق عدة فلترات خاصة بعدة طبقات إلتفافية من مستويات مختلفة فالنموذج م[5].

1.3. خصائص ال (Convolution):

+ مثل عملية الضرب، تبادلية (Commutative) و تجميعية (Associative) التعريف الرياضي للخاصيتين فالرياضيات. م[2]

معادلة توضح خاصيتي التبادلية والتجميعية لعملية الإلتفافية م[2]:

$$u * v = v * u$$
$$(u * v) * w = u * (v * w)$$

لدينا صورة (X) نريد أن نطبق عليها ال (Kernel) الأولى المسماة (Wblur) ثم النتيجة المتحصل عليها نطبق عليها ال (Kernel) الثانية المسماة (Whorz)، عند تطبيق خاصيتي التبادلية والتجميعية.

تكون حالات التطبيق الثلاثة المختلفة هاته متساوية النتيجة.

معادلة توضح إحدى أهم النتائج عند تطبيق خاصيتي التبادلية والتجميعية لعملية الإلتفافية م[2]:

$$(\boldsymbol{x} * \boldsymbol{w_{blur}}) * \boldsymbol{w_{horz}} = (\boldsymbol{x} * \boldsymbol{w_{horz}}) * \boldsymbol{w_{blur}} = \boldsymbol{x} * (\boldsymbol{w_{blur}} * \boldsymbol{w_{horz}})$$

- توضيح لكيفية تطبيق الحالات الثلاثة:

* الحالة الأولى: تطبيق ال (Kernel Wblur) على الصورة (X) ثم النتيجة نطبق عليها ال (Kernel Whorz).

* الحالة الثانية: تطبيق ال (Kernel Whorz) على الصورة (X) ثم النتيجة نطبق عليها ال (Kernel Wblur).

* الحالة الثالثة: نطبق ال (Kernel Wblur) على ال (Kernel Whorz) ثم نتيجتهم والتي هي عبارة على (Kernel) جديد نطبقها على الصورة (X).

* خطية:

معادلة توضح خاصية الخطية لعملية الإلتفافية [2]:

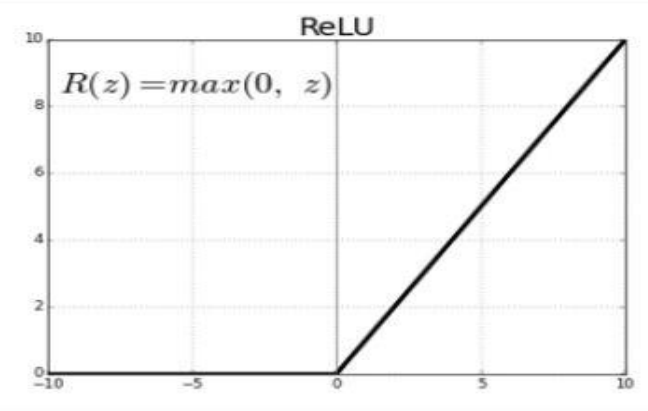
$$(ax + bx') * w = ax * w + bx' * w$$

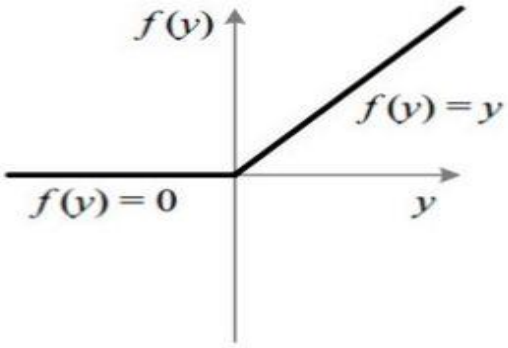
$$x * (aw + bw') = ax * w + bx * w'.$$

هاته الخاصية جيدة لأن العمليات الخطية غالبا سهل التعامل معها، نستنتج أن تسلسل من ال (Convolution Layers) هو عبارة عن عدة دوال خطية متتابعة تعطي نتائج خطية، لاكن الصور فالواقع لا تخضع للخطية لذلك يلزم عمليات أكثر تعقيدا لحل هاته المشكلة سنتطرق فيما بعد لمفهوم (Non-Linear Activation Function) التي تجعل ال (Convolution Layer) أكثر قوة بتحويل قيمها إلى قيم غير خطية وهي تطبق على كل ال (Convolution layers). م[2]

1.4. Relu function :

إحدى أهم ال (Non-Linear Activation Function) الأكثر شيوعا وإستخداما في نوع الشبكات العصبية الإلتفافية (CNN) ومجال ال (Deep Learning) وهي ال (Rectifier Function) وتدعى أيضا ب (Rectified Linear Unit) تختصر التسمية هكذا (Relu) هي دالة بسيطة تعطي قيمة 0 لأي مدخل سالب أو معدوم وتعطي نفس قيمة المدخل إذا كان المدخل موجب، الصور التالية توضح ذلك م[13].





الصورة رقم 09: توضح صيغة ومنحنى الدالة Relu م [13].

بكل بساطة هكذا تطبق هاته الدالة على قيم ال (Feature Map) الناتجة عن ال (Convolution) فالصورة التالية.

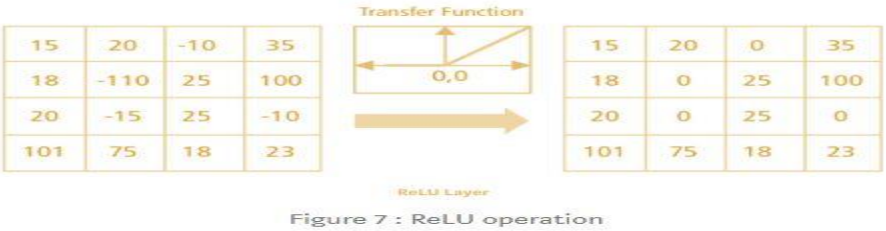


Figure 7 : ReLU operation

الصورة رقم 10: توضح تطبيق ال relu على نتائج الإلتفافية م[16].

مشتق ال (Relu Function) يكون 1 عند المدخلات الموجبة و 0 عند المدخلات السالبة أو المنعدمة.

الصيغة الرياضية لدالة ال Relu م [2]:

$$ReLU'(x) = \begin{cases} 1 & if\ x > 0 \\ 0 & otherwise \end{cases} = sgn(ReLU(x))$$

sgn إختصار لدالة الإشارة (Sign Function) وهي دالة تعطي إشارة المدخل مثلا المدخل سالب تعطي القيمة -1 المدخل موجب تعطي القيمة 1 المدخل معدوم تعطي القيمة 0.

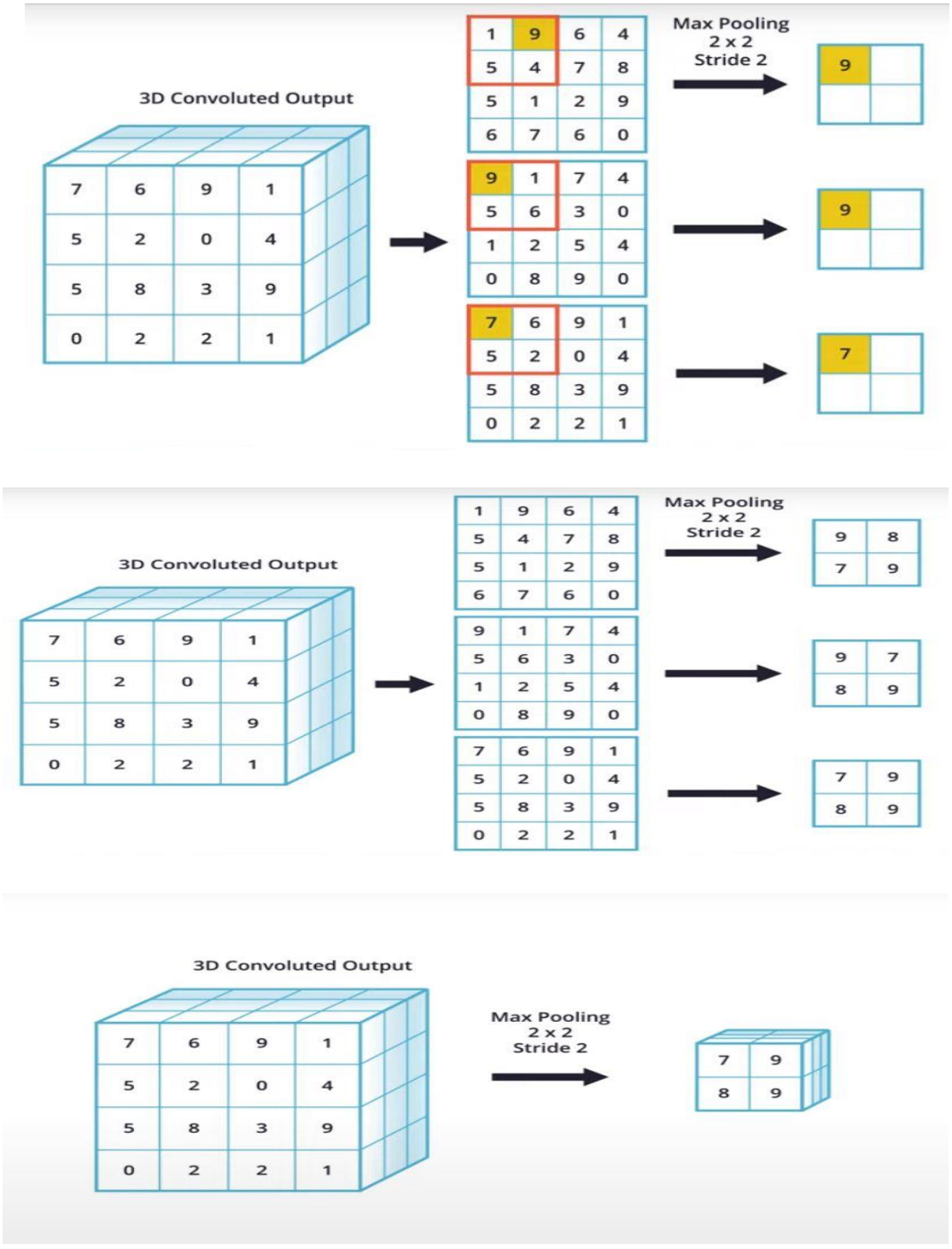
ما يجعلها مميزة عن بقية ال (Non-Linear Activation Function) هي أنها سريعة في عملية ال (Training) لأن مشتقها بسيط إما 0 أو 1 وهذا يسهل حساب ال (Gradient) في عملية ال (Back Propagation).

1.5. (Pooling Layer):

بعد إستخراج ال (Feature Maps) من عملية تطبيق ال (Filters Of Convolution) نقوم بإختزال وتقليل عدد ال (Features) فال (Feature Maps) ونبقي فقط على ال (Features) الأساسية وهذا ينتج لنا (Feature Maps) أقل أبعادا، (Max Pooling Layer) هو إحدى أنواع ال (Pooling Layer) الأكثر تداولا[12].

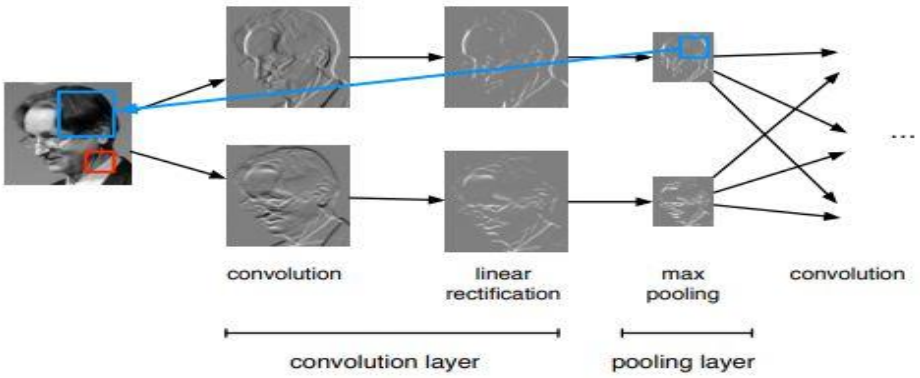
a) **<u>آلية تطبيق الـ (Max Pooling Layer) على الـ (Feature Map):</u>**

مثلا نقوم بتمرير ال (Kernel Of Max Pooling) ذات الأبعاد (Kernel Size = 2x2) والخطوات (Stride = 2,2) على ال (Feature Map) وفي كل مرة عند تموضع ال (Kernel Of Max Pooling) على مكان معين يتم إستخراج أكبر قيمة من القيم الأربعة وتوضع فال (Feature Map) الجديدة، يمكن أيضا إستخدام ال (Zero Padding) في ال (Pooling Layer) الصور أدناه توضح ذلك م[15].



الصورة رقم 11:توضح تطبيق ال max pooling على مخرجات عملية الإلتفافية [18].

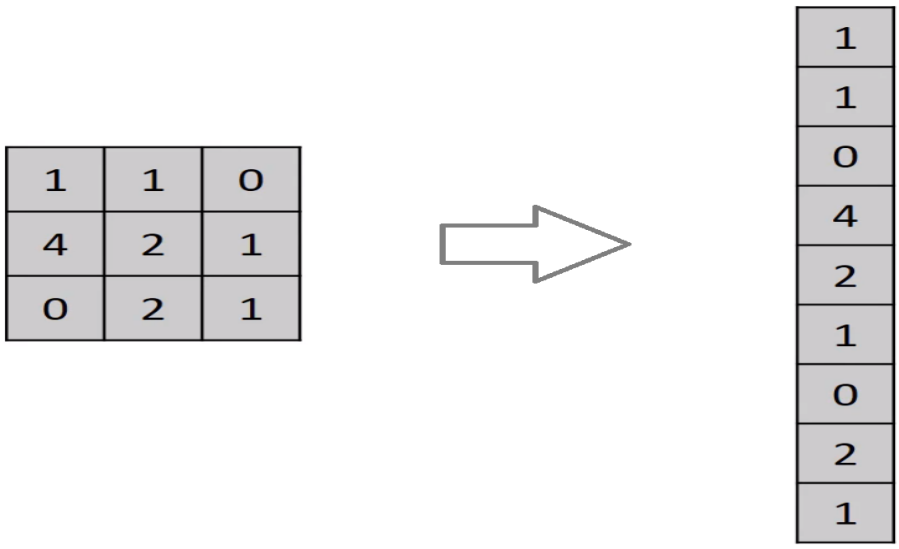
نتائج تطبيق ال (Convolution) ثم (Relu) ثم ال (Max Pooling) على صورة معينة.



الصورة رقم 12: نتائج تطبيق ال (Convolution) ثم (Relu) ثم ال (Max Pooling) على صورة معينة م[10].

.1.6 (Flattening Layer):

بعد أن قمنا بتطبيق هاته العمليات بالترتيب (Convolution), (Relu), (Max Pooling) على صورة معينة نتحصل على ال (Feature Maps) الأساسية لها وهي عدة مصفوفات ثنائية الأبعاد يلزم تحويلهم إلى متجه واحد (Vector) أحادي البعد حتى نتمكن من تمريريهم كمدخلات لل (Fully Connected Layer) فالمرحلة التالية، صورة توضح ال (Flattening Operation) على (Feature Map) واحدة من بين عدة (Feature Maps) للصورة م[12].



الصورة رقم 12: توضح تطبيق عملية ال Flattening على Feature Map م [15].

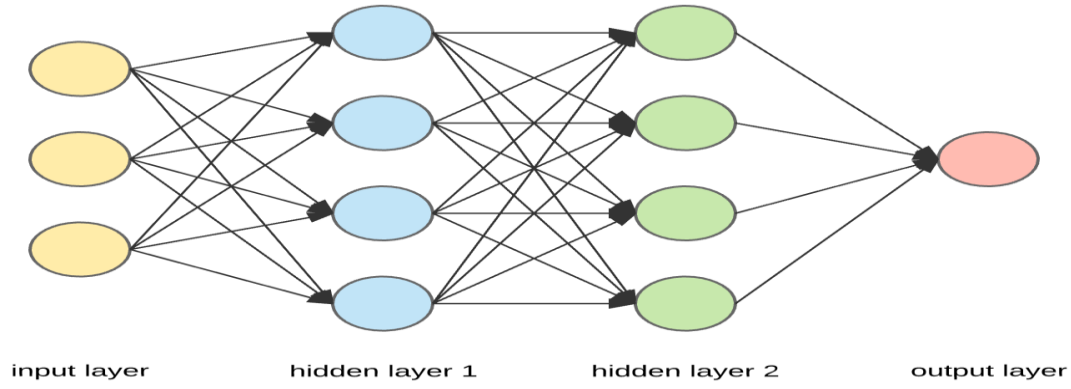
بعد تطبيق ال (Flattening Operation) على كل ال (Feature Maps) الخاصة بالصورة ينتج لنا عدة (Vectors) نقوم بدمجهم في (Vector) واحد ونمرره للمرحلة التالية.

في حالة كان ال (Batch Size > 1) يعني المدخلات لل (CNN) كانت أكثر من صورة واحدة فكل صورة يتم تحويل ال (Feature Maps) الخاصة بها إلى (Vectors) ثم دمجهم في (Vector) واحد.

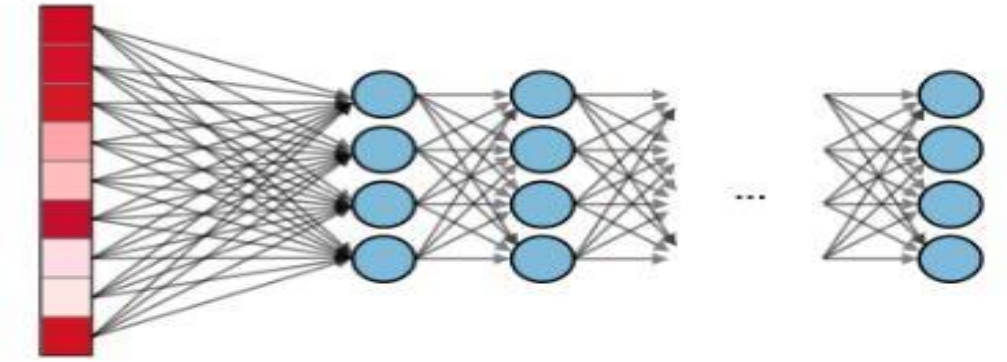
بعدها سنحصل على عدد من ال (Vectors) نفس عدد صور ال (Batch Size) نمررهم للمرحلة التالية.

1.7. (Fully Connected Layers) أو (Dense Layers):

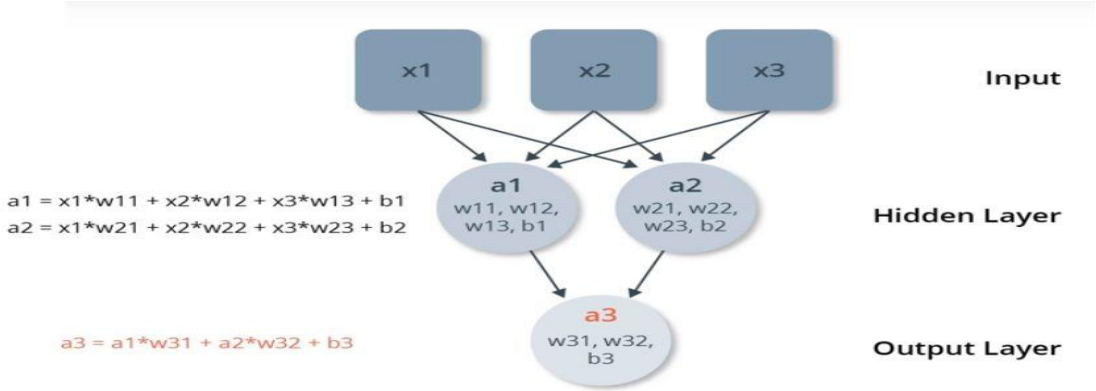
عادة تنتهي ال (CNN) بشبكة من ال (Fully Connected Layers) أول (Layer) من هاته الشبكة يدعى بال (Input Layer) ويكون متصل بكل قيم ال (Vector) الذي تحصلنا عليه سابقا في عملية ال (Flattening)، ال (Layers) التي وسط هاته الشبكة تدعى (Hidden Layers) وآخر (Layer) من هاته الشبكة يدعى (Output Layer) وهو المسؤول عن القيم التنبئية لكل صنف (Class)، كل (Layer) من هاته الشبكة يحتوي على عدة (Neurons) كل (Neuron) يكون متصل بال (Neurons) التابعة لل (Layer) الذي قبله نعني بالإتصال أنه يأخذ كل قيم ال (Neurons) فال (Layer) الذي قبله ويضرب كل قيمة في (Weight) معين ثم يجمع جميع القيم ويضيف لهم قيمة صغيرة تسمى (Bias) عادة تكون 1 من هاذه العملية نستنتج أن قيم ال (Neurons) خطية، إذا طبقنا على ال (Layer) إحدى ال (Activation Function) مثل (Relu Function) للتخلص من الخطية فإن كل ال (Neurons) داخله تحتوي على جزئين جزء خاص بال (Input) وجزء خاص بال (Output)، جزء ال (Input) يتغير فيه فقط ال (Weight) والـ (Bias) أثناء عملية ال (Training) وجزء ال (Output) يتم حسابه عن طريق تطبيق ال (Activation Function) على ال Input م[12][7].



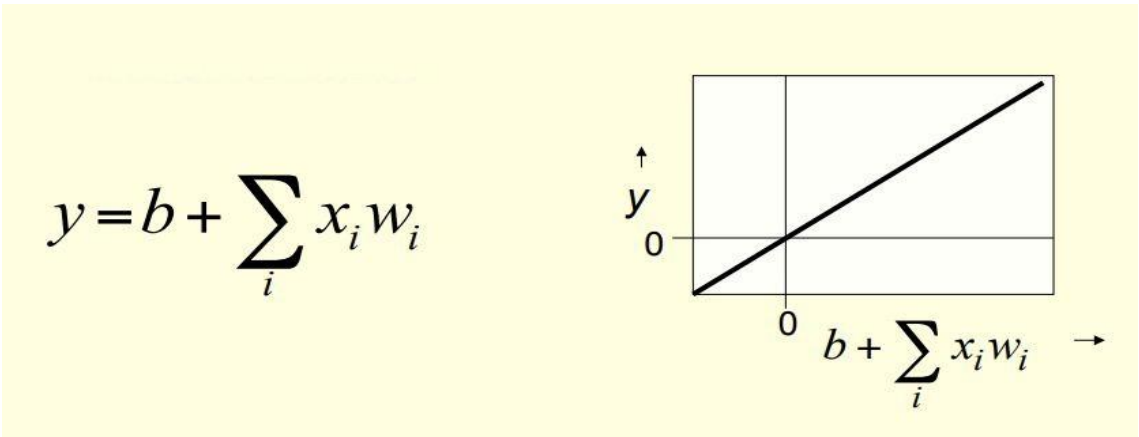
الصورة رقم 13: توضح شبكة عصبية من عدة طبقات [17].



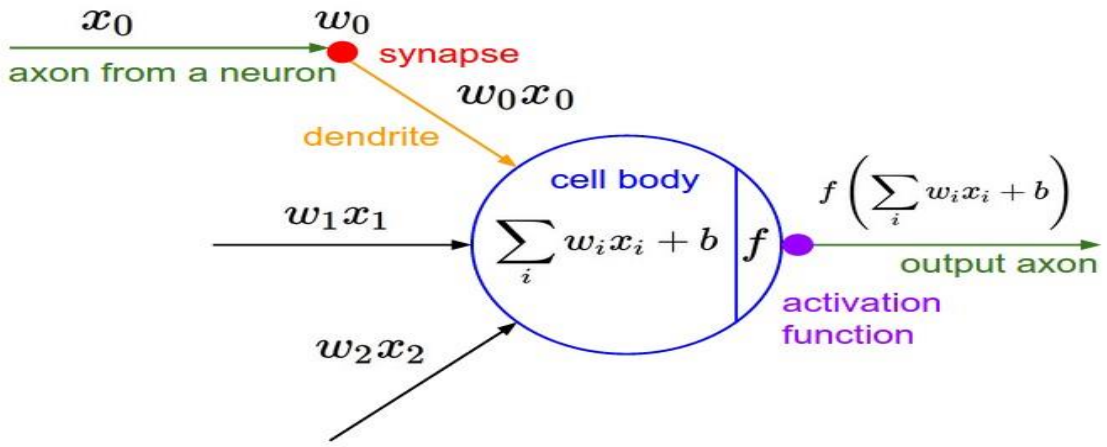
الصورة رقم 14:توضح شبكة عصبية من عدة طبقات مرتبطة بالمدخلات [7].



الصورة رقم 15: توضح كيفية حساب قيم كل عصب من الشبكة العصبية [18].

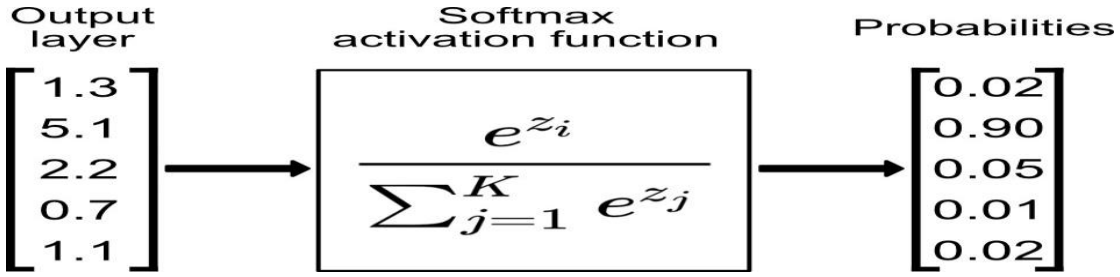


الصورة رقم 16: توضح أن قيمة العصبون هي قيمة خطية [11].



الصورة رقم 17: توضح مكونات العصبون وقيم كل من مدخلاته ومخرجاته [9].

آخر (Layer) من هاته الشبكة يحتوي على (Activation Function) مختلفة وهي (Softmax Activation Function) التي تمكننا من حساب الإحتمالات التنبئية لكل (Class).



الصورة رقم 18: توضح طريقة عمل ال Softmax Activation Function م[14].

فالصورة أعلاه يوجد (Vector) يحمل قيم ال (Output Of Neurons) التي في آخر (Layer) فالشبكة وهو (Output Layer) المقصود ب (Zi) هي قيمة من القيم التي فال (Vector Of Output Layer).

لحساب إحتمال (Class) معين نأخذ القيمة فال (Vector) المقابلة له ونحسب أسيتها، بعدها نقوم بقسمة الناتج على مجموع أسيات كل القيم هكذا نحصل على الإحتمال، نقوم بحساب الإحتمال لبقية ال (Classes) بنفس الطريقة.

مجموع القيم الناتجة من (Softmax Activation Function) يساوي الـ 1.

:(Dropout) .1.8

خوارزمية وإحدى تقنيات التنظيم (Regularization) لخفض أو التخلص من مشكلة ال (Overfitting) م[21] ، مشكلة ال (Overfitting) هي حفظ النموذج بدلا من تعلمه السمات العامة للمدخلات م[20]، تعتبر ال (Dropout) كأحد أنواع ال (Layers) في مكتبة (Keras) م[19] .

ملاحظة:-

لمنع ال (Overfitting) من الوقوع يوجد ثلاث طرق وهي م[19]:

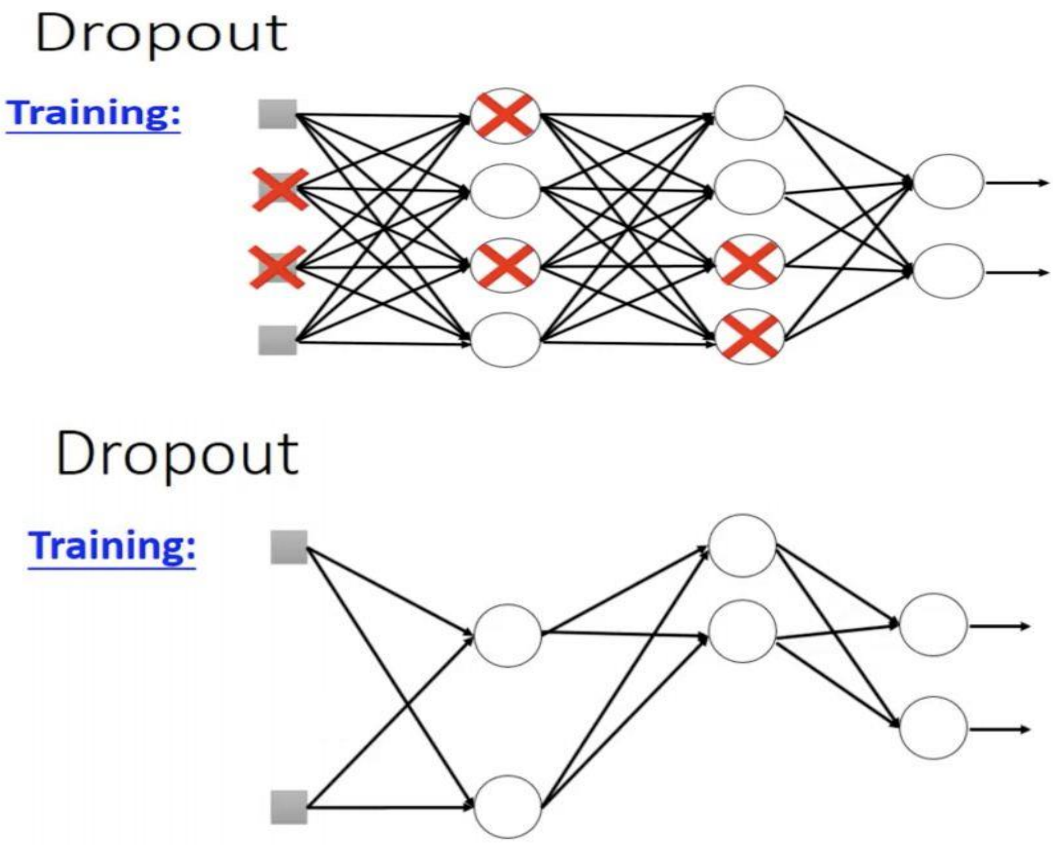
- تقليص حجم الشبكة الخاصة بالنموذج (Reduce NN Size).
- إستخدام إحدى تقنيات التنظيم (Regularization Techniques) مثل (Dropout, L1, L2).
- أو الحصول على بيانات تدريب أكثر.

a) <u>آلية عمل ال (Dropout Layer):</u>

ال (Dropout Layer) يطبق على أول (Layer) قبله مباشرة، في كل مرة قبل تحديث المتغيرات (Parameters) كل (Neuron) من ال (Layer) يكون لديه إحتمال عشوائي لل (Dropout) م[19]،تقرير تطبيق ال (Dropout) على (Neuron) معين من عدمه مربوط بقيمة تدعى معدل ال (Dropout) أي (Dropout Rate, P(drop)) مثلا إذا كان

(P(drop)=0.8) يعني أي (Neuron) كان لديه قيمة إحتمال أقل من ال 0.8 فستطبق عليه عملية ال (Dropout) م[20]، عندما تطبق عملية ال (Dropout) على (Neuron) معين من ال (Layer) هذا يعني أن كل من مدخلاته من طرف ال (Layer) الذي قبله ومخرجاته تحذف يصبح ال (Neuron) غير متصل بأي (Neurons) سواء فال (Layer) الذي قبله أو بعده، هذا يغير من بنية ال (CNN) الخاصة بنا لأن بعض ال (Neuron) تتعطل لبعض الوقت، نستخدم جزء صغير من ال (Training Dataset) يدعى (Mini Batch) لتدريب ال (Weights) الخاصة بالشبكة الجديدة، في كل (Mini Batch) نقوم بإعادة تحديد (Neurons) لنطبق عليهم ال (Dropout)، ال(Dropout) يخفض من أداء تعلم النموذج لاكنه يساعد في تحسين أداء إختبار النموذج لأنه يمنع ال (Overfitting)، ال (Underfitting) هو عندما يكون نموذجنا سيء الأداء أثناء التدريب، في حالة كنا نواجه مشكلة ال (Underfitting) أيضا في نموذجنا فإننا لا نستخدم ال (Dropout)، عند تطبيق ال (Dropout) على

النموذج يصبح كل (Weights) يتدرب بشكل أكثف لإلتقاط الميزات لأن بعض ال (Neurons) فالشبكة معطلة "إذا كان زملائي في الفريق لا يعملون بجد، لا بد لي من العمل بجدية أكبر." م[19].



الصورة رقم 19: توضح توقف بعض الأعصاب في الشبكة بسبب عمل dropout م[19].

Listing 4.8 Adding dropout to the IMDB network

```
model = models.Sequential()
model.add(layers.Dense(16, activation='relu', input_shape=(10000,)))
model.add(layers.Dropout(0.5)) ← dropout rate is 0.5
model.add(layers.Dense(16, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(1, activation='sigmoid'))
```

الصورة رقم 20: توضح كيفية إضافة ال dropout layer برمجيا م[19].

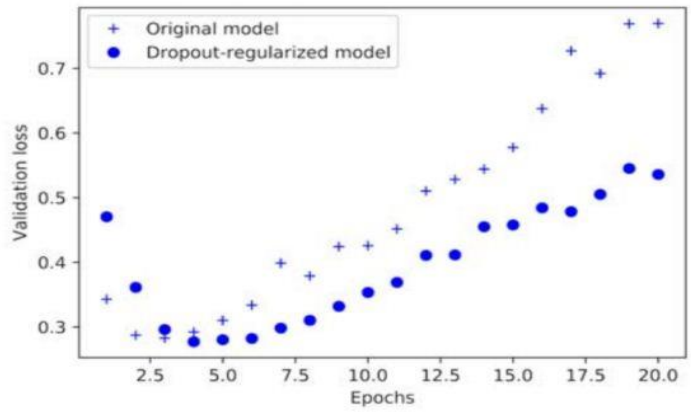


Figure 4.9 Effect of dropout on validation loss

الصورة رقم 21: توضح تأثير ال dropout على دقة مجموعة التحقق م[19].

2. (Dataset):

هي مجموعة من البيانات، الأكثر شيوعا هي أن تكون على شكل جدول من البيانات منسق حيث كل عمود فيه يمثل قيم متغير معين وكل سطر من الجدول يمثل عدة قيم متغيرات تتوافق مع متغير ما من ال (Dataset)، لا تقتصر مجموعة البيانات فقط على النصوص و الأرقام يمكن أن تكون على شكل صور أو مقاطع فديو م[22].

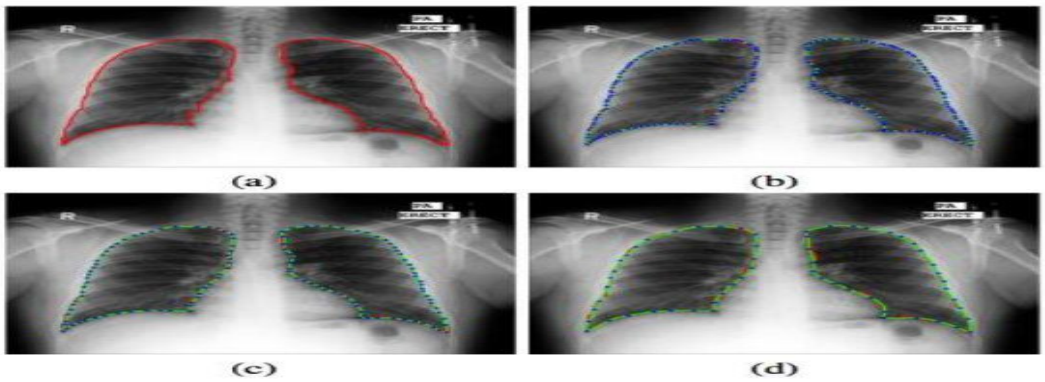
Table - dbo.Student | Summary

StudentSSN	FirstName	MiddleInitial	LastName	Address	City	StateCode	ZipCode	Ph
111-11-1111	Andrea	A	Able	100 S. Andrews St.	Edwardsville	IL	62025	61
222-22-2222	Bobby	B	Best	15 Beauford Way	Alton	IL	63000	61
333-33-3333	Claudia	C	Charming	Clover Avenue North	Maryville	IL	610004400	AL
888-88-8888	Henrietta	H	Headstrong	S. Howard Road, #8	Maryville	IL	610005600	61
999-99-9999	Inga	I	Indigo	5203 Icon Way	Alton	IL	63000	61
444-44-4444	Donald	D	Domingo	1804 Drewey Drive	Alton	IL	63000	61
555-55-5555	Edwardo	E	Easterwing	East South Road 188	Edwardsville	IL	62025	61
666-66-6666	Fredericko	F	Freewaydrive	Four Points South #2	Edwardsville	IL	62025	61
777-77-7777	Greta	G	Greathouse	S. Grand St. #15	Alton	IL	63000	61
111-11-1112	Arnold	A	Airhead	101 S. Andrews	Edwardsville	IL	62025	61
555-55-5554	Claude	C	Charming	Northwest Avenue	Alton	Il	63000	AL
111-11-1113	Harvey	H	Headstrong	105 Northwest St.	Edwardsville	IL	62025	61
999-99-9988	Amy	A	Airhead	17 Southend Drive	St. Louis	Mo	63115	31
333-33-3334	Ariana	A	Airhead	52 West 14th St.	St. Louis	Mo	631155600	31
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	AL

Row

Key Column

الصورة رقم 22: توضح ال (Dataset) على شكل جدول م[23]



الصورة رقم 23: توضح ال (Dataset) على شكل صور م[23]

في مجال التعلم الخاضع للإشراف (Supervised learning) ال (Dataset) تقسم إلى ثلاثة مجموعات كالتالي م[23]:

مجموعة بيانات التدريب (Train)، تستخدم لتدريب ال (Model)

مجموعة بيانات التطوير (Dev) أو التحقق (Valid)، لإختبار النموذج أثناء عملية تطويره،تكون هاته العملية في آخر ال (Epoch) ليس أثناء التدريب يعني عند نهاية كل (Epoch) من التدريب.

مجموعة الإختبار (Test)، تستخدم لإختبار آداء النموذج على البيانات التي لم يسبق له مشاهدتها م[23].

.2.1 (Data Normalisation):

أحيانا تكون ال (Features) الخاصة بال (Dataset) ذات نطاقات مختلفة ومتباعدة عن بعضها بشكل كبير لذلك قد يتم ترجيح الميزات ذات النطاقات الكبيرة ويكون لها تأثير كبير على عملية تدريب ال (Model) مقارنة بالميزات ذات النطاقات الصغيرة، فبالتالي تسوية نطاقات الميزات أمر مطلوب لجعل كل الميزات لها نفس التأثير تقريبا.

$$\hat{X} = \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)} (\max X_{new} - \min X_{new}) + \min\ X_i$$

المعادلة العامة ل (Min-Max normalization) م[24]

a) Linear Scaling to Unit Range:

هو حالة خاصة من (Min-Max normalization) بوضع MAXXnew=1 و MINXnew=0 وذلك لتحويل البيانات من مجال معين إلى المجال [0,1] رياضيا هكذا

$$\hat{X} = \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)}$$

معادلة توضح حالة خاصة من (Min-Max normalization) وهي ال(Linear Scaling) م[24]

كمثال تطبيقه على صورة:

- الصورة تحتوي على (Pixels) كل (Pixel) قيمته فالمجال [0,255] يعني أن

* min(Xi)= 0 و ال max(Xi)=255

* ونريد تحويل قيم البكسل إلى المجال من [0,1] يعني MAXXnew=1 و MINXnew=0

بالتعويض فالمعادلة العامة لل (Min-Max normalization) نحصل على علاقة تعطينا قيمة ال (Pixel) الجديدة إنطلاقا من قيمة ال (Pixel) القديمة حيث تكون جميع القيم الجديدة لل (Pixels) محصورة بين ال [0,1]

Linear Scaling:

$$\hat{X} = \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)} (\max X_{new} - \min X_{new}) + \min\ X_i$$

Data in range $\rightarrow$ $\max X_{new} = 1\ and \min X_{new} = 0. \rightarrow \hat{X} = \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)}$

$$\max(X_i) = 255 \qquad and\ \min(X_i) = 0 \qquad \rightarrow \quad \hat{X} = \frac{X_i}{255}$$

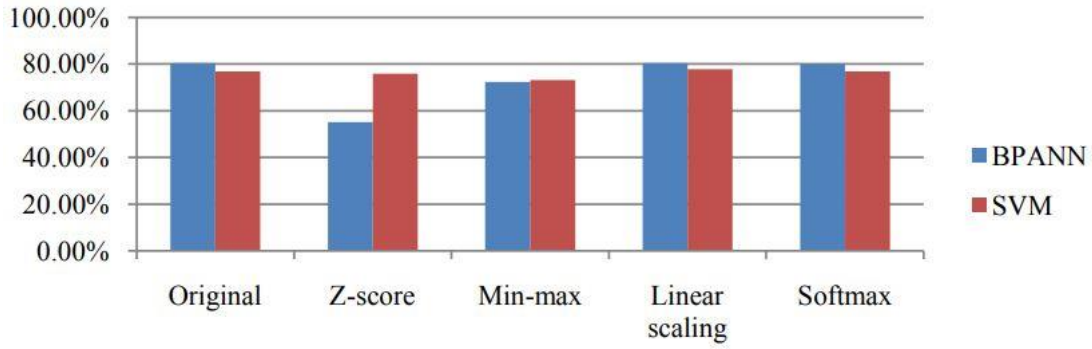
عملية ال (linear scaling) ل (Pixels) الصورة إلى المجال [0,1] م[24]

- و هاته بعض النتائج التي توضح تأثير تقنيات ال (Data Normalization) على دقة تصنيف ال (Model)

Classifier model	Original data	Normalization technique			
		Z-score	Min-max	Linear scaling	Softmax
BPANN	80.35%	55.2%	72.4%	80.35%	80%
SVM	76.89%	75.9%	73.1%	77.89%	76.8%

الشكل03: جدول يوضح دقة (SVM Model) و (BPANN Model) على البيانات الأصلية و على البيانات بتقنيات (Normalization) مختلفةم[24]

- كذلك هنا فقط التوضيح بأعمدة بيانية



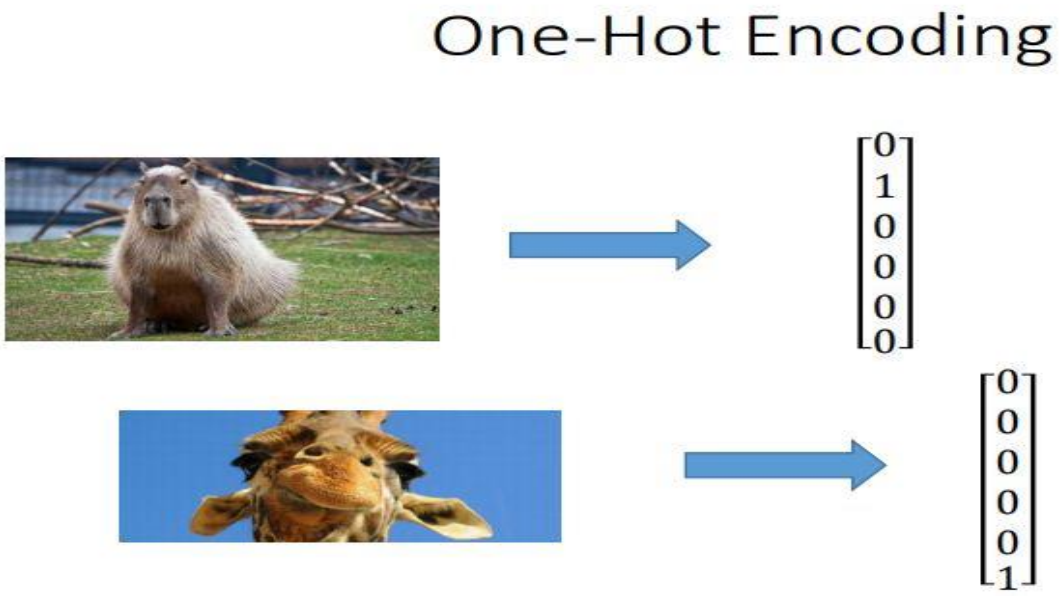
الشكل رقم 04: أعمدة بيانية توضح مقارنة مختلف تقنيات ال (Normalization) إعتمادا على أداء (SVM Mode) و (BPANN Model) م[24]

.2.2 (One-Hot Encoding):

يستخدم في التصنيف المتعدد الفئات (multi-class classification) لترميز تسميات الهدف

عند عملية التدريب نمرر لل (Model) صورة من ال (Dataset) بإسمها من أي صنف حتى يتدرب لاكن تمرير التسمية لا يكون بالشكل المعروف في حياتنا اليومية، مثلا لدينا (Model) يصنف ثلاثة أنواع من الحيوانات هي (القطط ،الطيور، الخيول) يعني لدينا ثلاثة فئات نريد أن يتدرب عليها كل صورة من ال (Training Dataset) إما أن تكون (قط أو طير أو خيل) إذا كانت الصورة من الصنف الأول قط تسمى (cat) إذا كانت من الصنف الثاني طير تسمى (bird) إذا كانت من الصنف الثالث خيل تسمى (horses) هاته التسميات العادية لاكن فالتسميات الفئوية الأمر مختلف تكون على شكل مصفوفة نستخدم ثلاث سمات رقمية لتمثيل الأصناف الممكنة للصورة من أي فئة حيث في الخانة التي رقمها يمثل صنف الصورة نضع 1 و البقية 0م[25] م[26]:

- تسمية القط [1,0,0]
- تسمية الطير [0,1,0]
- تسمية الخيل [0,0,1]



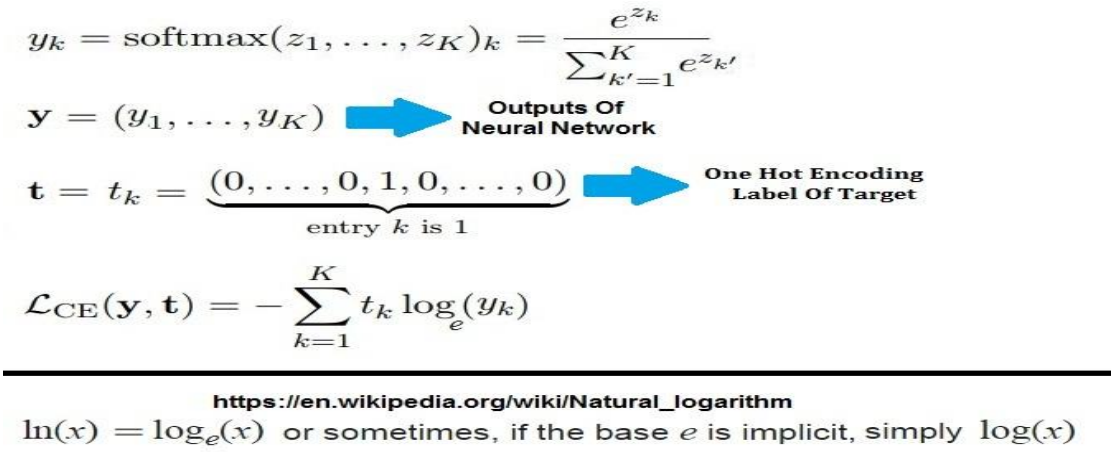
الصورة رقم 24: توضح التسمية الفئوية لحيوانين من بين 6 فئات حيوانيةم[26]

3. آلية تدريب النماذج :

3.1. مفاهيم أساسية لتطبيق عملية ال (Back propagation)

a) <u>(Loss Function):</u>

هي دالة تربط بين مخرجات ال (Neural network) والقيم المثالية التي نقيدها بها، نشتق هاته الدالة لنرى مدى تلبية ال (Neural Network) للقيود المفروضة على مخرجاتها حتى تكون قريبة من القيم المثاليةم[31]، وهي عدة أنواع أشهرهم وأكثرهم شيوعا هي ال (Cross entropy Loss Function) م[39].



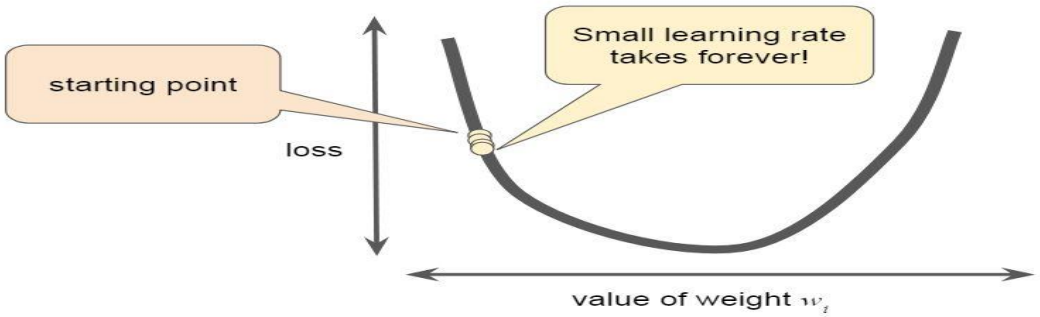
الصورة رقم 25: توضح صيغة مجموع قيم دالة الخسارة (Cross Entropy) لمخرجات ال (Neural Network) بدلالة التسمية الفئوية للهدف (One Hot Encoding Label) و (Softmax Activation Function) م[40]

ملاحظة: المقصود ب (Log) هو اللوغاريتم الطبيعي (Natural Logarithm) ذو الأساس (e) يعني (Ln(x)/Ln(e) = Ln(x)).

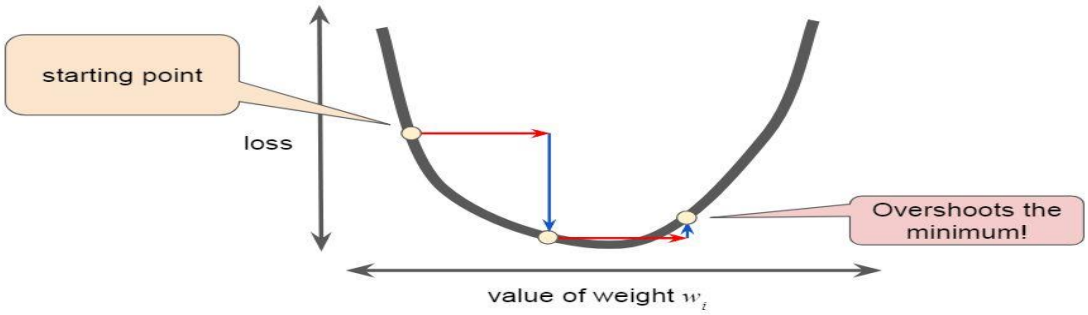
b) (Learning rate):

ال (Learning Rate) تختصر ب (Lr) ويسمى فالصيغ الرياضية عادة ب ألفا (<=> Alpha α) وهي معدل تحديث ال (Weights) فال (Model)، يعتبر ال (Learning Rate) إحدى ال (Hyperparameters) في مجال ال (Machine learning) حيث يقضي المبرمجون قدرا من الوقت لا بأس به في ضبط قيمة ال (Learning Rate)، إذا إخترنا قيمة (Learning Rate) صغيرة فسيستغرق تعلم ال (Model) زمن طويل كذلك إذا إخترنا قيمة (Learning Rate) كبيرة فقد نتخطى قيمة ال (Weights) المثالية التي تجعلنا فالنقطة الحدية لمنحنى دالة الخسارة والتي يكون عندها أصغر قيمة لدالة الخسارة م[32]، لحل هاته المشكلة نستخدم ال (Decay) الذي يقوم في كل مرة بإنقاص ال (Learning Rate) لأنه عندما نقترب من النقطة الحدية يجب أن يكون تحديث ال (Weight) بحذر ودقة حتى لا نتخطاها م[37].

ملاحظة: كل عملية تحديث ال (Weights) الخاصة بال (Model) تسمى (Iteration) وال (Iteration) تكون مساوية لعدد ال (Batch Size) الخاصة بال (Training Dataset).



الصورة رقم 26: توضح مدى تطور ال (Weight) بدلالة ال (Loss Fucntion) عند إستخدام قيمة (Learning Rate) صغيرةم[32]



الصورة رقم 27: توضح مدى تطور ال (Weight) بدلالة ال (Loss Fucntion) عند إستخدام قيمة (Learning Rate) كبيرةم[32]

$$\alpha_t = \alpha_0 e^{-t/\tau}$$

τ = is the decary timescale.

t = is the iteration count.

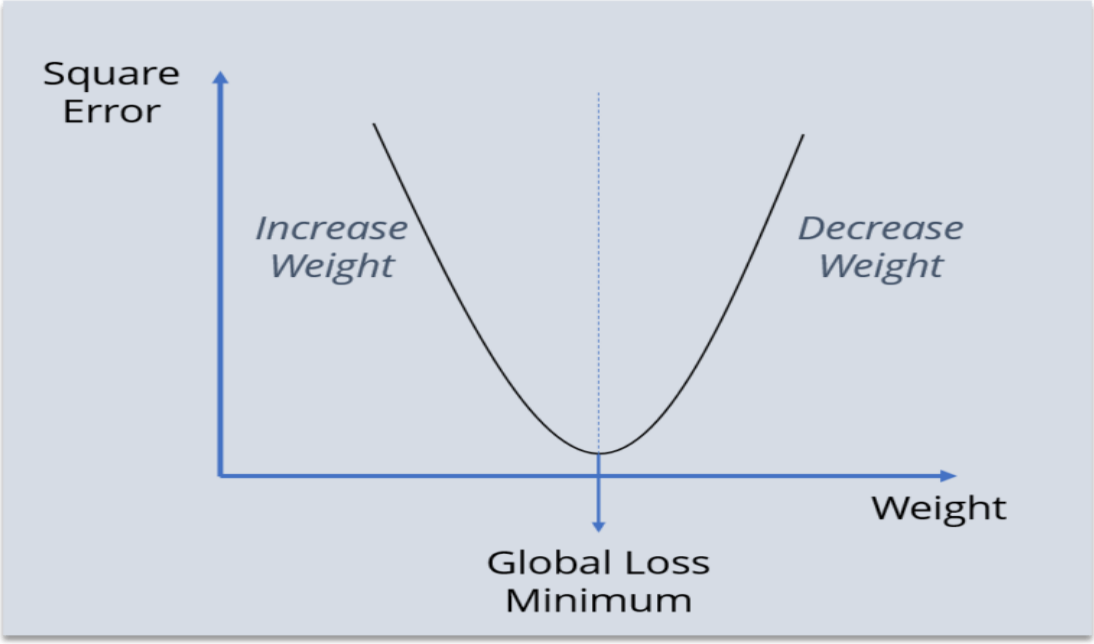
$t = 0$ Corresponds to the start of the decay.

معادلة تمثل تغير قيم الـ (Learning Rate) بدلالة الـ (Decay) والـ (Iteration) م[37]

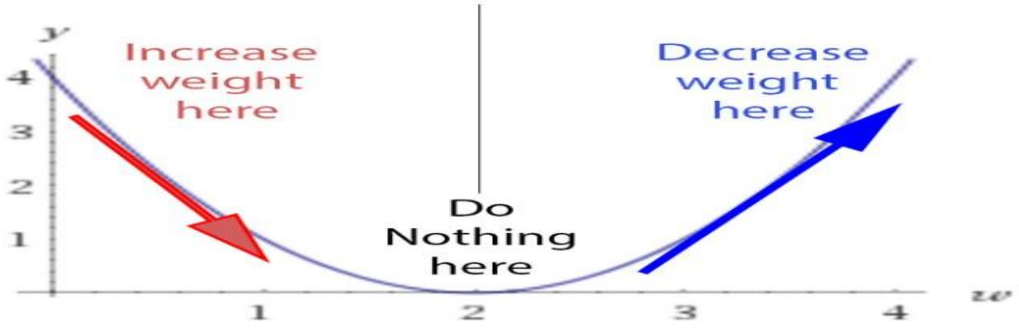
c) <u>(Gradient Descent):</u>

هي عملية حساب ال (Gradient) أو الميل (Slope) عند نقطة بداية معينة عادة تكون (0,0) لمنحنى تغير ال (Loss Function) بدلالة تغير قيمة ال (Weight) ثم إستنتاج طريقة تغيير قيمة ال (Weight) بأي إتجاه بهدف تغيير تموضع نقطة البداية إلى نقطة مختلفة من المنحنى يكون عندها قيمة

ال (Loss Function) أدنى و أقل من النقطة السابقة ، تستمر هاته العملية فالإنتقال من نقطة إلى نقطة من منحنى تغير ال (Loss Function) بدلالة تغير ال(Weight) حتى نحصل فالنهاية على إحداثيات النقطة من المنحنى التي بدورها تعطينا قيمة ال (Weight) المثالية التي يكون عندها أقل قيمة لل (Loss Function)، هاته العملية تحدث مع جميع ال (Weights) الخاصة بال (Model) ويكون ناتج ال (Gradient) لكل ال (Weights) على شكل متجه (Vector)، حتى نتحصل فالنهاية على جميع قيم الـ (Weights) الجديدة التي تجعل قيمة ال (Loss Function) في أدنى قيمة لها.م[33]



الصورة رقم 28: توضح كيفية تغير قيمة ال (Weight) إعتمادا على قيم الميل وإتجاهه في منحنى تغير ال (Weight) بدلالة تغير ال (Loss Fucntion) م[34]

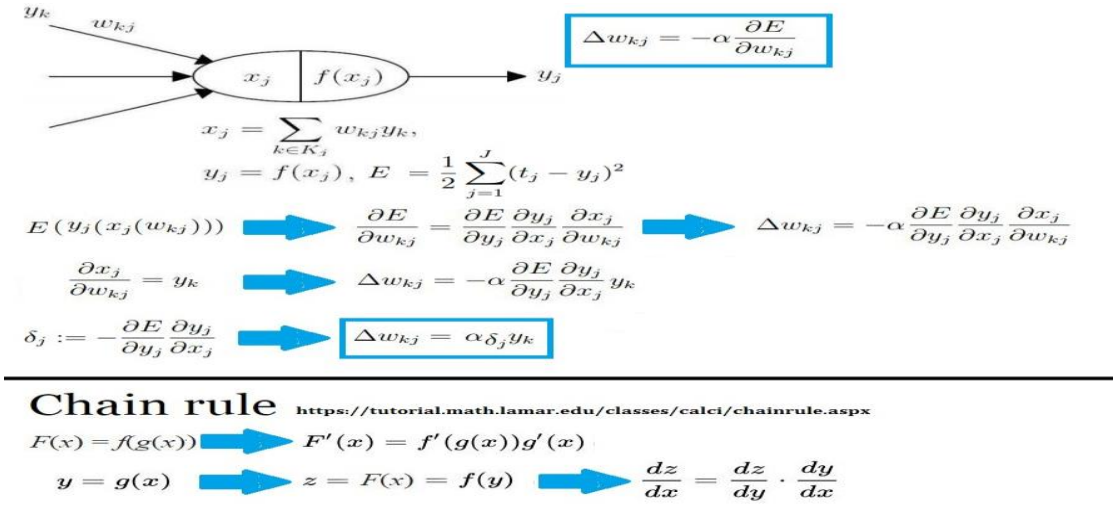


الصورة رقم 29: توضح كيفية تغير قيمة ال (Weight) إعتمادا على قيم الميل وإتجاهه في منحنى تغير ال (Weight) بدلالة تغير ال (Loss Fucntion) م[35]

:(Back Propagation) .3.2

هي خوارزمية تستخدم لتحديث ال (Weights) لشبكة عصبية (Neural network) ذات بنية ثابتة متعددة ال (Layers)، حتى تقلل مجموع أخطاء دالة الخسارة أو التكلفة (Cost Function or Error Function) بين مخرجات الشبكة والقيمة المثالية المستهدفة المحددة وبذلك يكون ال (Model) قد تدرب بشكل جيد، تعتمد هاته الطريقة على خوارزميات ال (Gradient Descent) م[30].

فالصورة التالية لدينا خطوات رياضية لكيف نقوم بحساب قيمة ال (Weight) الجديدة W^+(k,t) لل (Weight) القديمة W(k,t) التي تنتمي إلى أحد ال (Output Neurons) فال (Neural Network)، حيث ال (Activation Function) لل (Neuron) هي (Yj) و مجموع قيم ال (Loss Function) لمخرجات ال (Neural Network) هي (E) فالصورة إستخدم ال (Squard Error Function) ك (Loss Function):



الصورة رقم 30: توضح العمليات الحسابية لخوارزمية ال (Back Propagation) لحساب قيمة ال (Weight) الجديدة W^+(k,t). م[36]

لتطبيق عملية ال (Back Propagation) على ال (Model) نحتاج إلى خوارزميات ال (Gradient descent optimizations) وهم ثلاثة أنواع م[28]:

- ملاحظة :ال Sample هي الثنائية (image,label) أي صورة من ال (Trainig Dataset) مرفقة بإسمها م[27].

: Batch gradient descent

يستخدم كامل ال (Training Dataset) لحساب ال (Gradient) حتى تتم تحديث ال (Weights) لل (Model)

Stochastic gradient descent:

يستخدم كل مرة (Sample) واحدة فقط من ال (Training Dataset) لحساب ال (Gradient)

Mini-batch gradient descent:

يستخدم كل مرة جزء من ال (Training Dataset) ليس كلها وليس (Sample) واحدة فقط، لحساب ال (Gradient)

كلى الأنواع الثلاثة تمتلك عدة خوارزميات، الخوارزمية الأكثر شيوعا وإستخداما هيا (Adam) تنتمي إلى نوع ال (Mini-batch gradient descent) م[29].

خاتمة الفصل:

راينا في هذا الفصل مكونات النماذج وتعريفها كما رأينا خصائصها والياتها، كما رأينا محموعة البانات (Dataset) وكيف تسوى هذه البيانات وتصنيفها.

واخيرا وليس اخرا تحدثنا على اليات تدريب النماذج و المفاهيم الاساسية والخوارزميات المناسبة لهذه العملية.

في الفصل القادم سنحاول تطبيق ماتعلمناه في هذا الفصل فكيف سنقوم بذلك وماهي النتائج التي سنتحصل عليها ؟

الفصل الثالث: التصميم و الإنجاز

II. الفصل الثالث: التصميم و الإنجاز

مقدمة:

في هذا الفصل سنحاول تطبيق المفاهيم الاساسية لهذا العمل في برنامج ننشأه بلغة البايثون فماهي لغة البايثون وماهي استخداماته، وكيف سنطبق ذلك في العمل؟

1. البايثون (python):

لغة البايثون هي إحدى لغات البرمجة التي تم إنشاؤها في أواخر الثمانينات من القرن الماضي ، و تعد لغة البايثون من اللغات القوية و البسيطة حيث تسهل على المبرمجين تطوير التطبيقات بسرعة اضافة الى ذلك فإن لغة بايثون قابلة للتنفيذ في معظم أشهر أنظمة التشغيل منها windows و Mac و Os و.... ، و هذا يسمح لنا بتوزيع برامج python الخاصة بنا بطريقة مريحة بدون مطالبة المستخدمين بتثبيت مترجم python ، و يمثل معنى كلمة بايثون احدى أنواع الأفاعي .

كما تعد لغة بايثون لغة البتدئين و ذلك لبساطتها حيث أن هذه الأخيرة تتطلب عدد أقل من الأسطر البرمجية لتنفيذ المهمة مقارنة باللغات البرمجية الأخرى .

1.1. استخدامات البايثون :

- يمكن استخدام python تطوير البرمجيات .
- يمكن استخدام python على خادم لإنشاء تطبيقات الويب .
- يمكن استخدام python إلى جانب البرنامج للإنشاء مهام سير عمل .
- يمكن ل python الإتصال بأنظمة قواعد البيانات ، يمكنه أيضا قراءة الملفات و تعديلها .
- يمكن استخدام python للتعامل مع البيانات الضخمة و أداء الرياضيات المعقدة .
- يمكن استخدام python للنماذج الأولية السريعة أو لتطوير البرامج الجاهزة للإنتاج .

1.2. ال (Packages) المتوفرة على (Python) التي تمكن من إنشاء ال (Deep Learning Models):

❖ **(Keras):** مكتبة بايثون خاصة بالشبكات العصبية العميقة، متوفرة على عدة منصات مثل (Unix, Windows،...)، وتشتغل على عدة خلفيات مكاتب أخرى مثل (TensorFlow, Theano, CNTK،..)، وهي واجهة برمجة التطبيقات (API) عالية المستوى الرسمية ل

(TensorFlow) يمكن الولوج إليها ب (tensorflow.keras) م[45]، إنشاء النماذج بها بسيط للغاية عبارة عن تكديس عدة طبقات (Layers) وربط عملياتها الحسابية مع بعض م[42] تدعم كل من (Convolution, Recurrent layer، أو الجمع بينهما)، نمذجة سريعة كذلك تجعل المستخدم يتخطى تفاصيل تنفيذ ال (Back Propagation) أو كتابة وظائف التحسين بنفسه [43]، تدعم نماذج
(Keras) نوع بيانات من شكل مصفوفة (NumPy).

- **(Pandas):** مكتبة خاصة بتحليل ومعالجة البيانات[44]، سنستخدمها للتعامل مع ال (Dataset) الخاصة بنا كقرائة بيانات ال (Dataset) من الملفات ذات الصيغة (Csv) م[41].
- **(NumPy):**مكتبة خاصة بإنشاء مصفوفات متعددة الأبعاد تحتوي أيضا على عدة وظائف تطبق على المصفوفات[44]، سنستخدمها لإنشاء بعض المصفوفات التي سنضع فيهم صور و ال (Labels) الخاصة بال (Dataset).
- **(OpenCv):** مكتبة خاصة بمجال الرؤية الحاسوبية تحتوي على عدة مئات من خوارزميات الرؤية الحاسوبية [46] نحتاجها لقرائة الصور من مساراتها وعملية إعادة تحجيم الصور بأبعاد مختلفة كذلك بعض العمليات الأخرى.
- **(Matplotlib):** مكتبة خاصة برسومات الثنائية الأبعاد توفر طريقة سريعة لتصور البيانات في (Python) م[48]،سنستخدمها لرسم منحنيات تطور حالة التدريب (منحنيات دالة الخسارة لكل من ال (Validation) و ال (Training) كذلك منحنيات الدقة لكليهما أيضا) م[47].
- **(Scikit Learn):** مكتبة خاصة بخوارزميات ال (Machine learning) م[49]، سنستخدمها فقط لتقسيم بيانات ال (Dataset) للحصول على المجموعة الخاصة بال (Validation).

ملاحظة: يمكن تثبيت كل هاته ال (Packages) بإصدارات معينة تتوافق مع منصة العمل وبيئة البرمجة بسهولة، عن طريق مدير الحزم في بايثون (Pip) بتطبيق الأوامر التالية في موجه الأوامر.

```
pip install opencv
pip install numpy==1.16.4
```

pip install scikit-learn==0.22.2.post1

pip install matplotlib

pip install tensorflow==2.0.0-b1

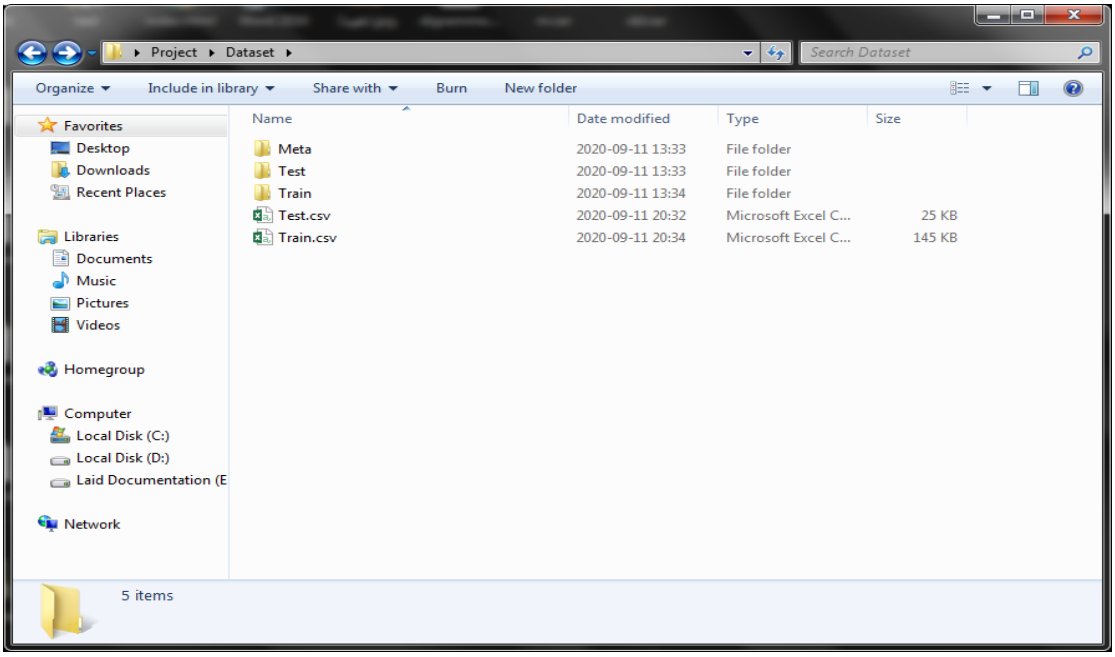
2. مرحلة التطبيق أو التنفيذ (Implementaton):

☒ فالبداية نقوم بعمل (Import) لجميع ال (Packages) التي نحتاجها.

```
import matplotlib.pyplot as plt
from tensorflow.python.keras.optimizers import Adam
from tensorflow.python.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
import cv2
import numpy as np
import argparse
import os
import pandas as pd
import tensorflow as tf

ap = argparse.ArgumentParser()
ap.add_argument("-d", "--dataset", required=True,
        help="path to our Dataset")
ap.add_argument("-o", "--output", required=True,
        help="path to output of training model")
args = vars(ap.parse_args())
```

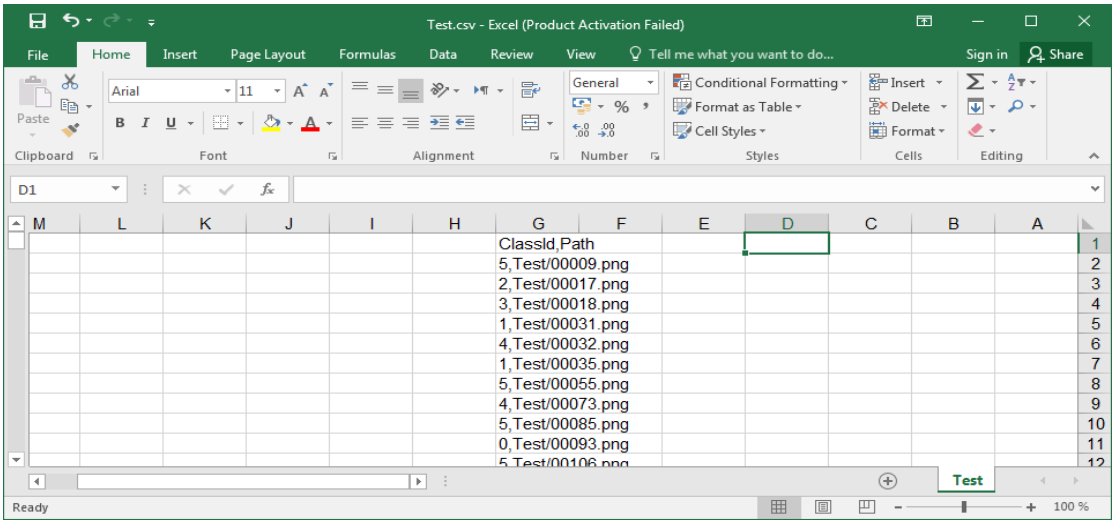
الصورة رقم 31: توضح عمل ال import لجميع المكاتب اللازمة

☒ ال (Dataset) التي سنستخدمها هي جزء صغير من ال (Dataset) المشهورة (German Traffic Sign Benchmarks Dataset) تتكون من 7 أنواع لوحات مرور، تحتوي على ملفين من نوع (Csv) هما (Test.csv) و (Train.csv) الأول خاص بال (Testing Dataset) والثاني خاص بال (Training Dataset)،كلى الملفان بهم عمودين عمود خاص بمسار الصور يسمى (Path) وعمود خاص بتصنيف الصور من أي فئة يسمى (Classid) يحتوي كلى الملفين عدة سطور كل سطر يحتوي على مسار الصورة وتصنيفها من أي (Class)، أيضا تحتوي ال (Dataset) على مجلدين خاصين بصور ال (Training Dataset) و (Testing Dataset)هما (Train) و (Test)، مجلد ال (Train) يحتوي على 7 مجلدات بتسميات مرقمة من ال 0 إلى 6 بهم صور مجموعهم 4470 صورة، كل مجلد منهم يحتوي على صور متنوعة لنوع واحد فقط من بين أنواع لوحات المرور الأخرى، بالنسبة لمجلد ال (Train) يحتوي على 1380 صورة متنوعة لل 7 لوحات مرور، يعني ال (Dataset) التي سنعمل عليها تحتوي على 5850 صورة.

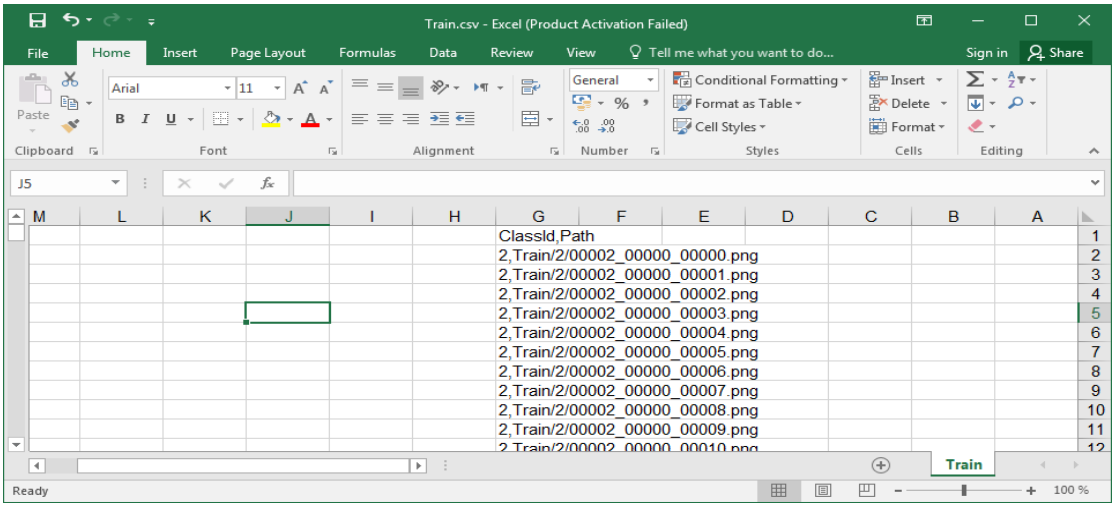


الصورة رقم 32: توضح ملفات الداتاسات

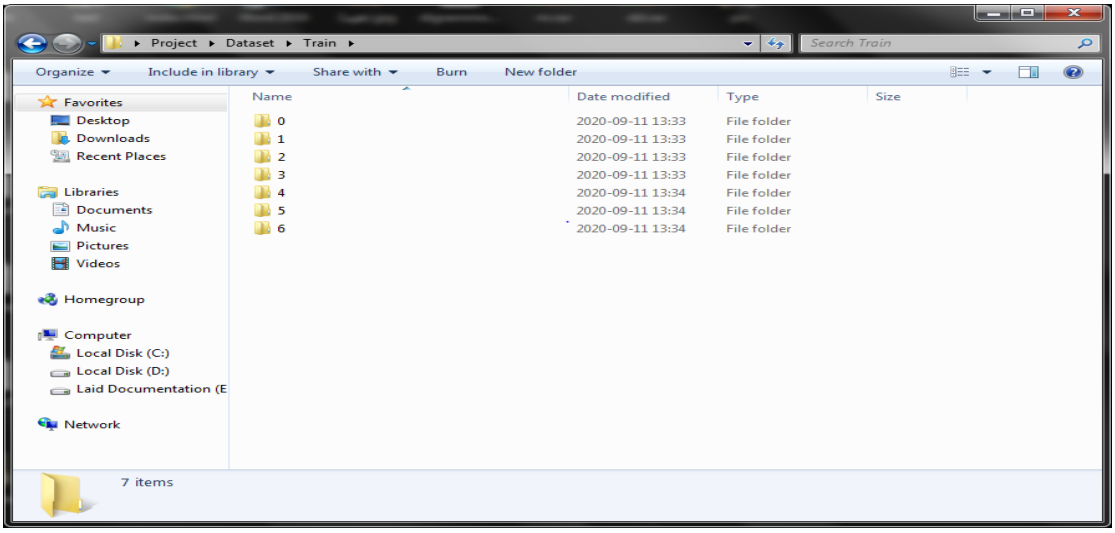
الصورة رقم 33: توضح لوحات المرور المتوفرة على الداتاسات



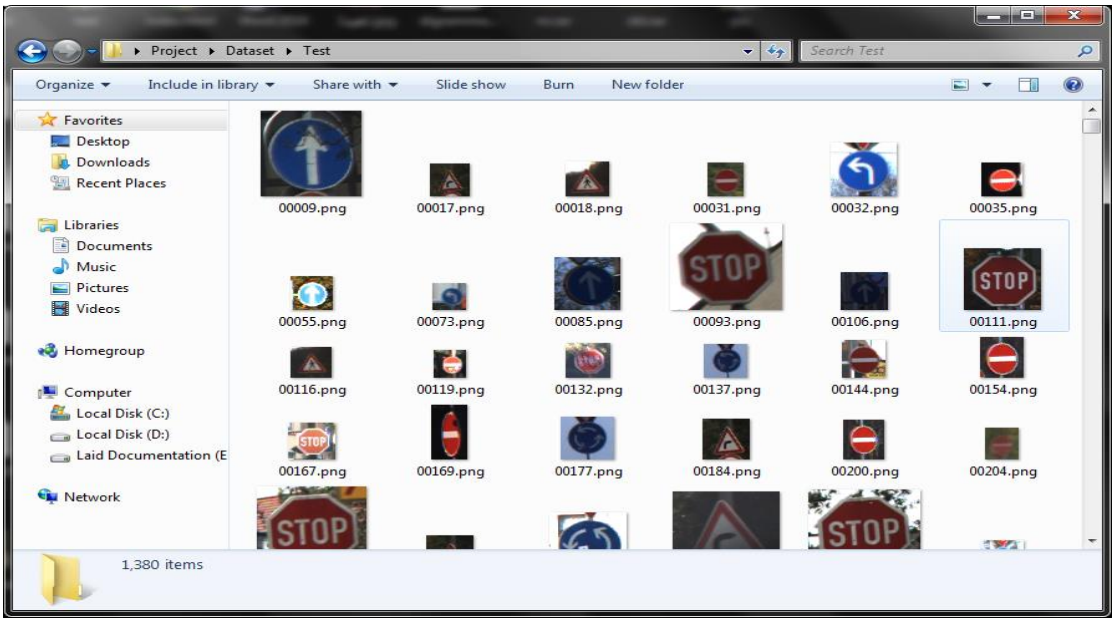
الصورة رقم 34: توضح محتوى ملف ال test.csv الخاص بالداتاسات



الصورة رقم 35: توضح محتوى ملف ال train.csv الخاص بالداتاسات



الصورة رقم 36: توضح محتوى مجلد ال train



الصورة رقم 37: توضح محتوى المجلد test الخاص بالداتاسات

☒ ال (Model) الذي سنستخدمه يتكون من 8 (Layers) متسلسلة وراء بعضها، حيث ال 5 (Layers) الأولى منه تعمل على إستخراج ال (Features) من الصور وتهيئتها لبقية ال 3 (Layers) التي تعمل على التصنيف (Classification)، نحدد أبعاد الصور الملونة التي يعالجها ال (Model) ب (32x32) نظرا لأن صور ال (Dataset) صغيرة كذلك لأن خوارزمية تصنيف الصور هي جزء من خوارزمية ال (Detection) يعني التصنيف يكون في أجزاء من الصور وليس في صور كبيرة كاملة لهذا فإن التصنيف (Classification) على العموم يتعامل مع صور صغيرة.

```
l0 = tf.keras.layers.Conv2D(32, kernel_size=(3,3), padding='same', activation='relu', input_shape=(32, 32, 3))
l1 = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))
l2 = tf.keras.layers.Conv2D(64, kernel_size=(3, 3), padding='same', activation='relu')
l3 = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))
l4 = tf.keras.layers.Flatten()
l5 = tf.keras.layers.Dense(256, activation='relu')
l6 = tf.keras.layers.Dropout(rate=0.5)
l7 = tf.keras.layers.Dense(7, activation='softmax')
model = tf.keras.models.Sequential([l0, l1, l2, l3, l4, l5, l6, l7])
```

الصورة رقم 38: توضيح هيكلة النموذج

☒ تستخرج وظيفة (load_data_preprocessing) من ملفات ال (Csv) مسارات الصور كذلك تصنيف الصور أو تسميتها ال (Labels)، بعد إستخراج جميع مسارات الصور تقوم الوظيفة بتحميل الصور من مساراتها في (List) ثم تحول هاته ال (List) إلى مصفوفة (Array) وتطبق عليها ال (Linear Scaling) لتجعل جميع قيم الصور فالمجال [0,1]، كذلك بالنسبة للتسميات بعد إستخراجهم من ملف ال (Csv) تقوم بوضعهم في (List) ثم تحول هاته ال (List) إلى مصفوفة (Array) وتطبق عليها عملية ال (One-Hot Encoding) لجعل التسميات (Labels) على شكل تسميات فئوية، تطبق هاته الوظيفة على ملف ال (Csv) الخاص بال (Training Dataset) وملف ال (Csv) الخاص بال (Testing Dataset) حتى نحصل فالنهاية على مجموعتين من البيانات الخاصة بال (Training) والخاصة بال (Testing).

```
def load_data_preprocessing(basePath, csvPath):
        data = []
        labels = []
        rows = pd.read_csv(csvPath)
        for i in range(rows.shape[0]):
                imagePath = os.path.sep.join([basePath, rows['Path'].values[i]])
                image = cv2.imread(imagePath)
                image = cv2.resize(image, (32, 32))
                data.append(image)
                labels.append(int(rows['ClassId'].values[i]))
        data = np.array(data)
        data = data.astype("float32")/255.0
        labels = np.array(labels)
        labels = to_categorical(labels, 7)
        return (data, labels)

print("[+] Loading training and testing data...")
trainPath = os.path.sep.join([args["dataset"], "Train.csv"])
(trainX, trainY) = load_data_preprocessing(args["dataset"], trainPath)

testPath = os.path.sep.join([args["dataset"], "Test.csv"])
(testX, testY) = load_data_preprocessing(args["dataset"], testPath)
```

الصورة رقم 39: توضح دالة تهيئة مجموعة البيانات قبل المعالجة

☒ تقسم مجموعة بيانات ال (Training) إلى مجموعتين هما مجموعة ال (Training) و (Validation) حيث حصتهم على التوالي 70% والأخرى 30%، نقوم بهاته العملية حتى نتحصل

على المجموعة الثالثة الخاصة ب (التحقق من التدريب) وهي على العموم تكون أصغر من مجموعة التدريب.

```
x_train,x_valid,y_train,y_valid = train_test_split(trainX,trainY,test_size = 0.3)
```

الصورة رقم 40: توضح تقسيم مجموعة التدريب للحصول على مجموعة تحقق

☒ نحدد نوع ال (Optimizer) الخاص بالتدريب وقيم ال (Decay) و ال (Learning Rate) الخاصة به ثم نقوم بعمل (Compile) للمودل بالوظيفة (compile) نرفق لها نوع ال (Loss Function) كذلك ال (Optimizer) الذي أنشأناه مسبقا وأيضا نوع المقاييس التي يتم رصد النموذج بها أثناء تدربه.

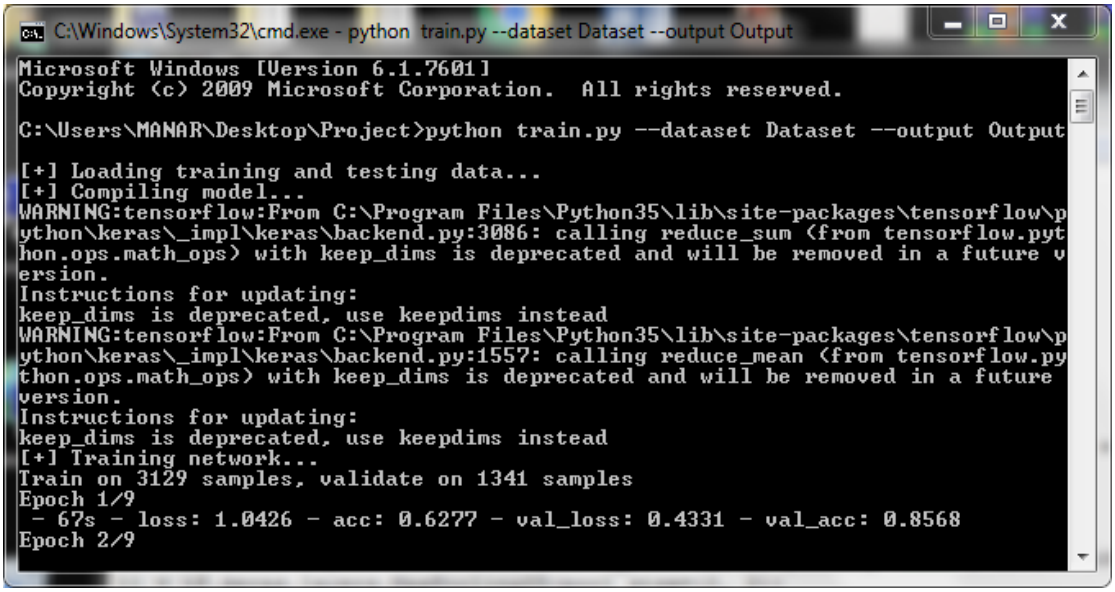
```
EPOCHS = 9
LR = 1e-3
OPT = Adam(lr=LR, decay=LR/EPOCHS)

model.compile(loss="categorical_crossentropy", optimizer=OPT, metrics=["accuracy"])
```

الصورة رقم 41: توضح تجديد قيم ال hyperparametrs و دالة ال optimizer ثم عمل compil للنموذج

☒ الآن تأتي مرحلة التدريب بالوظيفة (fit) نرفق لها بيانات التدريب (Training) وكذلك بيانات التحقق من التدريب (Validation) التي عالجناهم مسبقا، وأيضا قيمة ال (Batch Size) و عدد ال (Epochs)، ال (Verbose=2) للإضهار فالشاشة مراحل تطور التدريب دون تفاصيل كثيرة ونفعل الخاصية (shuffle) لكي يتم خلط ال (Batch) قبل تمريره للنموذج.

```
print("[+] Training network...")
H = model.fit(
        x_train,
        y_train,
        batch_size=64,
        validation_data=(x_valid, y_valid),
        epochs=EPOCHS,
        shuflle=True,
        verbose=2)
```

الصورة رقم 42: توضح الوظيفة fit الخاصة بتدريب النموذج و متغيراتها

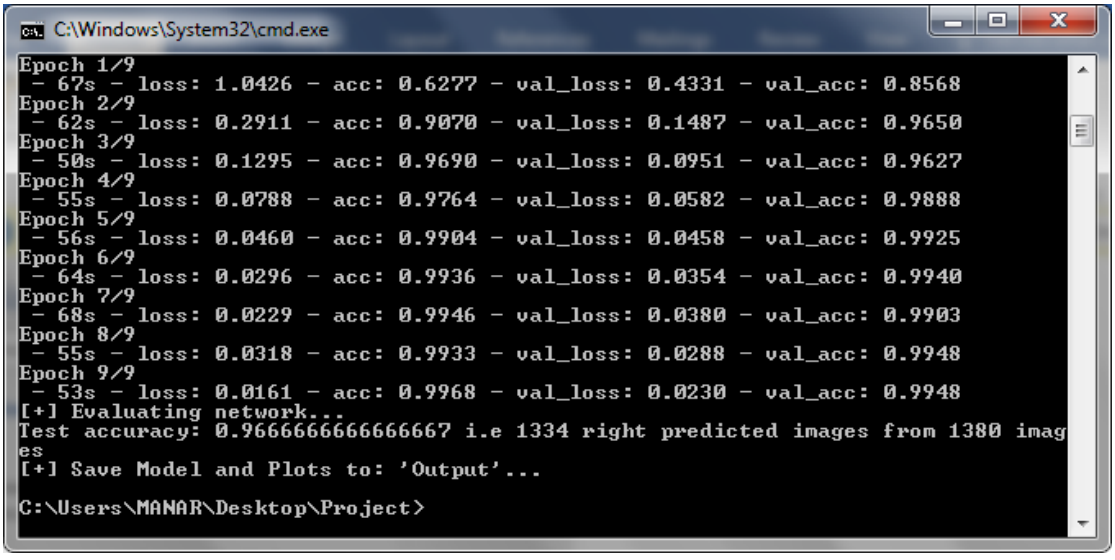


الصورة رقم 43: توضح بداية عملية تدريب النموذج

☒ بعد إنتهاء التدريب تأتي عملية تقييم النموذج على صور لم يراها مسبقا حتى نقيم آدائه، وذلك بالوظيفة (evaluate) نرفق لها بيانات ال (Testing) التي عالجناها مسبقا.

```
print("[+] Evaluating network...")
results = model.evaluate(testX, testY, verbose=0)
numTest = testX.shape[0]
prec = int(results[1]*numTest)
print("Test accuracy: {} i.e {} right predicted images from {} images".format(results[1],prec,numTest))
```

الصورة رقم 44: توضح الوظيفة evaluate الخاصة بتقييم النموذج و متغيراتها

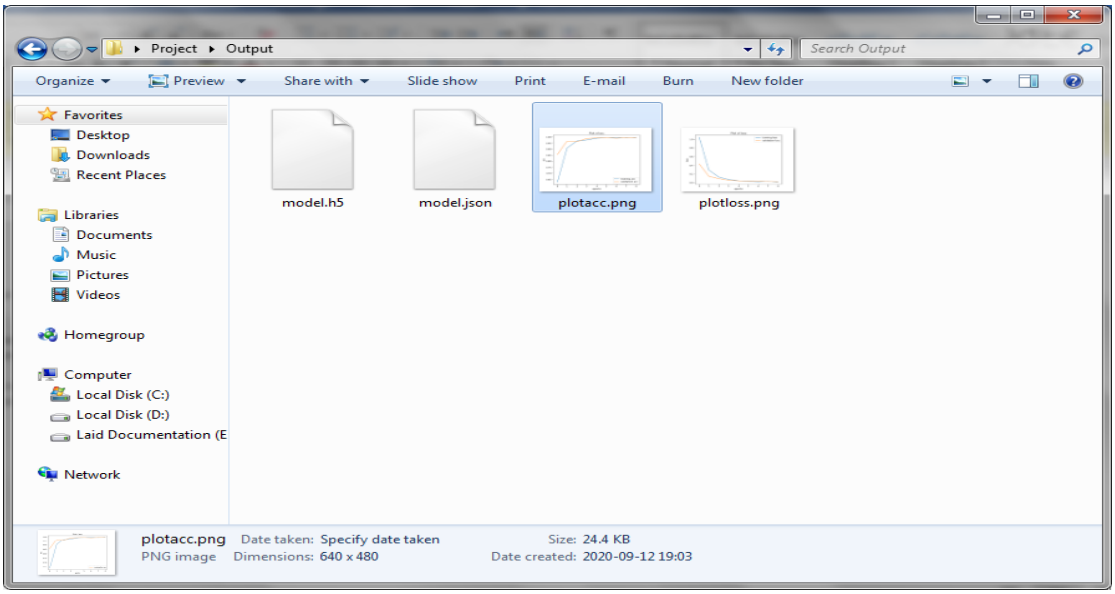


الصورة رقم 45: توضح النتائج النهائية للتدريب و تقييم النموذج

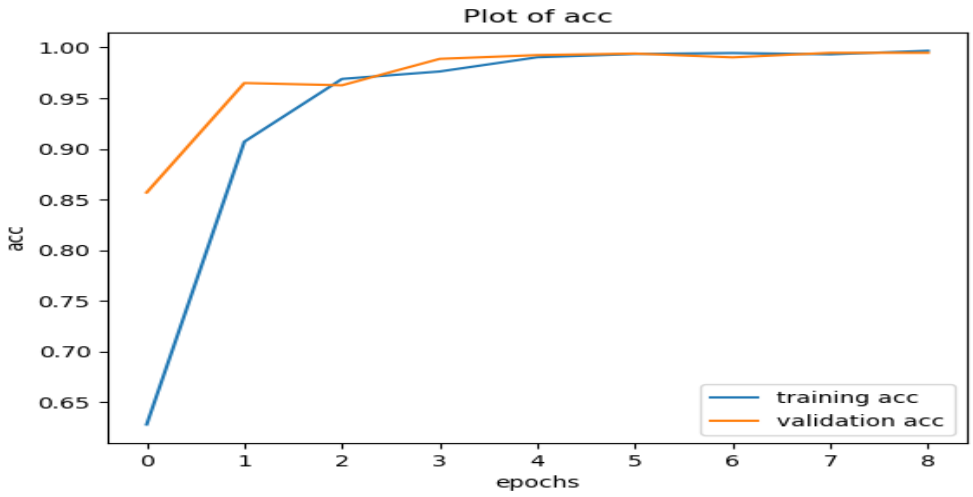
☒ أخيرا نقوم بحفظ النموذج المدرب حتى يتم تحميله مستقبلا متى أردنا التنبأ على صور لوحات مرور كذلك حفظ منحنيات ال (Accuracy) و ال (Loss Function) لكل من مجموعة ال (Training) و ال (Validation) على جميع ال (Epochs) كصور.

```
def plot(i):
  dic = {"0":'acc',"1":'loss'}
  plt.figure(i)
  plt.plot(H.history[dic[str(i)]], label='training '+ dic[str(i)])
  plt.plot(H.history['val_'+dic[str(i)]], label='validation '+dic[str(i)])
  plt.title("Plot of "+dic[str(i)])
  plt.xlabel('epochs')
  plt.ylabel(dic[str(i)])
  plt.legend()
  plt.savefig(args["output"]+"/plot"+dic[str(i)]+".png")

print("[+] Save Model and Plots to: '{}'...".format(args["output"]))
model_json = model.to_json()
with open(args["output"]+"/model.json", "w") as json_file:
    json_file.write(model_json)
model.save_weights(args["output"]+"/model.h5")
plot(0)
plot(1)
```

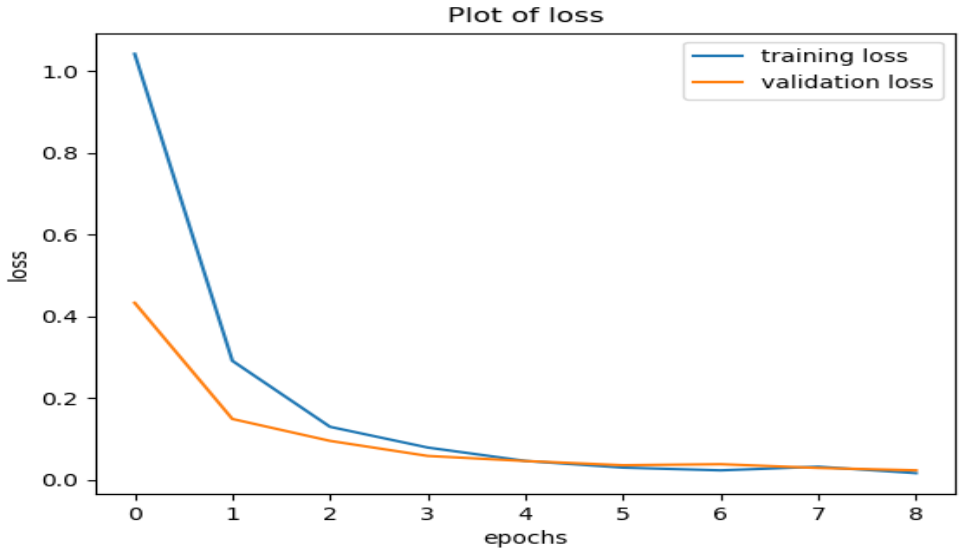
الصورة رقم 46: توضح وظيفة رسم مخططات التدريب و التحقق



الصورة رقم 47: توضح مخرجات محفوظات النموذج



الصورة رقم 48: توضح تزايد منحنيات الدقة لمجموعة التدريب و التحقق



الصورة رقم 49: توضح تناقص منحنيات دالة الخسارة لمجموعة التدريب و التحقق

خاتمة الفصل:

راينا في هذا الفصل تطبيق المعلومات المكتسبة في الفصل الثاني في عمل قائم على لغة البايثون، حيث عرفنا لغة البايثون وعرفنا استخداماتها، كماعرفنا الحزم والخوارزميات المتوفرة على البايثون، ثم ذهبنا الى مرحلة التطبيق والتنفيذ، حيث قمنا بكتابة النموذج ثم قمنا بتدريبه وبعد ذلك قيمنا النتائج.

خاتمة عامة:

اخيرا وليس اخرا، كان هذا المشروع المتمثل في بناء نموذج في مجال التعلم العميق لتصنيف صور لوحات المروربأستخدام الشبكة عصبونية التفافية (Convolutional neural network) ووتعلم الالة (Learning machine)، اعتمادا على ادوات منها مكتبات البايثون مثل: Pandas, TensorFlow, OpenCv, NumPy, Matplotlib, NumPy, SkLearn, .

حيث كان البرنامج يعمل بكفاءة فقد تحصلنا في النهاية على نتائج مرضية، حيث كانت الدقة 96%، الا انه يجد بعض الجوانب يمكن تحسينها مثل توسيع مجموعة البيانات بال (Augmentation) وبعض التقنيات الأخرى مثل (batchnormalization) .

وفي الختام نقول ان اهمية البرنامج تكمن في الفائدة التي يقدمها للانسان، و نسأل الله ان يوفقنا لأداء أعمالنا بأكمل وجه ونرجوا ان نكون قد قدمنا بعملنا هذا فائدة ولو بسيطة.

قائمة المصادر والمراجع:

[1] https://www.math.purdue.edu/~nwinovic/deep_learning.html

[2] http://www.cs.toronto.edu/~rgrosse/courses/csc321_2017/readings/L11%20Convolutional%20Networks.pdf

[3] https://docs.microsoft.com/en-us/learn/modules/train-evaluate-deep-learn-models/4-convolutional-neural-networks

[4] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[5] https://towardsdatascience.com/understanding-your-convolution-network-with-visualizations-a4883441533b

[6] https://classroom.udacity.com/courses/ud187

[7] https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

[8] https://web.stanford.edu/class/cs224n/readings/gradient-notes.pdf

[9] https://www.cs.toronto.edu/~lczhang/360/lec/w02/terms.html

[10] http://www.cs.toronto.edu/~rgrosse/courses/csc321_2017/readings/L11%20Convolutional%20Networks.pdf

[11] https://www.cs.toronto.edu/~hinton/coursera/lecture1/lec1.pdf

[12] https://docs.microsoft.com/en-us/learn/modules/train-evaluate-deep-learn-models/4-convolutional-neural-networks

[13] https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

[14] https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60

[15] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[16] https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[17] https://medium.com/@cdabakoglu/what-is-convolutional-neural-network-cnn-with-keras-cab447ad204c

[18] https://classroom.udacity.com/courses/ud187

[19] https://people.engr.tamu.edu/ajiang/636_06.pdf

[20] https://tigerprints.clemson.edu/cgi/viewcontent.cgi?article=3918&context=all_theses

[21] http://acsweb.ucsd.edu/~yuw176/report/ECE273.pdf

[22] https://its.fsu.edu/sites/g/files/upcbnu1011/files/ISPO%20Pages/What%20is%20a%20Dataset%201.1.pdf

[23] https://wp.wwu.edu/machinelearning/category/uncategorized/

[24] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.695.1851&rep=rep1&type=pdf

[25] https://www.cs.toronto.edu/~lczhang/360/lec/w05/lec09.pdf

[26] https://www.cs.toronto.edu/~guerzhoy/321/lec/W04/onehot.pdf

[27] https://public.lanl.gov/jt/Papers/mpml-R5-post.pdf

[28] http://niaohe.ise.illinois.edu/ie598/7_GD%20for%20DL_song.pdf

[29] http://www.cs.toronto.edu/~sajadn/sajad_norouzi/ECE1505.pdf

[30] https://www.cs.swarthmore.edu/~meeden/cs81/s10/BackPropDeriv.pdf

[31] http://web.cs.ucla.edu/~guyvdb/papers/XuArxiv17.pdf

[32] https://developers.google.com/machine-learning/crash-course/reducing-loss/learning-rate

[33] https://developers.google.com/machine-learning/crash-course/reducing-loss/gradient-descent

[34] https://www.edureka.co/blog/backpropagation/

[35] https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a-simple-way-f540a3611f5e

[36] https://www.cs.cornell.edu/courses/cs5740/2016sp/resources/backprop.pdf

[37] https://www.cs.toronto.edu/~lczhang/321/notes/notes08.pdf

[38] https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9

[39] https://towardsdatascience.com/cross-entropy-for-classification-d98e7f974451

[40] http://www.cs.toronto.edu/~rgrosse/courses/csc321_2017/readings/L04%20Training%20a%20Classifier.pdf

[41] http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset

[42] https://web.stanford.edu/class/cs20si/lectures/march9guestlecture.pdf

[43] https://www.crcv.ucf.edu/wp-content/uploads/2019/03/CAP6412_Spring2018_KerasTutorial.pdf

[44] https://johnfoster.pge.utexas.edu/numerical-methods-book/ScientificPython_Pandas.html

[45] https://keras.io/getting_started/intro_to_keras_for_engineers/

[46] https://docs.opencv.org/master/d1/dfb/intro.html

[47] https://matplotlib.org/tutorials/introductory/pyplot.html

[48] http://scipy-lectures.org/intro/matplotlib/

[49] https://towardsdatascience.com/an-introduction-to-scikit-learn-the-gold-standard-of-python-machine- learning-e2b9238a98ab

[50] تعلم الالة (Machine learning))، عبد الرحمان عثمان، نسخة غير منقحة قيد التطوير ،مارس 2018،اخر تطويرأبريل 2018،ص6

[51] https://www.nmthgiat.com/

[52] الذكاء الإصطناعي ،تأليف بلاي ويتباي، ترجمة قسم الترجمة بدار الفاروق –ط01- ،القاهرة للإستثمارات الثقافية (ش،م،م) سنة 2008، ص18

[53] https://matlab4engineers.com/2017/09/15/convolutional-neural-networks/?lang=en&v=fa3c7f2b5dae