Convolutional Neural Networks
CASA course
(09/11/2018)

D. Freire-Obregón, PhD.

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

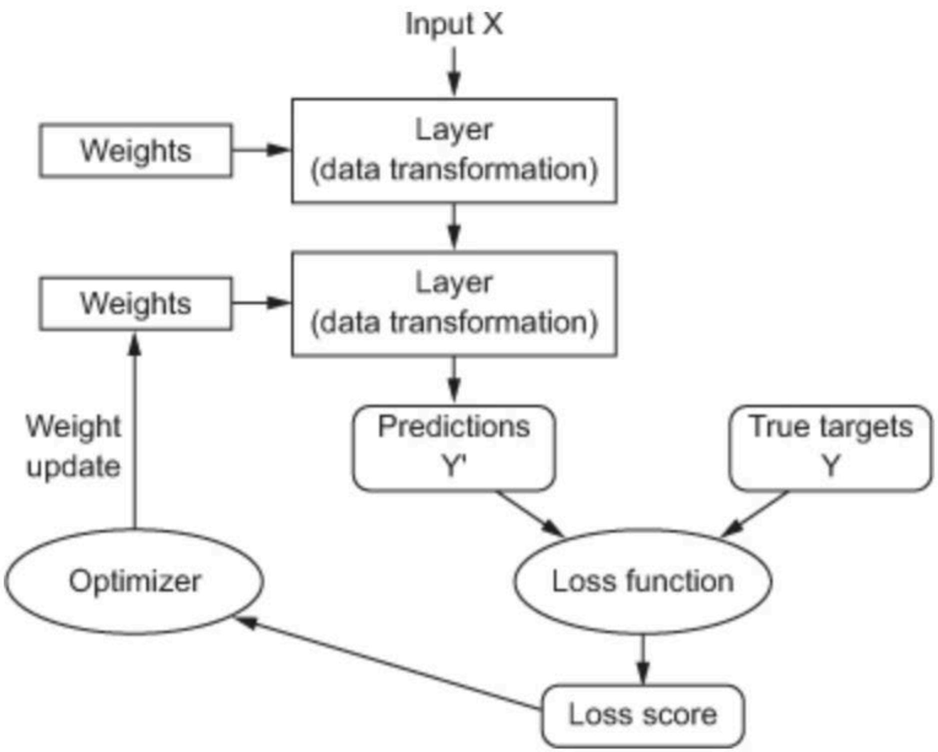
Plan for TS4

- Deep Learning
- Convolutional Operation
- Understanding border effects
- Max-pooling operation
- Fully Connected Layer
- What is happening?
- Where to use CNN?

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

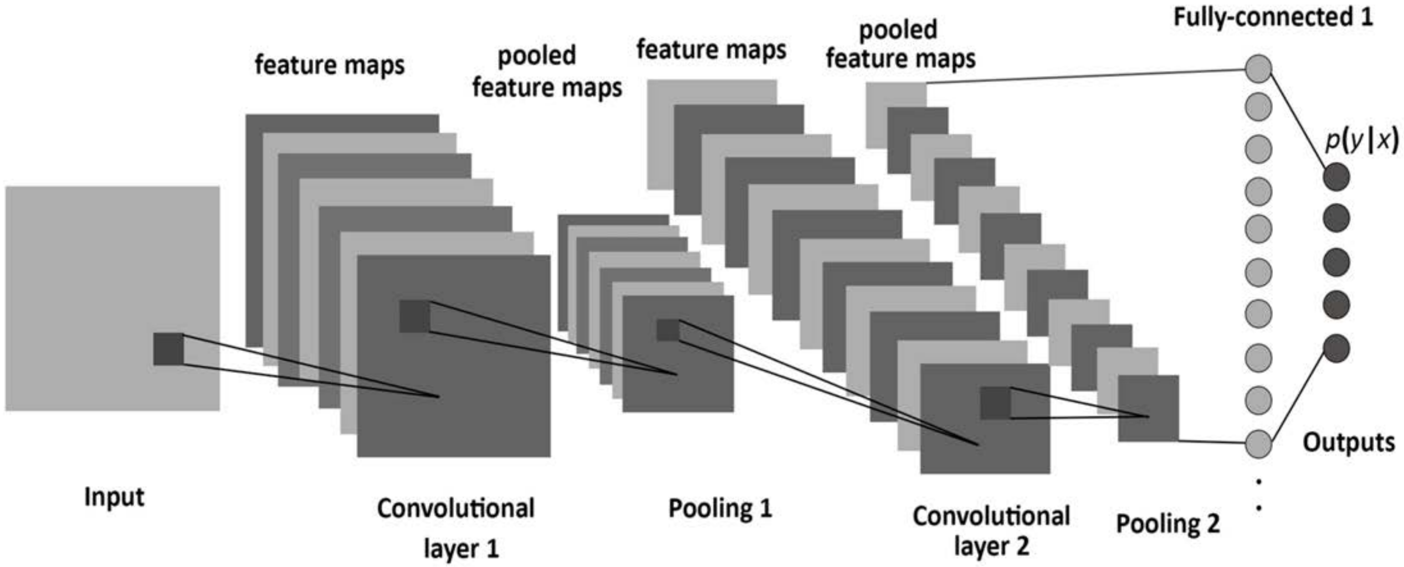
Deep Learning

Does this scheme change? No!



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Deep Learning



Convolutional Operation

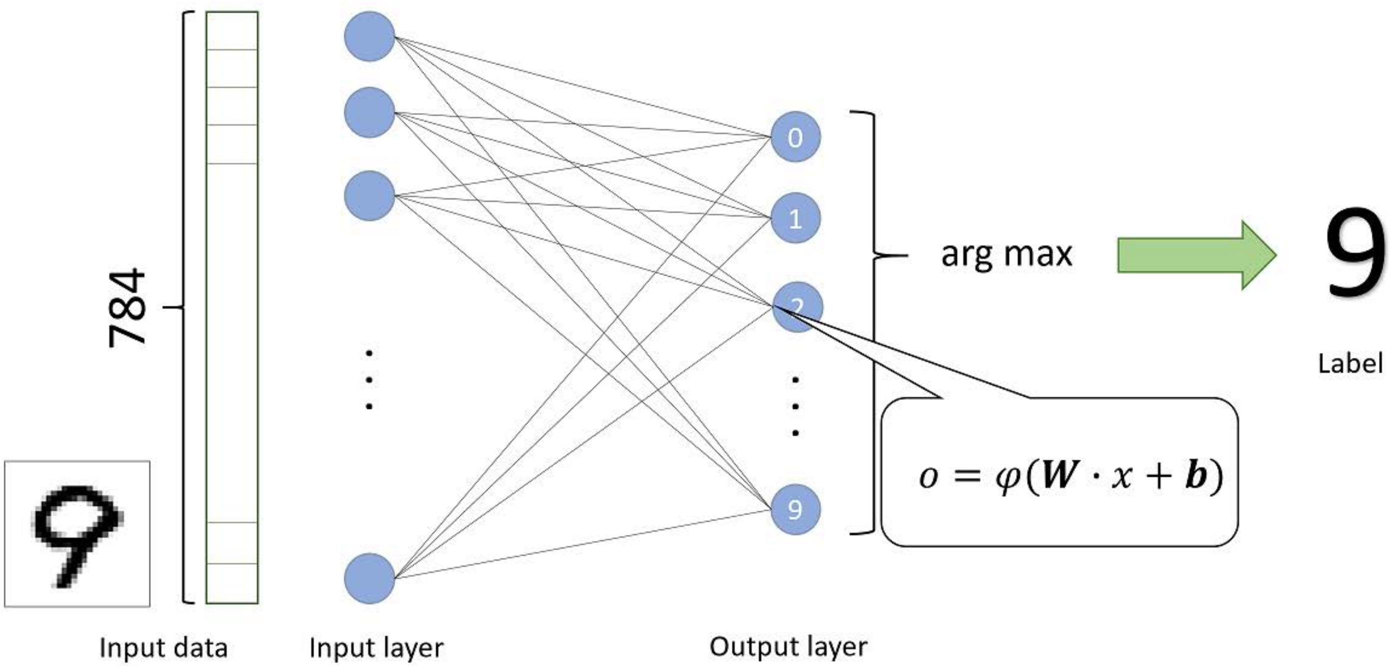
What is the difference between a densely connected layer and a convolution layer?

•Dense layers → Learn global patterns in their input space.

•Convolutional layers → Learn local patterns

Convolutional Operation

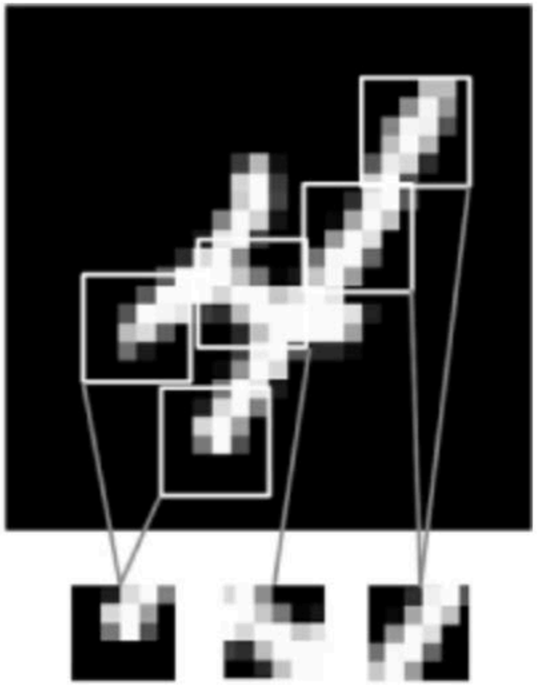
Dense layers:



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

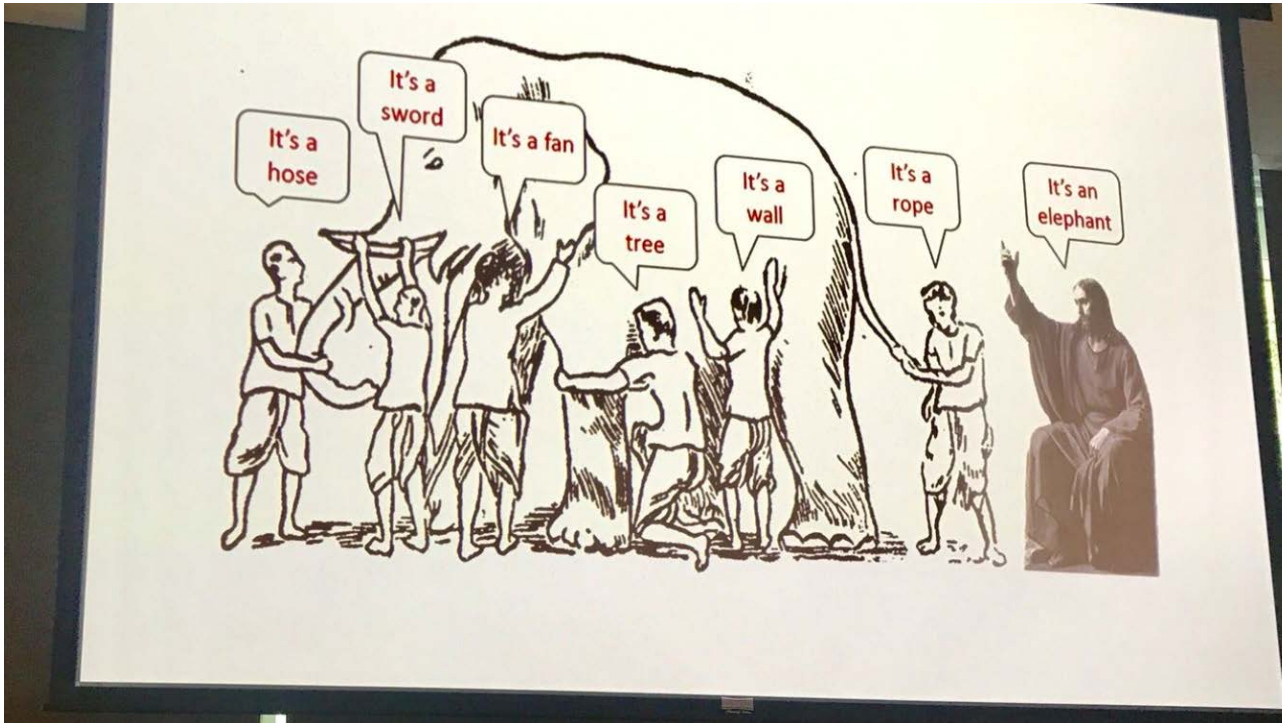
Convolutional Operation

Images can be broken into local patterns:
Edges, textures, etc…



Convolutional Operation

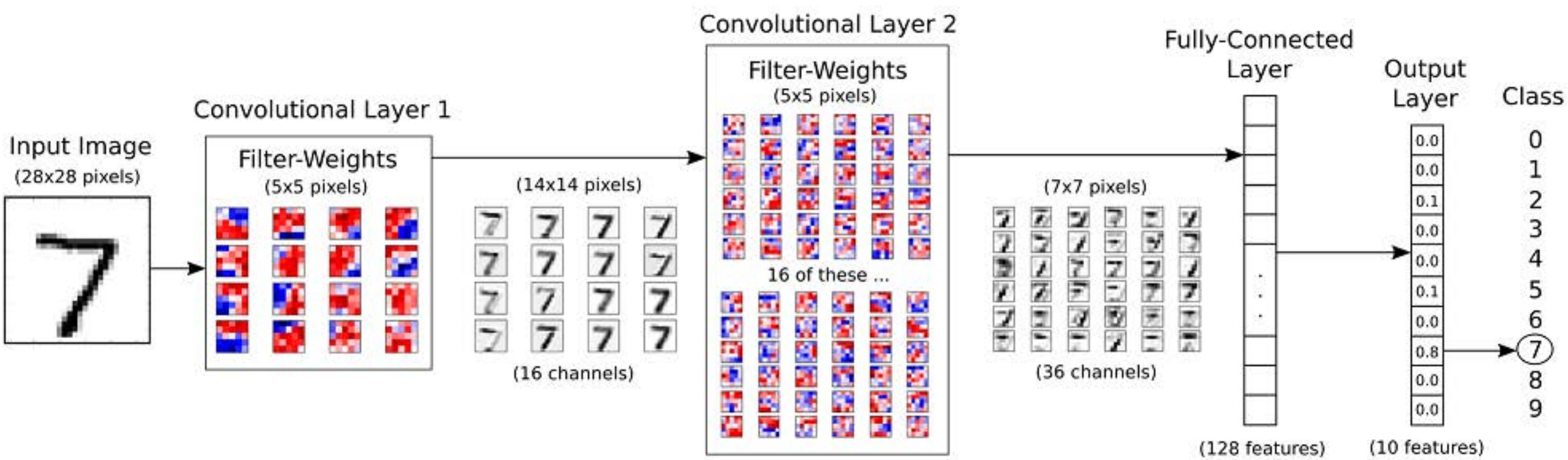
Convolutional layers:



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Convolutional Operation

Convolutional layers:



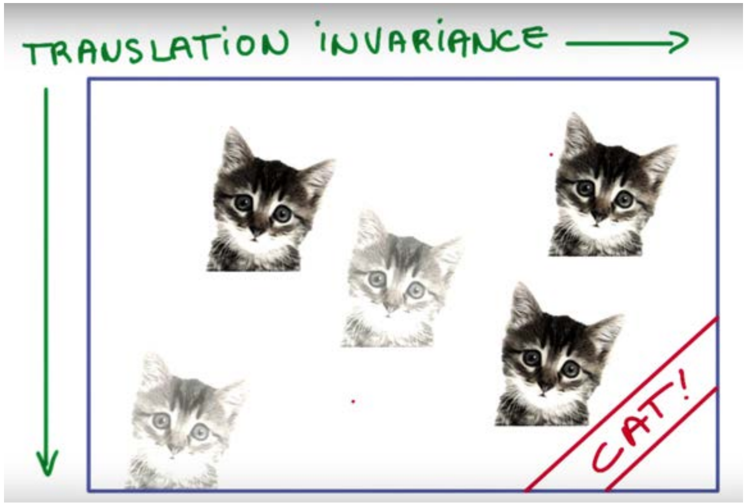
Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Convolutional Operation

The hability to break the image into local patterns gives convnets two properties:

1. The learned patterns are translation invariant → After learning a certain pattern in the top-left corner of the picture, a convnet can recognize it anywhere (p.e. Lower-right corner)

    The visual world is fundamentally translation invariant:



INSTITUTO UNIVERSITARIO SIANI INGENIERIA COMPUTACIONAL

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

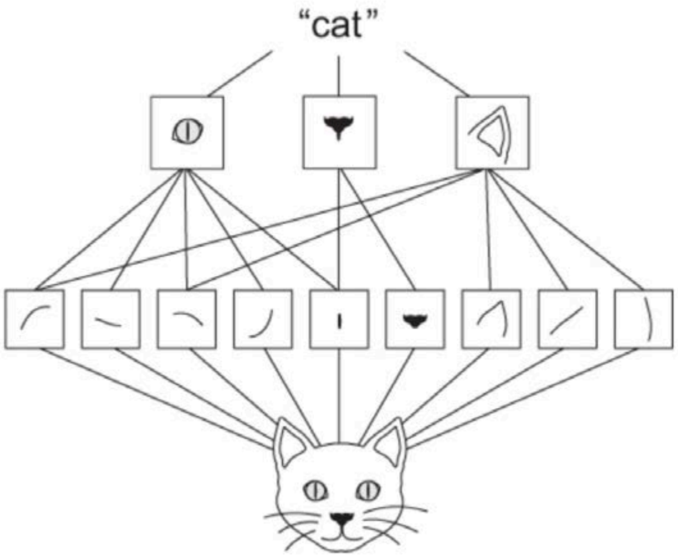
UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Convolutional Operation

The hability to break the image into local patterns gives convnets two properties:

2. They can learn spatial hierarchies of patterns. A first Conv. Layer will learn a small local patterns such as edges, then a second Conv. Layer will learn larger patterns made of the features of the first layer, and so on.

The visual world is fundamentally spatially hierarchical:



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Convolutional Operation

Operates over 3D tensors known as → Featured maps (height x width x depth)

For RGB image → depth = 3 at the first feature map (input)

The convolutional operation:

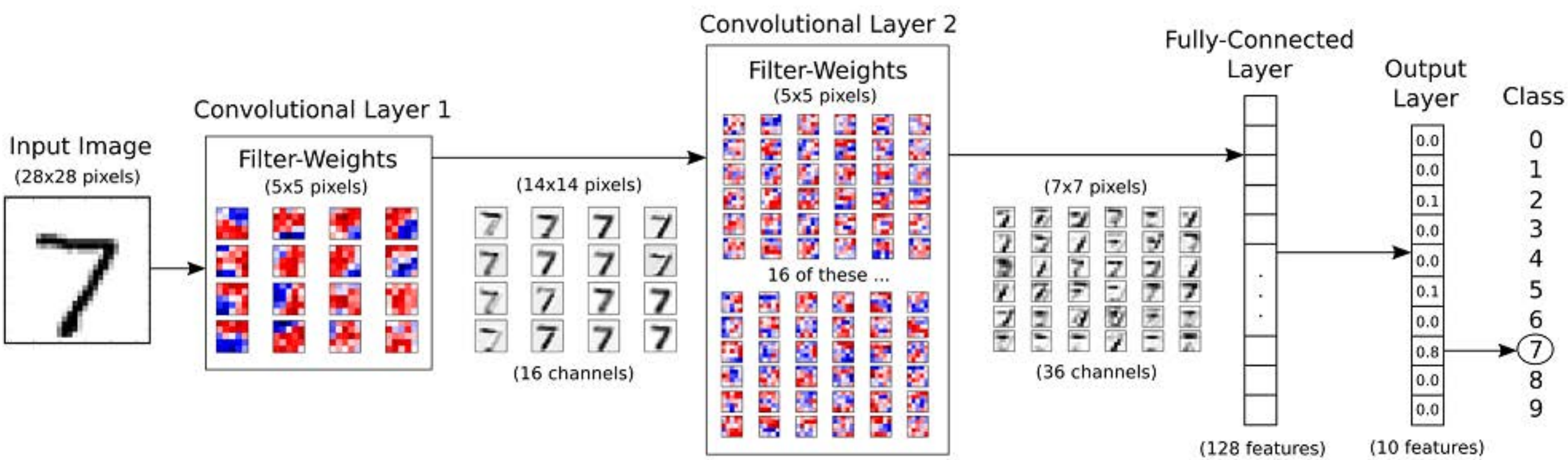
1. extracts patches from its input feature map.
2. Applies the same transformation to all this patches
3. Generates an output feature map (height x width x depth)

   depth → no longer RGB colors, now number of filters.

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Convolutional Operation

Convolutional layers:



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Convolutional Operation

MNIST example:

...

model = models.sequential()

model.add(layers.Conv2D(32, (3,3), activation='relu', input_shape=(28,28,1)))

...

The first convolutional layer takes a feature map of (28, 28, 1)

And outputs a feature map of size (26, 26, 32):

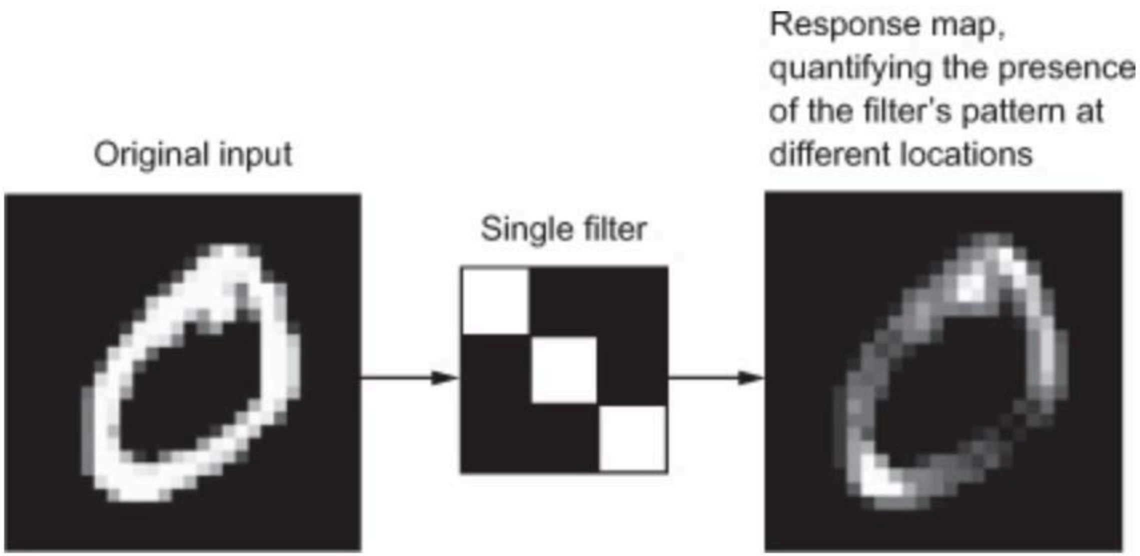
It computes 32 filters over the input.

Each feature map contains a 26x26 grid of values.

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Convolutional Operation

Feature map → is the 2D map of the response of this filter over the input.



Convolutional Operation

Convolutions are defined by two keys parameters:

1. Size of the patches extracted from the inputs → Typically 3x3 or 5x5.

2. Depth of the output feature map → Number of filters computed by the convolution. It is a very dynamic value, usually you start with a depth at the first layers and end with another depth at the final layers.

These parameters can be specified in Keras Conv2D function:

Conv2D(output_depth, (height, width))

Must *read: http://cs231n.github.io/convolutional-networks/*

Convolutional Operation

MNIST example:

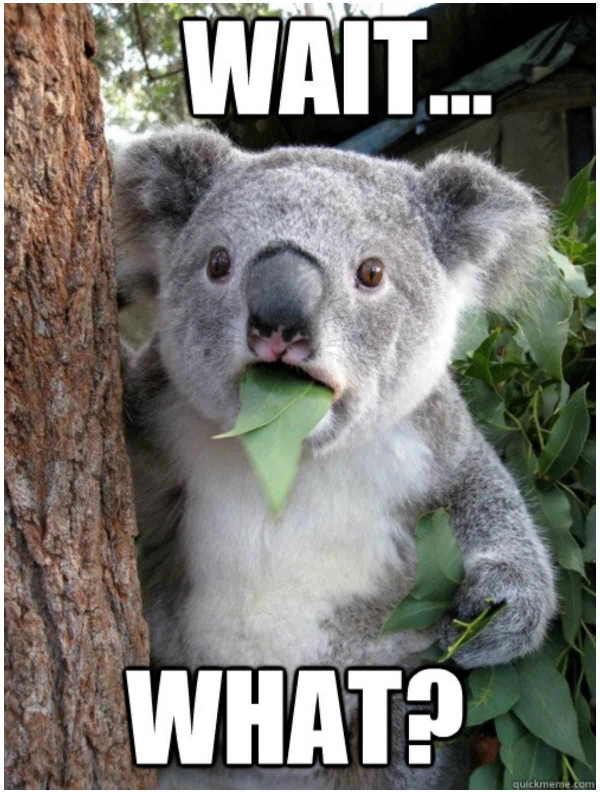
...

model = models.sequential()
model.add(layers.Conv2D(32, (3,3), activation='relu',
input_shape=(28,28,1)))

...

The first convolutional layer takes a feature map of (**28**, **28**, 1)
And outputs a feature map of size (26, 26, 32):

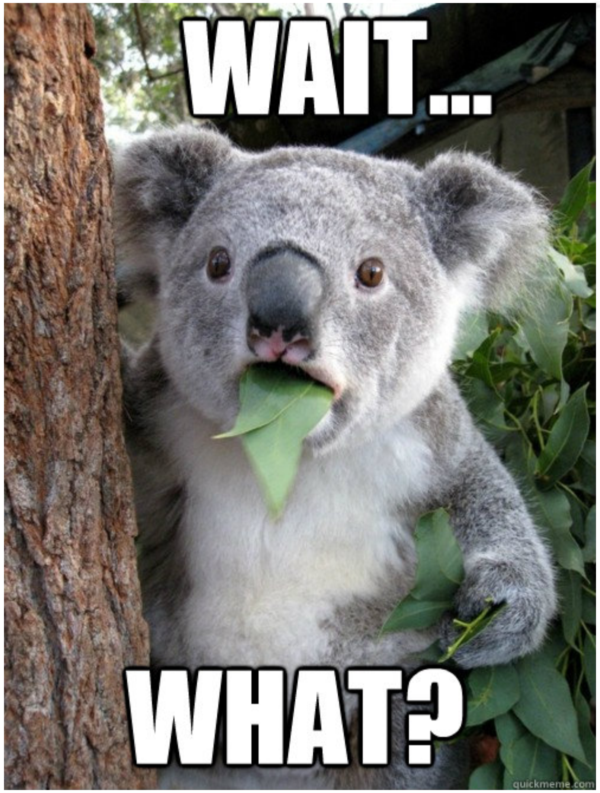
26x26???



Convolutional Operation

```
cnn_model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 32)        320       
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 32)        0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 11, 11, 64)        18496     
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 64)          0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 3, 3, 64)          36928     
_________________________________________________________________
flatten_1 (Flatten)          (None, 576)               0         
_________________________________________________________________
dense_3 (Dense)              (None, 64)                36928     
_________________________________________________________________
dense_4 (Dense)              (None, 10)                650       
=================================================================
Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0
_________________________________________________________________
```



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Understanding border effects

Note that the output width and height may differ from the input width and height.

Two possible reasons:

1. Border effects due to the padding of the input feature map.
2. The use of strides

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

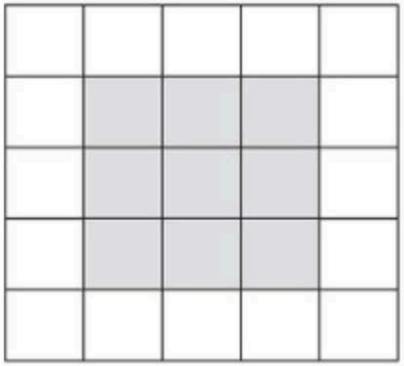
Understanding border effects: Padding

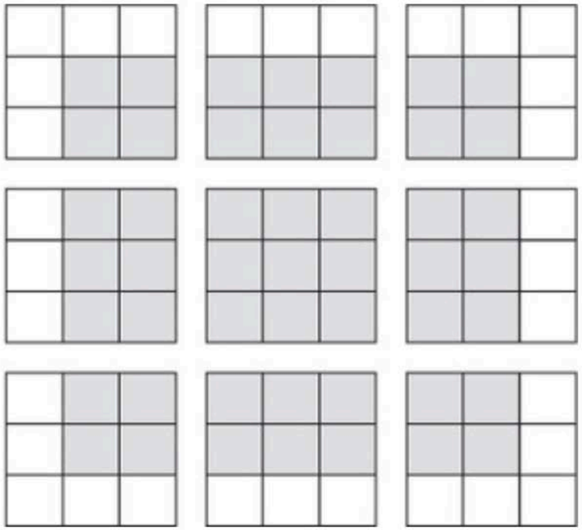
Let's consider:

A 5x5 feature map (25 pixels)

A 3x3 window

Then → There are only 9 possibilities to center the window.





The output feature map shrinks a little (2 pixels x dim)

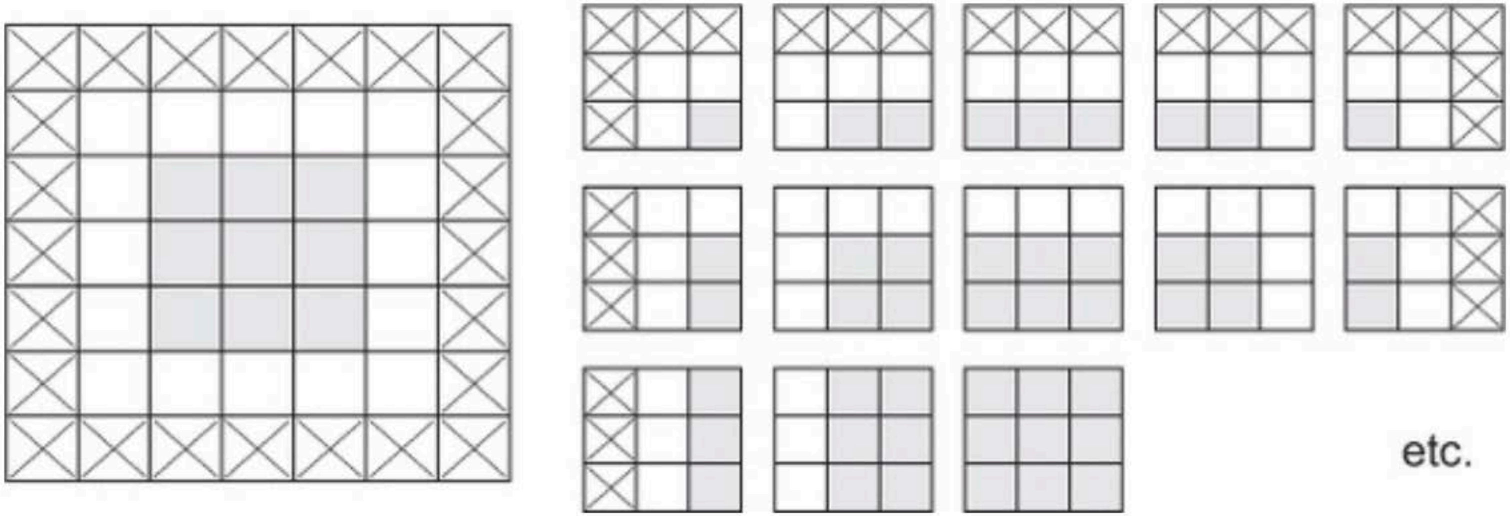
Same for our 28x28 previous example → 26x26 output dim.

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Understanding border effects: Padding

But, let's assume that you want to generate an output feature map with the same spatial dimension as the input → use the parameter *padding*.

Padding consists of adding an appropiate number of rows&columns each side of the input feature map in order to be able to centre perfectly the convolution window.



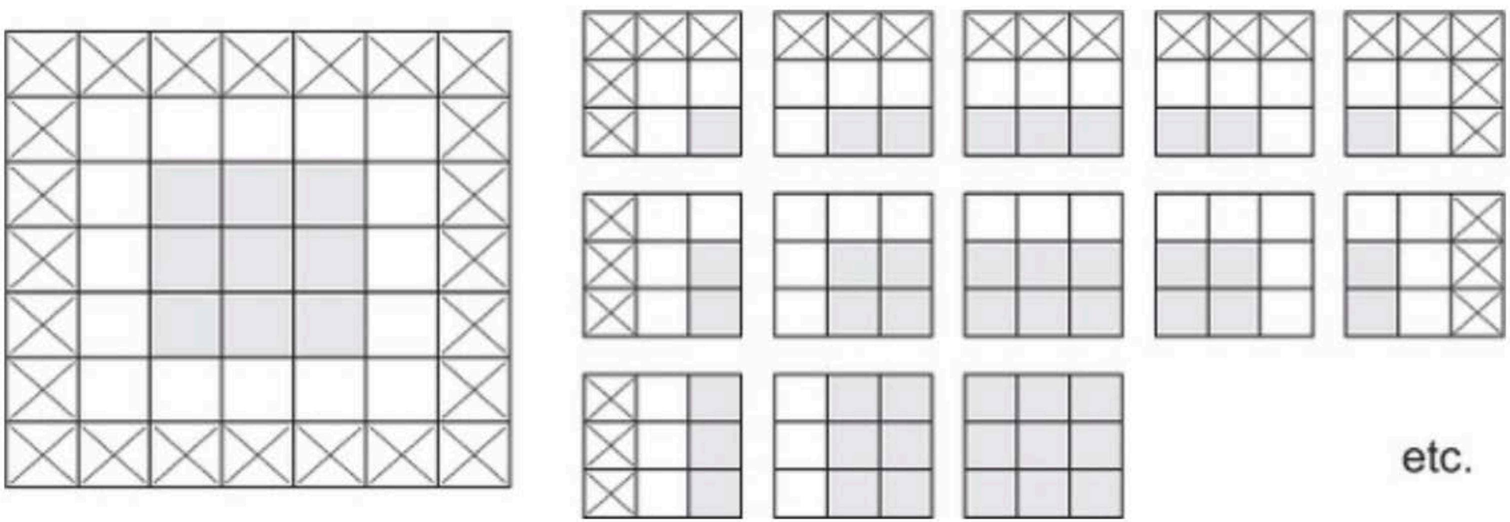
Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Understanding border effects: Padding

In Keras → Conv2D layers are padding configurable through the padding argument, that takes two values:

‘valid’ → Means no padding (by default)

‘same’ → Means “pad in such way as to have the same width&height as the input



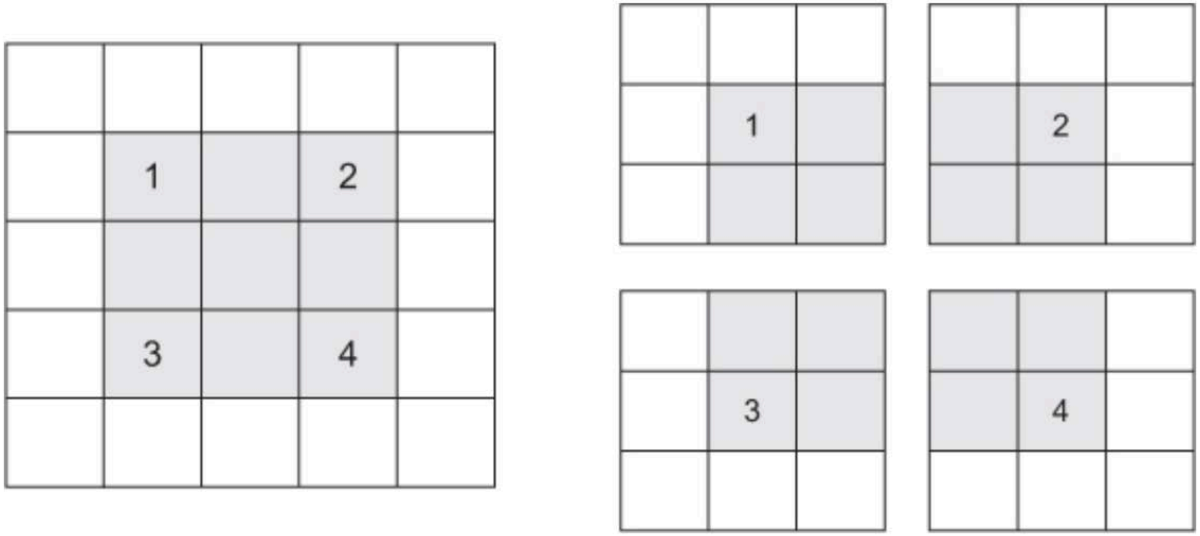
Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Understanding border effects: Strides

A factor that can influence the output size.

The description of convolution so far has assumed that the filters move in a contiguous way.

But the distance between two succesive windows is a Conv2D parameter → *strides*



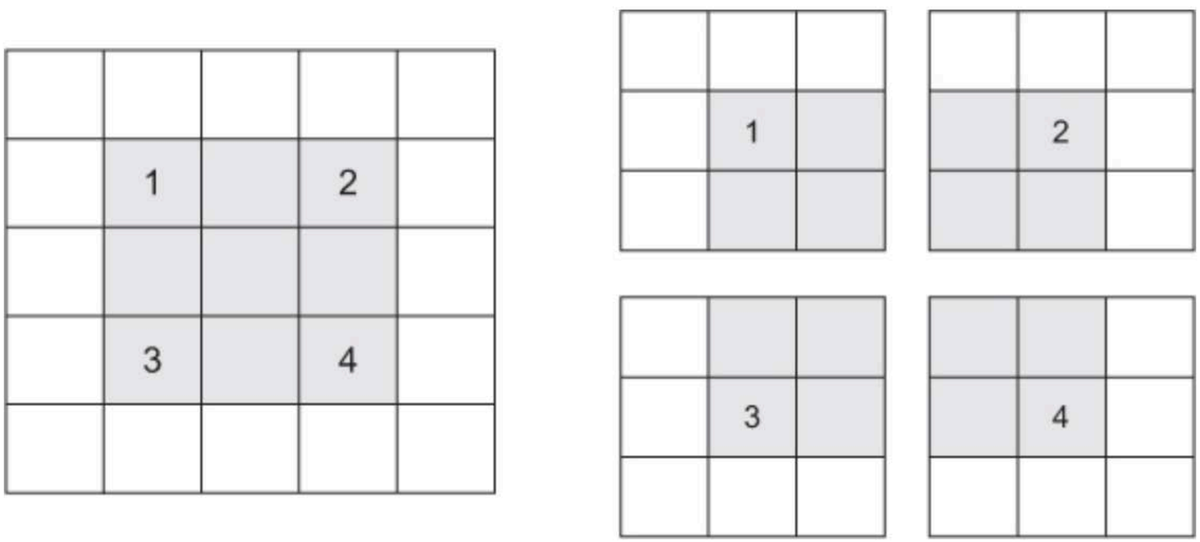
Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Understanding border effects: Strides

It is possible to have convolutions with a stride higher than 1.

P.e. Using stride 2 means the width&height of the feature map are downsampled by 2.

This is rarely used in practice → instead of strides, we tend to use max-pooling operation.

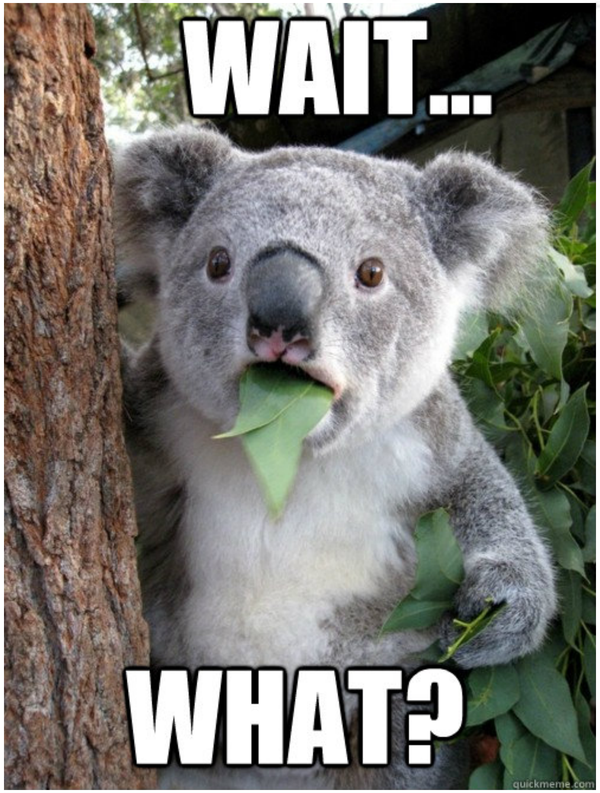


Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Max-pooling operation

```
cnn_model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 3, 3, 64)          36928
_________________________________________________________________
flatten_1 (Flatten)          (None, 576)               0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                36928
_________________________________________________________________
dense_4 (Dense)              (None, 10)                650
=================================================================
Total params: 93,322
Trainable params: 93,322
Non-trainable params: 0
_________________________________________________________________
```



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Max-pooling operation

MNIST example:

…

model = models.sequential()

model.add(layers.Conv2D(32, (3,3), activation='relu', input_shape=(28,28,1)))

model.add(layers.MaxPooling2D((2, 2)))

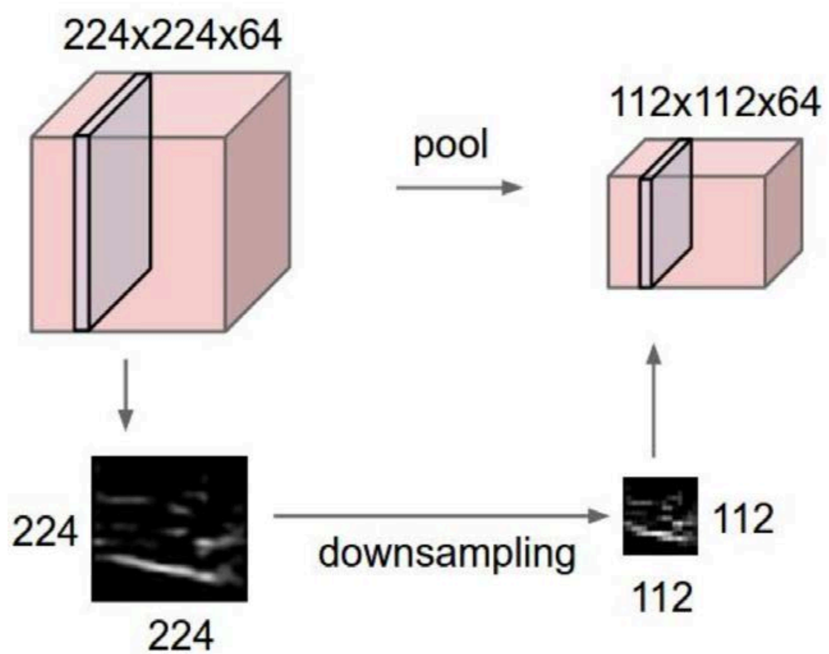
…

Max pooling → Aggressively downsample the feature maps (like strided convolutions)

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

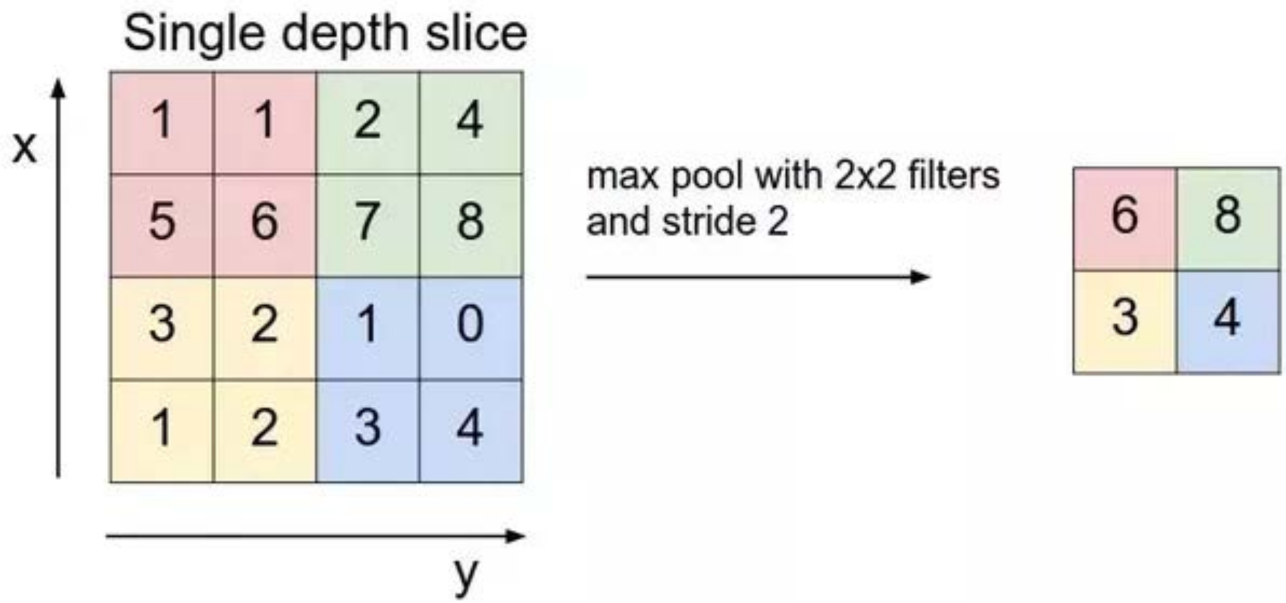
Max-pooling operation

Makes the representation smaller and more manageable
Operates over each feature map independently:



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Max-pooling operation



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Max-pooling operation

Why downsample feature maps?

Why don't we remove the max-pooling layers and keep large feature maps all the way up?

```
model_no_max_pool = models.Sequential()
model_no_maxpool.add(layers.Conv2D(32, (3, 3), activation='relu',input_shape=(28, 28, 1)))
model_no_max_pool.add(layers.Conv2D(64, (3, 3), activation='relu'))
model_no_max_pool.add(layers.Conv2D(64, (3, 3), activation='relu'))
```

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Max-pooling operation

Why downsample feature maps?

Why don't we remove the max-pooling layers and keep large feature maps all the way up?

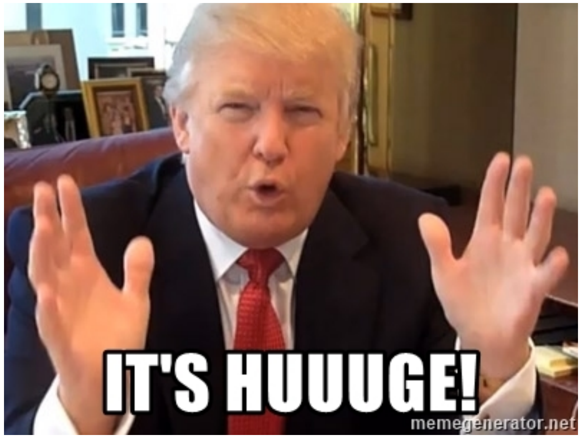
```
>>> model_no_max_pool.summary()

Layer (type)                     Output Shape          Param #
================================================================
conv2d_4 (Conv2D)                (None, 26, 26, 32)    320
________________________________________________________________
conv2d_5 (Conv2D)                (None, 24, 24, 64)    18496
________________________________________________________________
conv2d_6 (Conv2D)                (None, 22, 22, 64)    36928
================================================================
Total params: 55,744
Trainable params: 55,744
Non-trainable params: 0
```

Max-pooling operation

What is wrong with that setup?

The final feature map is 22 x 22 x 64 = 30,976 total coefficients per sample!!
That is huge!

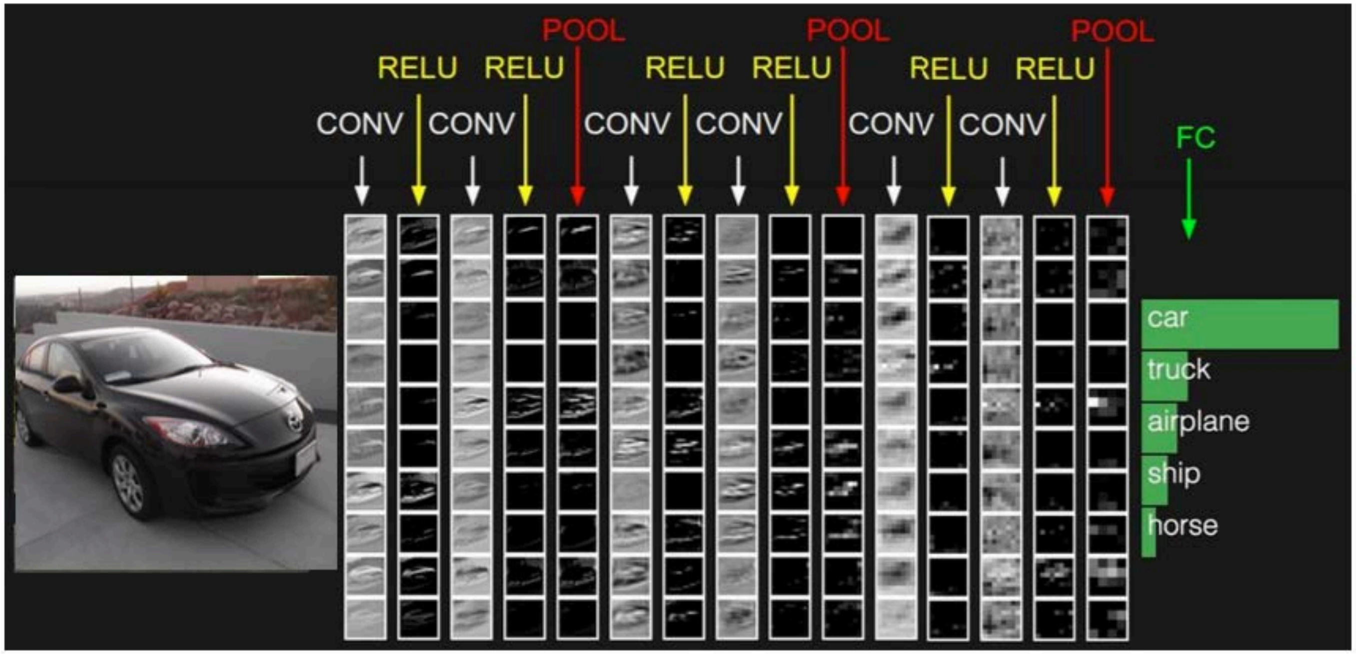
If we want to flatten it to stick a Dense layer of 512 units,



we would have **15.800.000** parameters

Fully Connected Layer

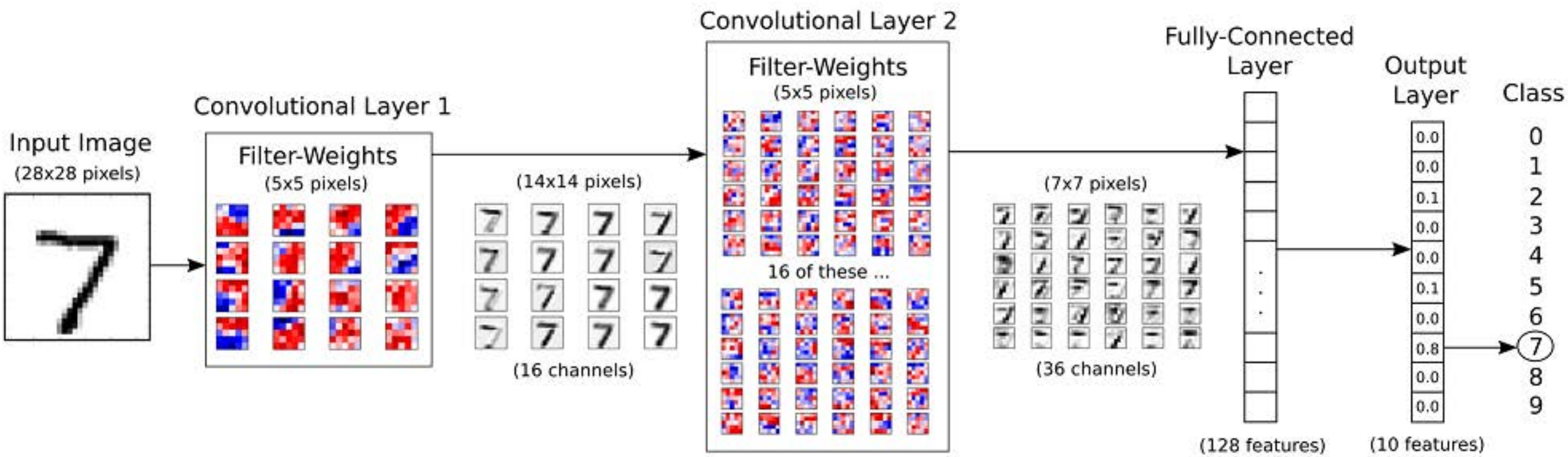
Contains neurons that connect to the entire input volume, as an ordinary NN.



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

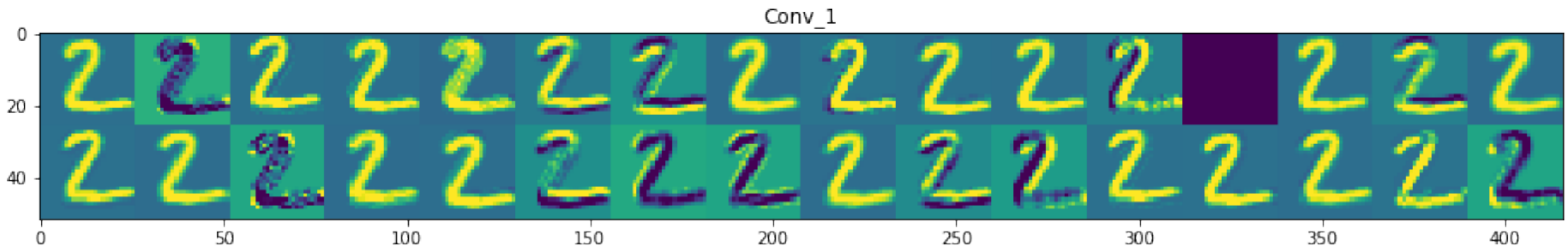
What is happening?

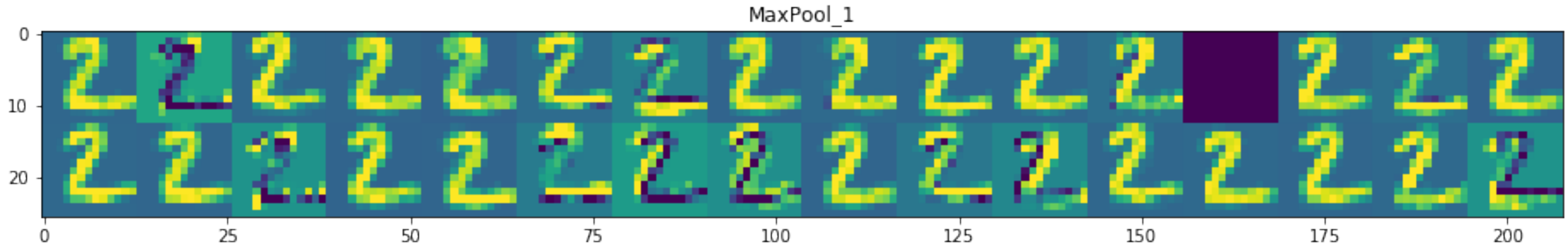
Convolutional layers:



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

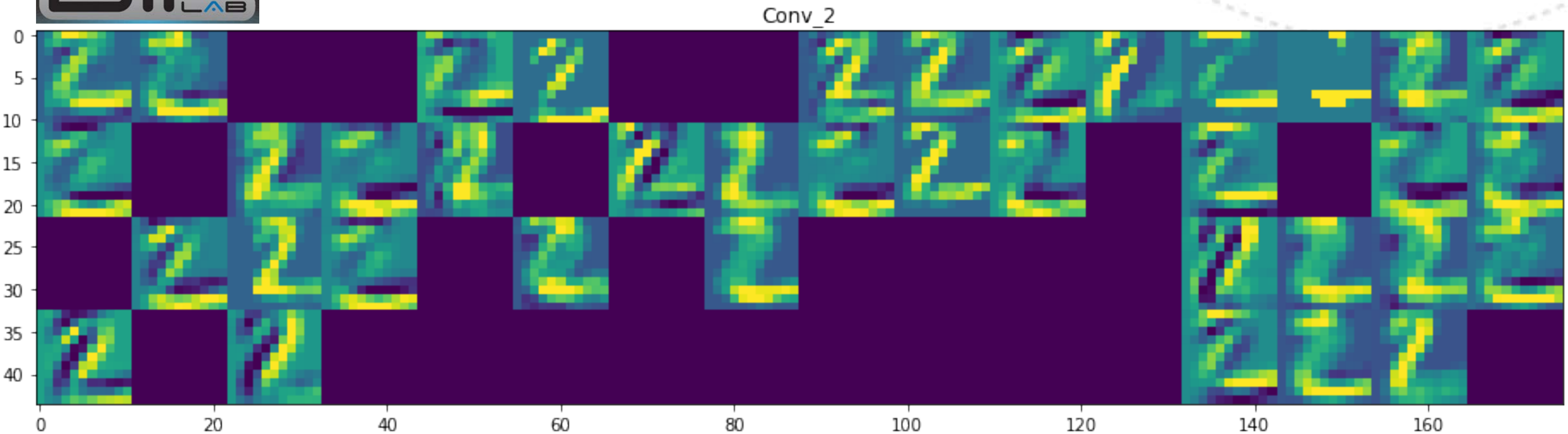
What is happening?
Feature Maps





Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

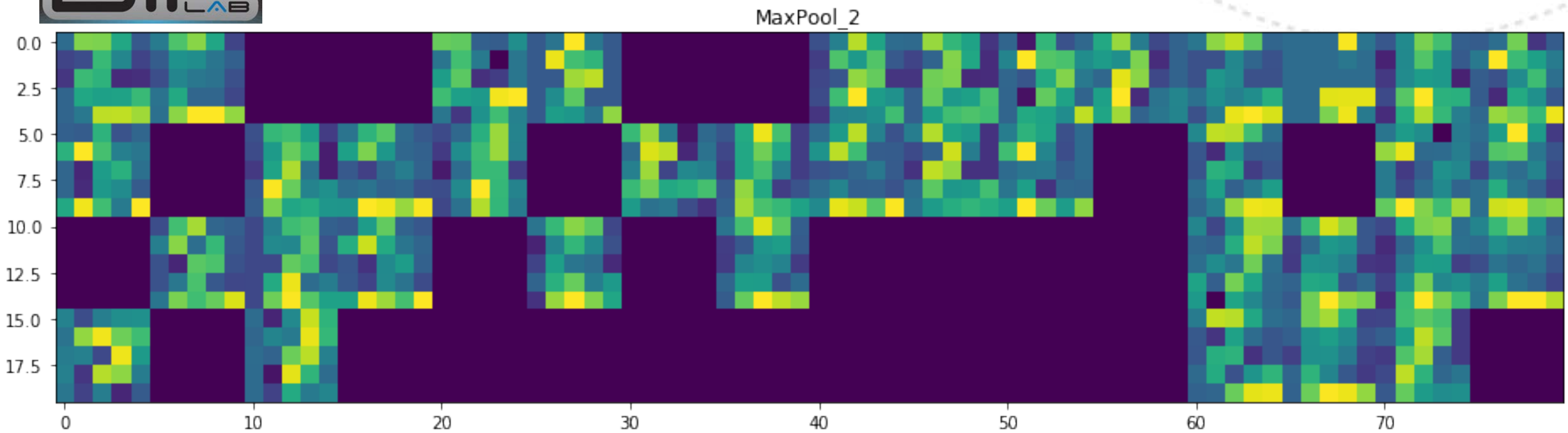
What is happening?
Feature Maps



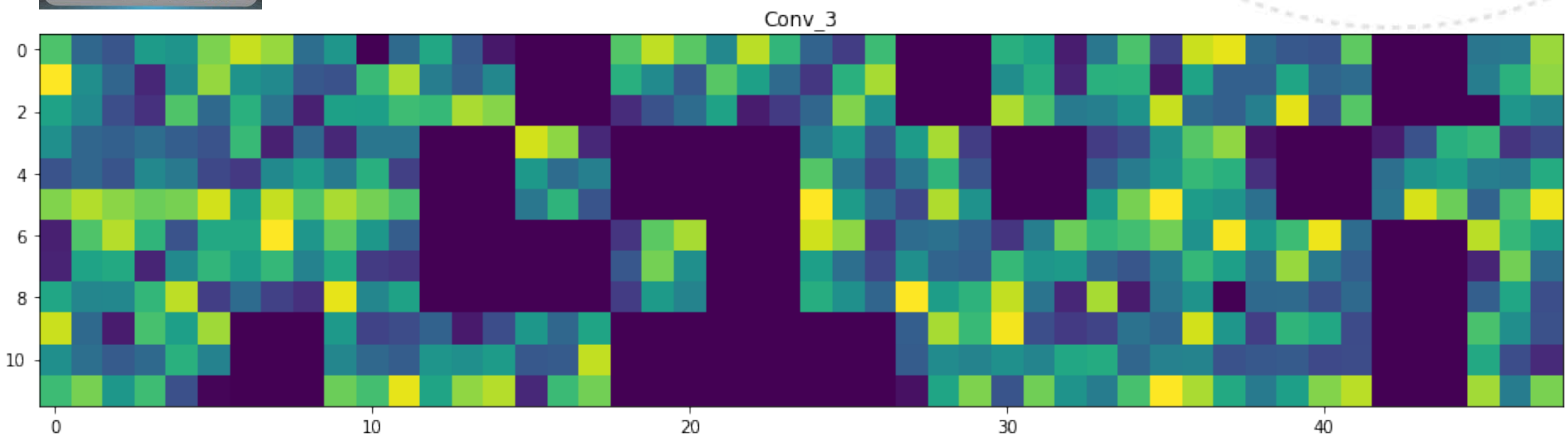
Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

What is happening?
Feature Maps



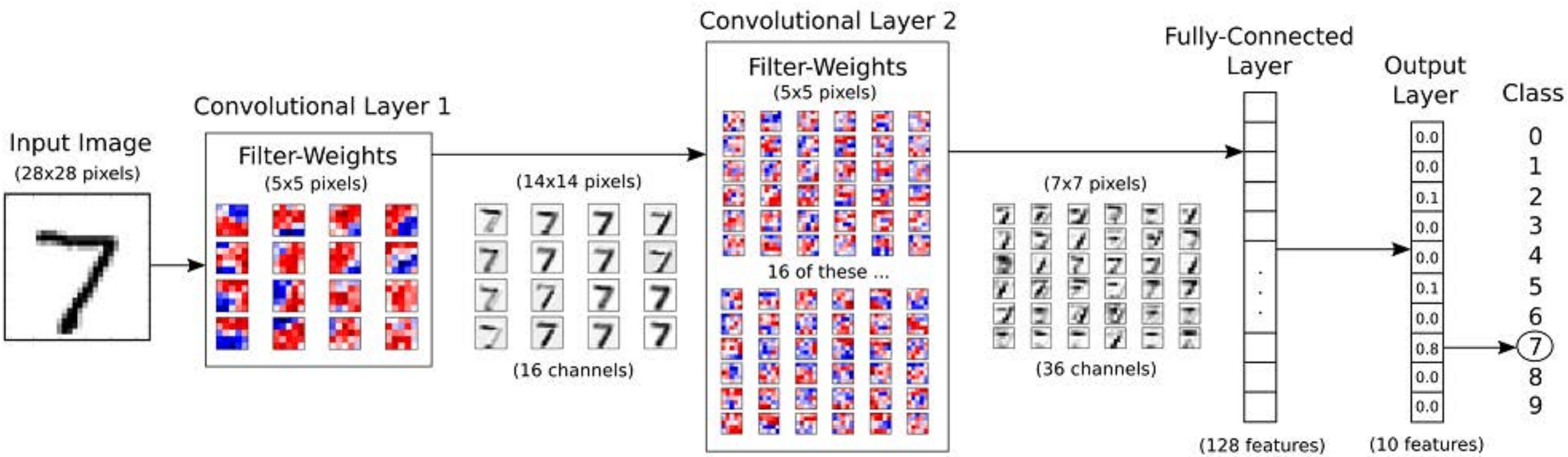
What is happening?
Feature Maps



These feature maps are flattened and stick into the Fully Connected layer!!

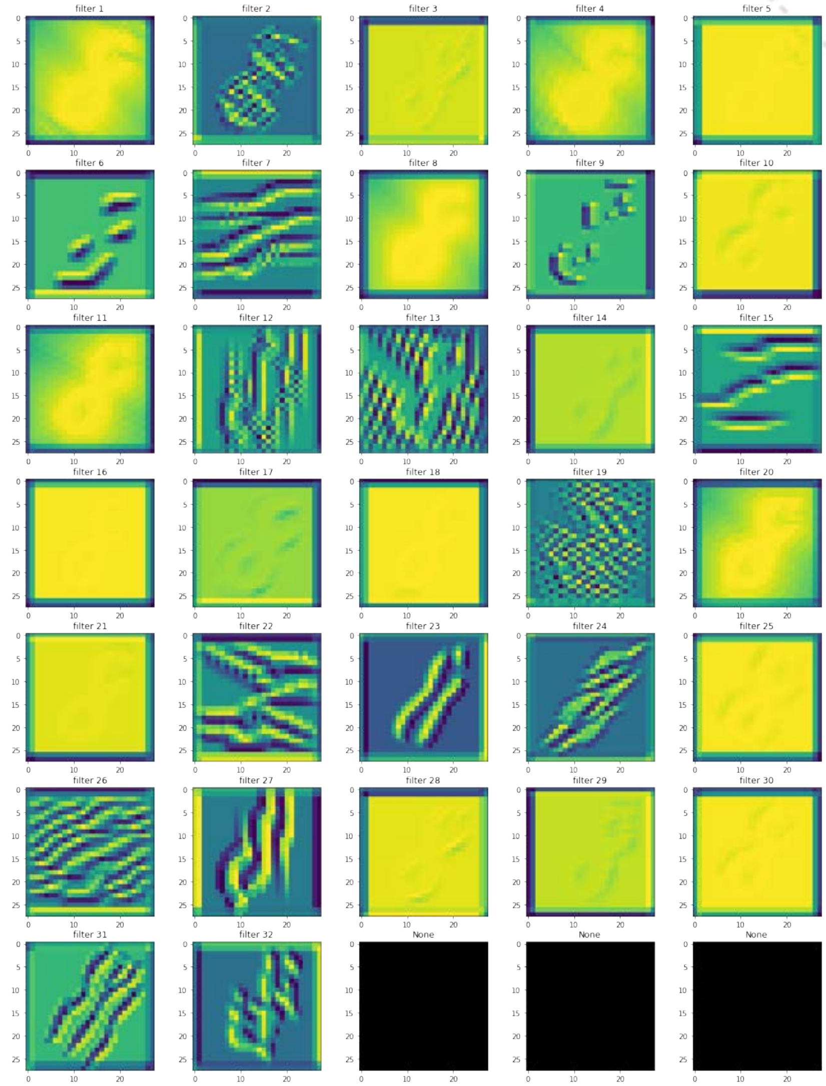
What is happening?

Convolutional layers:



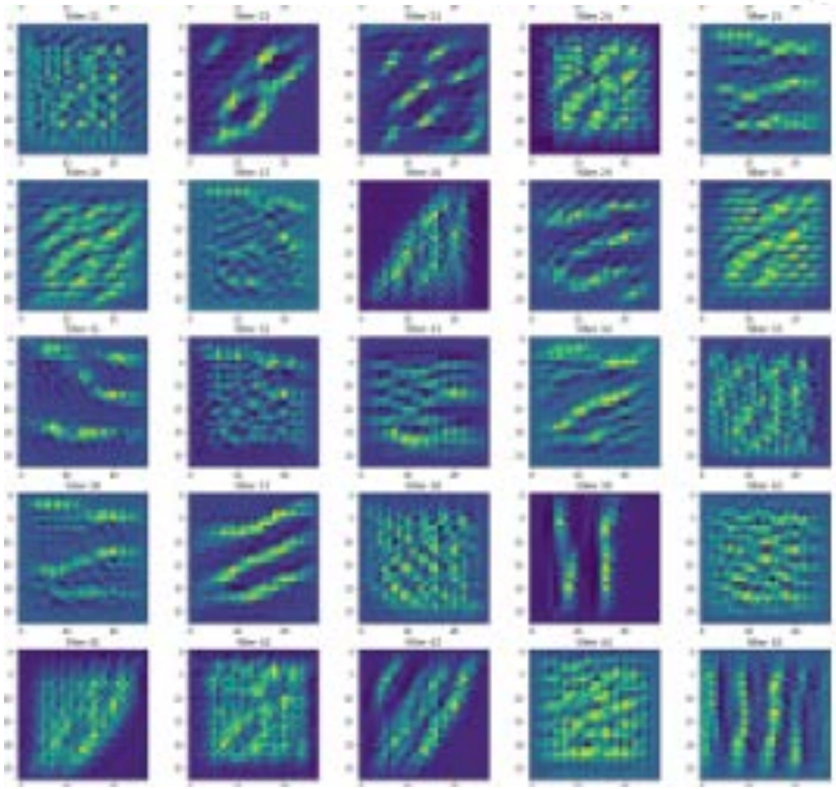
Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

What is happening?
Filters Conv1



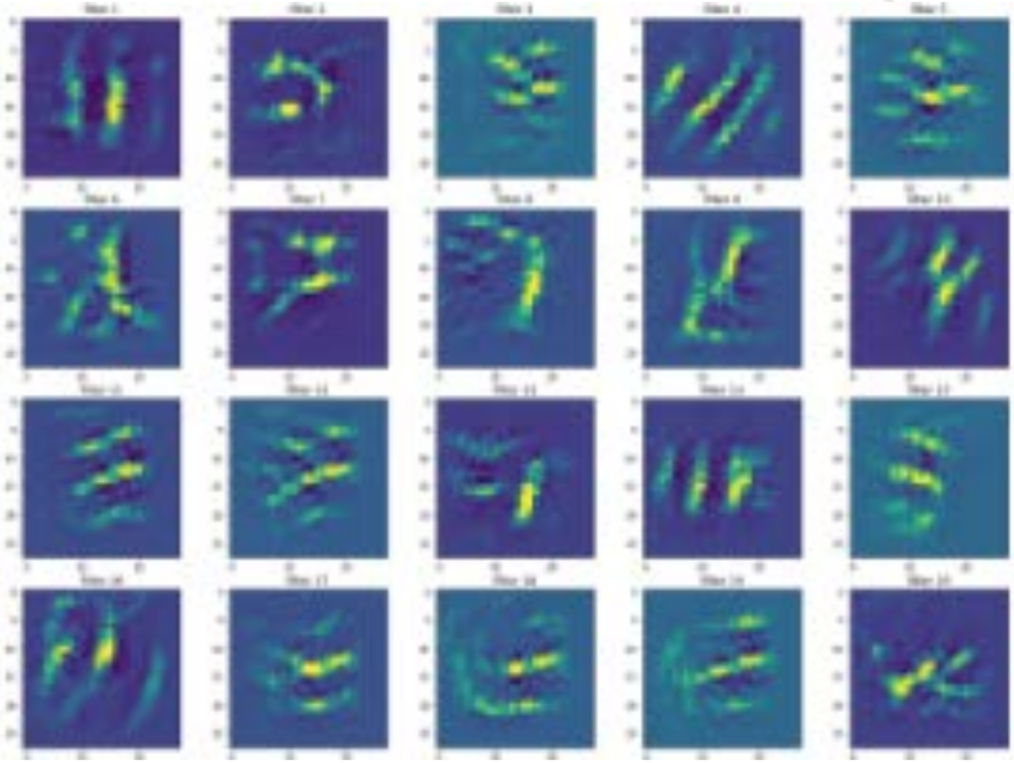
Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

What is happening?
Filters Conv2



Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

What is happening?
Filters Conv3



INSTITUTO UNIVERSITARIO
SIANI
INGENIERIA COMPUTACIONAL

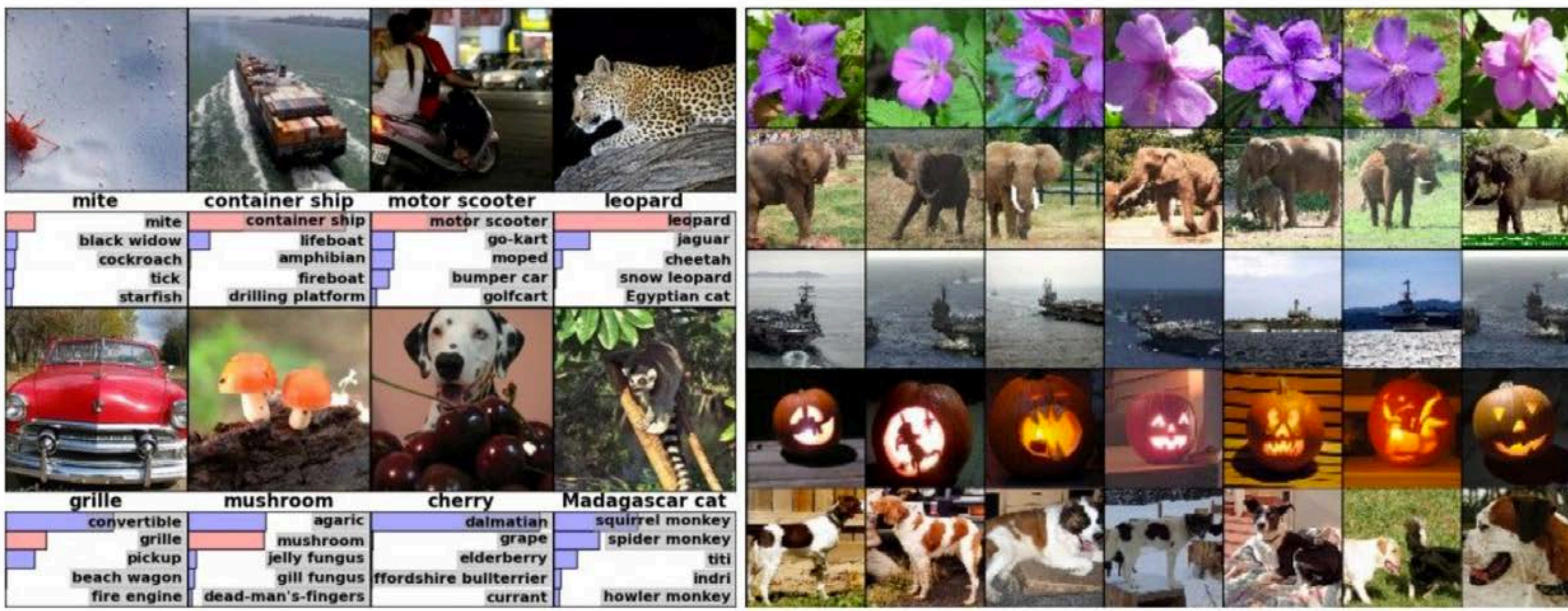
Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

Where to use CNN?

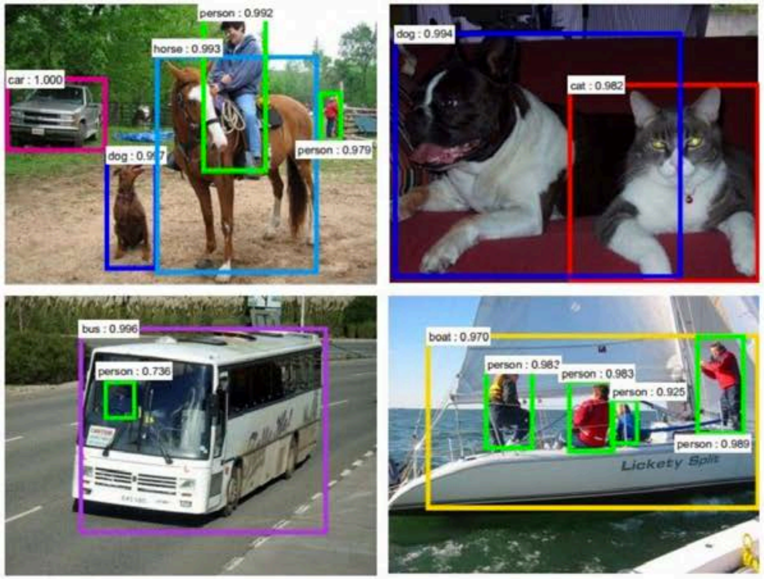
Classification

Retrieval



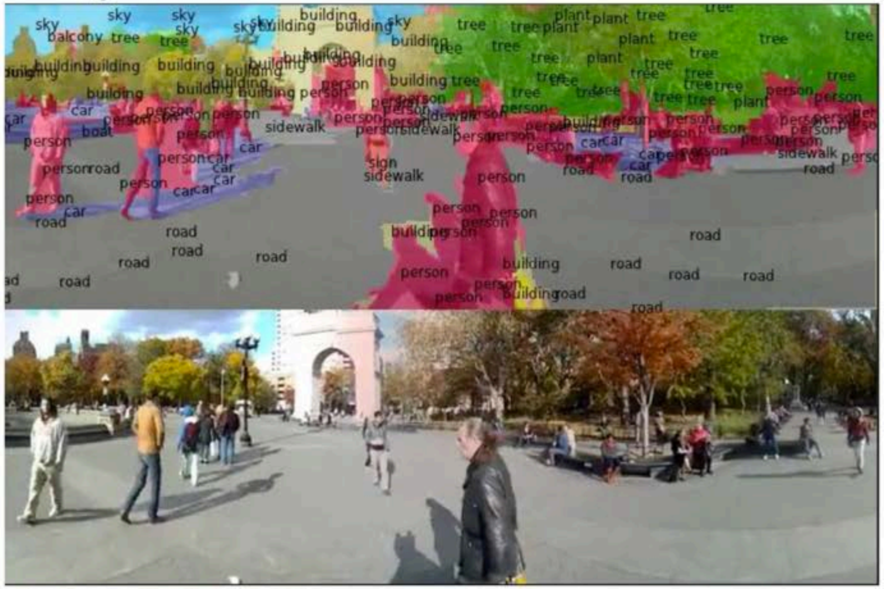
Where to use CNN?

Detection



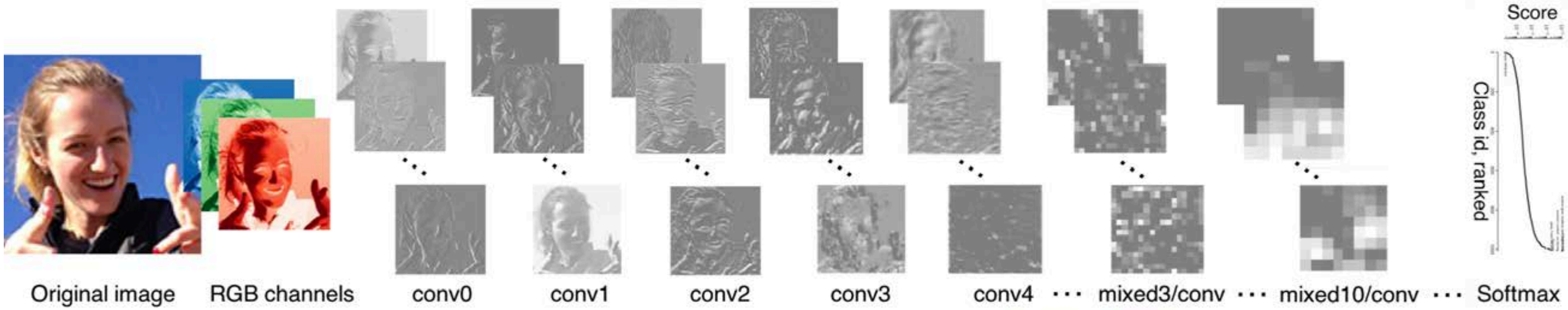
[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentation



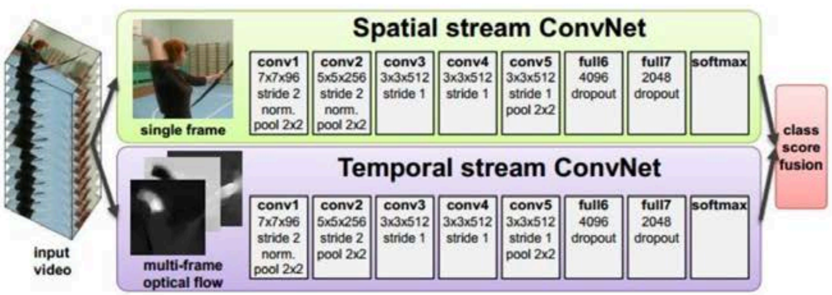
[Farabet et al., 2012]

Where to use CNN?



[Taigman et al. 2014]

Activations of inception-v3 architecture [Szegedy et al. 2015] to image of Emma McIntosh, used with permission. Figure and architecture not from Taigman et al. 2014.



[Simonyan et al. 2014]

Figures copyright Simonyan et al., 2014. Reproduced with permission.

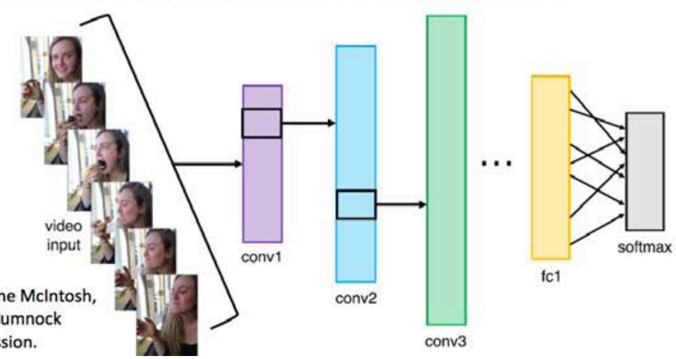


Illustration by Lane McIntosh, photos of Katie Cumnock used with permission.

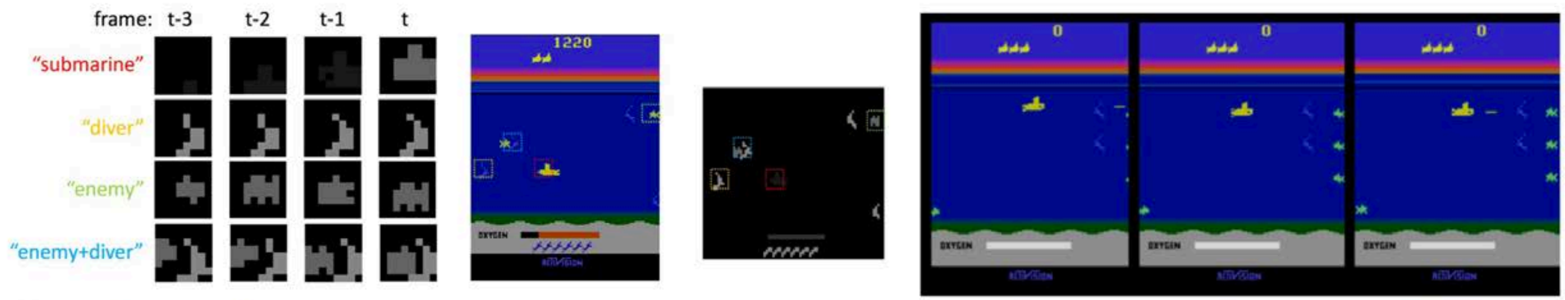
Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Where to use CNN?

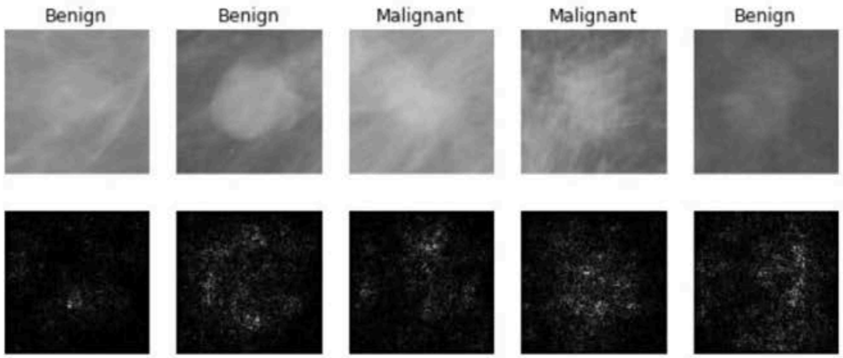
Images are examples of pose estimation, not actually from Toshev & Szegedy 2014. Copyright Lane McIntosh.

[Toshev, Szegedy 2014]



[Guo et al. 2014]

Where to use CNN?



[Levy et al. 2016]

Figure copyright Levy et al. 2016. Reproduced with permission.

[Dieleman et al. 2014]

From left to right: public domain by NASA, usage permitted by ESA/Hubble, public domain by NASA, and public domain.

[Sermanet et al. 2011]
[Ciresan et al.]

Photos by Lane McIntosh. Copyright CS231n 2017.

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Applications

https://www.youtube.com/watch?v=kkha3sPoU70

https://www.youtube.com/watch?v=U1vb-NI9sKE

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es

Credits

The 'Where to use CNN?' section is partially based on the Lecture 5: Convolutional Neural Networks. By Fei-Fei, Johnson and Yeung. April 18, 2017

Edificio Central del Parque Tecnológico
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
e-mail: info@siani.es · www.siani.es