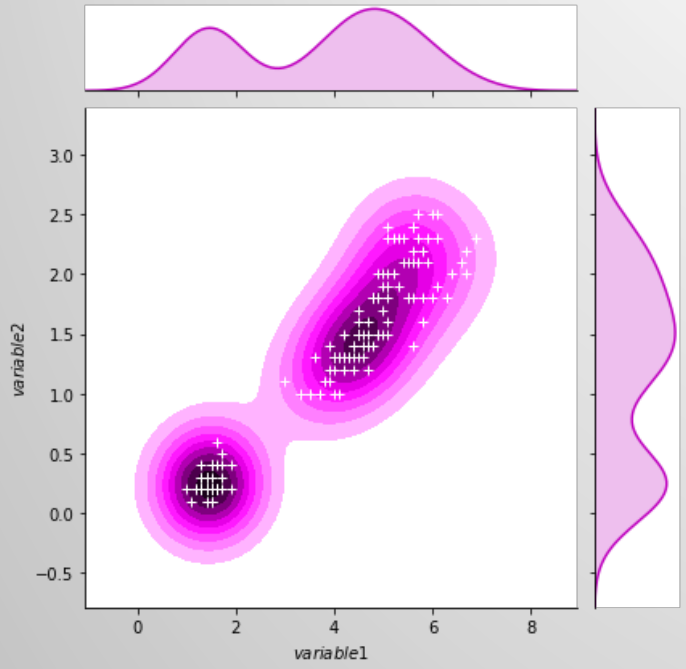
Hands on Introduction to Deep Learning

Raúl Benítez, PhD
raul.Benitez@upc.edu

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Barcelona East School of Engineering

Mathematics

Computer Science



$$
\begin{aligned}
p(\mathcal{D}|\theta) &= p(x_1, x_2, \ldots, x_n|\mu, \sigma^2) \\
&= \prod_{i=1}^{n} p(x_i|\theta) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right) \\
&= \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \exp\left(-\frac{\sum_{i=1}^{n}(x_i-\mu)^2 + n(\frac{1}{n}\sum_{i=1}^{n}x_i-\mu)^2}{2\sigma^2}\right)
\end{aligned}
$$

```python
import numpy as np
from sklearn import decomposition
from sklearn import datasets

iris = datasets.load_iris()
X = iris.data
y = iris.target

pca = decomposition.PCA(n_components=2)
pca.fit(X)
Xproj = pca.transform(X)
```

Text
Images
Multivariate numerical data
Genetics
Audio, video
HETEROGENEOUS

Statistics
Geometry
Optimization
Stochastic processes

Algorithms
Computational complexity
Information theory
Network analysis

An affair between computer science and biology

Back-propagation Applied to Handwritten Zip Code Recognition (1989)

Convolutional Networks For Images, Speech, And Time Series (1995)

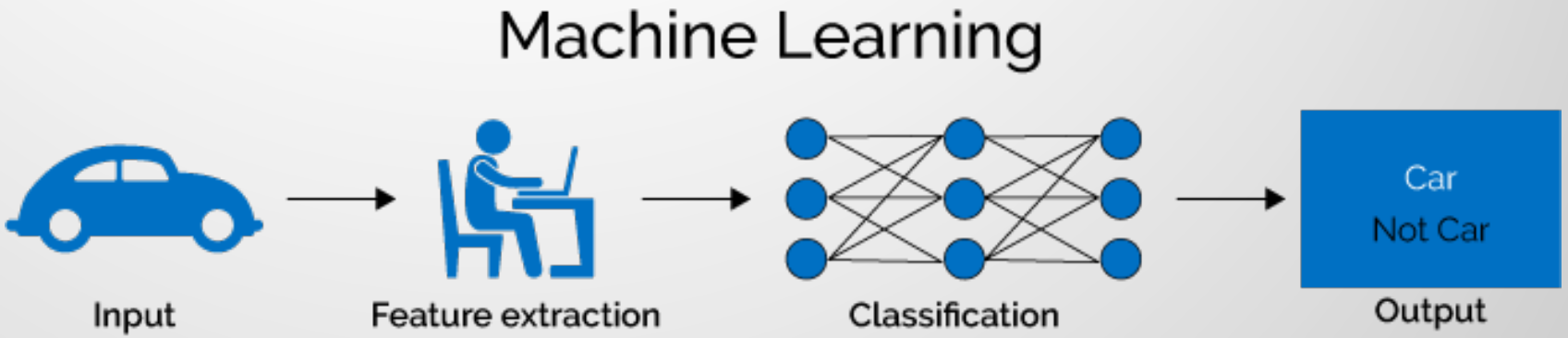
Gradient-based Learning Applied To Document Recognition (1998)

http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf
http://yann.lecun.com/exdb/publis/pdf/lecun-bengio-95a.pdf
http://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf

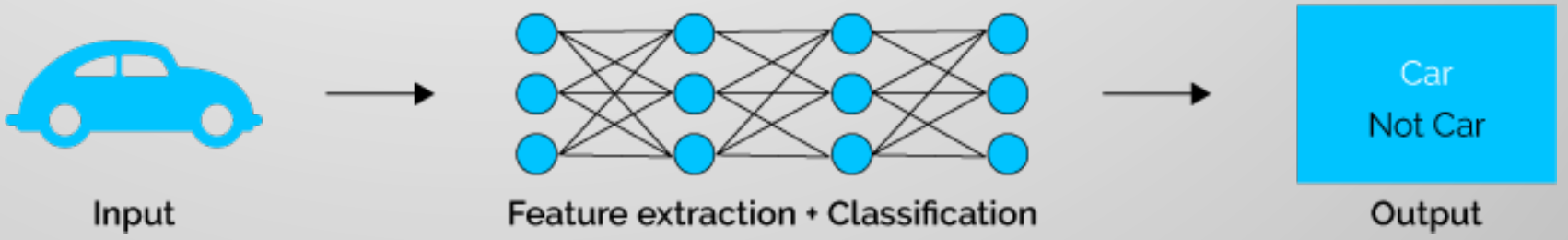
Cajal & Golgi Nobel Prize 1906
Hebbian learning 1949
Hodkin & Huxley 1952 Nobel Prize 1963
Hubel & Wiesel 1959 Visual Cortex Nobel Prize 1981

CHANGE IN PARADIGM

Tailored features
(shape, size, texture...)
meaningful



Deep features
useful

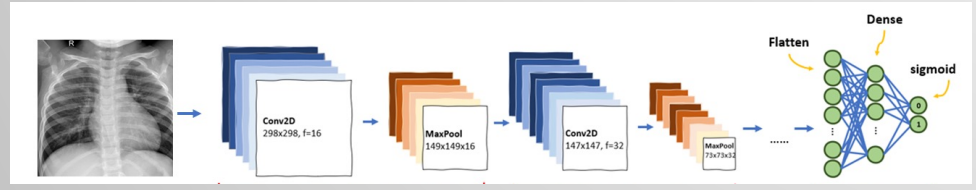


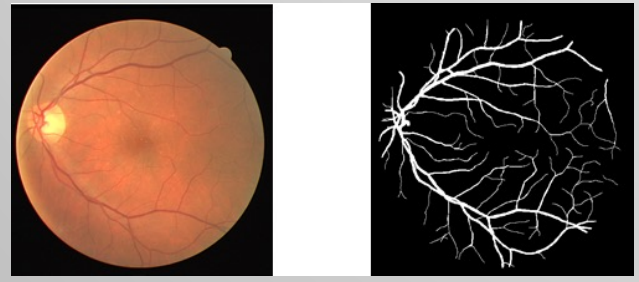
Deep Learning

Convolutional Neural Networks (CNNs)

Image Classification

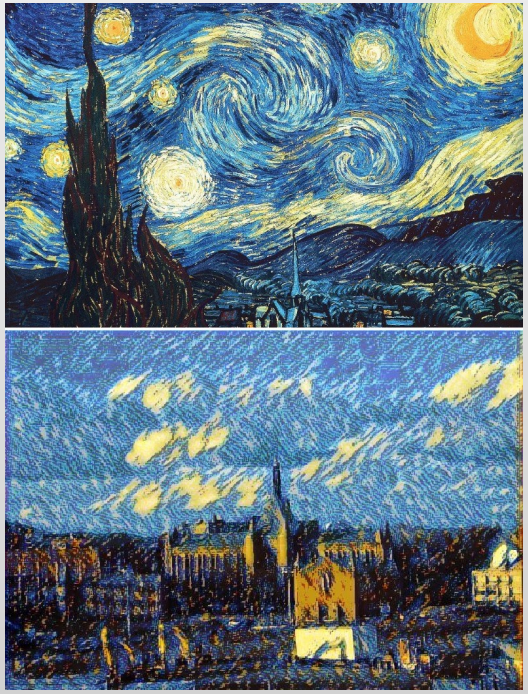
Image Segmentation





Generative Adversarial Networks (GANs)

Image generation

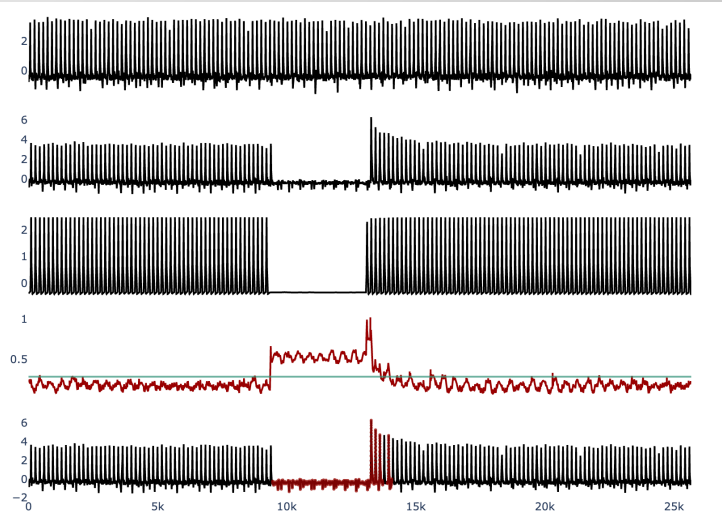


Recurrent Neural Networks (RNNs)

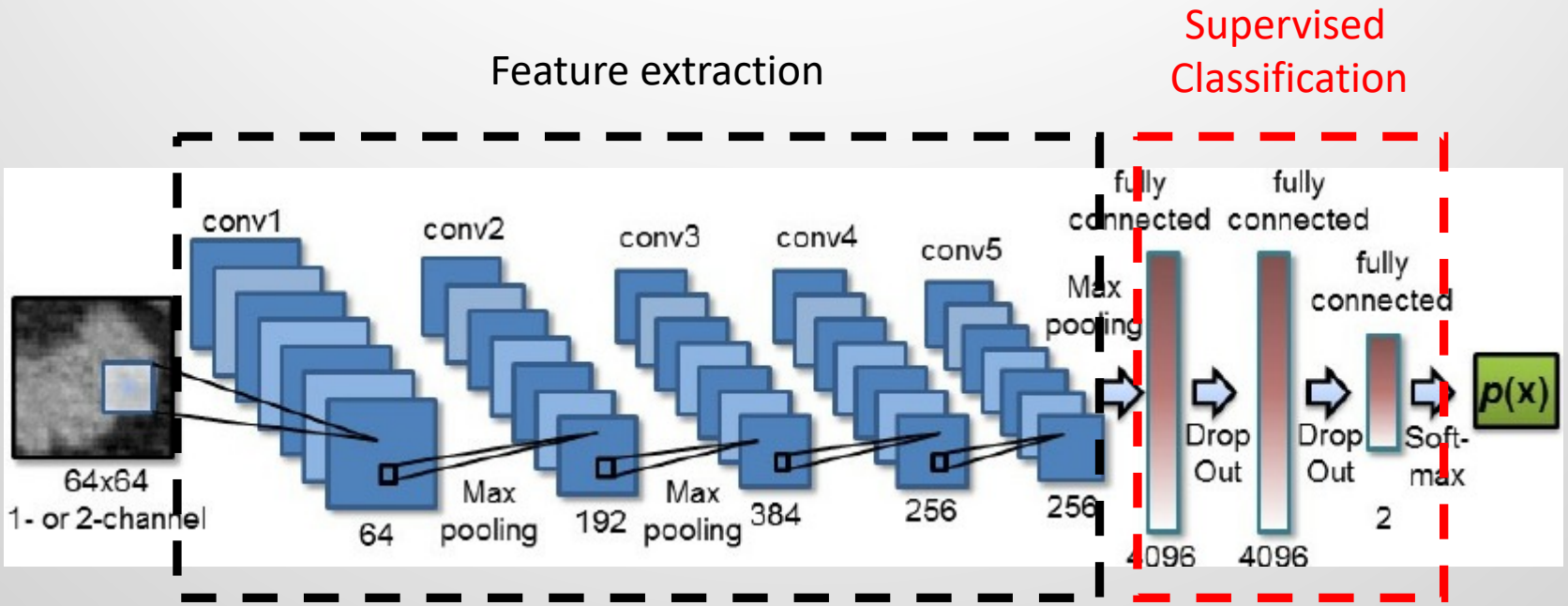
Time series analysis

Anomaly detection

Video analysis



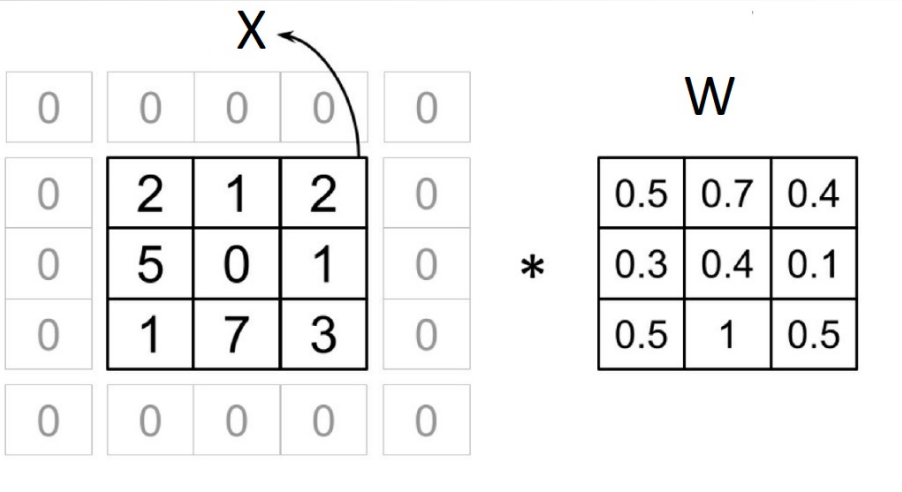
Convolutional Neural Networks – Image Classification



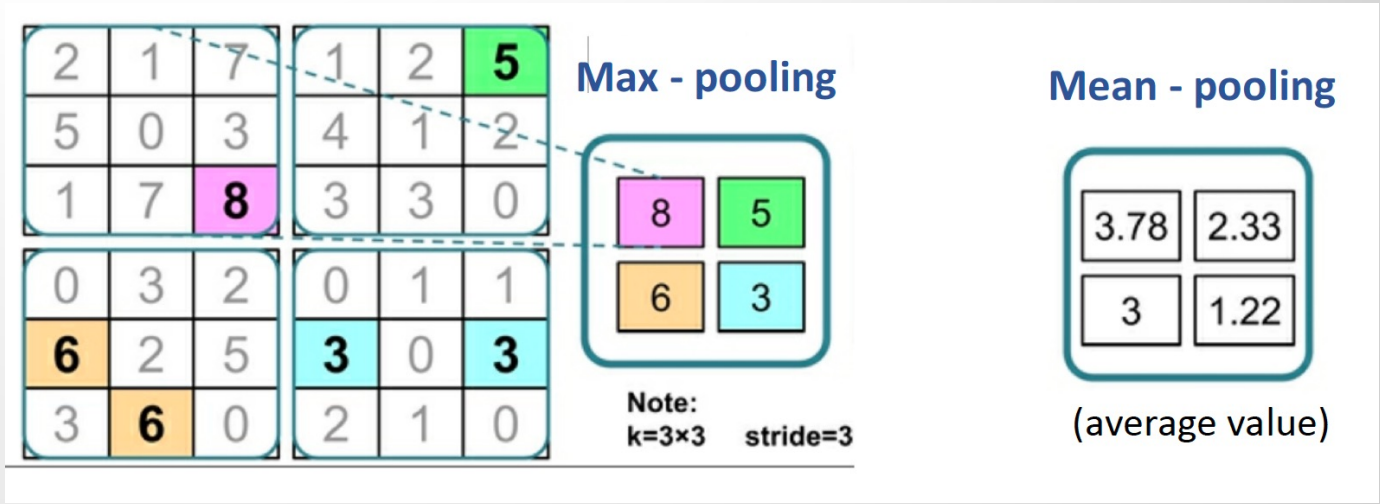
Learns:

- Which features are more relevant
- How to classify the images

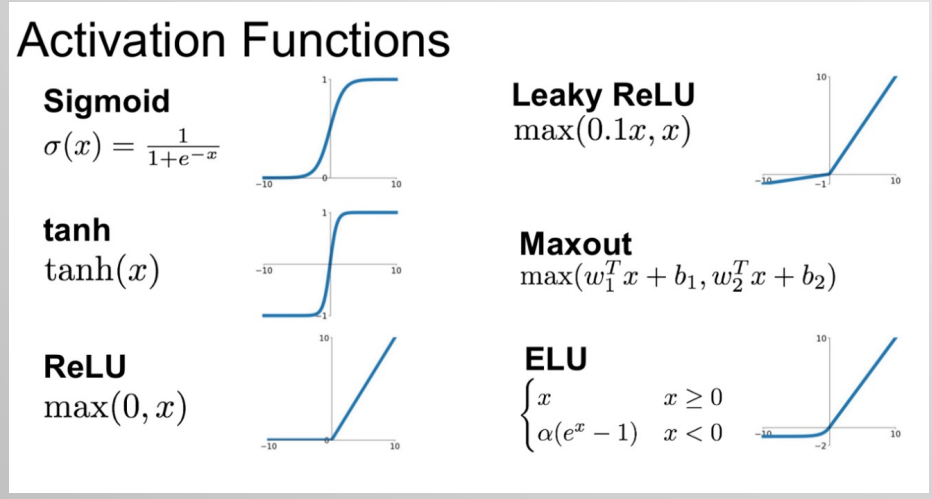
Convolutional layers
Bank of convolutiuonal filters



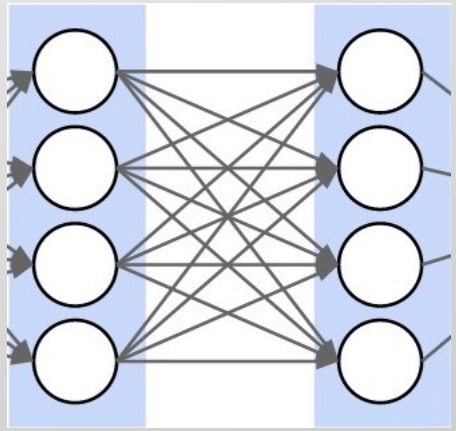
Pooling layers:
Sub-sampling by grouping, reduce overfitting



Activation layers: Connection between layers



Fully connected layers: Feature classification



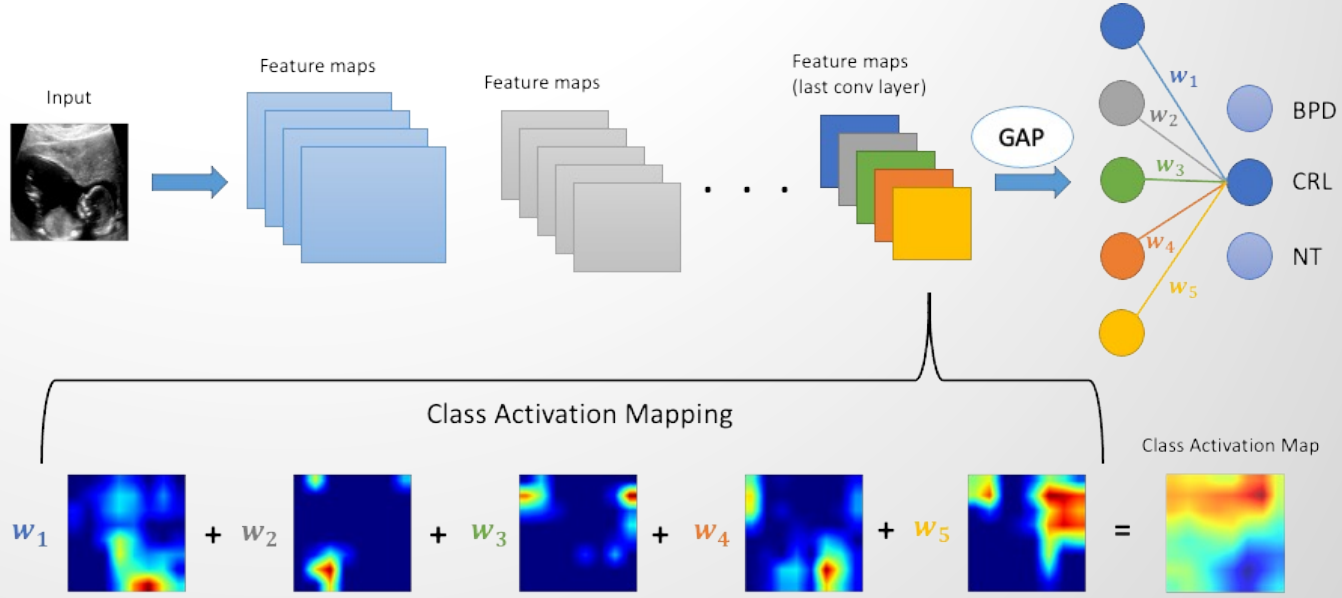
Transfer learning: Pre-trained models

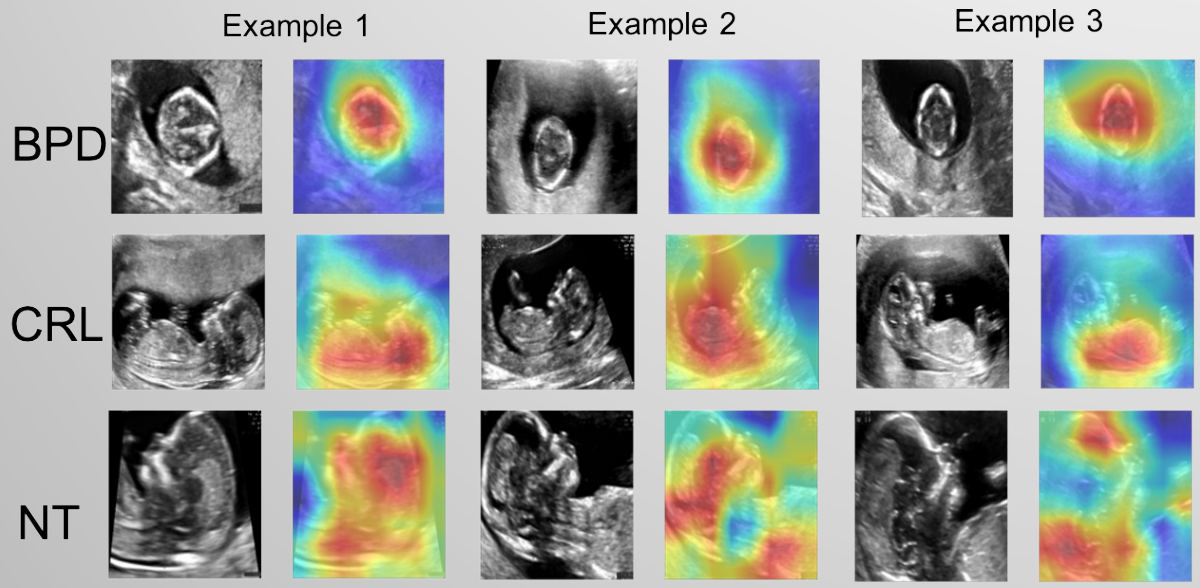

```
from keras.applications.inception_v3 import InceptionV3
from keras.layers import Input

# this could also be the output a different Keras model or layer
input_tensor = Input(shape=(224, 224, 3))  # this assumes K.image_data_format() == 'channels_last'

model = InceptionV3(input_tensor=input_tensor, weights='imagenet', include_top=True)
```

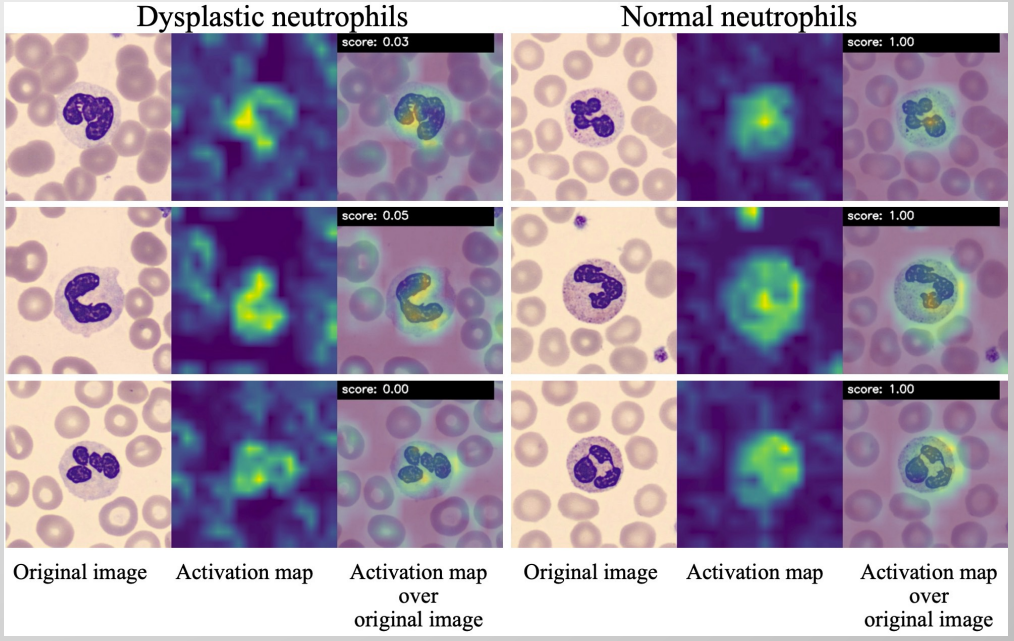
Documentation for individual models

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.715	0.901	138,357,544	23
VGG19	549 MB	0.727	0.910	143,667,240	26
ResNet50	99 MB	0.759	0.929	25,636,712	168
InceptionV3	92 MB	0.788	0.944	23,851,784	159
InceptionResNetV2	215 MB	0.804	0.953	55,873,736	572
MobileNet	17 MB	0.665	0.871	4,253,864	88
DenseNet121	33 MB	0.745	0.918	8,062,504	121
DenseNet169	57 MB	0.759	0.928	14,307,880	169
DenseNet201	80 MB	0.770	0.933	20,242,984	201

Deep Learning Interpretability







Hands-on tutorial

python + colab

TensorFlow K Keras

GitHub

MNIST

Labelled Faces in the Wild (LFW)

CIFAR-10