

The top left corner of the slide features several overlapping geometric shapes. These include a large orange-to-red gradient rectangle, a smaller dark blue triangle, and a larger dark blue hexagon. Other fainter shapes are visible in the background.

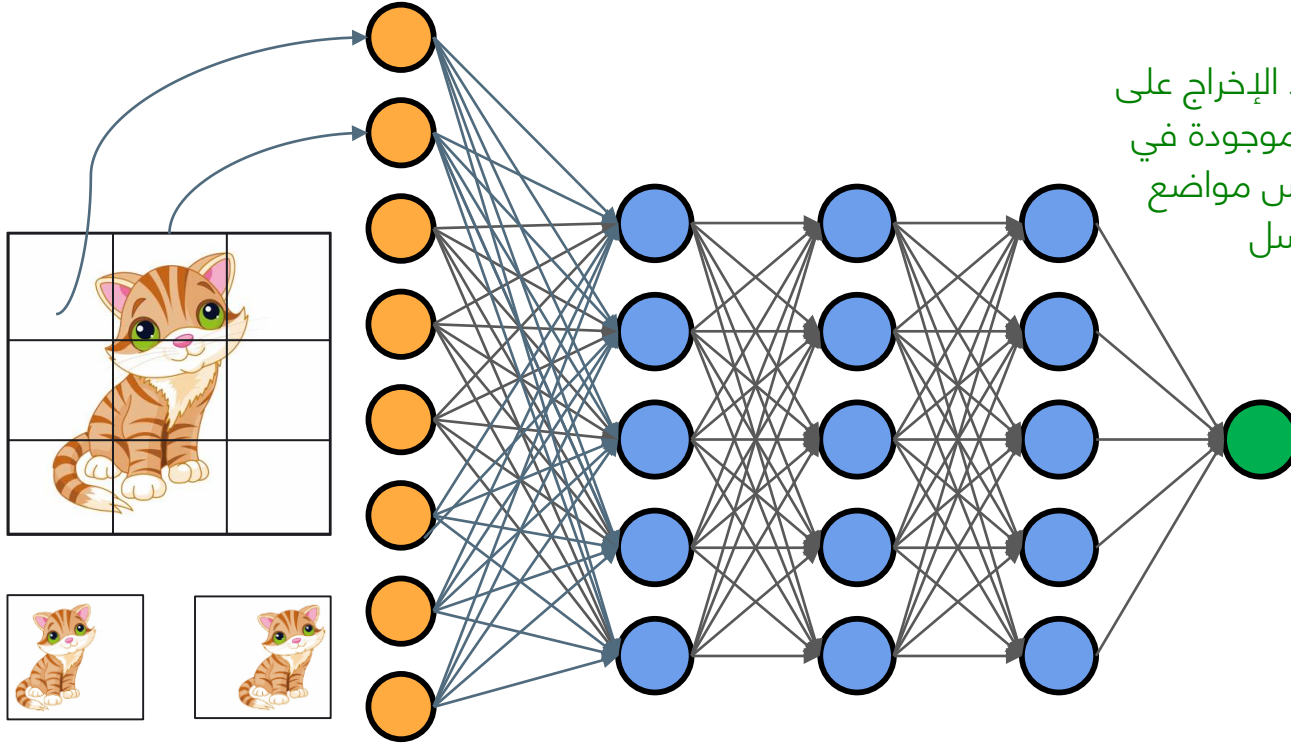
التعلم العميق لبيانات الصور

Deep Learning for Image Data

- ما هي الصورة
- أنواع الصور
- عمليات الصور
- رؤية الحاسب Computer Vision
- التعلم العميق لرؤية الحاسب
- الشبكات العصبية الترشيحية (CNN)
- إعداد البيانات لـ CNN
- التعلم المنقول Transfer Learning

المحتوى

بنية التعلم العميق

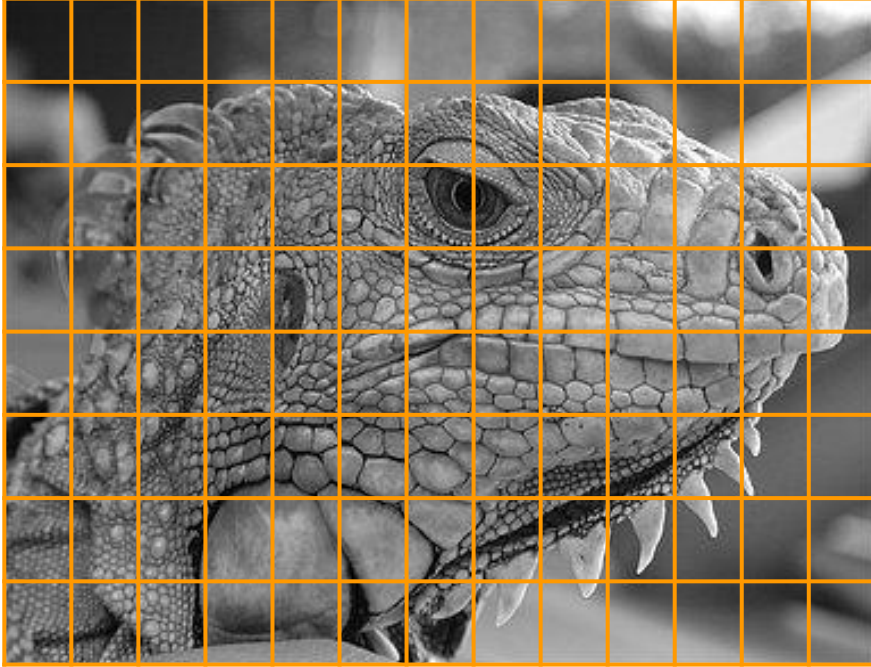


يجب أن يعتمد الإخراج على
الخصائص الموجودة في
الصورة وليس مواضع
البكسل

كلب أم قطة؟

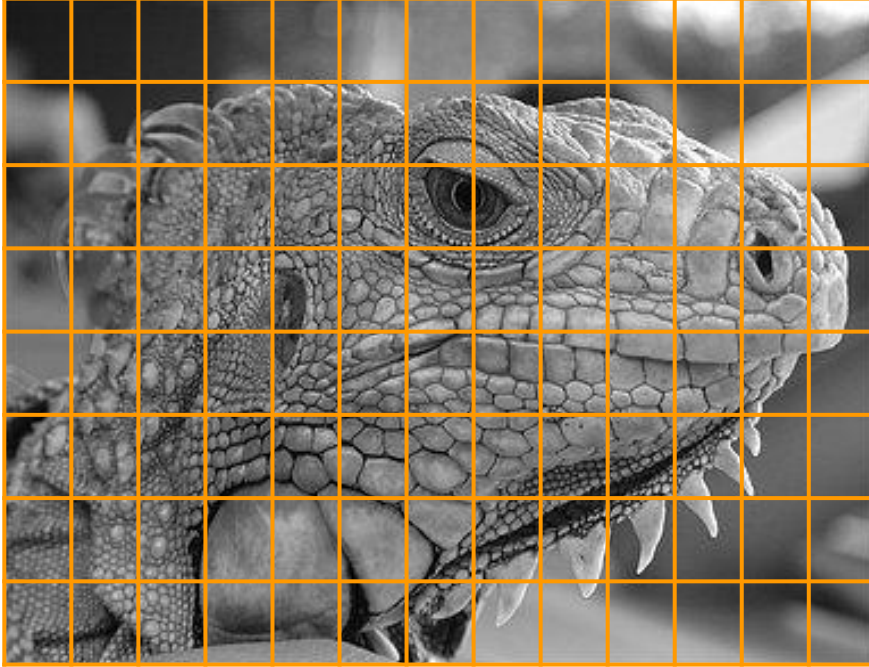
ما هي الصورة؟

ما هي الصورة؟



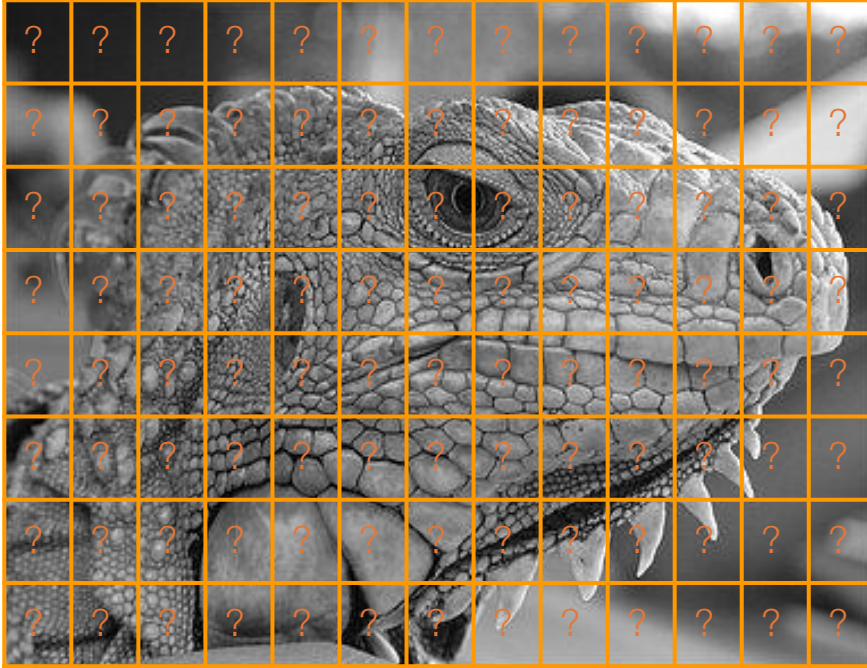
- الصورة تشبه الأحجية (puzzle) مع عدد كبير من القطع المجهزة معًا ، كل قطعة لها نفس الحجم ولكن يمكن أن يكون لها ألوان أو درجات مختلفة.
- كل من هذه القطع الصغيرة هي ما نسميه بالبكسل أو عنصر الصورة.
- إذن الصورة عبارة عن مصفوفة من وحدات البكسل.

ما هي الصورة؟



- إذن الصورة عبارة عن مصفوفة من وحدات البكسل.
- البيكسلات هي ببساطة أرقام تعكس درجة اللون في موقع معين.
- كلما زادت قيمة البكسل ، زادت درجة لون معين يمثله.

ما هي الصورة؟

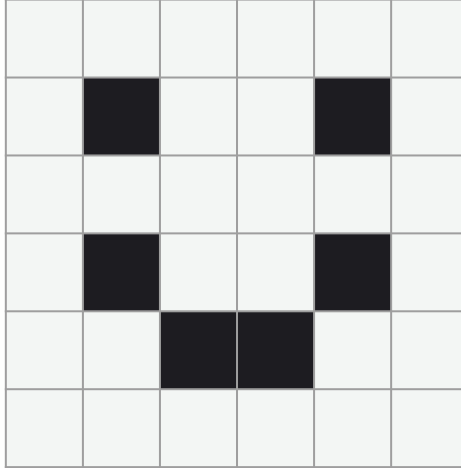


- إذن الصورة عبارة عن مصفوفة من وحدات البكسل.
- البيكسلات هي ببساطة أرقام تعكس درجة اللون في موقع معين.
- كلما زادت قيمة البكسل ، زادت درجة لون معين يمثلها.
- ولكن كيف نخصص قيمة للبكسل؟

أنواع الصور

صور باللون الأبيض والأسود

- نبدأ بالنوع الأساسي للصور وهو أبيض وأسود.
- قيمة البكسل في هذه الصور هي إما (0 أسود) أو (1 أبيض)



1	1	1	1	1	1
1	0	1	1	0	1
1	1	1	1	1	1
1	0	1	1	0	1
1	1	0	0	1	1
0	1	1	1	1	1

1	1	1	1	1	1
1	0	1	1	0	1
1	1	1	1	1	1
1	0	1	1	0	1
1	1	0	0	1	1
0	1	1	1	1	1

صور ذات التدرّج الرمادي

- اللونان المتطرفان هما الأسود (0) والأبيض (255).
- تظهر الأرقام الموجودة بينهما مستويات مختلف من اللون الرمادي.
- قم بتخزين كثافة البكسل في مصفوفة، وها هي لديك!



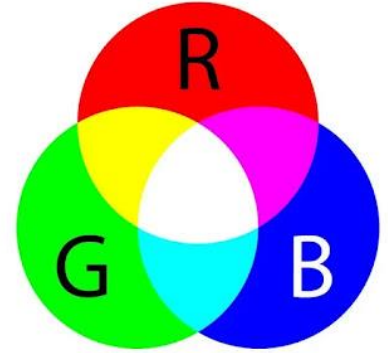
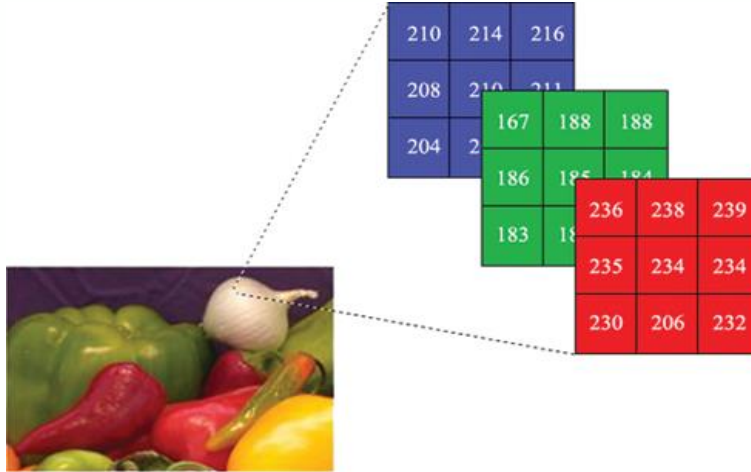
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Source: <https://ai.stanford.edu/~syue/cvweb/Pictures1/imagematrix.png>

صور ملوّنة (RGB-أحمر-أخضر-أزرق)

يتم تحديد لون كل بكسل من خلال دمج درجات الألوان الأحمر والأخضر والأزرق المخزنة داخل كل مستوى لوني في موقع البكسل.



<https://api.intechopen.com/media/chapter/51312/media/fig3.png>

صور ملوّنة (RGB أحمر-أخضر-أزرق)

قيم البكسل

		165	187	209	58	7
	14	125	233	201	98	159
253	144	120	251	41	147	204
67	100	32	241	23	165	30
209	118	124	27	59	201	79
210	236	105	169	19	218	156
35	178	199	197	4	14	218
115	104	34	111	19	196	
32	69	231	203	74		

لون البكسل في موقع: $(0,1)$

ثلاثية ال RGB المخزنة في $(0,1,0:2)$

$(0,1,0) = 144$ •

$(0,1,1) = 125$ •

$(0,1,2) = 187$ •

لون البكسل في موقع: $(0,1)$

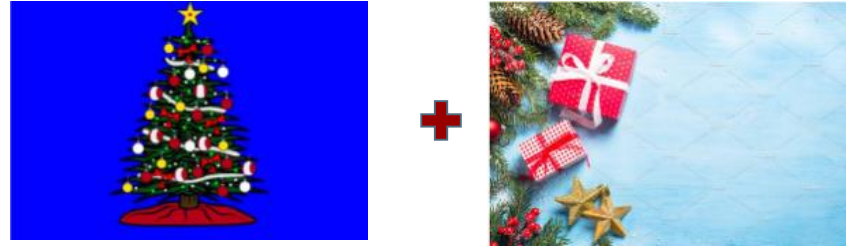
144, 125, 187

عمليات الصور

عمليات الصور

نظرًا لأن الصور هي ببساطة مصفوفات أرقام ، يمكننا إجراء العديد من العمليات عليها ، تمامًا مثل ما نفعله بالمصفوفات العادية.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} + \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix}$$



عمليات الصور

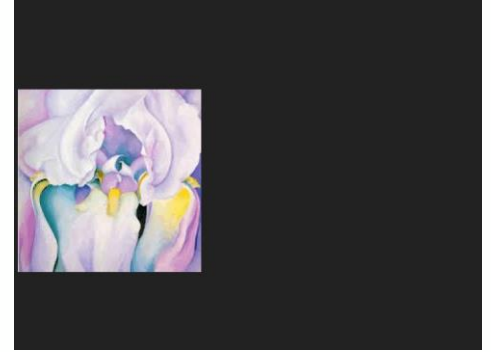
يمكن تطبيق الكثير من العمليات على الصور فقط لأنها عبارة عن مجموعة من الأرقام.



تقسيم ودمج قنوات الصور



مزج الصور



تدوير الصور

عمليات الصور

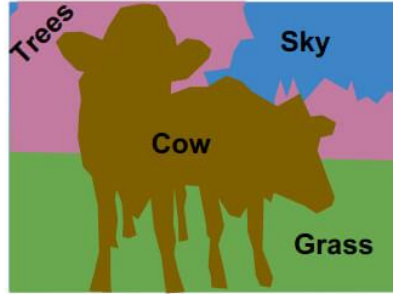
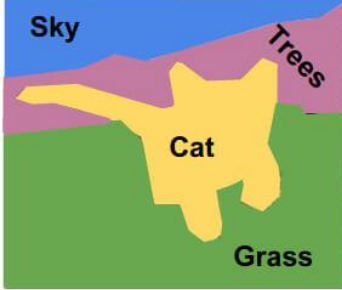
- للقيام بهذه التحولات ، وأكثر من ذلك ، يمكنك استخدام مكتبة OpenCV.
- أصبحت مكتبة OpenCV معيارًا للتعامل مع المهام المتعلقة بالصور.



رؤية الحاسب

رؤية الحاسب

- يعتبر مجال رؤية الحاسب مجالاً يتعامل مع استخراج المعلومات من الصور.
- مهام رؤية الحاسب الشائعة: تصنيف الصور، اكتشاف الأشياء والتجزئة الدلالية.

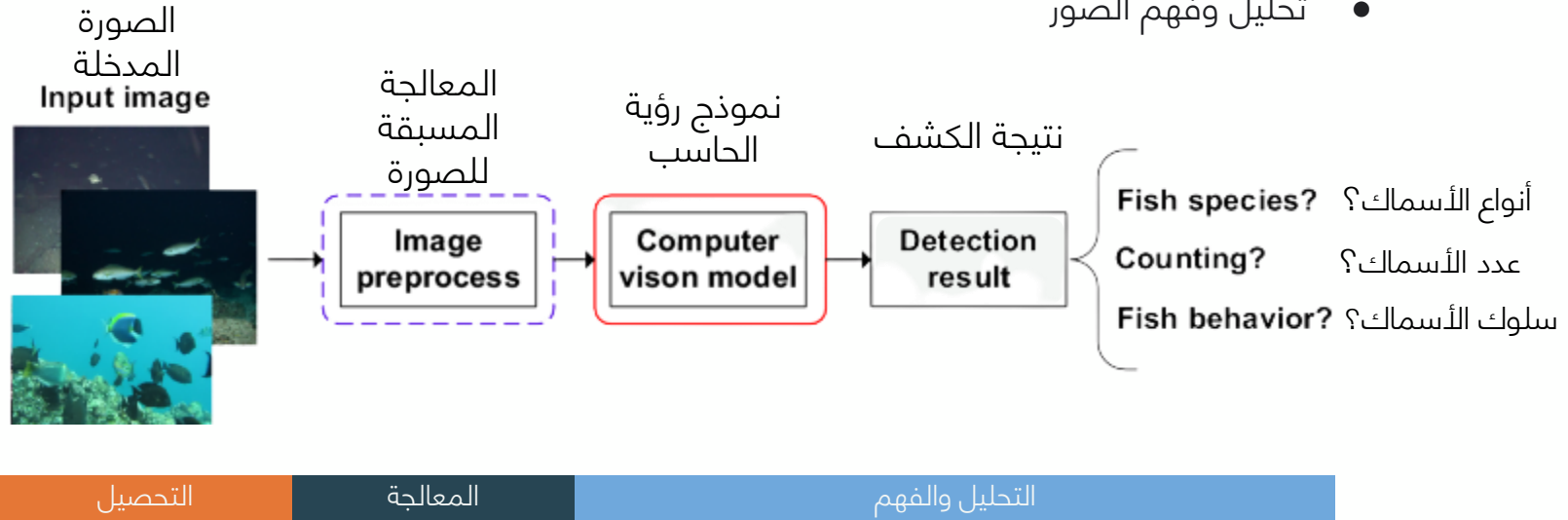


رؤية الحاسب ليست بمعالجة للصور

- معالجة الصور
 - صورة ←صورة(عملية إنشاء صورة جديدة من صورة موجودة)
 - أمثلة :إزالة التشويش واكتشاف الحواف وما إلى ذلك.
- رؤية الحاسب
 - الصورة ←المحتوى (الملاحظات)
 - أمثلة :اكتشاف الوجه وتتبع الأشياء وما إلى ذلك.
- قد يتطلب نظام رؤية الحاسب تطبيق معالجة الصور على المدخلات الأولية ، على سبيل المثال معالجة الصور قبل ادخالها على الخوارزمية.

إطار رؤية الحاسب

- تحصيل الصور
- معالجة الصور
- تحليل وفهم الصور



تحديات رؤية الحاسب

بعد عقود من البحث، تظل رؤية الحاسب دون حل، على الأقل من حيث تلبية قدرات رؤية البشر.

لماذا؟

- ليس لدينا فهم قوي لكيفية رؤية البشر.
- التعقيد المتأصل في العالم المرئي (اتجاه مختلف في العديد من ظروف الإضاءة مع أي نوع من الحجب من أشياء أخرى وما إلى ذلك)



التحديات: الإضاءة

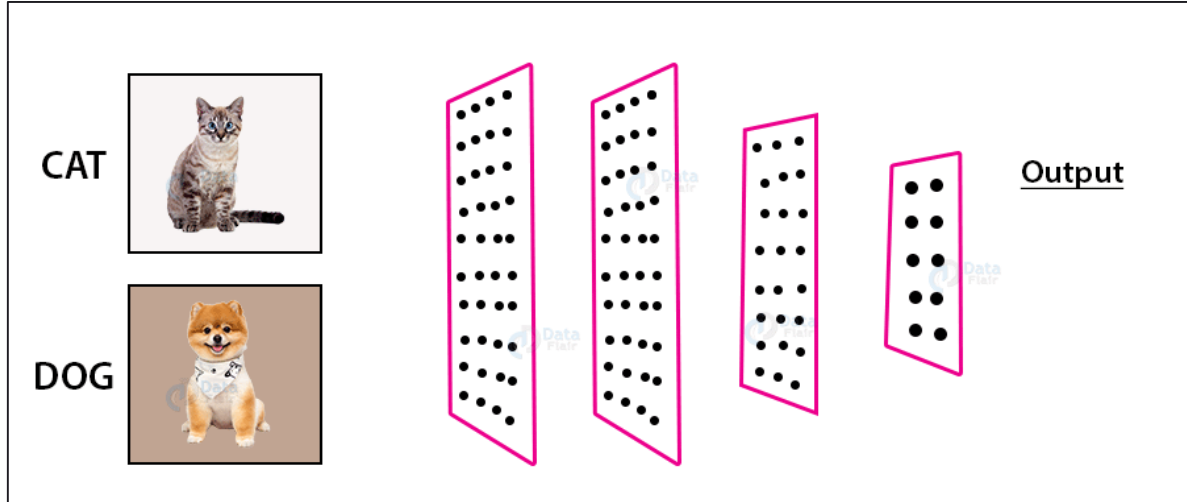


التحديات: الحجب



تطبيقات رؤية الحاسب

تصنيف الصور

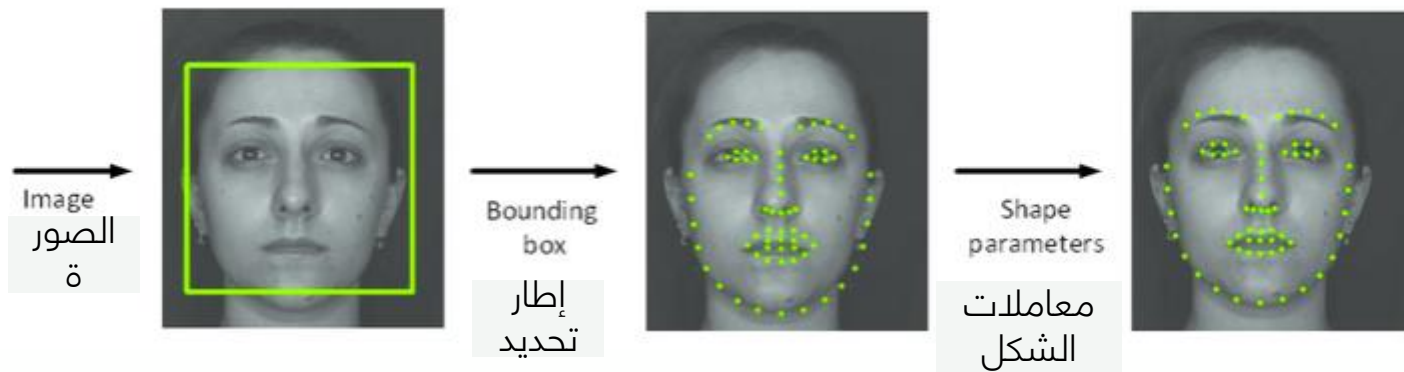


التحقق من الأشياء



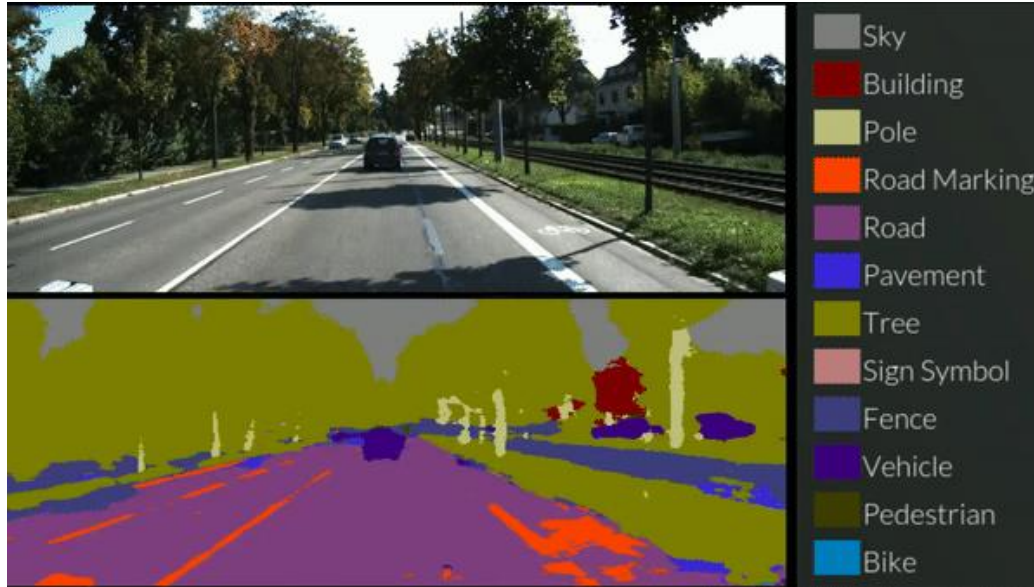
اكتشاف معالم الأشياء

ما هي النقاط الرئيسية للأشياء في الصور؟



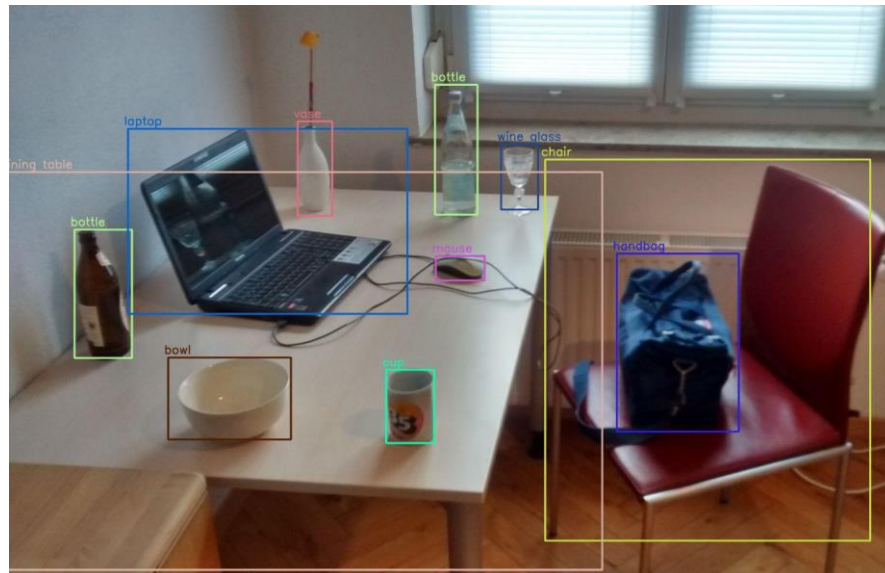
تقسيم الصور

البكسلات المنتمية لأشياء بالصورة.



اكتشاف الأشياء

ما هي الأشياء في الصورة؟

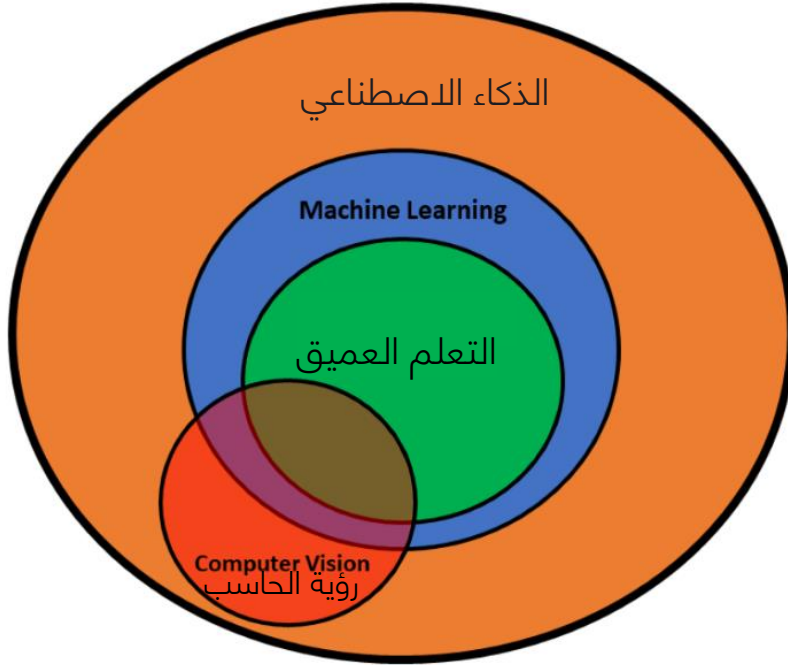


التعلم العميق لرؤية الحاسب

التعلم العميق لرؤية الحاسب

نظرًا لأن رؤية الحاسب تدور حول استخراج المعلومات من الصور ، يمكن أن يساعد التعلم العميق في أداء بعض مهام رؤية الحاسب:

- اكتشاف الأشياء
- تصنيف الصور
- التقسيم الدلالي
- ...



التعلم العميق لرؤية الحاسب

- سيتم صياغة المسألة في هذه الحالة على أنها مسألة تعلم عميق عادية، والفرق الوحيد هو أنه من المفترض أن يكون المدخل صورة بدلاً من بيانات مجدولة.
- مثال لتصنيف الصور.



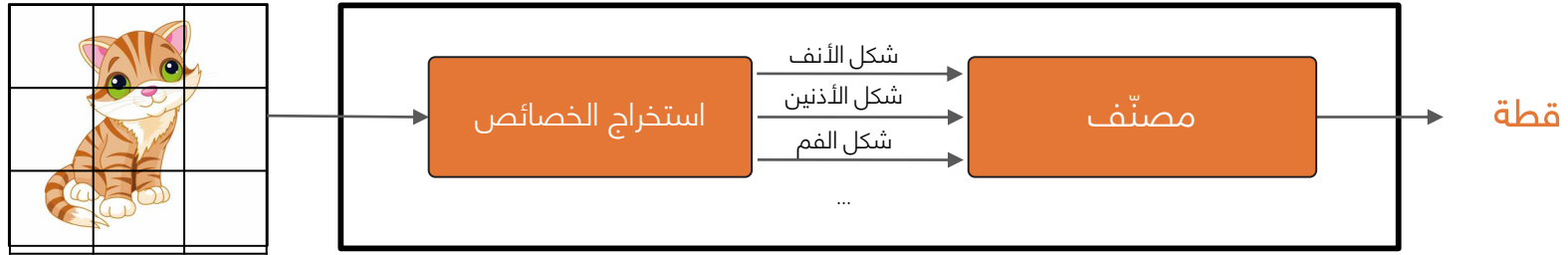
التعلم العميق لرؤية الحاسب

- نريد تشريح نموذج التعلم العميق ومعرفة ما يحتويه وكيف يتعامل مع الصورة.
- في نهاية الأمر، يجب أن يكون لدينا مصنف عادي يتنبأ إذا ماكان بالصورة قطة أو كلب.



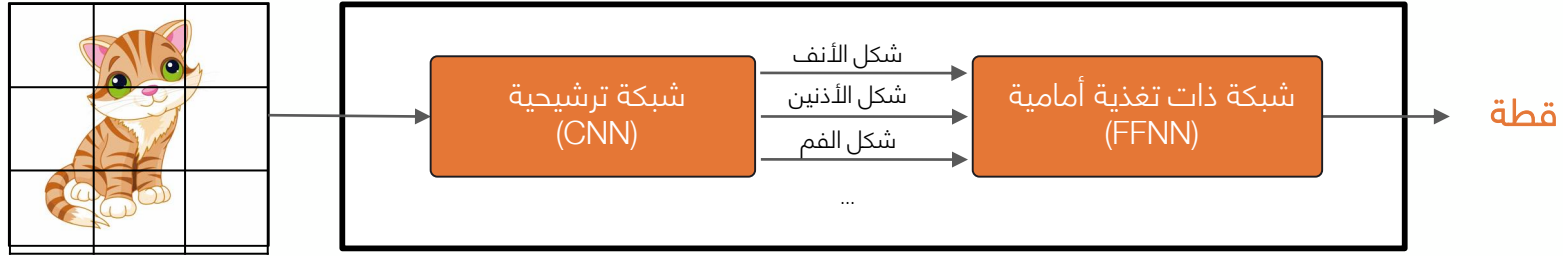
التعلم العميق لرؤية الحاسب

- يتنبأ المصنف بشكل مثالي وفقاً لعدة خصائص بالصورة مثل شكل الأنف وشكل الأذنين وما إلى ذلك.
- يجب استخراج هذه الخصائص من الصورة المدخلة.



التعلم العميق لرؤية الحاسب

- لاستخراج الخصائص بطريقة تلقائية، نستخدم ما نسميه بالشبكات العصبية الترشيحية (CNN)، والمصنف ببساطة عبارة عن شبكة عصبية ذات تغذية أمامية (FFNN).



The background features several abstract geometric shapes, including triangles and hexagons, in shades of blue, teal, and orange, scattered across the dark blue background.

الشبكات العصبية الترشيحية

Convolutional Neural Networks (CNNs)

الشبكات العصبية الترشيحية

- الشبكات العصبية الترشيحية (CNN) هي شبكات عصبية تعمل على الصور وتستخرج الخصائص منها.
- لفهم شبكات CNN، علينا أن نفهم بنيتها الأساسية وهي عملية الترشيح.



عملية الترشيح

الترشيح هو عبارة عن تطبيق مرشح على صورة مدخلة ليستخرج منها معلومات محددة







مرشح



عملية الترشيح

- المرشحات هي ببساطة مصفوفات تساعدنا على تحويل صورة أصلية إلى معلومات محددة منها اعتمادًا على قيم هذه المرشحات.

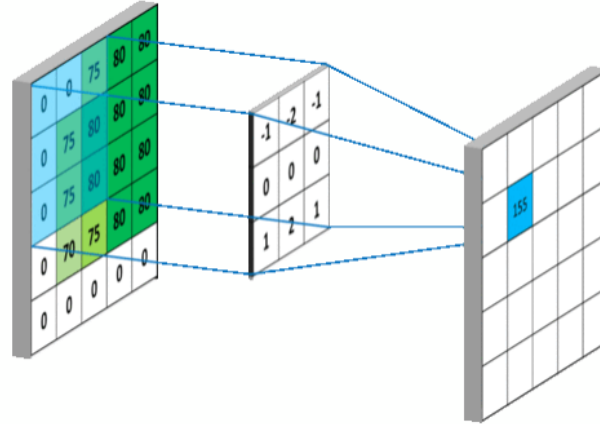


الصورة الأصلية	تمويه جاوسي	زيادة الحدة	اكتشاف الحواف
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
			

https://miro.medium.com/max/1206/1*ZPXWZDIHFbTxs-6KVPS5gg.png

عملية الترشيح

- إن عملية الترشيح "رياضيًا" هي ببساطة عبارة عن تمرير مرشح على صورة جمع لضرب العناصر بين المرشح ومنطقة الصورة التي يغطيها



مثال لعملية الترشيح

0 _{x0}	0 _{x1}	1 _{x0}	1 _{x0}	0 _{x1}	0 _{x0}
0 _{x0}	0 _{x0}	1 _{x1}	1 _{x0}	0 _{x1}	0 _{x0}
0 _{x0}	0 _{x0}	1 _{x1}	1 _{x0}	0 _{x1}	0 _{x0}
0 _{x0}	0 _{x0}	1 _{x1}	1 _{x0}	0	0
0	0	1	1	0	0
0	0	1	1	0	0

$$\begin{aligned}
 &= 0 \times 0 + 1 \times 0 + 0 \times 1 \\
 &+ 0 \times 0 + 1 \times 0 + 0 \times 1 \\
 &+ 0 \times 0 + 1 \times 0 + 0 \times 1
 \end{aligned}$$

0	1	0
0	1	0
0	1	0

النواة أو
المرشح

0	3	3	0
0	3	3	0
0	3	3	0
0	3	3	0

النتائج

معاملات الترشیح

حجم المرشح

بالإمكان تغيير حجم المرشح.

0	0	1	2	1	0	0
0	3	13	22	13	3	0
1	13	59	97	59	13	1
2	22	97	159	97	22	2
1	13	59	97	59	13	1
0	3	13	22	13	3	0
0	0	1	2	1	0	0

7x7

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

5x5

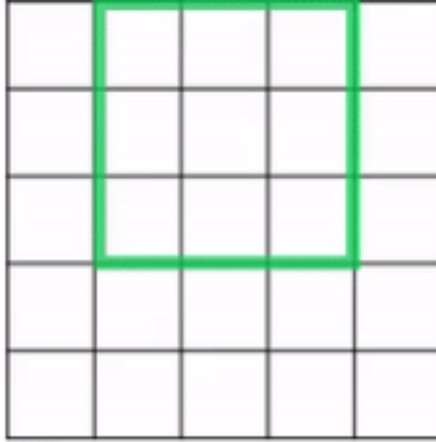
1	2	1
2	4	2
1	2	1

3x3

مشكلة تأثير الحدود The Border Effect

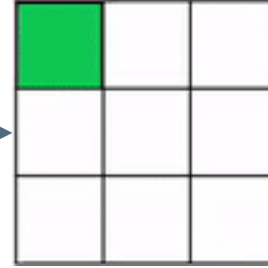
- صورة بحجم 5x5 يتم تصغيرها إلى 3x3 بعد تطبيق مرشح 3x3 عليها، وبالتالي تضيع المعلومات من على حدود الصورة الأصلية.

الصورة الأصلية (5x5)

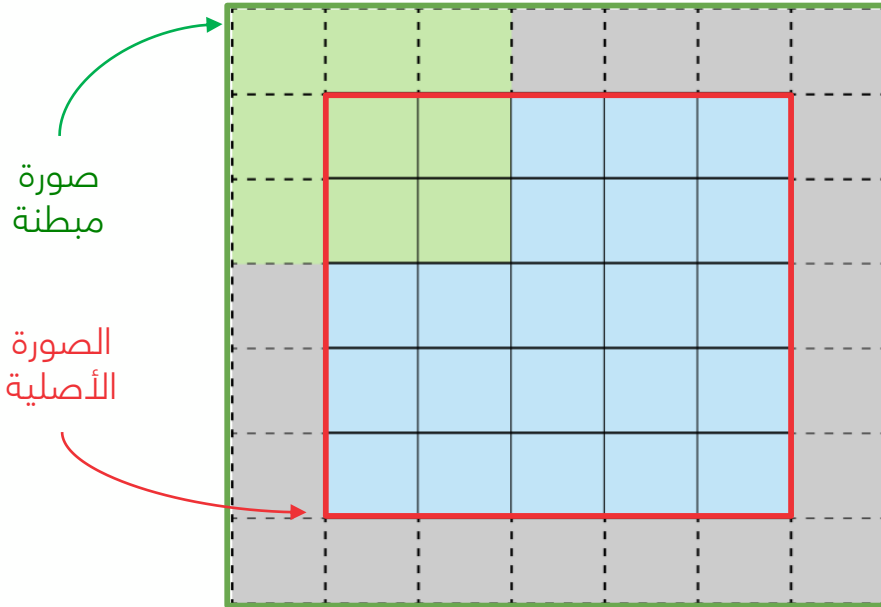


ترشيح (3x3)

الصورة الناتجة (3x3)



علاج مشكلة تأثير الحدود بالتبطين Padding



الصورة الناتجة

مثال على التبطين

0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

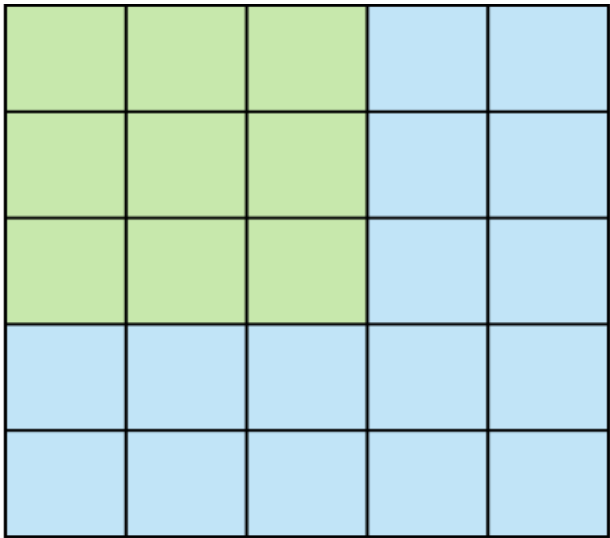
نواة المرشح

0	-1	0
-1	5	-1
0	-1	0

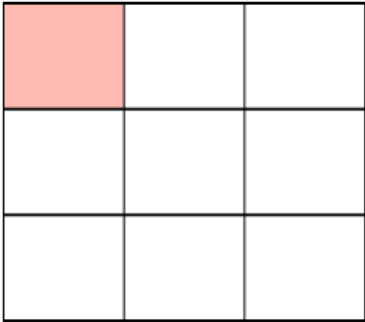
114				

تقليل عينات الإدخال بتكبير خطوة المرشح

Downsample Input with Stride

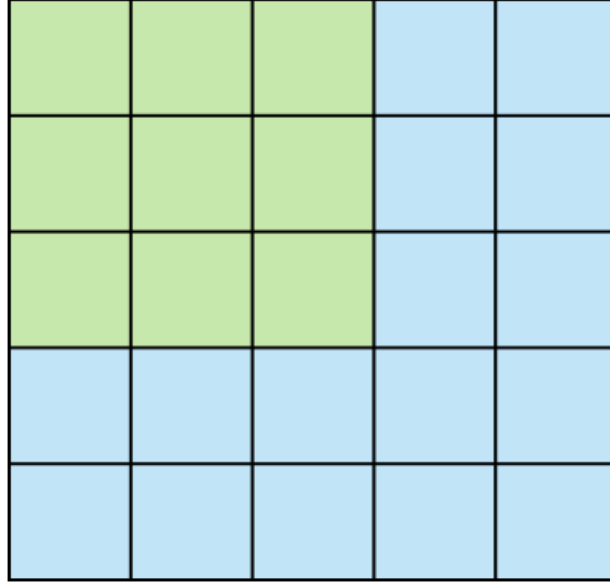


الخطوة 1 =

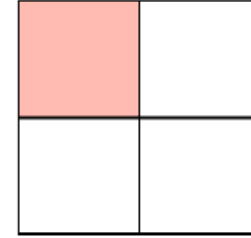


خريطة الخصائص
النتيجة

تقليل عينات الإدخال بتكبير خطوة المرشح



الخطوة 2 =

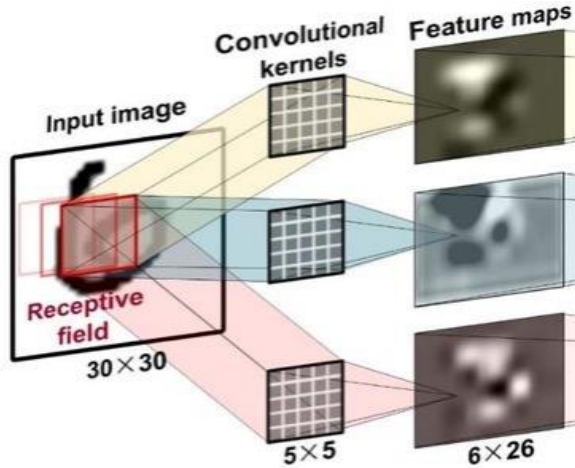


خريطة الخصائص
النتيجة

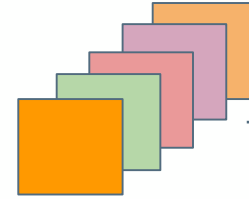
الطبقات الترشيحية

الطبقات الترشيحية Convolutional Layers

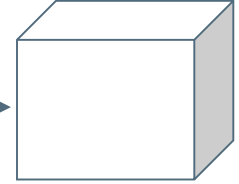
اعتمادًا على قيم النواة، يختلف الإخراج، لهذا لكل إدخال، يمكن تطبيق العديد من المرشحات للحصول على معلومات متعددة من صورة الإدخال – وهذا ما نشير إليه بالطبقة الترشيحية.



صورة الإدخال



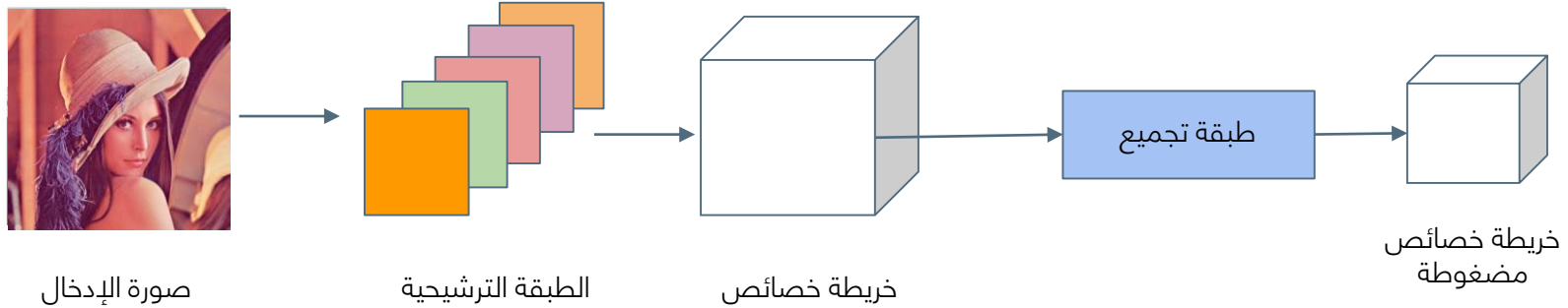
الطبقة الترشيحية



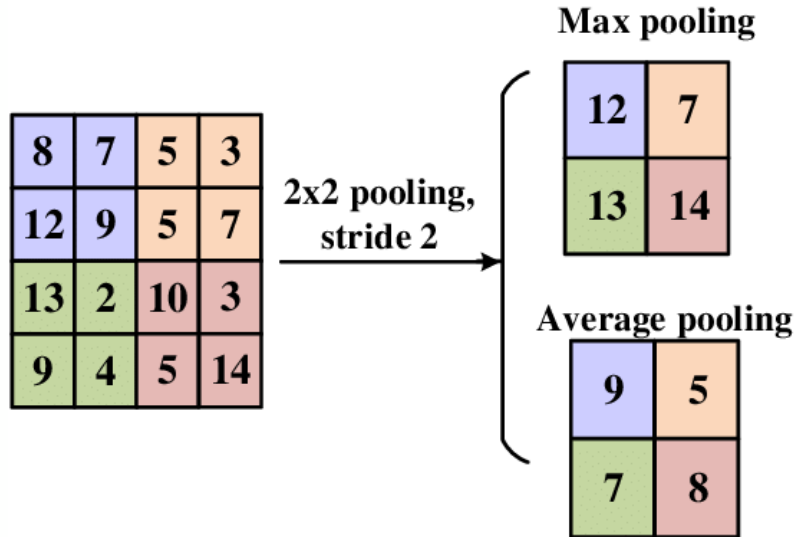
خريطة خصائص
ثلاثية الأبعاد

طبقات التجميع Pooling Layers

- بعد استخراج خريطة الخصائص، نحتاج إلى ضغط أو تكديس هذا الناتج حتى يتمكن المصنف النهائي من استخدامه للتنبؤ بطريقة أسهل.
- للقيام بذلك نستخدم ما نسميه طبقات التجميع.



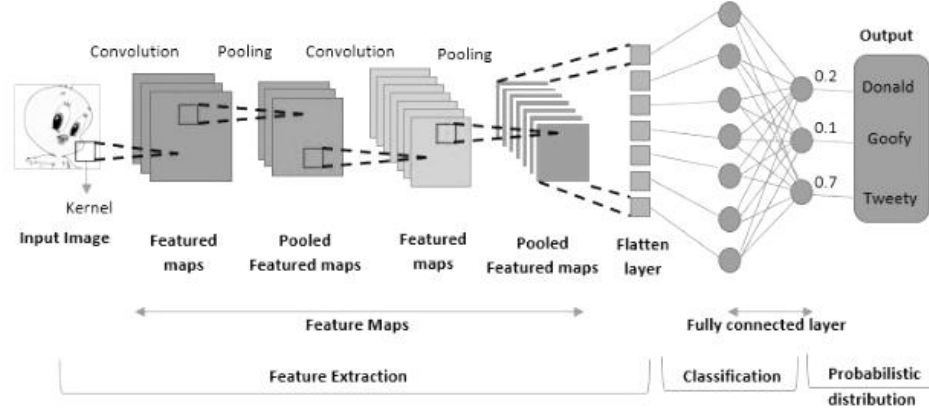
طبقات التجميع



- هناك شكلان رئيسيان من التجميع: تجميع القيمة القصوى (Max Pooling) وتجميع القيمة المتوسطة (Average Pooling).
- يأخذ Max Pooling القيمة القصوى لمنطقة الصورة.
- يأخذ Average Pooling القيمة المتوسطة لمنطقة الصورة.

الشبكة العصبية الترشيحية

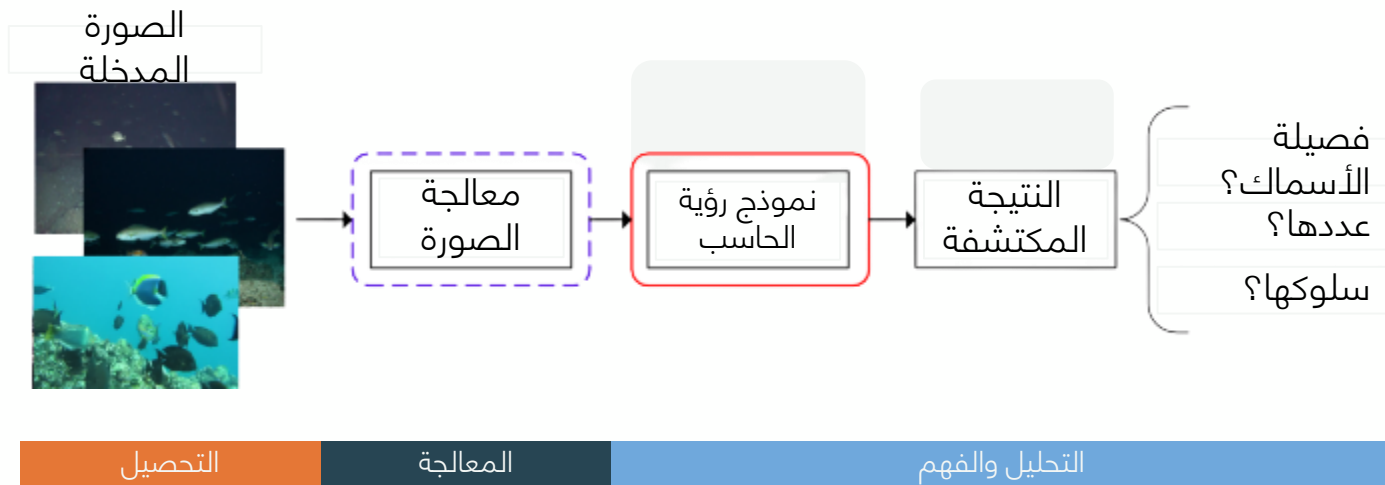
- عندما نقوم بتكديس كل هذه الطبقات نحصل على الشبكة العصبية الترشيحية (CNN).
- الهدف من هذه الشبكة هو معرفة المجموعة المثلى لقيم المرشح جنبًا إلى جنب مع الأوزان في الجزء المتصل بالكامل.



إعداد البيانات للشبكات العصبية الترشيحية

إعداد البيانات

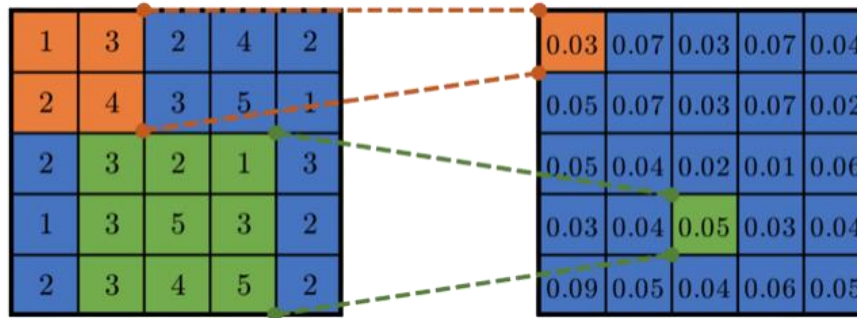
- تمامًا مثل أي مشكلة أخرى في التعلم العميق، بمجرد تحديد هيكل النموذج، يتعين علينا إعداد البيانات بطريقة يمكننا من تدريب النموذج عليها.



إعداد البيانات

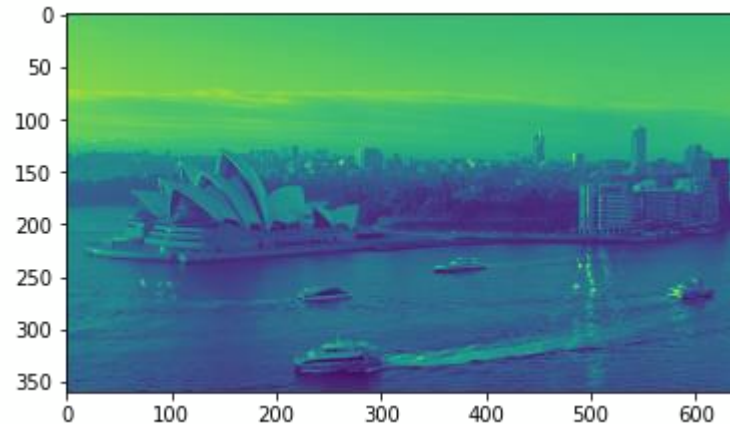
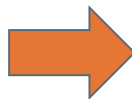
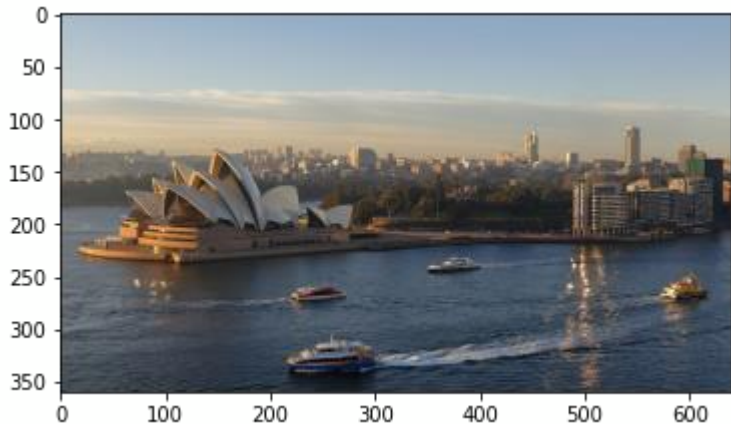
- تسوية قيم البكسلات:

- على هذا النحو ، فمن الممارسات الجيدة تسوية قيم البكسل بحيث يكون لكل بكسل قيمة بين 0 و1.
- نظرًا لأن قيم البكسل تتراوح بين 0 و255، يمكن أن نقوم بالتسوية عبر قسمة القيم على 255.



إعداد البيانات

- تمرکز قيم البكسل:
يتطلب التمرکز أن يتم حساب متوسط قيمة البكسل قبل طرحها من قيم البكسل.



مولد بيانات الصور (Image Data Generator)



توفر فئة Keras مجموعة من التقنيات لقياس قيم البكسل في مجموعة بيانات الصور الخاصة بك قبل النمذجة:

- يقوم بتغليف مجموعة بيانات الصورة الخاصة بك
- ثم (عند الطلب) يقوم بإرجاع الصور على دفعات إلى الخوارزمية أثناء التدريب أو التحقق من الصحة أو التقييم.
- يقوم بتطبيق عملية تغيير القياس في الوقت المناسب.

مولد بيانات الصور (Image Data Generator)

استخدام مولد بيانات الصور (ImageDataGenerator) كالتالي:

- قم بتحميل مجموعات بيانات الصور.
- قم بتكوين فئة ImageDataGenerator
- حساب إحصائيات الصورة عند اللاقتضاء، عن طريق استدعاء الدالة `fit()`.
- قم بإنشاء مكررات بيانات لكل تقسيم بيانات (تدريب، اختبار، تحقق من الصحة) ووجههم إلى مجموعات البيانات.
- استخدم مكررات البيانات لملاءمة النموذج (قم بتمرير المثيل إلى دالة `model.fit()`).
- استخدم مكررات البيانات لتقييم النموذج (قم بتمرير المثيل إلى دالة `model.evaluate()`).

مولد بيانات الصور (ImageDataGenerator)

```
# قم ببناء مثيل المولد بالمعامل المناسبة
datagen = ImageDataGenerator(...)

# بحساب إحصائيات بيانات الصور من تقسيم البيانات المختص بالتدريب
datagen.fit(trainX)

# قم ببناء مثيل لمكرر بيانات التدريب
train_it = datagen.flow(trainX, trainY, batch_size=64)
# قم ببناء مثيل لمكرر بيانات الاختبار
test_it = datagen.flow(testX, testY, batch_size=64)

# قم بتدريب النموذج
model.fit(train_it, validation_data=test_it, ...)
```

تحميل مجموعة بيانات الصور من ال Directory

هيكل مجلدات وملفات مجموعة البيانات:

```
data/  
data/train/  
data/test/  
data/validation/
```



```
data/  
data/train/  
data/train/red/  
data/train/blue/  
data/test/  
data/test/red/  
data/test/blue/  
data/validation/  
data/validation/red/  
data/validation/blue/
```



```
data/train/red/car01.jpg  
data/train/red/car02.jpg  
data/train/red/car03.jpg  
...  
data/train/blue/car01.jpg  
data/train/blue/car02.jpg  
data/train/blue/car03.jpg
```



ملفات مختلفة

تحميل الصور بشكل تدريجي

```
# قم ببناء مثيل المولد بالحجج المناسبة
datagen = ImageDataGenerator(...)

# قم ببناء مثيل لمكرر بيانات التدريب
train_it = datagen.flow_from_directory("dataset/train", class_mode="binary", batch_size=64)

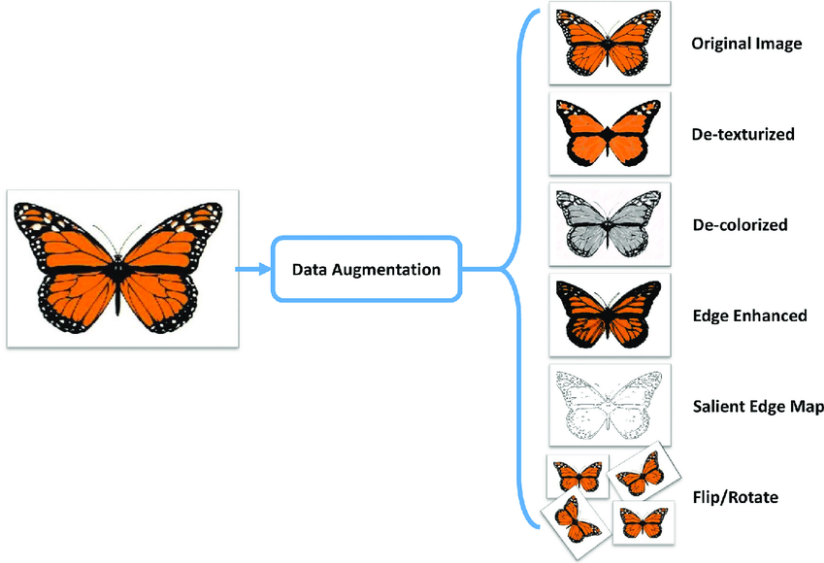
# قم ببناء مثيل لمكرر بيانات الاختبار
test_it = datagen.flow_from_directory("dataset/test", class_mode="binary", batch_size=64)

# قم بتدريب النموذج
model.fit(train_it, validation_data=test_it, ...)
```

زيادة البيانات

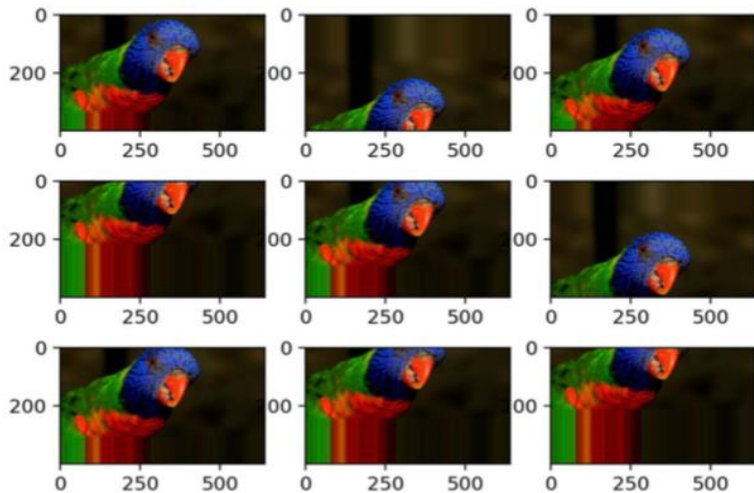
زيادة البيانات باستخدام
ImageDataGenerator:

- ترحيل أو تحريك الصور
- قلب الصور
- تدوير الصور
- سطوع الصور
- تكبير الصور

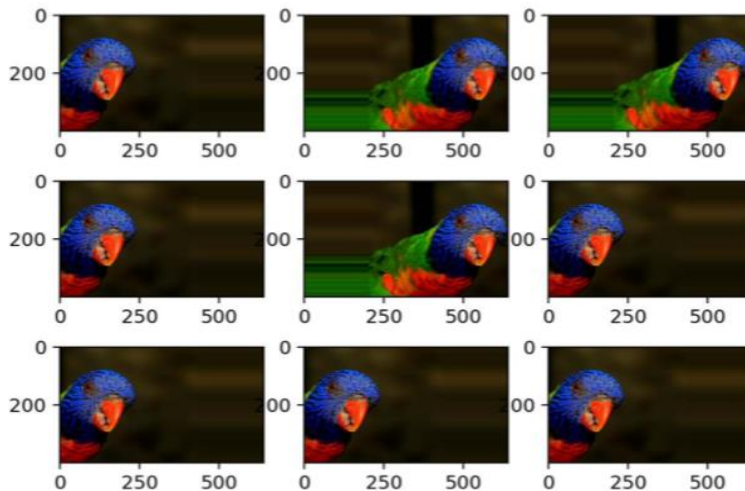


ترحيل أو تحريك الصور

Height_shift_range = [min, max]

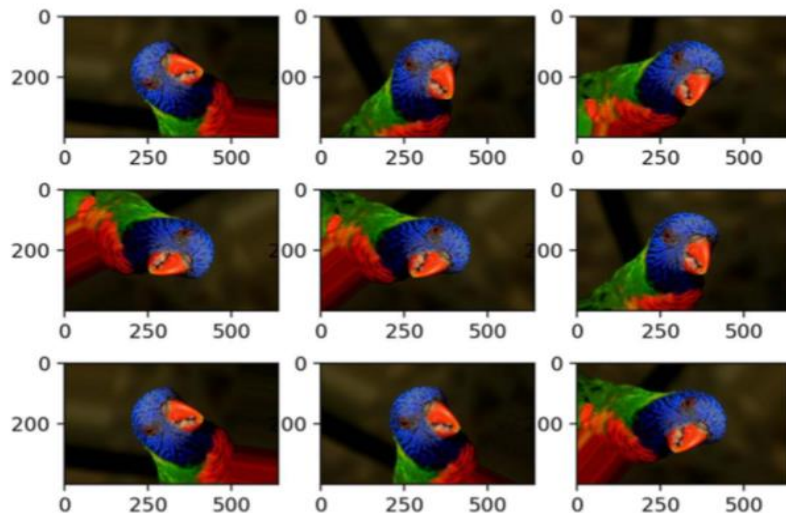


Width_shift_range = x

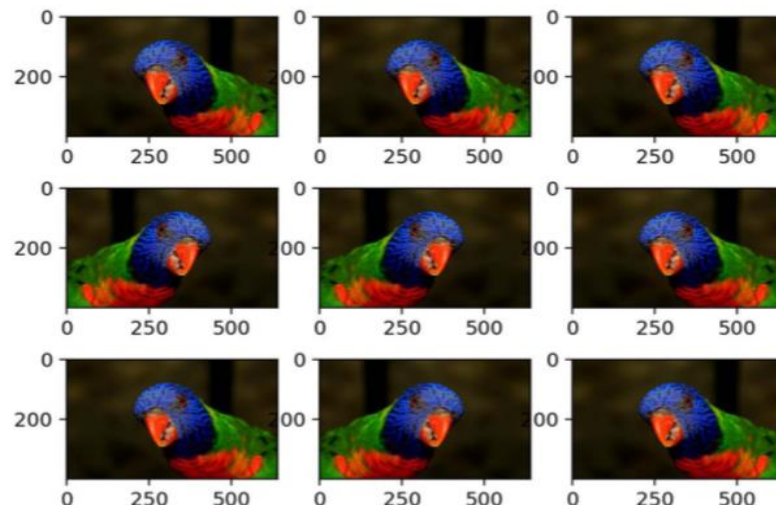


قلب الصور

horizontal_flip=True

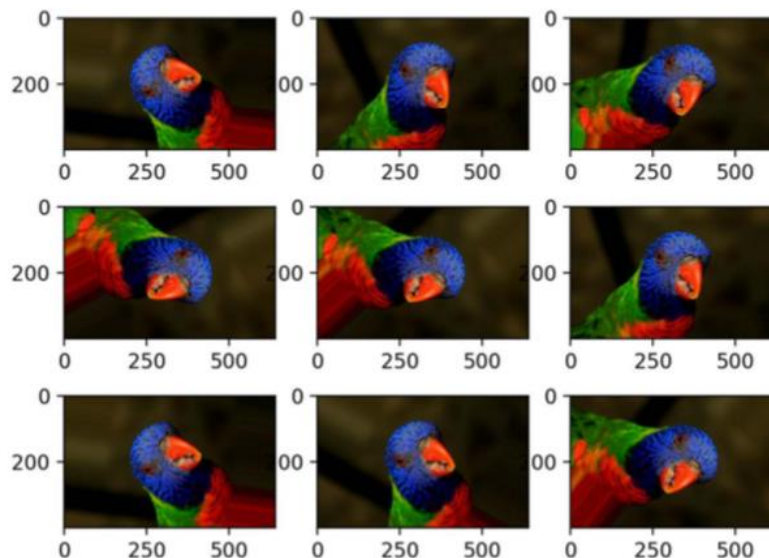


vertical_flip=True



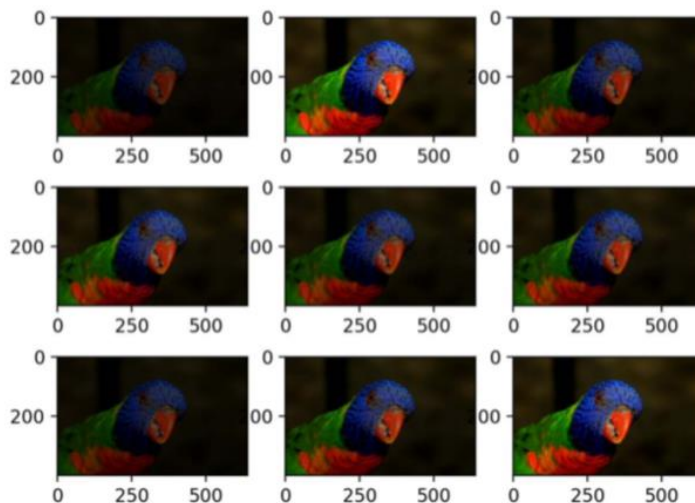
تدوير الصور

rotation_range=*angle*



سطوع الصورة

```
brightness_range=[min,max]
```



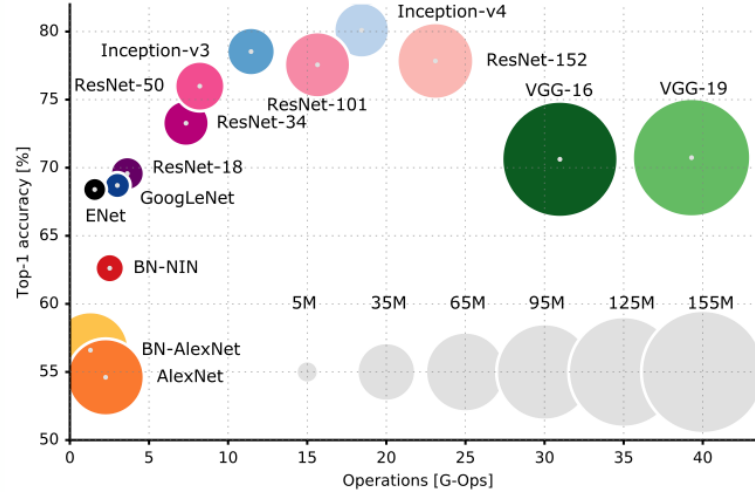
The background features several abstract geometric shapes in various shades of blue, teal, and orange, scattered across the top and sides of the frame.

تدريب عملي: إتقان بنى الشبكات العصبية الترشيحية Mastering CNN Architectures

الشبكات العصبية الترشيحية المتقدمة

بنى الشبكات العصبية الترشيحية المتقدمة

لطالما كانت الشبكات العصبية الترشيحية موضوعًا مشوقًا، ولهذا السبب، تم اقتراح العديد من البنى المتقدمة التي توفر أحدث أداء.

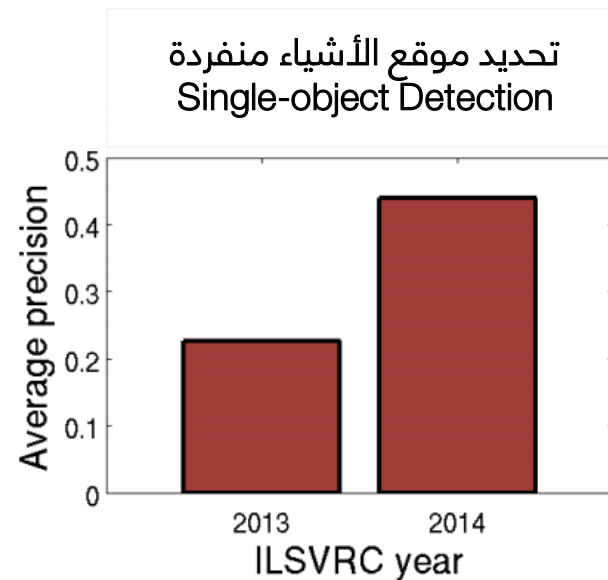
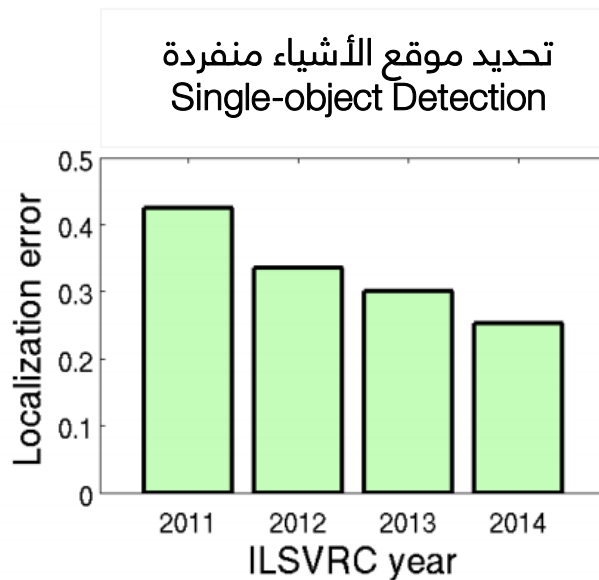
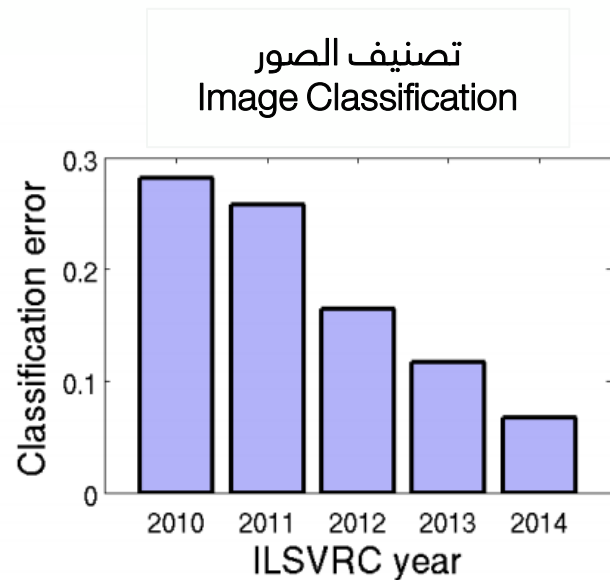


مجموعة بيانات ImageNet



- تتطلب البنى المتقدمة مجموعة بيانات كبيرة للتدريب.
- ImageNet عبارة عن مجموعة كبيرة من الصور المسماة يدوياً والمستخدم لتطوير خوارزميات رؤية الحاسب.
- يعد تحدي ImageNet للتعرف البصري واسع النطاق (ILSVRC) عبارة عن مسابقة سنوية تستخدم مجموعات فرعية من ImageNet لتعزيز تطوير وقياس الخوارزميات المتقدمة.

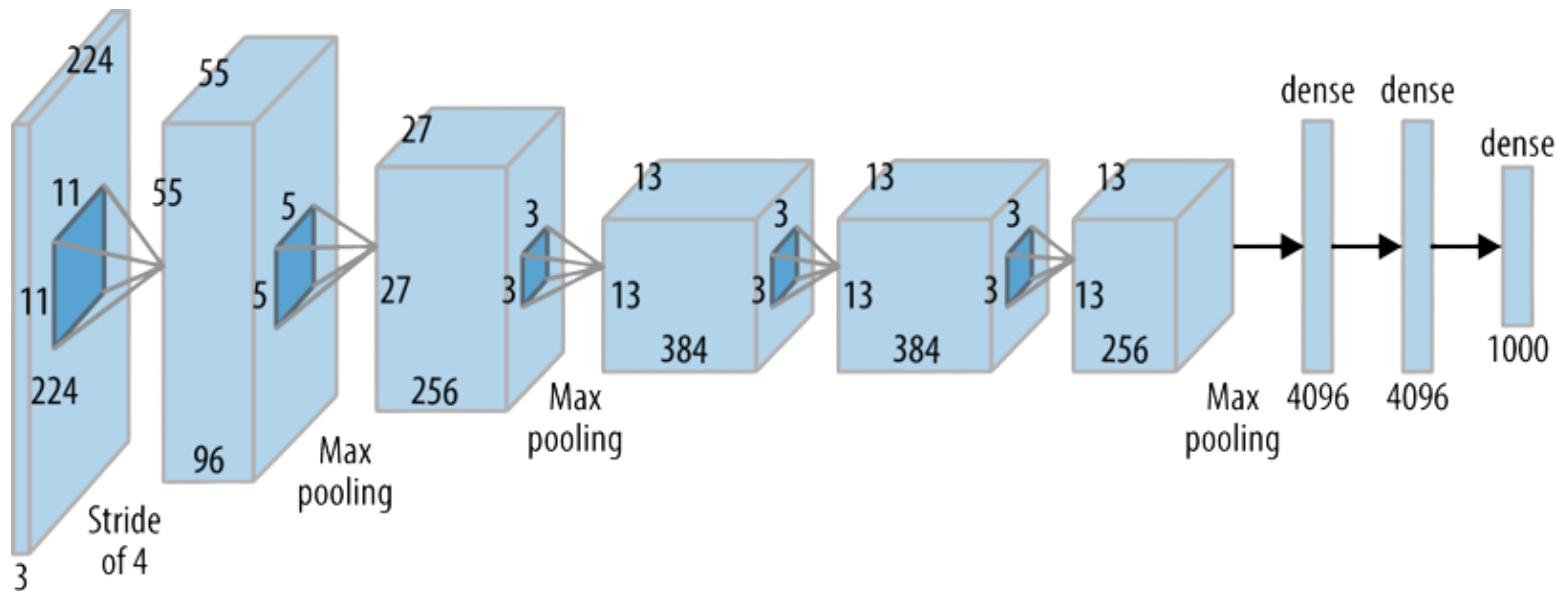
تحدي التعرف البصري على نطاق واسع من ImageNet



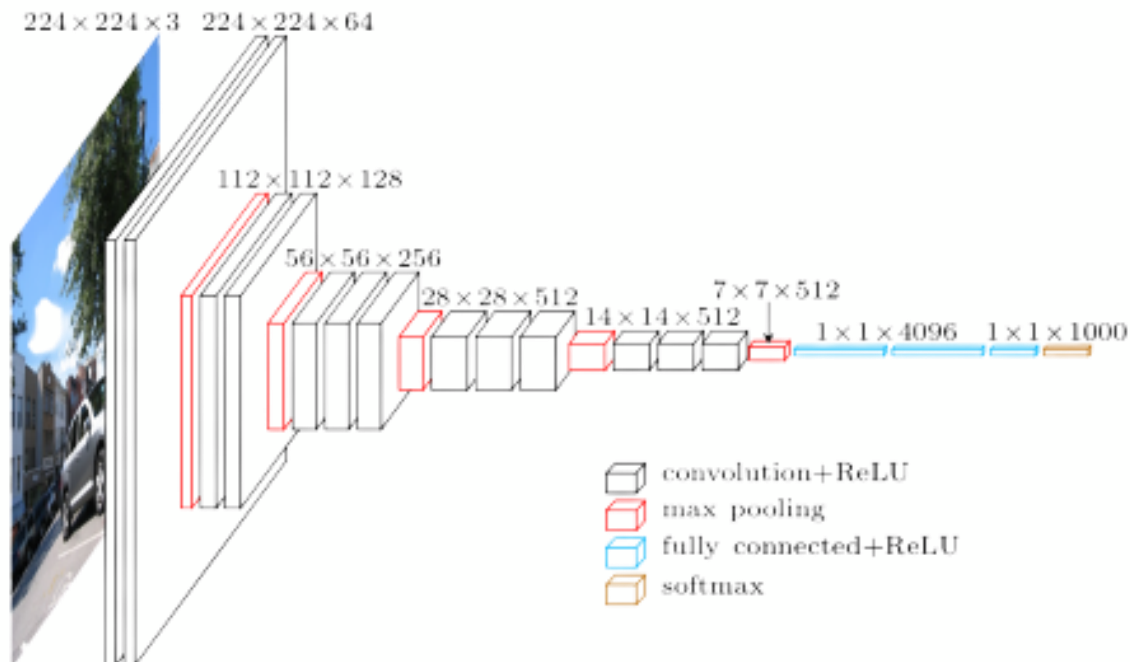
نتائج ILSVRC

- 2012: AlexNet (SuperVision)
- 2013: ZFNet (Clarifai)
- 2014 (اكتشاف الأشياء): Inception (GoogLeNet)
- 2014 (تصنيف الصور): VGG
- 2015: ResNet

AlexNet

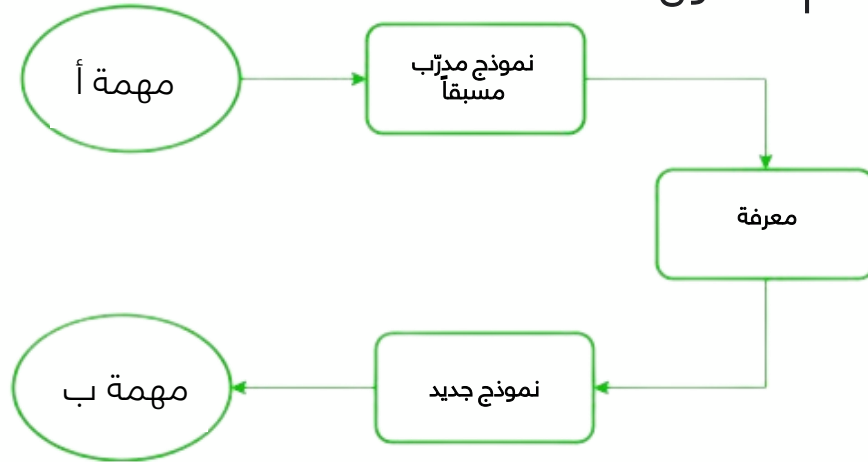


VGG 19



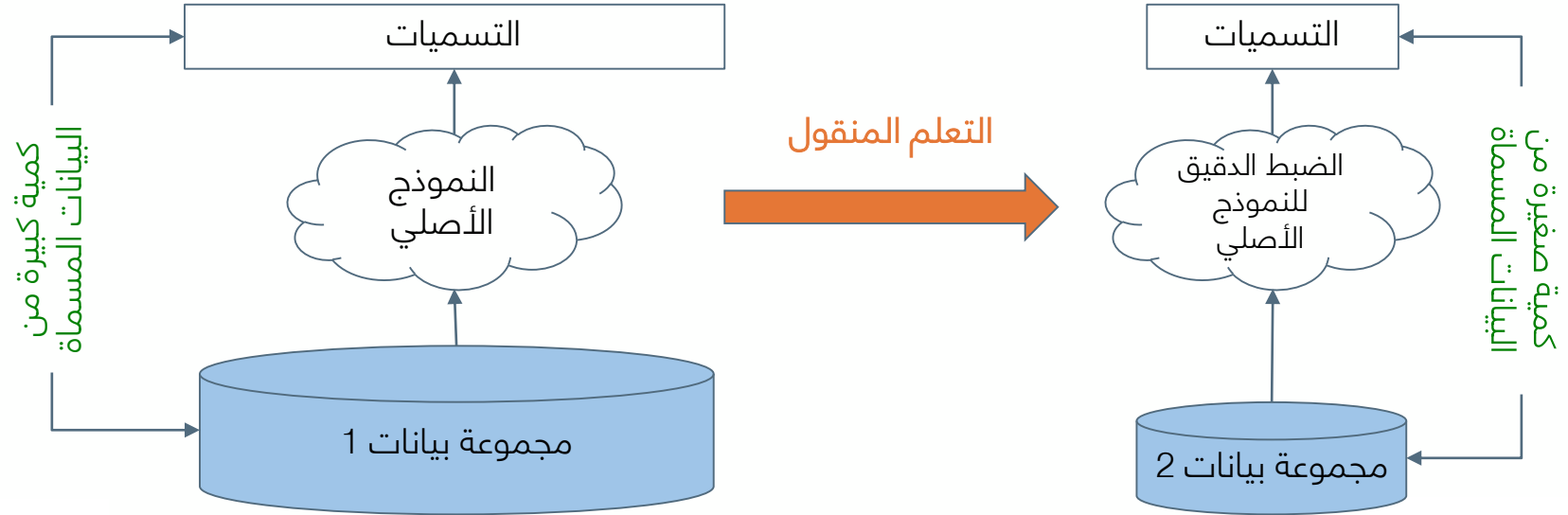
لماذا البنى المتقدمة؟

- يتم تدريب هذه النماذج على مجموعة بيانات ضخمة ، مما يعني أنها اكتسبت معرفة كبيرة .
يمكننا استخدام هذه المعرفة والارتكاز عليها لبناء نماذج بدلاً من تدريب الأشياء من الصفر.
- هذا ما نسميه بالتعلم المنقول.



التعلم المنقول للبيانات المحدودة

التعلم المنقول هو أسلوب قوي للاستخدام عند التعامل مع كمية محدودة من البيانات.



التعلم المنقول للبيانات المحدودة



شكراً لكم

Thank you



SDAIA

الهيئة السعودية للبيانات
والذكاء الاصطناعي
Saudi Data & AI Authority