BSC~Microsoft Research Centre

Advanced Deep Learning for Computer Vision

Tassadaq Hussain

Professor Namal University

Director Centre for AI and Big Data

Collaborations:

Barcelona Supercomputing Center Barcelona, Spain

European Network on High Performance and Embedded Architecture and Compilation

Pakistan Supercomputing Center

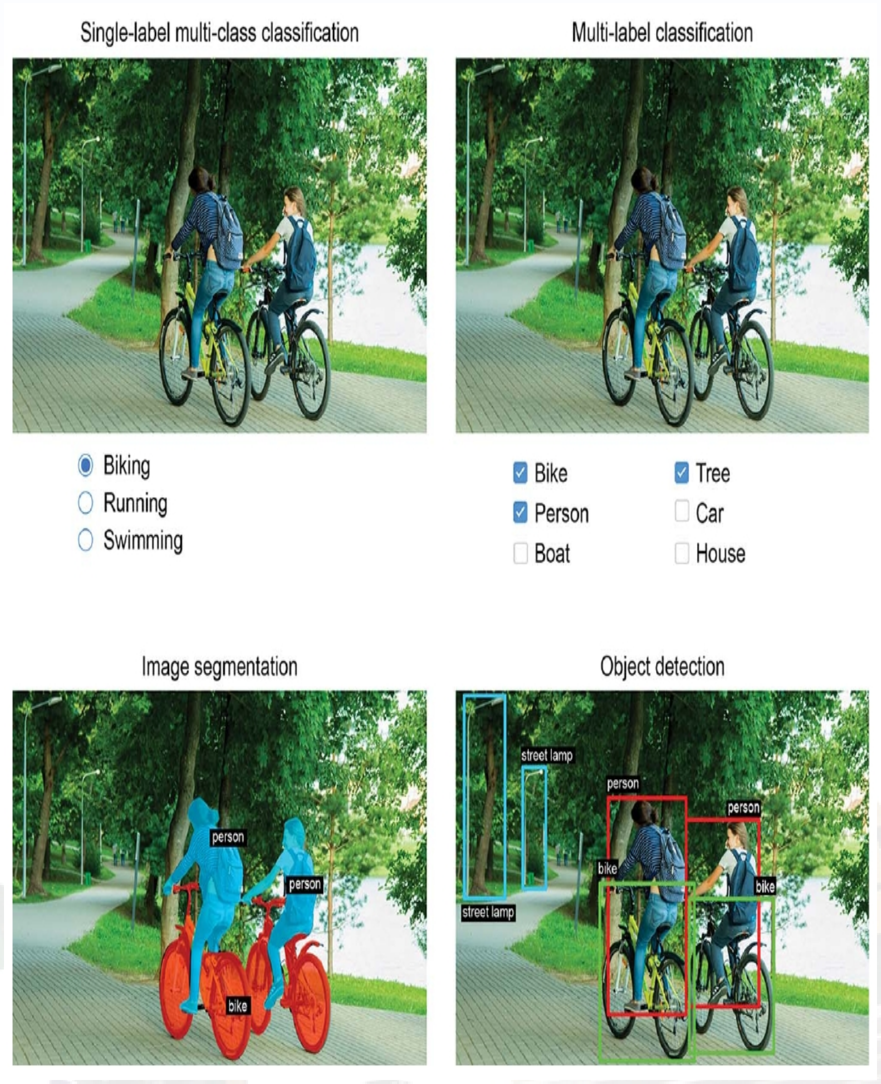
UCERD
Gathering Intellectuals
www.ucerd.com

```
from tensorflow import keras
from tensorflow.keras import layers
inputs = keras.Input(shape=(28, 28, 1))
x = layers.Conv2D(filters=32, kernel_size=3, activation="relu")(inputs)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=64, kernel_size=3, activation="relu")(x)
x = layers.MaxPooling2D(pool_size=2)(x)
x = layers.Conv2D(filters=128, kernel_size=3, activation="relu")(x)
x = layers.Flatten()(x)
outputs = layers.Dense(10, activation="softmax")(x)
model = keras.Model(inputs=inputs, outputs=outputs)
```

UCERD

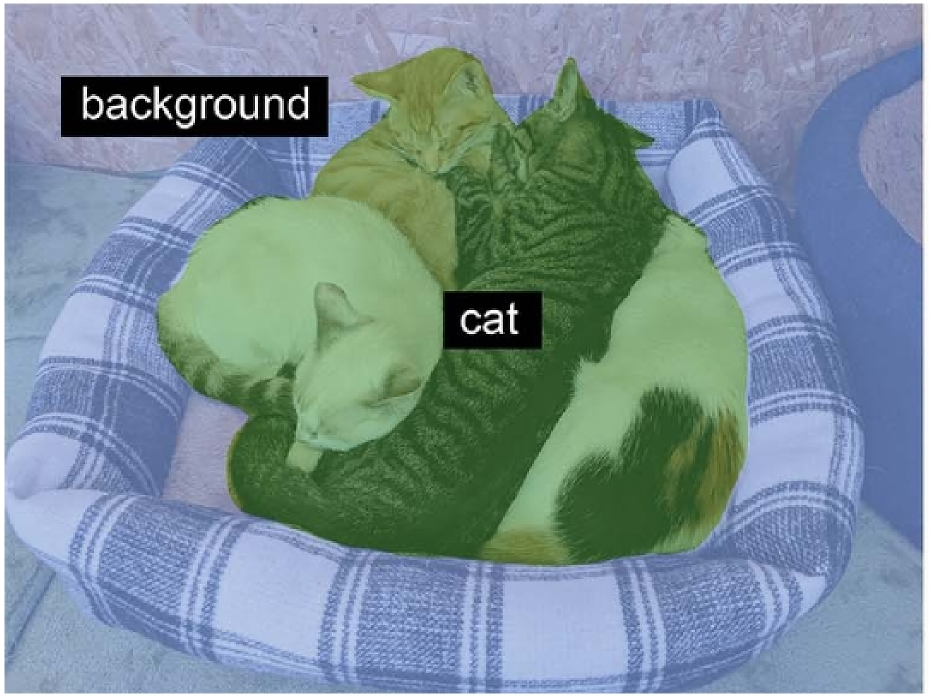
Three essential computer vision tasks

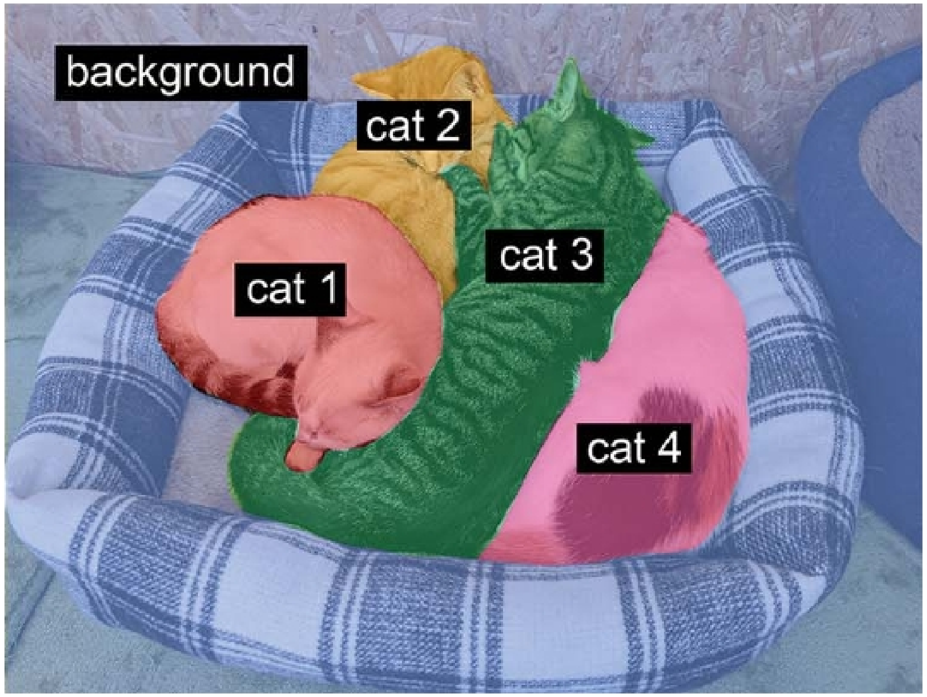
- Image classification: It may be either single-label classification (an image can only be in one category, excluding the others), or multi-label classification.
- Image segmentation—Where the goal is to "segment" or "partition" an image into different areas, with each area usually representing a category.
- Object detection—Where the goal is to draw rectangles (called bounding boxes) around objects of interest in an image, and associate each rectangle with a class.



UCERD

Image Segmentation





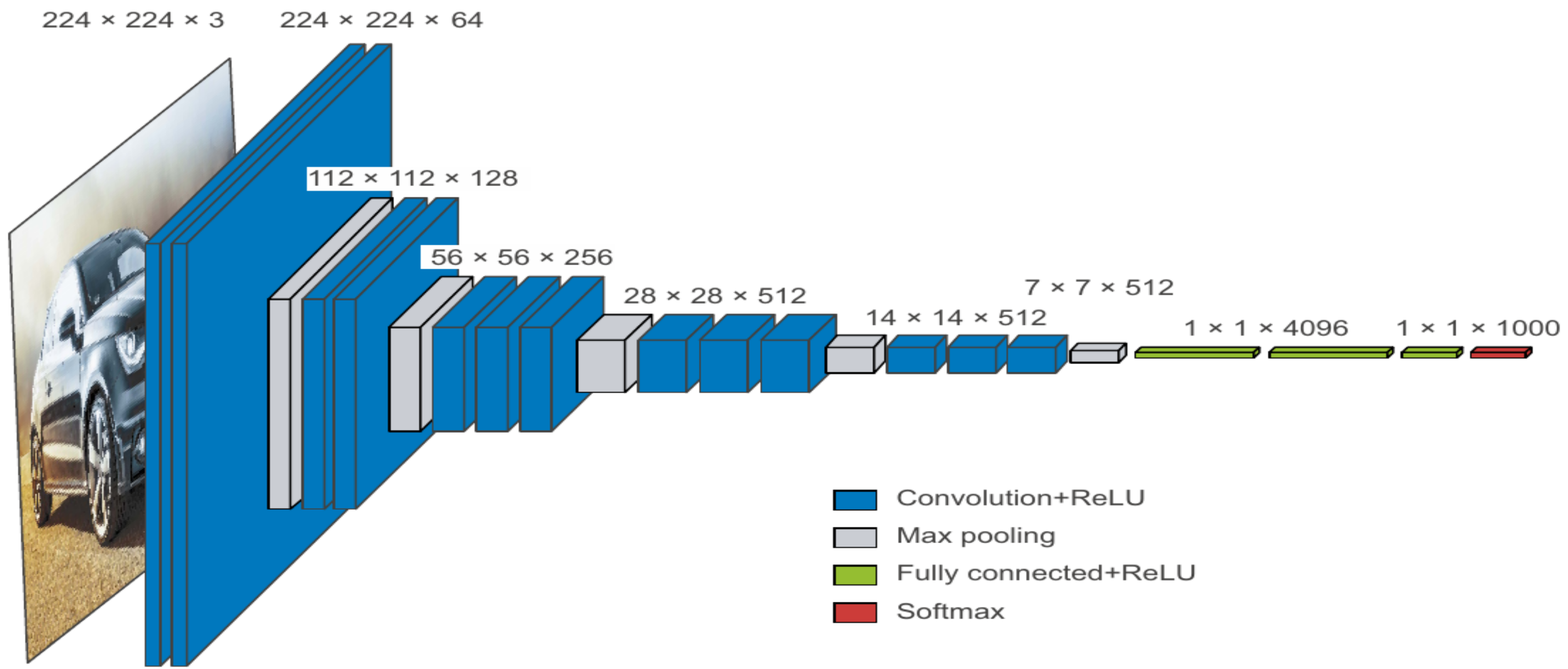
Data and Annotations

```
import os
input_dir = "images/"
target_dir = "annotations/trimaps/"
input_img_paths = sorted(
[os.path.join(input_dir, fname)
for fname in os.listdir(input_dir)
if fname.endswith(".jpg")])
target_paths = sorted(
[os.path.join(target_dir, fname)
for fname in os.listdir(target_dir)
if fname.endswith(".png") and not fname.startswith(".")])
```

UCERD

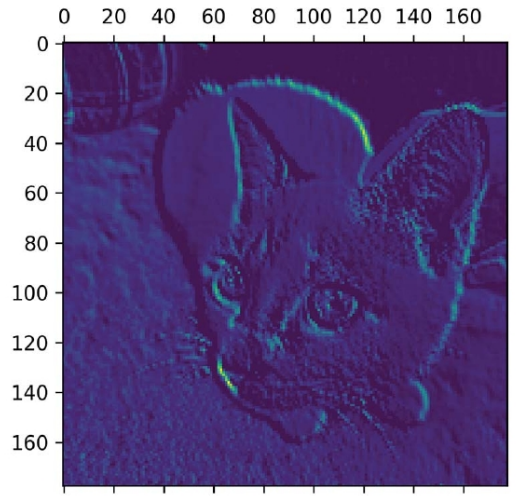
- Path of Images Directory
- Define Classes

- Your model should be organized into repeated blocks of layers, usually made of multiple convolution layers and a max pooling layer.
- The number of filters in your layers should increase as the size of the spatial feature maps decreases.
- Deep and narrow is better than broad and shallow.
- Introducing residual connections around blocks of layers helps you train deeper networks.
- It can be beneficial to introduce batch normalization layers after your convolution layers.
- It can be beneficial to replace Conv2D layers with SeparableConv2D layers, which are more parameter-efficient.



Visualizing

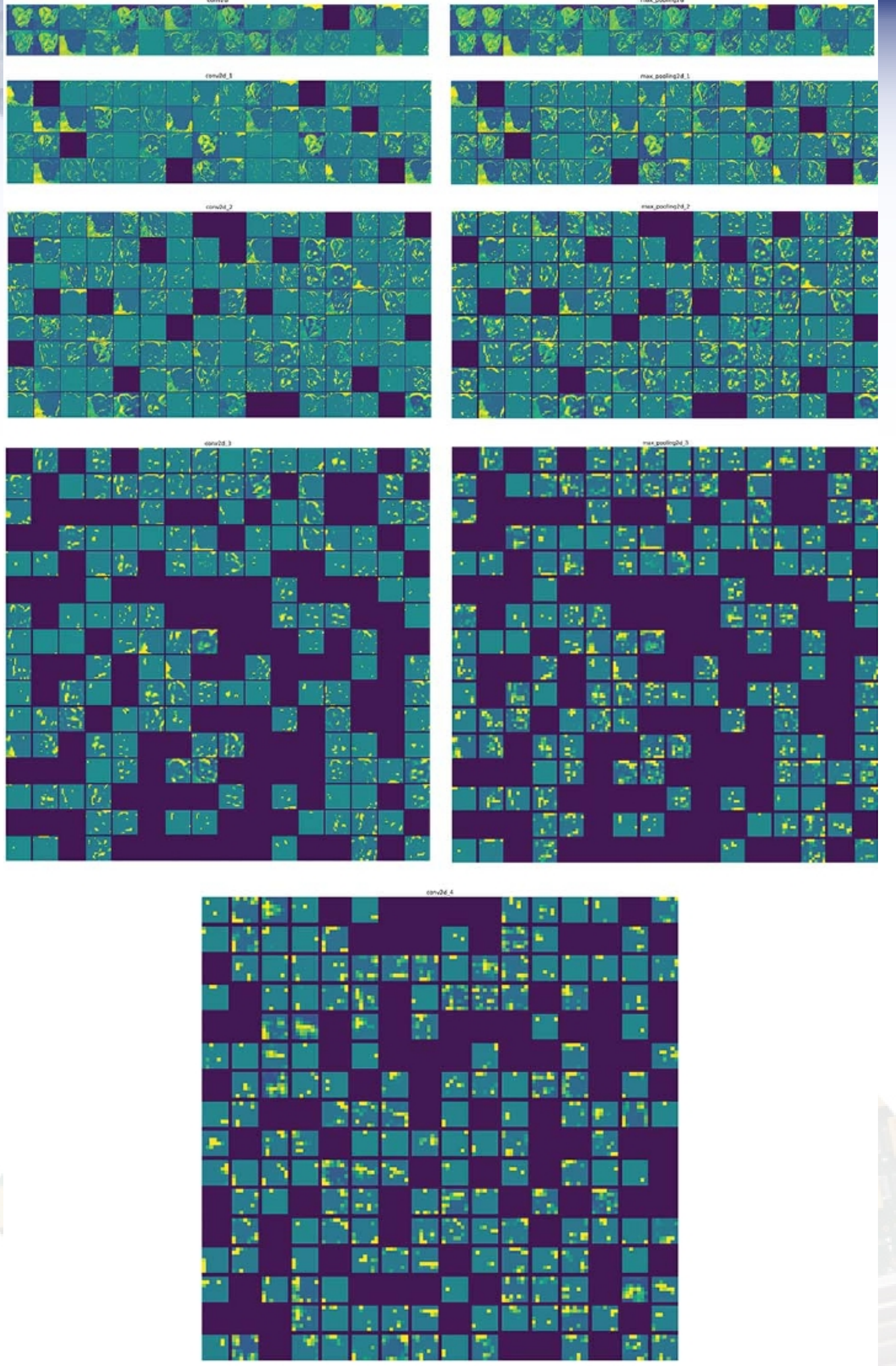
- Visualizing intermediate convnet outputs (intermediate activations)—Useful for understanding how successive convnet layers transform their input, and for getting a first idea of the meaning of individual convnet filters
- Visualizing convnet filters—Useful for understanding precisely what visual pattern or concept each filter in a convnet is receptive to
- Visualizing heatmaps of class activation in an image—Useful for understanding which parts of an image were identified as belonging to a given class, thus allowing you to localize objects in images



Every channel of every layer activation on the test cat picture:
The first layer acts as a collection of various edge detectors. At that stage, the activations retain almost all of the information present in the initial picture.

As you go deeper, the activations become increasingly abstract and less visually interpretable. They begin to encode higher-level concepts such as “cat ear” and “cat eye.” Deeper presentations carry increasingly less information about the visual contents of the image, and increasingly more information related to the class of the image.

The sparsity of the activations increases with the depth of the layer: in the first layer, almost all filters are activated by the input image, but in the following layers, more and more filters are blank. This means the pattern encoded by the filter isn’t found in the input image.


conv2d_1
max_pooling2d_1
conv2d_2
max_pooling2d_2
conv2d_3
max_pooling2d_3
conv2d_4