



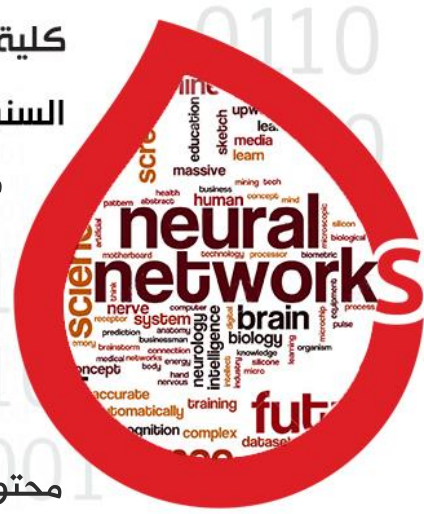
كلية الهندسة المعلوماتية
السنة الرابعة - الذكاء الصناعي

محتوى مجاني غير مخصص للبيع التجاري

CNN

د. ياسر خضرا

محتوى مجاني غير مخصص للبيع التجاري



RB Informatics;

الشبكات العصبونية

الشبكات العصبونية الالتفافية Convolutional Neural Network

وهو نوع من الشبكات العصبونية التي تعمل على الصور حيث يكون الدخل لها عبارة عن صورة.

ما هي الصورة؟

الصورة كما نراها بأعيننا هي عبارة عن ألوان وأشكال ولكن الحاسوب يراها ويتعامل معها على أنها مصفوفة، كل بكسل فيها يحمل درجة لونية معينة حسب نوع الصورة (ملونة - أبيض وأسود - رمادية...الخ).



What we see

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

أنواع الصور:

1. الصورة الرمادية: Grayscale

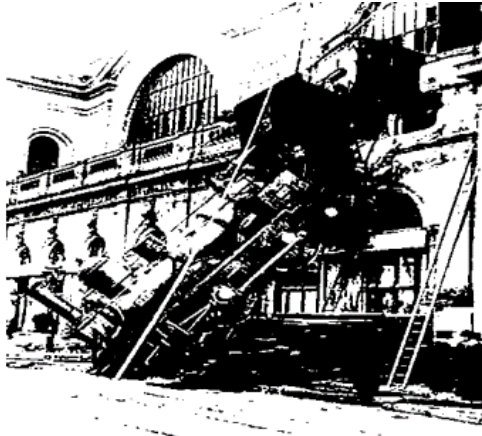
وهي عبارة عن صورة وجميع ألوانها هي تدرجات الرمادي، وتتمثل بمصفوفة قيمة كل عنصر فيها بين ال 0 وال 255 ويتخزن ب 8 بت، وبالتالي كل لون يتمثل ب بت واحد، ويستخدم هذا النوع من الصور لمشاهدة بريق الصورة.



محتوى مجاني غير مخصص للبيع التجاري

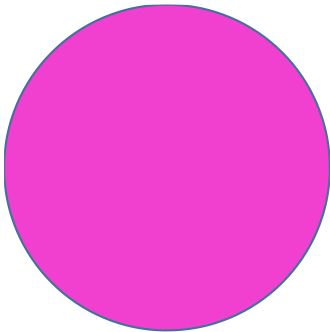
2. الصورة الثنائية: Binary image

وهي صورة تتمثل بمصفوفة قيم عناصرها إما 0 أو 255 (0 أسود، 255 أبيض)، وتستخدم هذه الصور في اكتشاف الحدود بالصورة، الرسم الإلكتروني و...الخ.

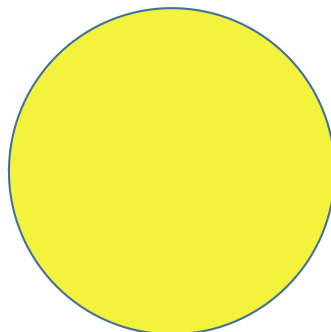


3. الصور الملونة: Colour image

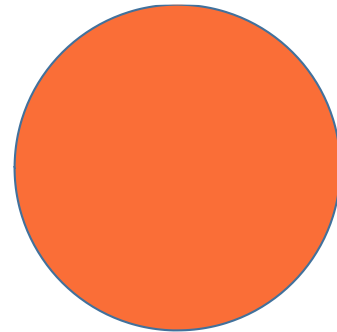
وتتألف من مصفوفة كل عنصر فيها يتألف من ثلاث قيم تمثل كل قيمة تدرج اللون من الألوان الأساسية الثلاث (أحمر - أزرق - أخضر RGB)، وكل قيمة تكون محصورة بين ال 0 و 255 وبالتالي سيتم تخزين كل عنصر من المصفوفة ب 32 بت ($8 \times 3 = 32$) وتغييرات القيم بين هذه القيم الثلاث يولد لدينا الألوان الأخرى، حيث أن لكل لون درجة معينة من كل لون أساسي.



RED : 241
GREEN : 63
BLUE : 207



RED : 243
GREEN : 243
BLUE : 61



RED : 250
GREEN : 110
BLUE : 55

مقارنة بين الأنواع الثلاث:



Color image
24 bits (RGB)

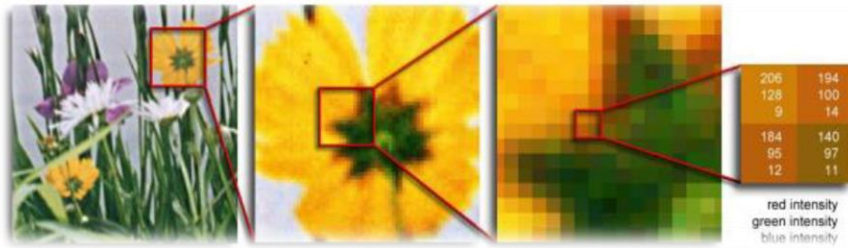
Gray image
8 bits

Binary image
1 bit

والصورة الملونة عبارة عن ثلاث صور رمادية، حيث كل مركبة من المركبات الثلاث لعناصر المصفوفة الممثلة للصورة الملونة يمكن التعبير عنها بصورة رمادية ولكل لون قناة تعبر عن درجة هذا اللون في الصورة، للتوضيح:



نلاحظ درجات الألوان في الصورة حيث في الصورة RED يعبر اللون الأبيض عن درجة اللون الأحمر في هذا البكسل من الصورة واللون الأسود عن اللون الغير أحمر، وتدرجات الرمادي هي تدرجات اللون الأحمر في هذا البكسل من الصورة وكذلك بالنسبة للصورة GREEN و BLUE، ويمكن التعبير عن هذه المركبات الثلاث بما يسمى القيمة اللونية للبكسل



إحداثيات البكسلات بالصورة:

لكل بكسل في الصورة إحداثيات معينة، ولنفرض تبدأ الإحداثيات من الزاوية العلوية اليسرى ويكون البكسل فيها قيمته 0,0، مثال: لتكن لدينا صورة أبعادها 4×4 فتمثل بالشكل التالي:

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3



والصورة اللونية هي عبارة عن ثلاث مصفوفات متراكبة فوق بعضها.

ومن أجل تدريب الشبكات العصبونية قد تم إنشاء مكتبة صور تدعى Image.net
<https://www.image.org.net>

وهي مكتبة ضخمة من الصور مجانا وتوجد فيها مجموعات صور مدربة، أي أن الصورة تكون مع الوصف أو التوقع الخاص لها، مثال:



وبالتالي القدرة على معرفة ماهية الصورة، وتدريب الشبكة العصبونية بعينات تدريب مختبرة وصحيحة.

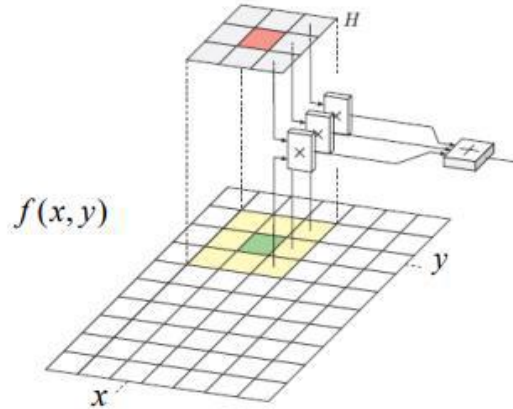
المرشحات filters

وتعني إضافة تأثير معين على الصورة مثل جعلها باللون الأبيض والأسود إذا كانت ملونة، جعل الصورة أكثر إضاءة ... والعديد من التقنيات.

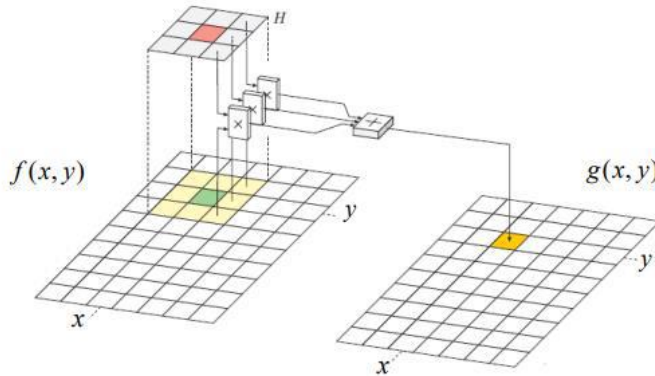
✓ نعلم بأن الصورة تتكون من بكسلات ولكل بكسل بكسلات مجاورة وتساعد هذه العملية في تحديد البكسلات التي ستتأثر بالفلتر.

🔗 تطبيق الفلتر:

نقوم بإنشاء ما يسمى بالقناع Mask وهو عبارة عن مصفوفة بنفس أبعاد مصفوفة الصورة التي نريد تطبيق الفلتر عليها وقد تختلف بالحجم.



في الصورة أعلاه مصفوفة تعبر عن الصورة التي نريد تطبيق الفلتر عليها ومصفوفة قناع 3×3 وتتم عملية الفلتر بإنشاء مصفوفة جديدة بنفس المصفوفة المعبرة عن الصورة التي نريد تطبيق الفلتر عليها ونضرب كل قيمة في بكسل الصورة القديمة بقيمة البكسل القناع المطبق على هذه البكسلات (حسب حجم القناع).



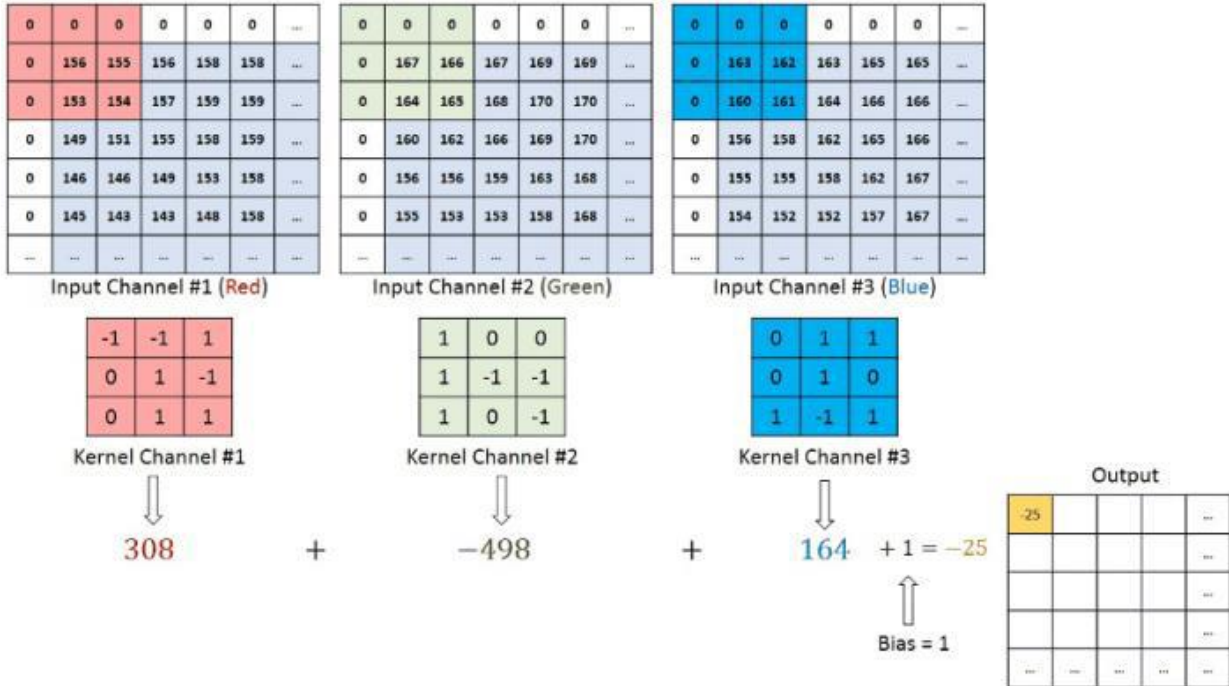
في حال كانت الصورة ملونة والقناع يحوي على ثلاث قنوات (أحمر - أزرق - أخضر) فإننا نضرب كل قيمة بالقناع مع القناة الموافقة له ونجمع القيم الثلاث وينتج لدينا القيمة الجديدة للبكسل:

وبالتالي يمكن التعبير عن الفلتر بالعلاقة التالية:

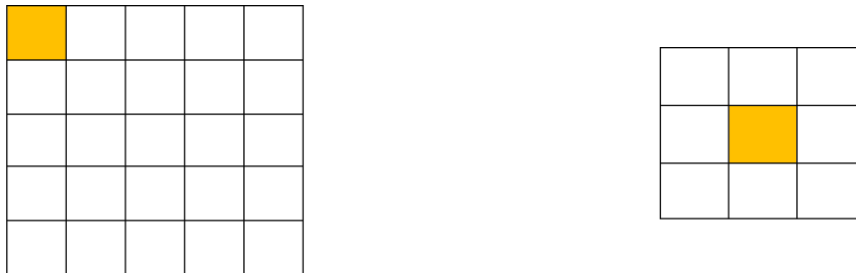
$$\text{Kernel Size} = (2n - 1) \times (2m - 1)$$

$$g(x, y) = f(x, y) * H = H * f(x, y) = \sum_{i=-(n-1)}^{n-1} \sum_{j=-(m-1)}^{m-1} f(x - i, y - j) \cdot H(i, j)$$

حيث:

 g : الصورة الجديدة بعد الفلتر H : القناة المطبق على الصورة f : الصورة التي طبقنا عليها الفلترو $f * H$ تعني طي الصورة f حسب القناة H و x, y تدل على موقع البكسل في الصورة

تقوم العملية على وضع مركز القناة على أول بكسل بالصورة على فرض الصورة والقناة من الشكل:



هنا ستواجهنا مشكلة وهي بأن القناة لم يغطي الصورة بشكل كامل.

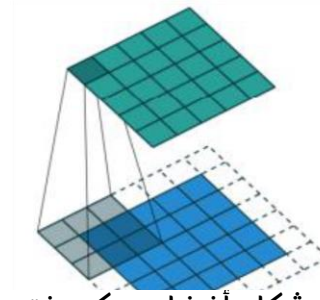
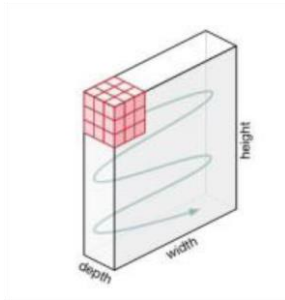
0	0	0
0	136	155
0	130	156

فكما نلاحظ بأن القناة لم يغطي الصورة ولذلك فإننا نضع ما يسمى Padding وهو عبارة عن عناصر فارغة نعبئها بالقيمة صفر والغرض منها هي أن تبقى حدود القناة بداخل الصورة.

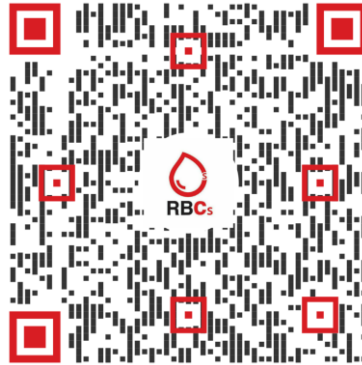
0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...



نقوم بإضافة أطراف للصورة وقيمتها 0 وبالتالي القناع سيبقى ضمن حدود الصورة.
يتحرك القناع إلى اليمين وينفذ العملية على البكسل التالي حتى الوصول لنهاية الصورة والحركة تكون من اليسار إلى اليمين.



ولمشاهدة الحركة بشكل أفضل يمكن فتح الصورة:

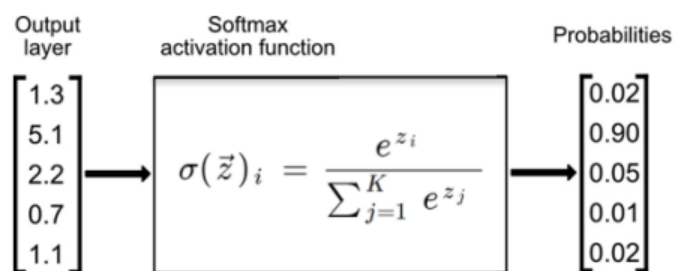


■ ملاحظة: في القناع السابق العملية كانت ضرب وجمع القيم، ولكن من الممكن أن يكون القناع يقوم بأمور مختلفة مثل توسيط القيم المطبق عليها وبالتالي ستتشابه الدرجات اللونية في البكسلات المتجاورة وتقل درجة الحدود وهو الفلتر المستخدم في تنعيم الوجه.

Softmax

وهو تابع يحول مصفوفة من القيم الحقيقية الى مصفوفة من القيمة الحقيقية حيث يكون مجموعها 1 (احتمالات).

مثال:



وبالتالي يستخدم بالشبكات العصبونية التي تتعامل مع الاحتمالات حيث كلما ارتفعت القيمة ارتفعت الاحتمالية.

مثال على استخدام التابع:

ليكن لدينا الشعاع التالي: $\begin{bmatrix} 8 \\ 5 \\ 0 \end{bmatrix}$
بالبداية يجب حساب المقام:

$$\sum_{j=1}^K e^{z_j} = e^{z_1} + e^{z_2} + e^{z_3} = 2981.0 + 148.4 + 1.0 = 3130.4$$

ومن ثم حساب البسط لكل قيمة:

$$e^{z_1} = e^8 = 2981.0$$

$$e^{z_2} = e^5 = 148.4$$

$$e^{z_3} = e^0 = 1.0$$

ومن ثم نطبق علاقة التابع عند كل قيمة من قيم الشعاع للحصول على القيم الجديدة:

$$\sigma(\vec{z})_1 = \frac{2981.0}{3130.4} = 0.9523$$

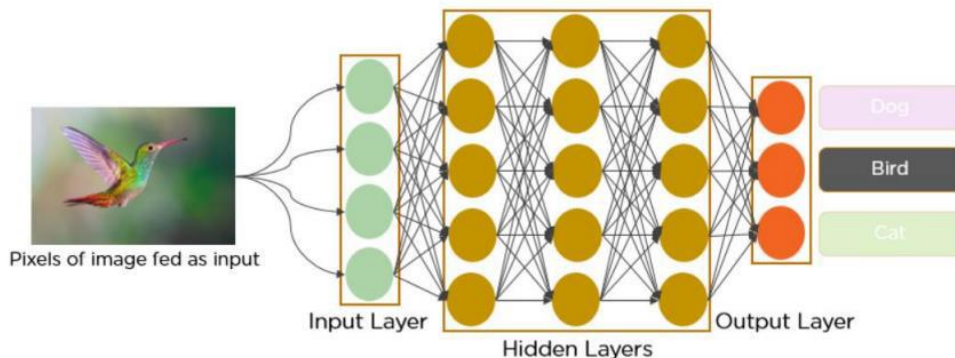
$$\sigma(\vec{z})_2 = \frac{148.4}{3130.4} = 0.0474$$

$$\sigma(\vec{z})_3 = \frac{1.0}{3130.4} = 0.0003$$

وبالتالي الشعاع الناتج $\begin{bmatrix} 0.9523 \\ 0.0474 \\ 0.0003 \end{bmatrix}$ ونلاحظ أن مجموع القيم الناتجة هو 1.

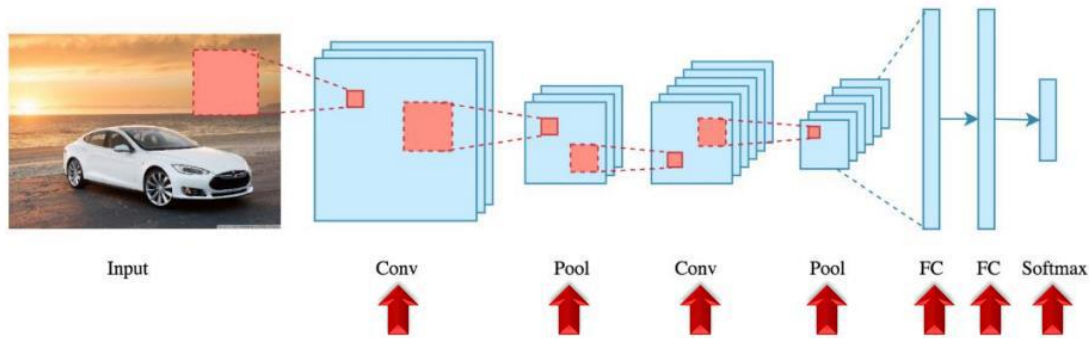
Convolutional Neural Network (CNN)

وهي نوع من الشبكات العصبونية التي تحاكي رؤية الانسان، ودخلها صورة وخرجها تصنيف على سبيل المثال. مثال لشبكة عصبونية تقوم بتحديد نوع الحيوان بالصورة المعطاة لها.



✓ وكما نلاحظ وجود أنواع الطبقات الثلاث (دخل - مخفية - خرج) ولكن عدد العصبونات يختلف، إن عدد العصبونات في كل شبكة يختلف حسب الحاجة ويوجد العديد من التصميمات والهيكلية للشبكة العصبونية ولكل منها فوائد ومزايا، وأحد الأمثلة على البنية هي بنية الهرم وتكون فيها الشبكة متخذة

شكل الهرم وقاعدة الهرم هي طبقة الدخل وذروة الهرم هي طبقة الخرج وفي كل طبقة يكون عدد العصبونات أقل من الطبقة السابقة، ولهذه التصميم دراسات عديدة ستمر معنا لاحقاً. ✓ وكما تكلمنا سابقاً لتكون الشبكة قادرة على إعطاء نتائج صحيحة قدر الإمكان فإن اختيار سمات جيدة أمر مهم، فلو كانت المعالجة لتحديد صور طائر أو غير طائر فإن خاصية وجود جناح مهمة وخاصية وجود منقار مهمة وخاصية وجود عيون مهمة وإنما غير كافية وبالتالي يجب أن نبحث عن سمات جوهرية تفصل نوع الغرض المراد البحث عنه عن غيره من الأغراض. ✓ ويمكن جعل السمات بشكل ثنائي القيمة، أي تكون لكل قيمة إما القيمة 1 أو 0، مثل وجود العيون، وجود منقار، وجود أجنحة... الخ وبالتالي فالقيمة تعبر عن وجود هذه السمة بالصورة أو لا. مراحل عملية تحديد الصور:



وهي مجموعة مراحل تتم على الصورة للحصول على المعلومات منها. المرحلة الأولى:

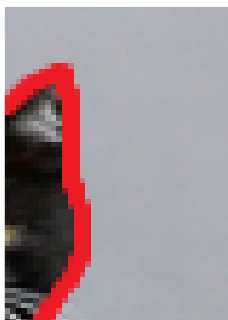
1. عملية الطي Convolution

وهي عملية تنعيم للصورة وتعمل على محاولة إزالة الضجيج من الصورة.

الضجيج هو أي مؤثر غير مرغوب على الصورة يؤدي إلى جعل المعلومات غير مفهومة بالصورة مثل الضجيج الذي يحصل عند أخذ صورة بالظلام، ولكل نوع من الضجيج يوجد طريقة معينة لإزالته.

2. كشف الحواف:

وهي عملية فصل الأغراض بالصورة للحصول على المعلومات عن كل عنصر بالصورة، فلو كان لدينا الصورة التالية:

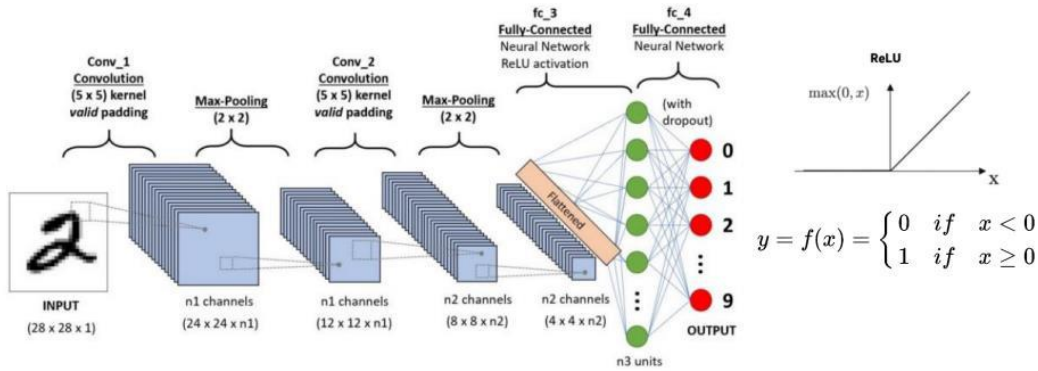


من الضروري تحديد حواف القطعة في الصورة لدراساتها بشكل منفصل ومعرفة فيما إذا كانت مواصفاتها توافق مواصفات القطعة المحددة لدينا، ويمكن التعرف على اللون عن طريق الفروقات اللونية، في الصورة أعلاه

يمكن التمييز بين لون القطعة ولون الشارع وبالتالي القدرة على تحديد الحدود بشكل مبدئي.
وفكرة كشف الحواف هي تفاضل، والحادفة هي حين يكون التفاضل عالي، وبالتالي وظيفتنا إيجاد قناع قادر على إيجاد الحدود للأغراض في مشكلتنا..

The role of ConvNet

وتعني هذه القاعدة أن نجعل الصورة قابلة للمعالجة بشكل أكثر سهولة، ولكن مع الحفاظ على السمات الأساسية للصورة التي سنقوم باستخدامهم للتعرف على الصورة، وهذه الطريقة مفيدة لتصغير حجم المعالجة عند العمل على مجموعة بيانات عملاقة.
مثال شبكة للتعرف على المحارف:



A CNN sequence to classify handwritten digits

شرح بسيط لطريقة من إحدى الطرق للتعرف على الأحرف:

بالبدء سنقوم باستخراج السمات المفيدة من الصورة، ومن ثم سنقوم بمقارنة النتيجة مع أحرف اللغة الأنكليزية وسينتج لدينا عدة توقعات، مثل:

أحتمال أن يكون a هي 80 %

أحتمال أن يكون d هي 10 %

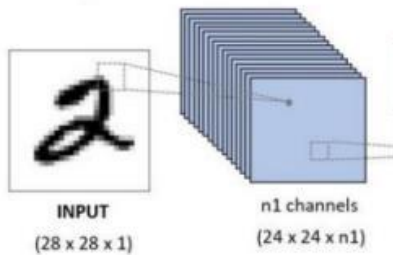
وهكذا...

وبالتالي أعلى احتمال سيكون هو النتيجة الأقرب للصحة.

شرح الشبكة في الصورة أعلاه:

الصورة بحجم 28×28

✓ في البداية سنقوم بعمل Padding للصورة وبالتالي ستصبح بحجم 24×24 وفائدة هذه العملية هي الفرض أن العنصر المراد البحث عنه موجود بوسط الصورة وبالتالي، الأطراف لن تفقدنا.

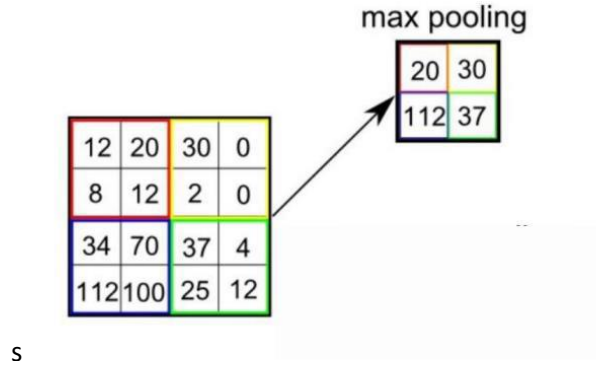


n_1 هي عدد عمليات المعالجة ويمكن القول بأنها مرتبطة بعدد المداخل.

✓ ثم سنقوم بتخفيض أبعاد الصورة وذلك بما يسمى عملية pooling وهي عملية لتخفيض أبعاد الصورة وبالتالي تقليل المعالجة المحتاجة، و يوجد نوعان من ال pooling وهما:

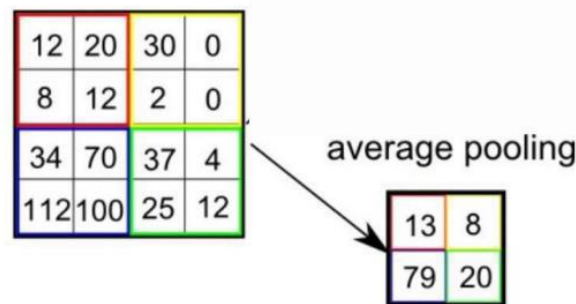
1. Max pooling

وتعني بأن نأخذ إطار بحجم معين وأخذ القيمة العليا في هذا الإطار، مثال:

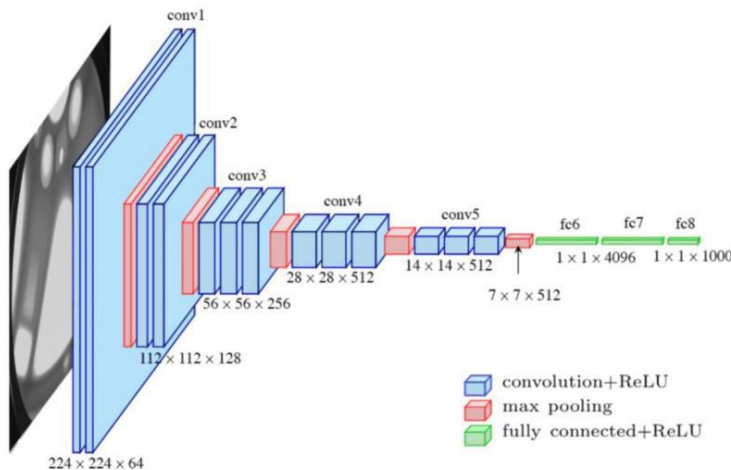


2. Average Pooling:

وتعني بأن نأخذ إطار بحجم معين وأخذ القيمة المتوسط في هذا الإطار، مثال:



✓ وبالتالي فإننا نقوم بعملية Convolution و pooling حتى ينتج لدينا كامل المعلومات من الصورة، مثال توضيحي للعملية:

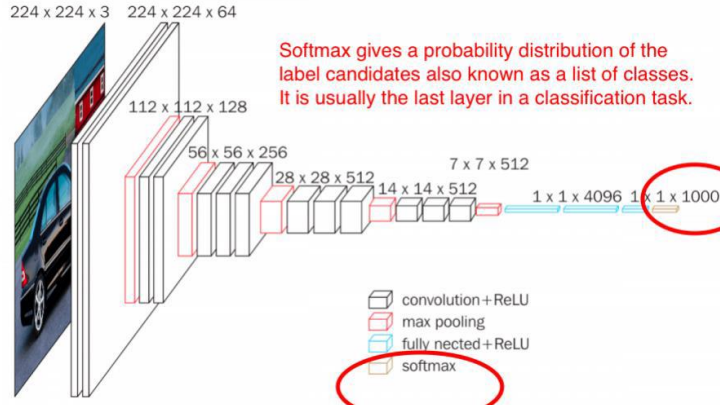


وتعني المرحلة الأولى بأننا قمنا بالحصول على 64 صورة من الصورة الأصلية، كل صورة بأبعاد 224×224

و من ثم عمل pooling وهكذا حتى الحصول على شعاع من السمات يدخل لشبكة العصبونية $(1 \times 1 \times 1000)$.

- تذكرة: fully connected تعني بأن جميع العصبونات متصلة بكامل العصبونات في الطبقة التي تليها.
- ملاحظة: تابع Relu هو تابع يعطي القيم الموجبة بقيمتها الحقيقية و باقي القيم (السالبة و 0) تعطى كقيمة 0 فالقيم غير السالبة غير معرفة.

مثال على صورة ملونة:



والفرق بأننا سنحتاج تابع softmax وغالبا ما يضاف في النهاية.

معلومة:

لتحويل الصورة الملونة لصورة رمادية فإننا نضرب المركبات الثلاث في البكسل الواحد بنسب معينة

أخضر: 0.587

أحمر: 0.299

أزرق: 0.114

مثال بكسل يحوي المركبات التالية:

$$R: 244, G: 100, B: 180$$

نضرب كل مركبة بالنسبة ونجمعها:

$$GrayColor = 244 \times 0.299 + 100 \times 0.587 + 180 \times 0.114 \sim = 152$$

و هو ما يسمى قانون Y

Performance Matrices of ANN

مقدمة: عند انتهائنا من تصميم الشبكة العصبونية والبنية المطلوبة التي ستقوم بحل المشكلة المعروضة، لابد من اختبار هذه الشبكة ورؤية النتائج وتسجيلها لمعرفة سلوك الشبكة وحالات الخطأ ونسبته ومتى حدوثه.

ما أهمية هذه العملية؟

- هذه العملية لمحاولة فهم آلية الخطأ ونسبته و محاولة تجاوزه أو إيجاد حل قادر على تجنب الوقوع بهذا الخطأ مثال: على فرض قمنا بتطوير نظام يكشف العملة المزورة وقام هذا النظام بعمله ونسبة كشف صحيحة جيدة على فرض 95%، ولكن بنسبة الخطأ الـ 5% كان هناك ثغرة يمكن استغلالها وهذا ما يؤدي إلى خسائر كبيرة جداً ومكلفة وبالتالي عملية فحص البنية والنظام وتقييمه عملية مهمة ومفيدة في اكتشاف الثغرات.
- وهو علم بحد ذاته وتقوم به شركات مع مختصين بهذا المجال بواسطة طرق وخوارزميات تتبعها وفي حال كان النظام قادر على تجاوز هذه الاختبارات فإن يتم منح شهادة وتأكيد على فعالية النظام بعمله.

Confusion Matrix

وهي عبارة عن مصفوفة ثنائية بأبعاد $N \times N$ حيث يعبر الصف عن التصنيف الصحيح للحالة والأعمدة هي التصنيف الذي توقعه النظام الذي يقوم بعملية تصنيف.

<i>predicted</i> → <i>real</i> ↓	<i>Class</i> ₁	<i>Class</i> ₂
<i>Class</i> ₁	85	5
<i>Class</i> ₂	9	1

المثال أعلاه يعني وجود حالتين للتصنيف وبالتالي سنقوم برسم مصفوفة 2×2 والقيم تعني:

- ✓ 85 من العينات التي تصنيفت على أنها من الصف الأول وهي من الصف الأول (تصنيف صحيح).
- ✓ 5 من العينات تصنيفت على أنها صف ثاني وهي صف أول (تصنيف خاطئ).
- ✓ 9 من العينات تصنيفت على أنها صف أول وهي صف ثاني (تصنيف خاطئ).
- ✓ 1 من العينات تصنيفت على أنها صف ثاني وهي صف ثاني (تصنيف صحيح).

ملاحظة: مجموع العينات (85, 5, 9, 1) هو عدد عينات التدريب الكلي وهو بالمثال السابق 100 عينة.

في حال كان لدينا ثلاث أصناف سنقوم بإنشاء مصفوفة 3×3

<i>predicted</i> → <i>real</i> ↓	<i>Class</i> ₁	<i>Class</i> ₂	<i>Class</i> ₃
<i>Class</i> ₁	80	19	1
<i>Class</i> ₂	23	55	22
<i>Class</i> ₃	13	26	61

وتعني القيم كما في المثال السابق، وبالتالي القاعدة العامة هي:

	$Class_1$	$Class_2$...	$Class_m$
$Class_1$	n_{11}	n_{12}	...	n_{1m}
$Class_2$	n_{21}	n_{22}	...	n_{2m}
...
...
...
$Class_m$	n_{m1}	n_{m2}	...	n_{mm}

استنتاج: في هذه المصفوفة دائما ما يكون القطر الرئيسي لهذه المصفوفات هو التوقع الصحيح.

airplane	923	4	21	8	4	1	5	5	23	6
automobile	5	972	2					1	5	15
bird	26	2	892	30	13	8	17	5	4	3
cat	12	4	32	826	24	48	30	12	5	7
deer	5	1	28	24	898	13	14	14	2	1
dog	7	2	28	111	18	801	13	17		3
frog	5		16	27	3	4	943	1	1	
horse	9	1	14	13	22	17	3	915	2	4
ship	37	10	4	4		1	2	1	931	10
truck	20	39	3	3			2	1	9	923
	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck

Predicted class

لتسهيل الحسابات تم تعريف أربع مصطلحات تدل على حالة العينات في كل تصنيف ومعرفة الفرق بين المصدر (حالة العينة) والتوقع (الحالة التي توقعها النظام للعينة) وهي:

		Predicted	
		Positives(1)	Negatives(0)
Actual	Positives(1)	TP	FN
	Negatives(0)	FP	TN

:True Positives (TP)

وتعني بأن التصنيف Positive وتصنف على أنه Positive وبالتالي فهو صحيح (True)

:True Negative (TN)

وتعني بأن التصنيف Negative وتصنف على أنه Negative وبالتالي فهو صحيح (True).

:False Positives (FP)

وتعني بأن التصنيف Negative وتصنف على أنه Positives وبالتالي فهو خاطئ (False).

False Negative (FN)

وتعني بأن التصنيف Positives وتصنف على أنه Negative وبالتالي فهو خاطئ (False).

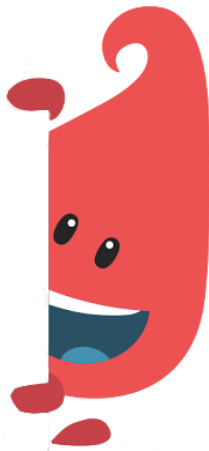
مثال: تطبيق لتصنيف الإيميلات على أنها احتيالية (Spam) أو لا (Normal):
معلومة: يتم اعتبار الإيميل على أنه احتيالي حسب ورود كلمات فيه أو روابط، وعدة عوامل تحدد فيما إذا كان هذا الإيميل احتيالي أو لا ولذلك قد نجد إيميلات مهمة في لائحة الإيميلات الاحتيالية وإيميلات احتيالية في قائمة الرسائل العادية وذلك بأن النظام ليس دقيق بشكل كامل.

سنعتبر بأن حالة ال Positive هي حالة إيميل احتيالي و Negative هي حالة إيميل عادي وبالتالي سينتج لدينا الجدول:

	Predicted class POSITIVE (spam 📧)	Predicted class NEGATIVE (normal 📧)
Actual class POSITIVE (spam 📧)	TRUE POSITIVE (TP) 📧 📧 320	FALSE NEGATIVE (FN) 📧 📧 43
Actual class NEGATIVE (normal 📧)	FALSE POSITIVE (FP) 📧 📧 20	TRUE NEGATIVE (TN) 📧 📧 538

في المثال السابق قام التطبيق بتصنيف 320 إيميل على أنه احتيالي وهو احتيالي وهذا يعني تطابق المصدر مع التوقع وبالتالي الحالة TP، وباقي الحالات تتبع نفس القاعدة.

❖ في حال كان النظام يقوم بتصنيف لعدة حالات (احتياالي - إعلان - طبيعي) سيتم تطبيق نفس القاعدة مع تغيير بالشكل، حيث ستكون مصفوفة ثنائية بأبعاد 3×3 وسنصطلح بأن حالة ال Positive هي حالة إيميل احتيالي (Spam) وحالة ال Negative هي حالة إيميل عادي أو إعلان (not-spam)، وسيكون الجدول بالشكل التالي:



	Predicted class POSITIVE (spam 📧)	Predicted class NEGATIVE (ad 📧)	Predicted class NEGATIVE (normal 📧)
Actual class POSITIVE (spam 📧)	TRUE POSITIVES 27	FALSE NEGATIVES 286	40
Actual class NEGATIVE (ad 📧)	1	37	9
Actual class NEGATIVE (normal 📧)	FALSE POSITIVES 5	16	TRUE NEGATIVES 500

- ❖ حيث أن النظام يشير بأن 27 إيميل احتيالي تم تصنيفهم على أنهم احتيالي (حالة نجاح تصنيف TP).
- ❖ و $562 = 37 + 9 + 16 + 500$ إيميل هي إيميل عادي أو إعلان وتصنف على أنه إيميل عادي أو إعلان وبشكل أبسط فإن النظام قام بالتصنيف على أنه spam أو not-spam وبالتالي أصبحت حالة إيميل عادي وإيميل إعلان هي حالة not-spam.
- ❖ المربعات باللون الأحمر خالفت القاعدة وبالتالي هي تصنيف خاطئ.

مثال على تصنيف ثنائي:

تطبيق يقوم بتصنيف صور على أنها صورة قطة أو ليست صور قطة

- ✓ Cat
- ✓ Not-Cat

على فرض لدينا 1100 صورة ونريد تصنيفها بهذا التطبيق وبعد انتهاء التصنيف نتج التالي:

		Predicated Class	
		Cat	Non-Cat
Actual Class	Cat	90	10
	Non-Cat	60	940

إذا كان اهتمامنا بأن يكون تصنيف النظام للصور على أن حالة Cat هي positive فيمكن كتابة النتائج بالشكل:

$$Cat: \begin{bmatrix} TP = 90 & FN = 10 \\ FP = 60 & TN = 940 \end{bmatrix}$$

و في حال اهتمامنا بالحالة الثانية Not-cat وسنعتبرها هي حالة Positive فستكون النتائج بالشكل:

$$Not_Cat: \begin{bmatrix} TP = 940 & FN = 60 \\ FP = 10 & TN = 90 \end{bmatrix}$$

لاحظ تبديل كل عنصر بالمصفوفة مع مقابله بالقطر

$$Cat: \begin{bmatrix} TP = 90 & FN = 10 \\ FP = 60 & TN = 940 \end{bmatrix}$$

ما أهمية هذه القاعدة في تقييم الشبكة العصبونية؟

هذه القيم هي المفتاح الرئيسي لتقييم الشبكة من عدة نواحي حسب قوانين معينة، وتبين قوة الشبكة وضعفها أهم المعاملات التي ستساعدنا في تقييمها، على سبيل المثال فإن الشبكة من الممكن أن تكون جيدة في تصنيف معين وسيئة في آخر، كأن تكون الشبكة جيدة جداً في تصنيف صور الطيور ولكن سيئة في باقي الأصناف.

أهم المعايير في تقييم شبكتنا:

1. الدقة Accuracy:

وفي مسائل التصنيف تكون عدد التوقعات الصحيحة TP, TN على مجموع كافة التوقعات (عدد العينات).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (ACC)$$

وفي حال كنا نريد معرفة دقة النظام في تصنيف صور القطط بشكل صحيح فحسب القانون:

$$Accuracy_{cat} = \frac{90 + 940}{90 + 60 + 10 + 940} = 93.6\%$$

و في حال أردنا معرفة دقة النظام في باقي التصنيفات فحسب القانون:

$$Accuracy_{notcat} = \frac{90 + 940}{90 + 60 + 10 + 940} = 93.6\%$$

نلاحظ بأن النتيجة لم تتغير وذلك لأنه تصنيف ثنائي وعملية الجمع تبديلية.

متى نستخدم عامل الدقة؟

يعد عامل جيد في حال كانت التصنيفات المستهدفة متقاربة من بعضها نوعاً ما.

2. التفصيل Precision:

وتستخدم عند الاهتمام بعدد تصنيفات ال Positive لحالة معينة ويسمى أيضاً PPV (positive predictive value)، وهي تعني كل العينات التي صنف حسب الصنف الذي نهتم به، وبالجدول فإنه يتمثل بالعמוד الأول:

	Predicted	
	Positives(1)	Negatives(0)
Actual Positives(1)	TP	FN
Negatives(0)	FP	TN



والقانون من الشكل:

$$Precision = \frac{TP}{TP + FP}$$

في حال الاهتمام ب cat:

$$Precision_{cat} = \frac{90}{90 + 60} = 60\%$$

في حال Not-cat

$$Precision_{notcat} = \frac{940}{940 + 10} = 98.9\%$$

من النتائج نستنتج بأن المصنف أفضل في تصنيف الصور الـ Not-cat من تصنيف الصور الـ Cat.

3. Recall or Sensitivity

تهتم بالتوقعات الـ Positive للصنف الذي نهتم به أي ما هو Positive وتصنف Positive وما هو Negative وتصنف Positive ويتمثل بالجدول بالسطر الأول.

		Predicted	
		Positives(1)	Negatives(0)
Actual	Positives(1)	TP	FN
	Negatives(0)	FP	TN

والقانون من الشكل:

$$Recall = \frac{TP}{TP + FP} \Rightarrow \begin{cases} Recall_{cat} = \frac{90}{90 + 10} = 90\% \\ Recall_{notcat} = \frac{940}{940 + 60} = 94\% \end{cases}$$

4. Specificity الخصوبة

يكون اهتمامي بالمتمم لتوقع الصنف المهتمين به ويتمثل بالسطر الثاني من الجدول:

		Predicted	
		Positives(1)	Negatives(0)
Actual	Positives(1)	TP	FN
	Negatives(0)	FP	TN

والقانون:

$$Specificity = \frac{TN}{TN + FP} \Rightarrow \begin{cases} Specificity_{cat} = \frac{940}{940 + 60} = 94\% \\ Specificity_{notcat} = \frac{90}{90 + 10} = 90\% \end{cases}$$

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	70	20	Positive Predictive Value	0.70
	Negative	30	80	Negative Predictive Value	0.80
		Sensitivity	Specificity	Accuracy = 0.75	
		0.78	0.73		

بالبداية نرسم شكل الحالات الأربعة لتساعدنا على تطبيق القوانين بشكل صحيح:

		Predicted	
		Positives(1)	Negatives(0)
Actual	Positives(1)	TP	FN
	Negatives(0)	FP	TN



بتطبيق القوانين:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Precision(Positive Predictive Value) = \frac{TP}{TP + FP}$$

$$Recall(Specificity) = \frac{TP}{TP + FP}$$

$$Specificity = \frac{TN}{TN + FP}$$

ويمكن الاهتمام بتصنيف معين وحساب المعايير عنده، مثال:

		Predicted		
		A	B	C
Actual	A	30	50	20
	B	20	60	20
	C	10	10	8

For Label A

نلاحظ وجود ثلاث تصنيفات ولحساب المعايير عند أحد التصنيفات نكتب:

$$Recall = \frac{TP_A}{TP_A + FN_A} = \frac{TP_A}{Total\ Gold\ Label\ A} = \frac{30}{30 + (50 + 20)} = 0.3$$

$$Precision = \frac{TP_A}{TP_A + FP_A} = \frac{TP_A}{Total\ Predicted\ as\ A} = \frac{30}{30 + (20 + 10)} = 0.5$$

وفي حال الثلاث حالات وأكثر فإن التصنيف سيصبح A or $Not - A$ وبالتالي تم جمع الخانات كما شرحنا سابقاً في مثال الإيميل الاحتمالي والإعلان والعادي.

F-Score

وهو متغير ثابت يجمع ال Precision و Recall ويعطى بالقانون التالي:

$$F_{\beta} = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

وعندما تكون بيتا تساوي ال 1 يسمى F1-Score وتصبح العلاقة من الشكل:

$$F1 - score = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

نطبق على مثال تصنيف صور القطط السابق:

$$1 - score_{cat} = 2 \cdot \frac{\text{Precision}_{cat} \times \text{Recall}_{cat}}{\text{Precision}_{cat} + \text{Recall}_{cat}} = 2 \cdot \frac{0.6 \times 0.9}{0.6 + 0.9} = 72\%$$



The end