



## الشبكات العصبية

وهو نوع من الشبكات العصبية 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 ويعامل معها على أنها مصفوفة، كل بكسل فيها يحمل درجة لونية معينة حسب نوع الصورة (ملونة - أبيض وأسود - رمادية ... الخ).



What we see

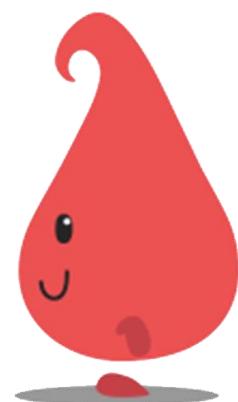
0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

### أنواع الصور:

#### 1. الصورة الرمادية:

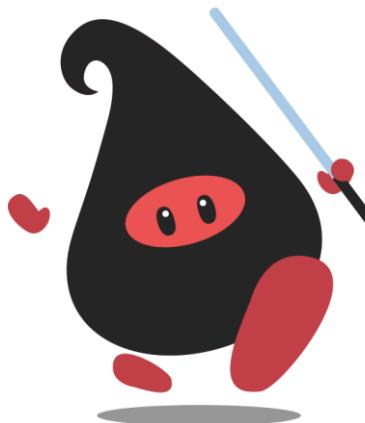
وهي عبارة عن صورة وجميع ألوانها هي تدرجات الرمادي، وتتمثل بمصفوفة قيمة كل عنصر فيها بين ال 0 وال 255 ويختزن ب 8 بت، وبالتالي كل لون يتمثل ب بت واحد، ويستخدم هذا النوع من الصور لمشاهدة برق الصورة.





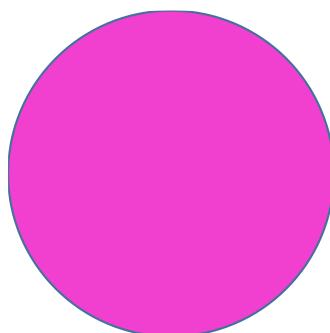
## 2. الصورة الثنائية: Binary image

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

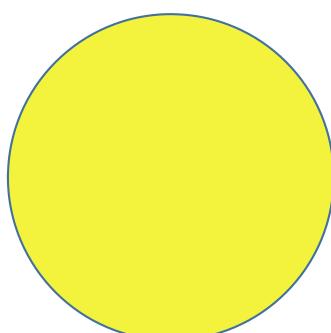


## 3. الصور الملونة: Colour image

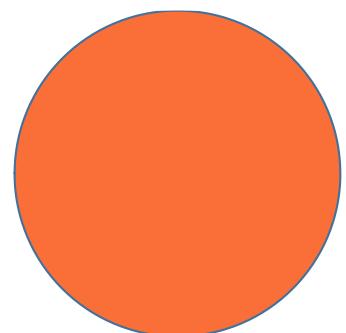
وتتألف من مصفوفة كل عنصر فيها يتكون من ثلاثة قيم تمثل كل قيمة درج اللون من الألوان الأساسية الثلاث (أحمر - أزرق - أخضر RGB)، وكل قيمة تكون محصورة بين الـ 0 و 255 وبالتالي سيتم تخزين كل عنصر من المصفوفة بـ  $32 \times 3 = 32$  بت (8) وتغييرات القيم بين هذه القيم الثلاث يولّد لدينا الألوان الأخرى، حيث أن لكل لون درجة معينة من كل لون أساسى.



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

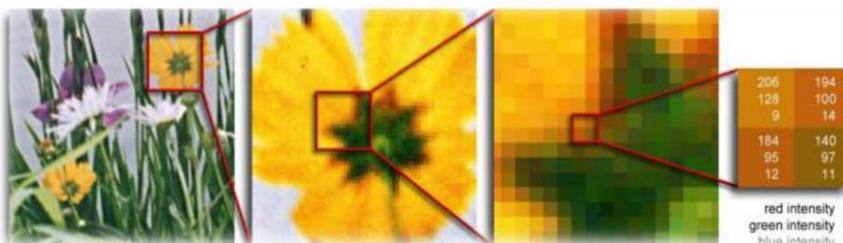
Binary image  
1 bit

محتوى مجاني غير مخصص للبيع التجاري

والصورة الملونة عبارة عن ثلاثة صور رمادية، حيث كل مركبة من المركبات الثلاث لعناصر المصفوفة الممثلة للصورة الملونة يمكن التعبير عنها بصورة رمادية وكل 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، ودرجات 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:

كل بكسل في الصورة إحداثيات معينة، ولنفرض تبدأ الإحداثيات من الزاوية العلوية اليسرى ويكون البكسل فيها قيمته 0,0، مثلاً: لتكن لدينا صورة أبعادها  $4 \times 4$  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 ثلاثة مصفوفات متراكبة فوق بعضها.

ومن أجل تدريب الشبكات العصبية قد تم إنشاء مكتبة صور تدعى [Image.net](https://www.image.org.net)

وهي مكتبة خدمة من الصور مجاناً وتوجد فيهامجموعات صور مدربة، أي أن الصورة تكون مع الوصف أو التوقع الخاص لها، مثلاً:



وبالتالي القدرة على معرفة ماهية الصورة، وتدريب الشبكة العصبية بعينات تدريب مختبرة وصحيبة.

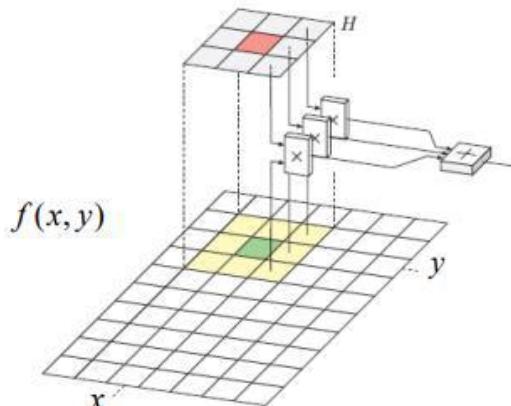
## filters

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

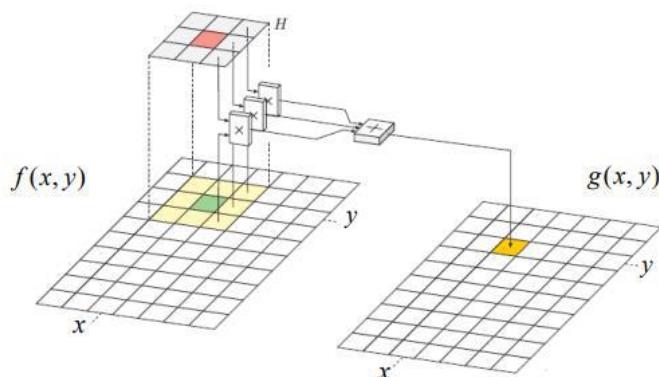
- ✓ نعلم بأن الصورة تكون من بكسلات وكل بكسيل بكسيلات مجاورة وتساعد هذه العملية في تحديد البكسيلات التي ستتأثر بالفلتر.

### ⇨ تطبيق الفلتر:

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



في الصورة أعلاه مصفوفة تعبّر عن الصورة التي نريد تطبيق الفلتر عليها ومصفوفة قناع أبعادها  $3 \times 3$  وتم عملية الفلترة بإنشاء مصفوفة جديدة بنفس المصفوفة المعبرة عن الصورة التي نريد تطبيق الفلتر عليها ونضرب كل قيمة في بكسيل الصورة القديمة بقيمة بكسيل القناع المطبق على هذه البكسيلات (حسب حجم القناع).



في حال كانت الصورة ملونة والقناع يحوي على ثلاثة قنوات (أحمر - أزرق - أخضر) فإننا نضرب كل قيمة بالقناع مع القناة الموافقة له ونجمع القيم الثلاث وينتج لدينا القيمة الجديدة للبكسيل: وبالتالي يمكن التعبير عن الفلتر بالعلاقة التالية:

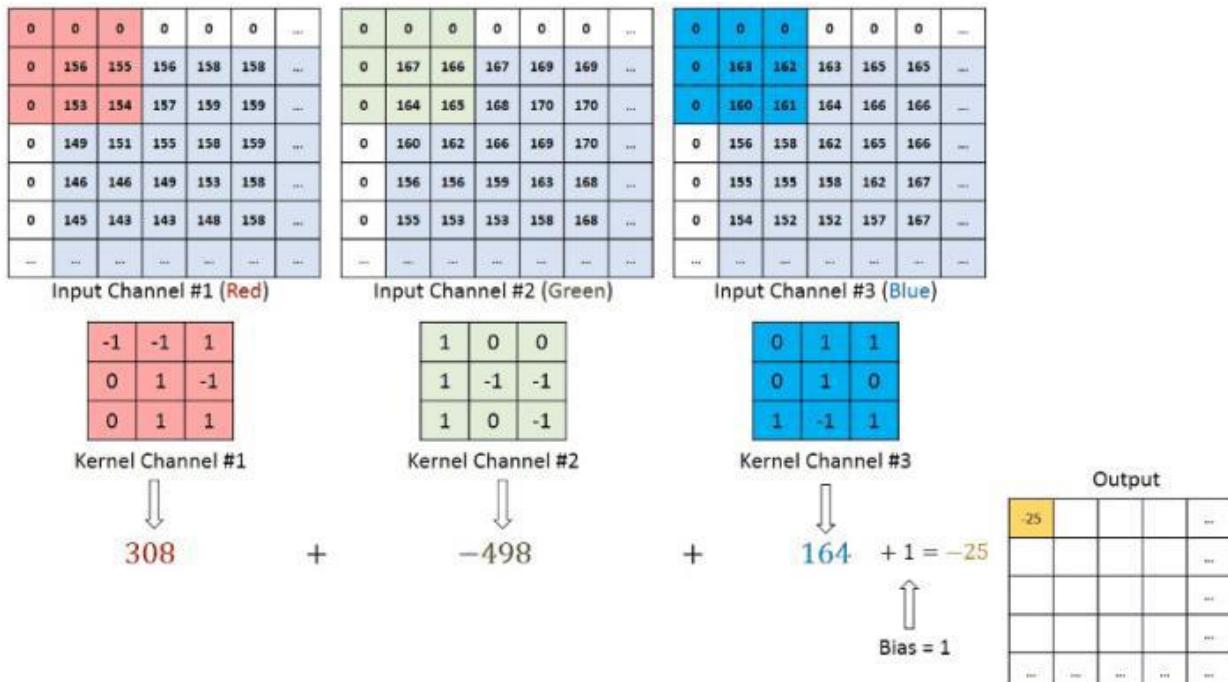
$$\text{Kernel Size} = (2n - 1) \times (2m - 1)$$

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

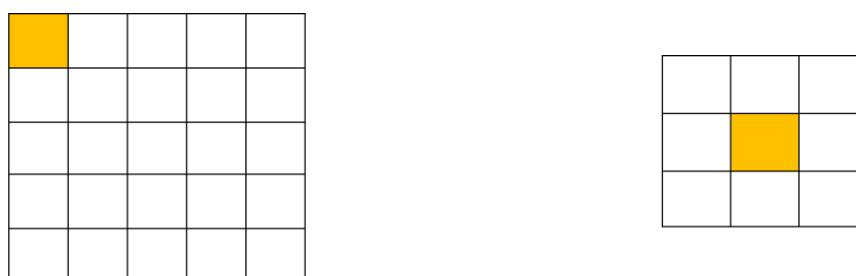


حيث:

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



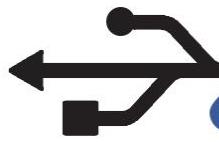
تقوم العملية على وضع مركز القناع على أول بكسل بالصورة على فرض الصورة والقناع من الشكل:



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

0	0	0
0	136	155
0	130	156

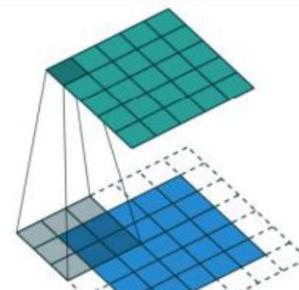
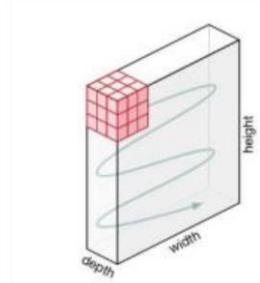
فكم نلاحظ بأن القناع لم يغطي الصورة ولذلك فإننا نضع ما يسمى Padding وهوعبارة عن عناصر فارغة نعبر عنها بالقيمة صفر والغرض منها هي أن تبقى حدود القناع بداخل الصورة.



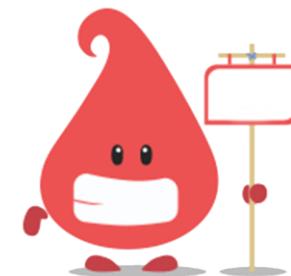
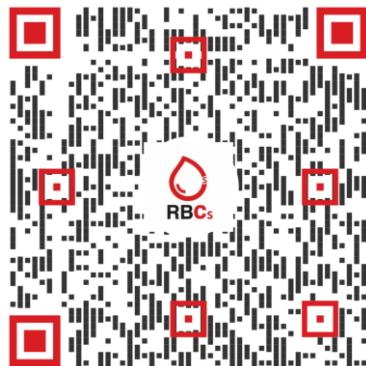
0	0	0	0	0	0	0	...
0	156	155	156	158	158	158	...
0	155	154	157	159	159	159	...
0	149	151	155	158	159	159	...
0	146	146	149	153	158	158	...
0	145	143	143	148	158	158	...
...	...	...	...	...	...	...	...



نقوم بإضافة أطراف للصورة وقيمها 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 توسسيط القيم المطبقة عليها وبالتالي ستتشابه الدرجات اللونية في البكسلات المجاورة وتقل درجة الحدود وهو الفلتر المستخدم في تنعيم الوجه.

## Softmax

وهوتابع يحول مصفوفة من القيم الحقيقة الى مصفوفة من القيمة الحقيقة حيث يكون مجموعها 1 (احتمالات).

مثال:

Output layer	Softmax activation function	Probabilities
$\begin{bmatrix} 1.3 \\ 5.1 \\ 2.2 \\ 0.7 \\ 1.1 \end{bmatrix}$	$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$	$\begin{bmatrix} 0.02 \\ 0.90 \\ 0.05 \\ 0.01 \\ 0.02 \end{bmatrix}$

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

مثال على استخدام التابع:

$$\begin{bmatrix} 8 \\ 5 \\ 0 \end{bmatrix}$$

ليكن لدينا الشعاع التالي:

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:

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

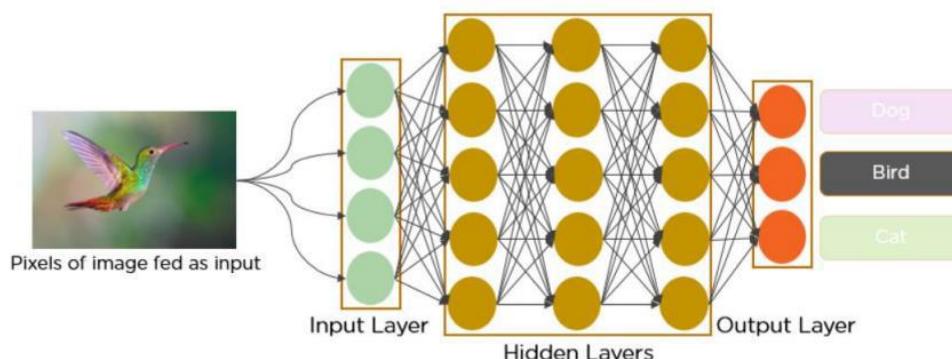
$$\sigma(\vec{z})_2 = \frac{148.4}{3130.4} = 0.0474$$

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

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 العصبية التي تحاكي رؤية الإنسان، ودخلها صورة وخرجها تصنيف على سبيل المثال. مثال لشبكة عصبية تقوم بتحديد نوع الحيوان بالصورة المعطاة لها.



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

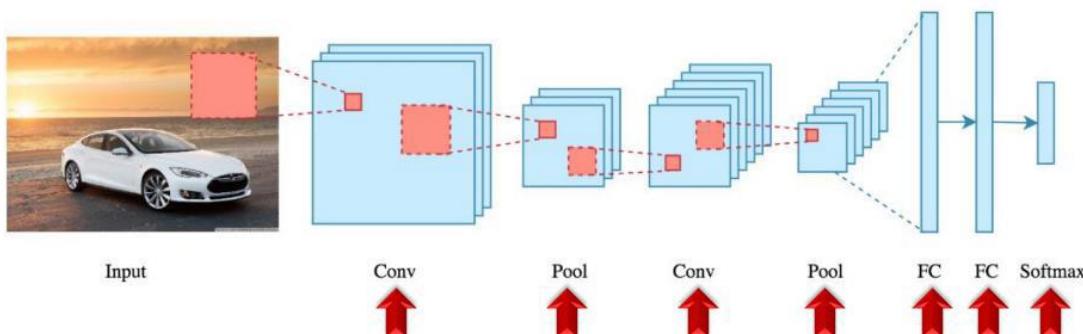
شكل الهرم وقاعدة الهرم هي طبقة الدخول وذروة الهرم هي طبقة الخرج وفي كل طبقة يكون عدد العصبونات أقل من الطبقة السابقة، وهذه التصاميم دراسات عديدة ستمر معنا لاحقاً.

✓ 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 وخاصة وجود منقار مهمة وخاصة وجود عيون مهمة وإنما غير كافية وبالتالي يجب أن نبحث عن سمات جوهرية تفصيل نوع الغرض المراد البحث عنه عن غيره من الأغراض.

✓ ويمكن جعل السمات بشكل ثانوي القيمة، أي تكون لكل قيمة أما 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 تم على الصورة للحصول على المعلومات منها.  
المراحل الأولى:

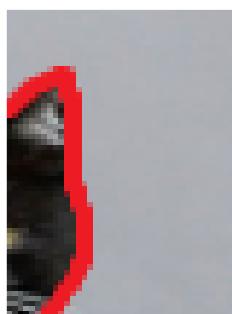
#### 1. عملية الطي Convolution

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، وكل نوع من الضجيج يوجد طريقة معينة لإزالته.

#### 2. كشف الحواف Edge Detection

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

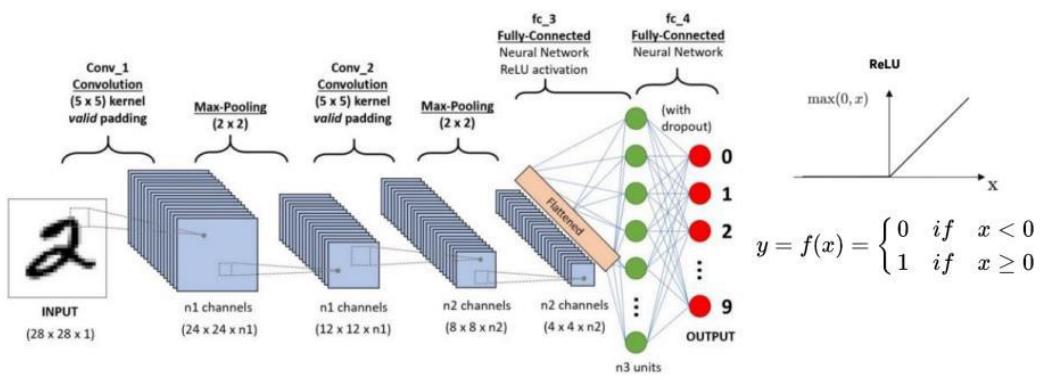


من الضروري تحديد حافة القطعة في الصورة لدراستها بشكل منفصل ومعرفة فيما إذا كانت مواصفاتها تتوافق مواصفات القطعة المحددة لدينا، ويمكن التعرف على اللون عن طريق الفروقات اللونية، في الصورة أعلاه

يمكن التمييز بين لون القطة ولون الشارع وبالتالي القدرة على تحديد الحدود بشكل مبدئي.  
وفكرة كشف الحواف هي تفاضل، والحافة 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:



### شرح سطير طريقة من إحدى الطرق للتعرف على الأحرف:

بالبداية سنقوم باستخراج السمات المفيدة من الصورة، ومن ثم سنقوم بمقارنة النتيجة مع أحرف اللغة الانكليزية وسيتوجب لدينا عدة توقعات، مثل:

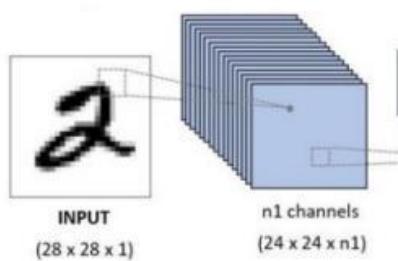
احتمال أن يكون  $a$  هي 80 %  
احتمال أن يكون  $d$  هي 10 %  
وهكذا...

وبالتالي أعلى احتمال سيكون هو النتيجة الأقرب للصحة.

### شرح الشبكة في الصورة أعلاه:

الصورة بحجم  $28 \times 28$

✓ في البداية سنقوم بعمل Padding للصورة وبالتالي ستصبح بحجم  $24 \times 24$  وفائدة هذه العملية هي الفرض أن العنصر المراد البحث عنه موجود بوسط المجموعة وبالتالي، الأمثلاء، لن تفردنا.

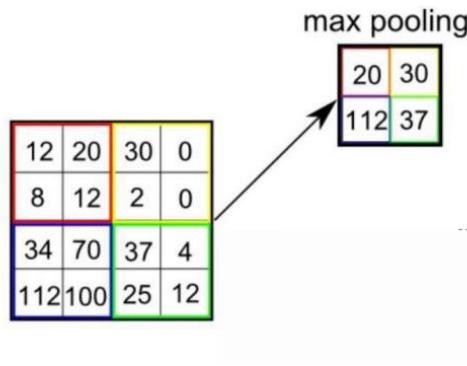


و  $n_1$  هي عدد عمليات المعالجة ويمكن القول بأنها مرتبطة بعدد المدخل.

✓ 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:

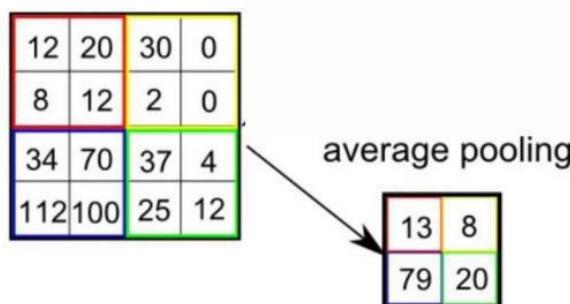
### 1. Max pooling

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

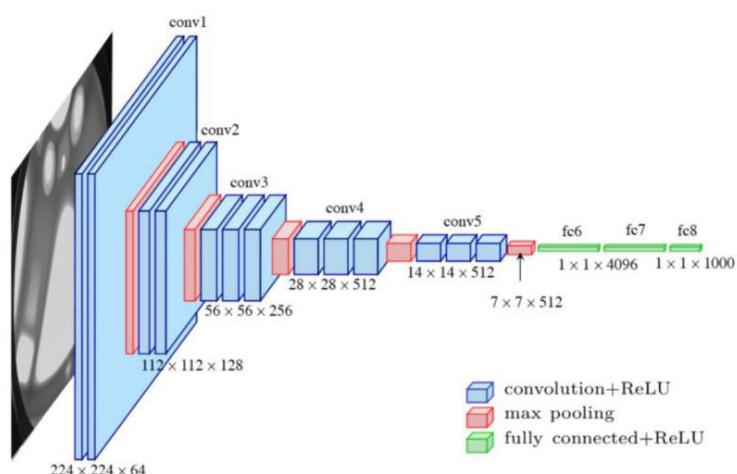


### 2. Average Pooling:

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



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



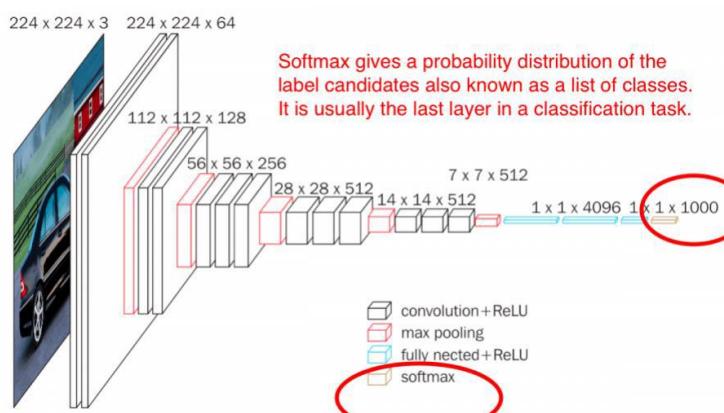
وتعني المرحلة الأولى بأننا قمنا بالحصول على 64 صورة من الصورة الأصلية، كل صورة بأبعاد  $224 \times 224$

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

**تذكرة:** تعني بأن جميع العصبونات متصلة بكامل العصبونات في الطبقة التي تليها.

**ملاحظة:** تابع Relu هوتابع يعطي القيمة الموجبة بقيمها الحقيقة و باقي القيم (السلبية و 0) تعطى قيمة 0 فالقيمة غير السلبية غير معرفة.

مثال على صورة ملونة:



والفرق بأننا سنحتاج تابع softmax و غالباً ما يضاف في النهاية.

معلومات:

لتحويل الصورة الملونة لصورة رمادية فإننا نضرب المركبات الثلاث في البكسل الواحد بنسبة 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 انتهاءنا من تصميم الشبكة العصبية والبنية المطلوبة التي ستقوم بحل المشكلة المعروضة، لابد من اختبار هذه الشبكة ورؤيه النتائج وتسجيلها لمعرفة سلوك الشبكة وحالات الخطأ ونسبة ومتى حدوثه.

### ما أهمية هذه العملية؟

- ◀ هذه العملية لمحاولة فهم آلية الخطأ ونسبة ومحاولة تجاوزه أو إيجاد حل قادر على تجنب الوقوع بهذا الخطأ مثال: على فرض قمنا بتطوير نظام يكشف العملة المزورة وقام هذا النظام بعمله وبنسبة 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 الصفر عن التصنيف الصحيح للحالة والأعمدة هي التصنيف الذي توقعه النظام الذي يقوم بعملية تصنيف.

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

المثال أعلاه يعني وجود حالتين للتصنيف وبالتالي سنقوم برسم مصفوفة  $2 \times 2$  والقيم تعني:

- ✓ 85 من العينات التي صنفت على أنها من الصنف الأول وهي من الصنف الأول (تصنيف صحيح).
- ✓ 5 من العينات صنفت على أنها صنف ثانٍ وهي صنف أول (تصنيف خاطئ).
- ✓ 9 من العينات صنفت على أنها صنف أول وهي صنف ثانٍ (تصنيف خاطئ).
- ✓ 1 من العينات صنفت على أنها صنف ثانٍ وهي صنف ثانٍ (تصنيف صحيح).

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

في حال كان لدينا ثلاثة أصناف سنقوم بإنشاء مصفوفة  $3 \times 3$

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

محتوى مجاني غير مخصص للبيع التجاري

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

	<i>Class<sub>1</sub></i>	<i>Class<sub>2</sub></i>	...	<i>Class<sub>m</sub></i>
<i>Class<sub>1</sub></i>	$n_{11}$	$n_{12}$	...	$n_{1m}$
<i>Class<sub>2</sub></i>	$n_{21}$	$n_{22}$	...	$n_{2m}$
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
<i>Class<sub>m</sub></i>	$n_{m1}$	$n_{m2}$	...	$n_{mm}$

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

	airplane	4	21	8	4	1	5	5	23	6
actual class	airplane	923	4	21	8	4	1	5	23	6
	automobile	5	972	2				1	5	15
	bird	26	2	892	30	13	8	17	5	4
	cat	12	4	32	826	24	48	30	12	5
	deer	5	1	28	24	898	13	14	14	2
	dog	7	2	28	111	18	801	13	17	
	frog	5		16	27	3	4	943	1	1
	horse	9	1	14	13	22	17	3	915	2
	ship	37	10	4	4		1	2	1	931
	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 Positive وتصنف على أنه Negative وبالتالي فهو خاطئ (False).

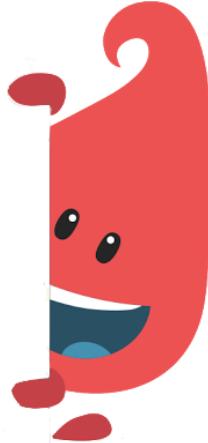
مثال: تطبيق لتصنيف البريد الإلكتروني على أنها احتيالية (Spam) أو لا (Normal):  
 معلومة: يتم اعتبار البريد الإلكتروني على أنه احتيالي حسب ورود كلمات فيه أو روابط، وعدة عوامل تحدد فيما إذا كان هذا البريد الإلكتروني أو لا ولذلك قد نجد إيميلات مهمة في قائمة البريد الإلكتروني  
 الاحتيالية وإيميلات احتيالية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

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

	Predicted class POSITIVE (spam ✉ )	Predicted class NEGATIVE (normal ✉ )
Actual class POSITIVE (spam ✉ )	TRUE POSITIVE (TP)  <b>320</b>	FALSE NEGATIVE (FN)  <b>43</b>
Actual class NEGATIVE (normal ✉ )	FALSE POSITIVE (FP)  <b>20</b>	TRUE NEGATIVE (TN)  <b>538</b>

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

❖ في حال كان النظام يقوم بالتصنيف لعدة حالات (احتياطي - إعلان - طبيعي) سيتم تطبيق نفس القاعدة مع تغيير الشكل، حيث ستكون مصفوفة ثنائية بأبعاد  $3 \times 3$  وسنصلح بأن حالة ال Positive هي حالة إيميل احتيالي (Spam) ونسبة 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	286 FALSE NEGATIVES	40
Actual class NEGATIVE (ad 📢 )	1 FALSE POSITIVES	37 TRUE NEGATIVES	9
Actual class NEGATIVE (normal ✉ )	5	16 TRUE NEGATIVES	500

- ❖ حيث أن النظام يشير بأن 27 إيميل احتيالي تم تصنيفهم على أنهما احتيالي (حالة نجاح تصنيف TP).
- ❖ و  $27 + 37 + 9 + 16 + 500 = 562$  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 خالفة 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 وزرير تصنيفها بهذا التطبيق وبعد انتهاء التصنيف نتج التالي:

		Predicted 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 بالقطدر

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

ما أهمية هذه القاعدة في تقييم الشبكة العصبية؟  
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 تصنیف معین وسیئة في آخر، لأن تكون الشبكة جيدة جداً في تصنیف صور الطيور ولكن سیئة في باقی الأصناف.

### أهم المعايير في تقييم شبكتنا:

#### 1. الدقة :Accuracy

وفي مسائل التصنیف تكون عدد التوقعات الصحیحة TP,TN على مجموع كافة التوقعات (عدد العینات).

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

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

$$\text{Accuracy}_{cat} = \frac{90 + 940}{90 + 60 + 10 + 940} = 93.6\%$$

و في حال أردنا معرفة دقة النظام في باقی التصنیفات فحسب القانون:

$$\text{Accuracy}_{notcat} = \frac{90 + 940}{90 + 60 + 10 + 940} = 93.6\%$$

نلاحظ بأن النتيجة لم تتغير وذلك لأنه تصنیف ثنائی وعملية الجمع تبدیلية.

متى نستخدم عامل الدقة؟

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

#### 2. التفصیل :Precision

وتستخدم عند الاهتمام بعدد تصنیفات ال Positive لحالة معینة ويسمى ايضاً PPV (positive predictive value)، وهي تعنی كل العینات التي صفت حسب الصنف الذي نهتم به، وبالجدول فإنه يتمثل بالعامود الأول:

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



والقانون من الشكل:

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

في حال الاهتمام ب cat

محتوى مجاني غير مخصص للبيع التجاري

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

في حال Not-cat

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

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

### :Recall or Sensitivity .3

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

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

والقانون من الشكل:

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

### Specificity الخصوصية .4

يكون اهتمامي بالمتعمم لتوقع الصنف المهمتين به ويتمثل بالسطر الثاني من الجدول:

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

والقانون:

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

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 الأربع لتساعدننا على تطبيق القوانيين بشكل صحيح:

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



بتطبيق القوانيين:

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

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

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

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

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 ثلاثة 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  $A \text{ or } Not - A$  وبالتالي تم جمع الخانات كما شرحنا سابقاً في مثال الإيميل الاحتيالي والإعلان والعادي.

## F-Score

وهو متغير ثابت يجمع ال Recall و Precision 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:

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



The end