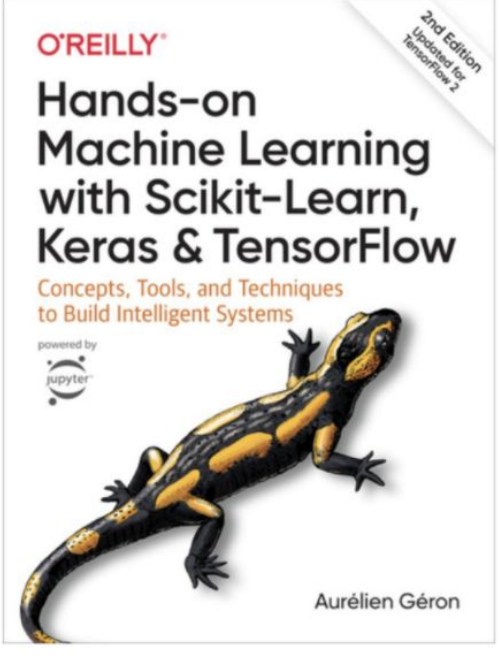
# Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition

★★★★★ 92 reviews

By Aurélien Géron




**TIME TO COMPLETE:**
24h 18m

**TOPICS:**
Machine Learning

**PUBLISHED BY:**
O'Reilly Media, Inc.

**PUBLICATION DATE:**
September 2019

**PRINT LENGTH:**
848 pages

| Chapter | Title |
|---|---|
| 1 | The Machine Learning Landscape |
| 2 | End-to-End Machine Learning Project |
| 3 | Classification |
| 4 | Training Models |
| 5 | Support Vector Machines |
| 6 | Decision Trees |
| 7 | Ensemble Learning and Random Forests |
| 8 | Dimensionality Reduction |
| 9a | Unsupervised Learning Techniques: Clustering |
| 9b | Unsupervised Learning Techniques: Gaussian Mixtures |
| 10 | Introduction to Artificial Neural Networks with Keras |
| 11 | Training Deep Neural Networks |
| 12 | Custom Models and Training with TensorFlow |
| 13 | Loading and Preprocessing Data with TensorFlow |
| 14 | Deep Computer Vision Using Convolutional Neural Networks |
| 15 | Processing Sequences Using RNNs and CNNs |
| 16 | Natural Language Processing with RNNs and Attention |
| 17 | Representation Learning and Generative Learning Using Autoencoders and GANs |
| 18 | Reinforcement Learning |
| 19 | Training and Deploying TensorFlow Models at Scale |

Chapter 14 : Deep Computer Vision Using Convolutional Neural Networks
Discussion led by George Zoto

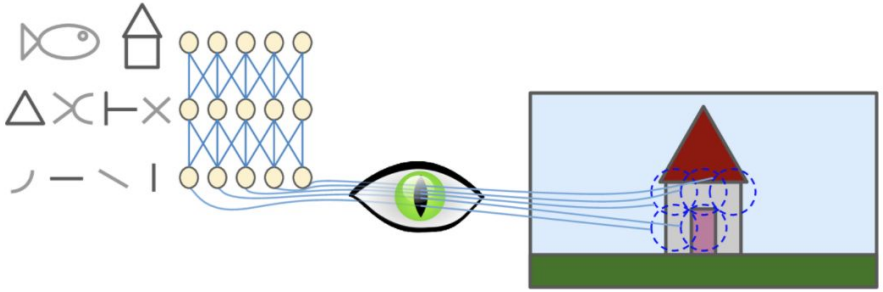
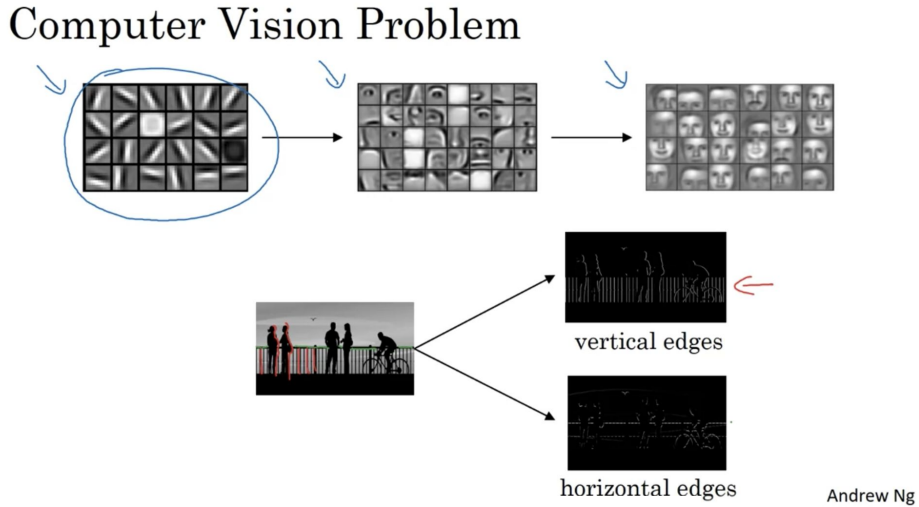
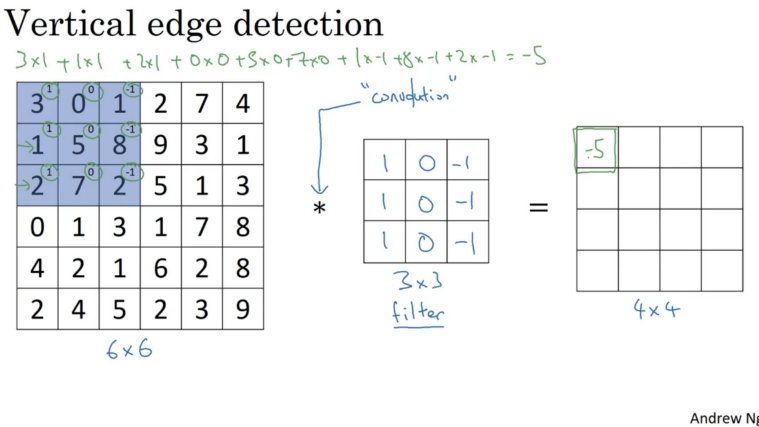
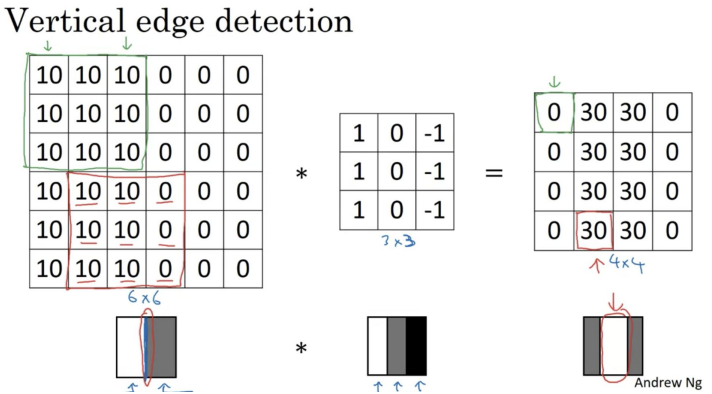


*Figure 14-1. Biological neurons in the visual cortex respond to specific patterns in small regions of the visual field called receptive fields; as the visual signal makes its way through consecutive brain modules, neurons respond to more complex patterns in larger receptive fields.*

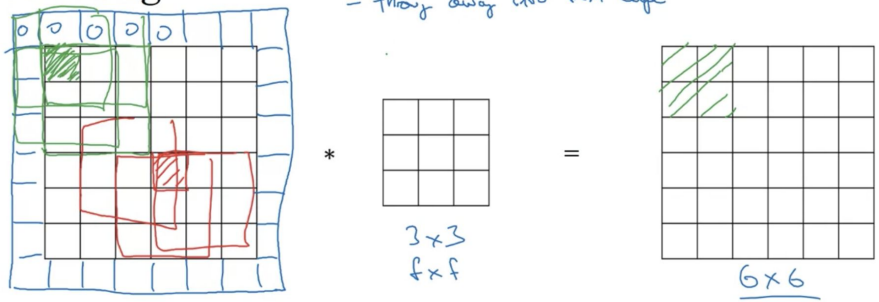


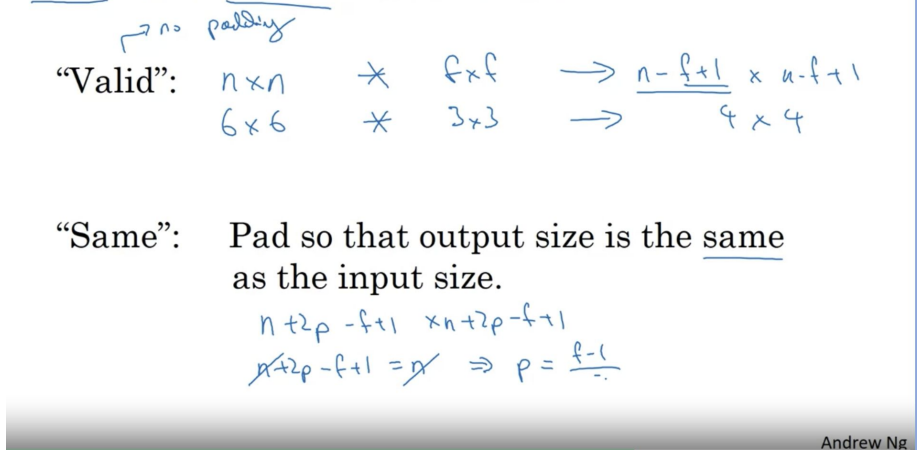

Source: https://www.coursera.org/learn/convolutional-neural-networks/lecture/4Trod/edge-detection-example

# Padding



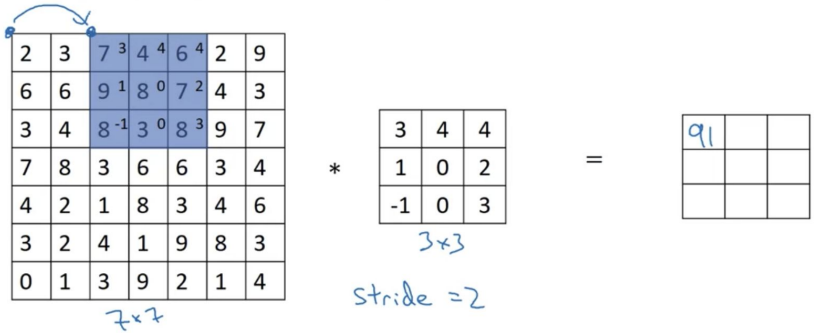

Andrew Ng

# Valid and Same convolutions




Andrew Ng

# Strided convolution




# Summary of convolutions

$n \times n$ image $\quad$ $f \times f$ filter

padding $p$ $\quad$ stride $s$

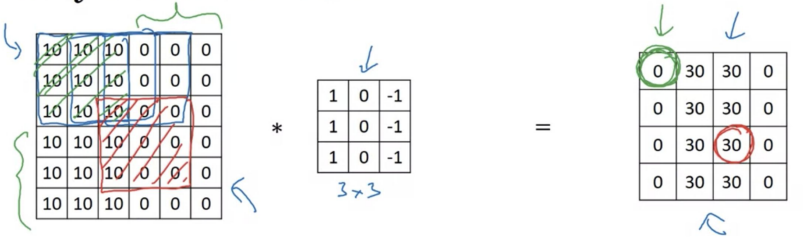
$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

Andrew Ng

# Enhancing Vision with Convolutional Neural Networks

## Why convolutions

translation invariance




**Parameter sharing:** A feature detector (such as a vertical edge detector) that's useful in one part of the image is probably useful in another part of the image.

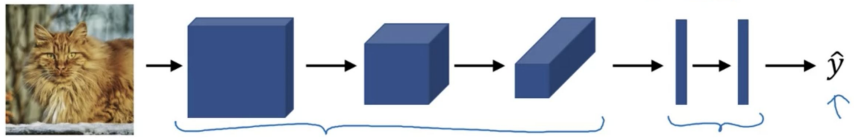
→ **Sparsity of connections:** In each layer, each output value depends only on a small number of inputs.

Andrew Ng

## Putting it together

Training set $(x^{(1)}, y^{(1)}) \ldots (x^{(m)}, y^{(m)})$.

w, b



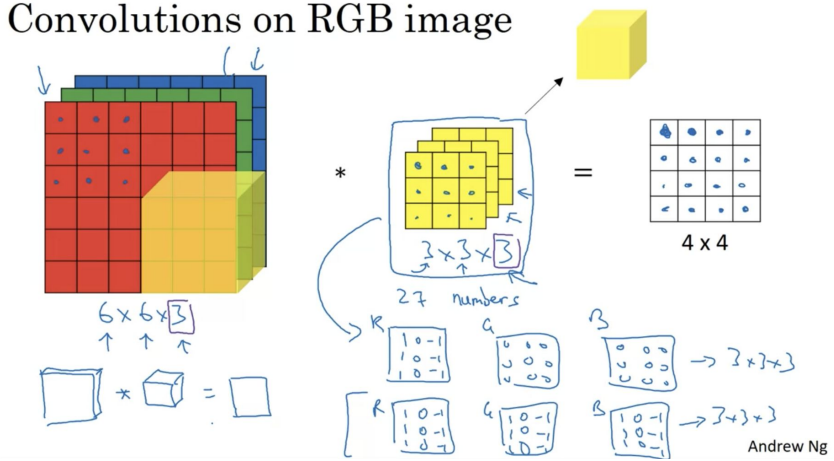

Cost $J = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$

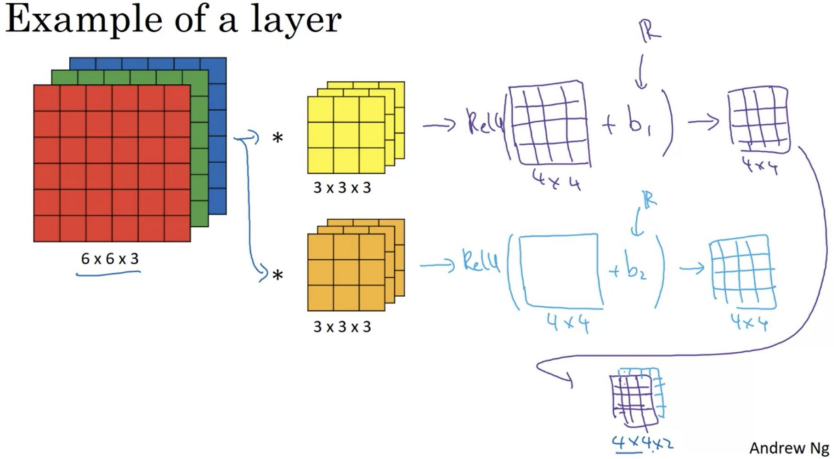
Use gradient descent to optimize parameters to reduce $J$

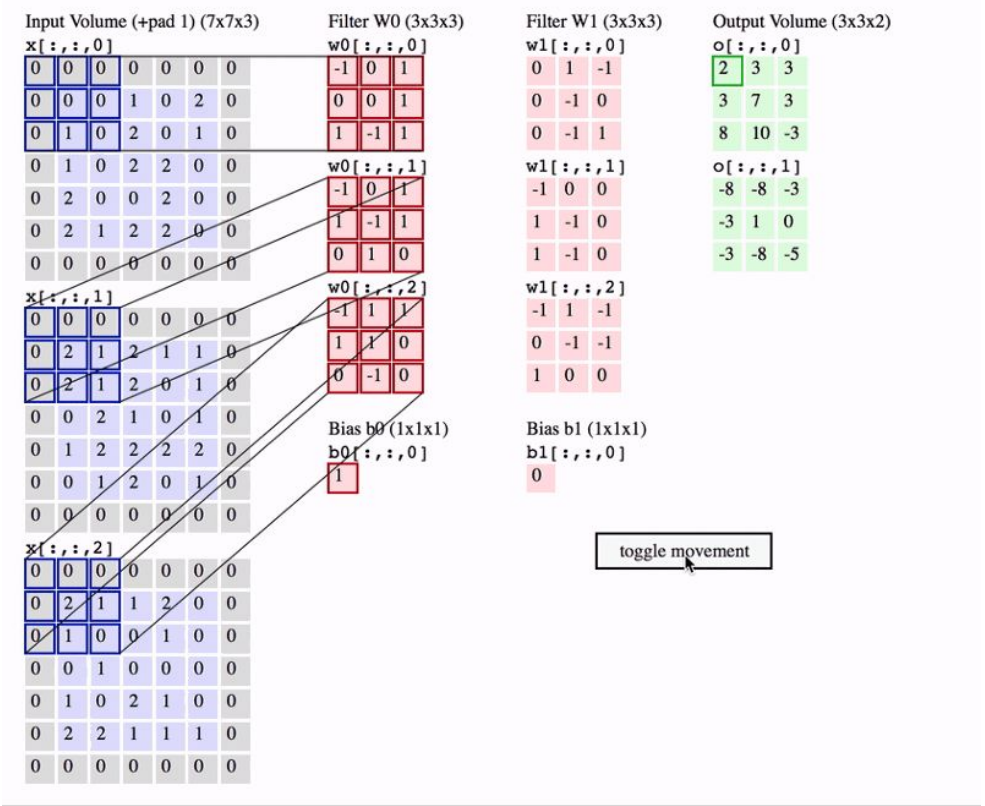
Andrew Ng

Source: https://www.coursera.org/lecture/convolutional-neural-networks/why-convolutions-Xv7B5

# Content added after Week 1 and 2, based on group feedback




Source: https://www.coursera.org/learn/convolutional-neural-networks
https://images.app.goo.gl/Z4xDvjzi3Gcerpm27

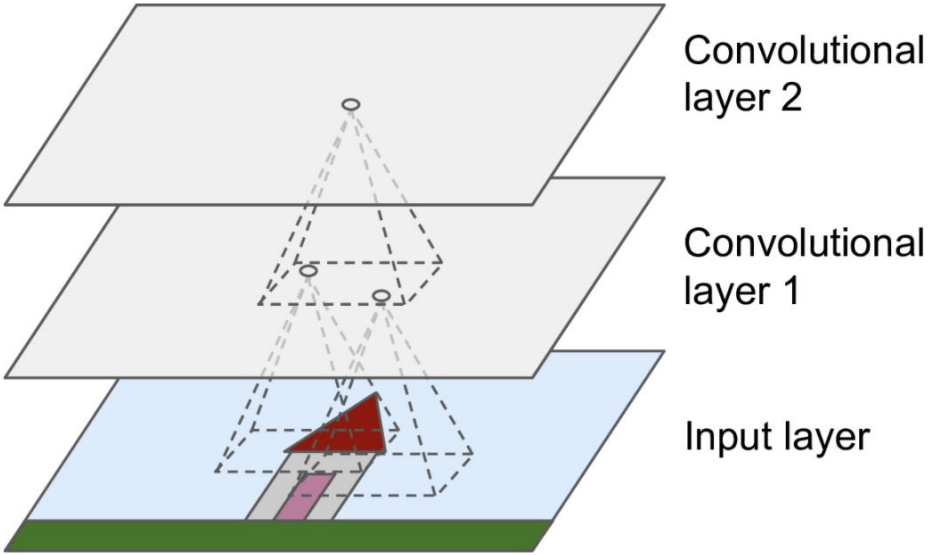



*Figure 14-2. CNN layers with rectangular local receptive fields*

```
model = keras.models.Sequential([
    keras.layers.Conv2D(64, 7, activation="relu", padding="same",
                        input_shape=[28, 28, 1]),
    keras.layers.MaxPooling2D(2),
    keras.layers.Conv2D(128, 3, activation="relu", padding="same"),
    keras.layers.Conv2D(128, 3, activation="relu", padding="same"),
    keras.layers.MaxPooling2D(2),
    keras.layers.Conv2D(256, 3, activation="relu", padding="same"),
    keras.layers.Conv2D(256, 3, activation="relu", padding="same"),
    keras.layers.MaxPooling2D(2),
    keras.layers.Flatten(),
    keras.layers.Dense(128, activation="relu"),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(64, activation="relu"),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(10, activation="softmax")
])
```

```
model.layers
```

```
convolutional.Conv2D
pooling.MaxPooling2D
convolutional.Conv2D
pooling.MaxPooling2D
convolutional.Conv2D
pooling.MaxPooling2D
convolutional.Conv2D
pooling.MaxPooling2D
convolutional.Conv2D
pooling.MaxPooling2D
core.Flatten at 0x7f
core.Dense at 0x7f07
core.Dense at 0x7f07
```

```
model.layers[0].Conv2D
```

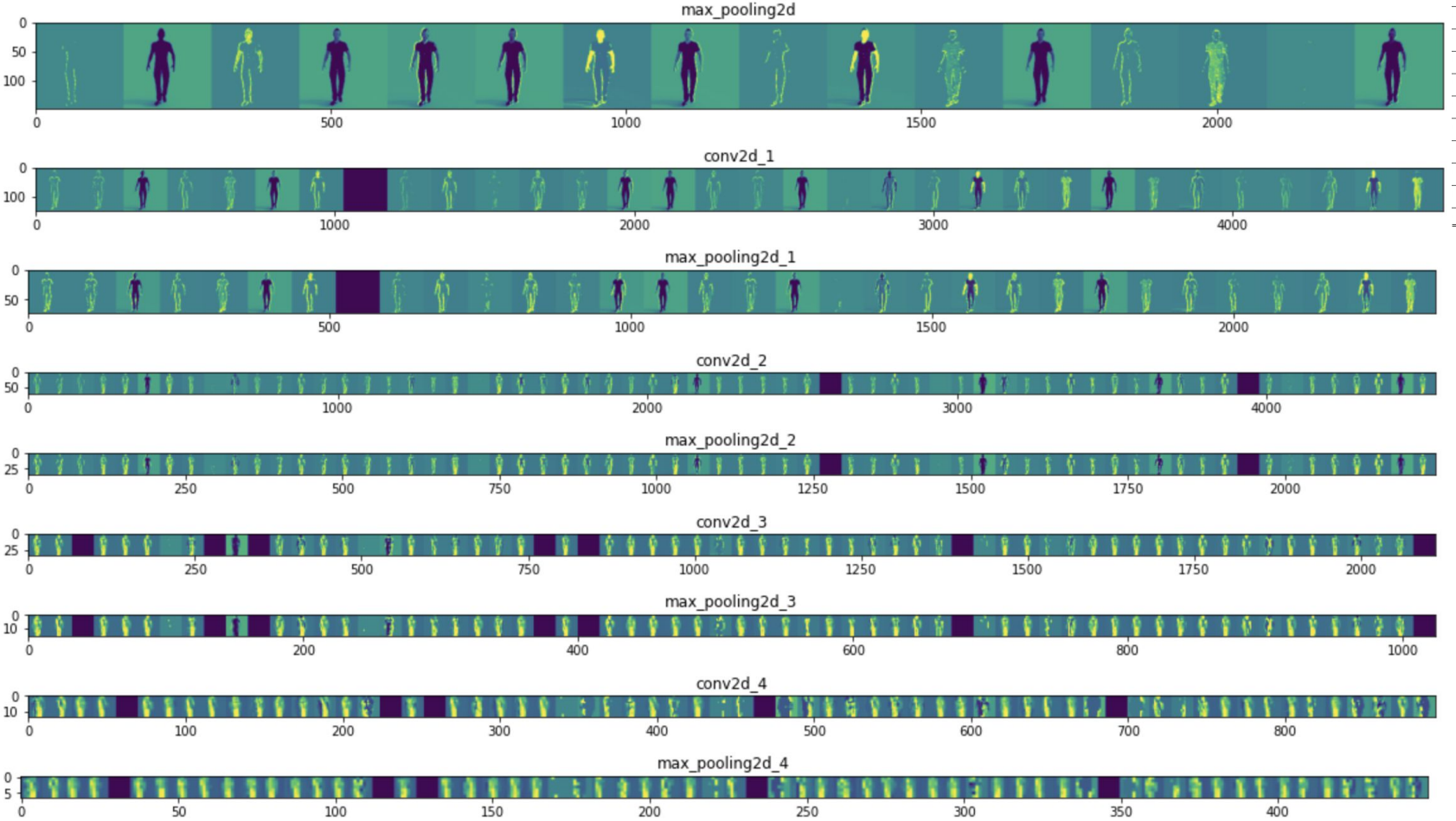
- **valhuman01-00.png**(image/png) -

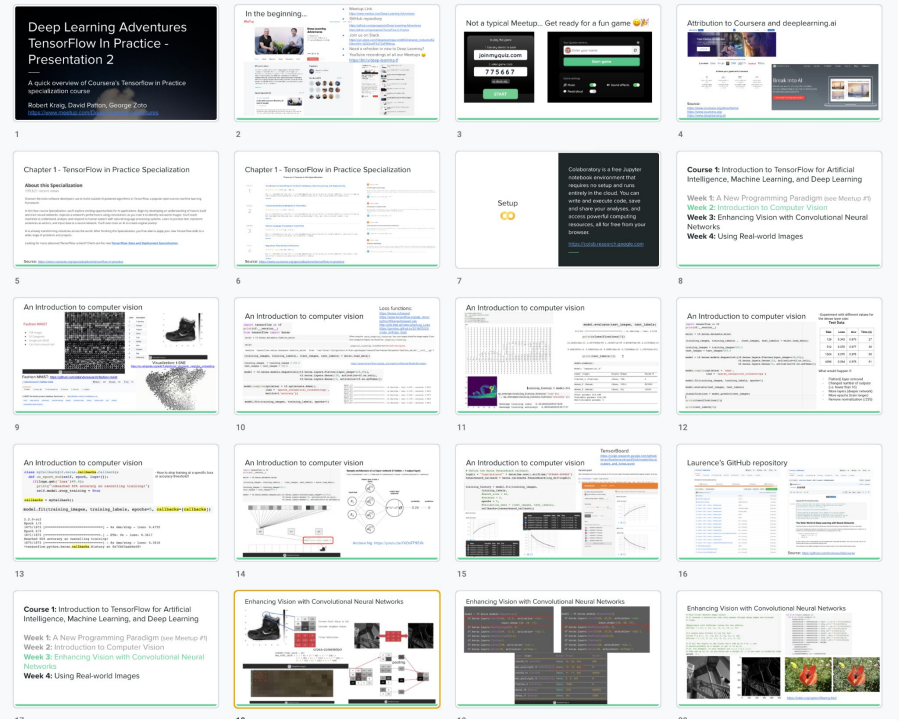
```
Saving valhuman01-00.png to
[1.]
valhuman01-00.png is a human
```

[1.]

```
successive_outputs = [layer.output for layer in model.layers[1:]]
```

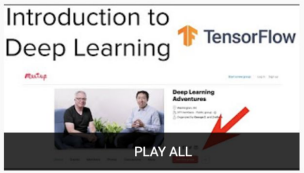
| Layer (type) | Output Shape |
|---|---|
| conv2d (Conv2D) | (None, 298, 298, 16) |
| max_pooling2d (MaxPooling2D) | (None, 149, 149, 16) |
| conv2d_1 (Conv2D) | (None, 147, 147, 32) |
| | 73, 73, 32) |
| | 71, 71, 64) |
| | 35, 35, 64) |
| | 33, 33, 64) |
| | 16, 16, 64) |
| | 14, 14, 64) |
| | 7, 7, 64) |
| | 3136) |
| | 512) |
| | 1) |






[Deep-Learning-Adventures-Tensorflow-In-Practice-Presentation-2]

[http://bit.ly/dla-youtube]




Deep Learning Adventures - Tensorflow In Practice Specialization

13 videos • 6,564 views • Last updated on Jul 10, 2020

Public

Join our Deep Learning Adventures community and become an expert in Deep Learning, TensorFlow, Computer Vision, Convolutional Neural Networks, Kaggle

SORT


Introduction to Deep Learning and TensorFlow

George Zoto


Introduction to Computer Vision and Convolutional Neural Networks

George Zoto


Convolutional Neural Networks and ImageDataGenerator

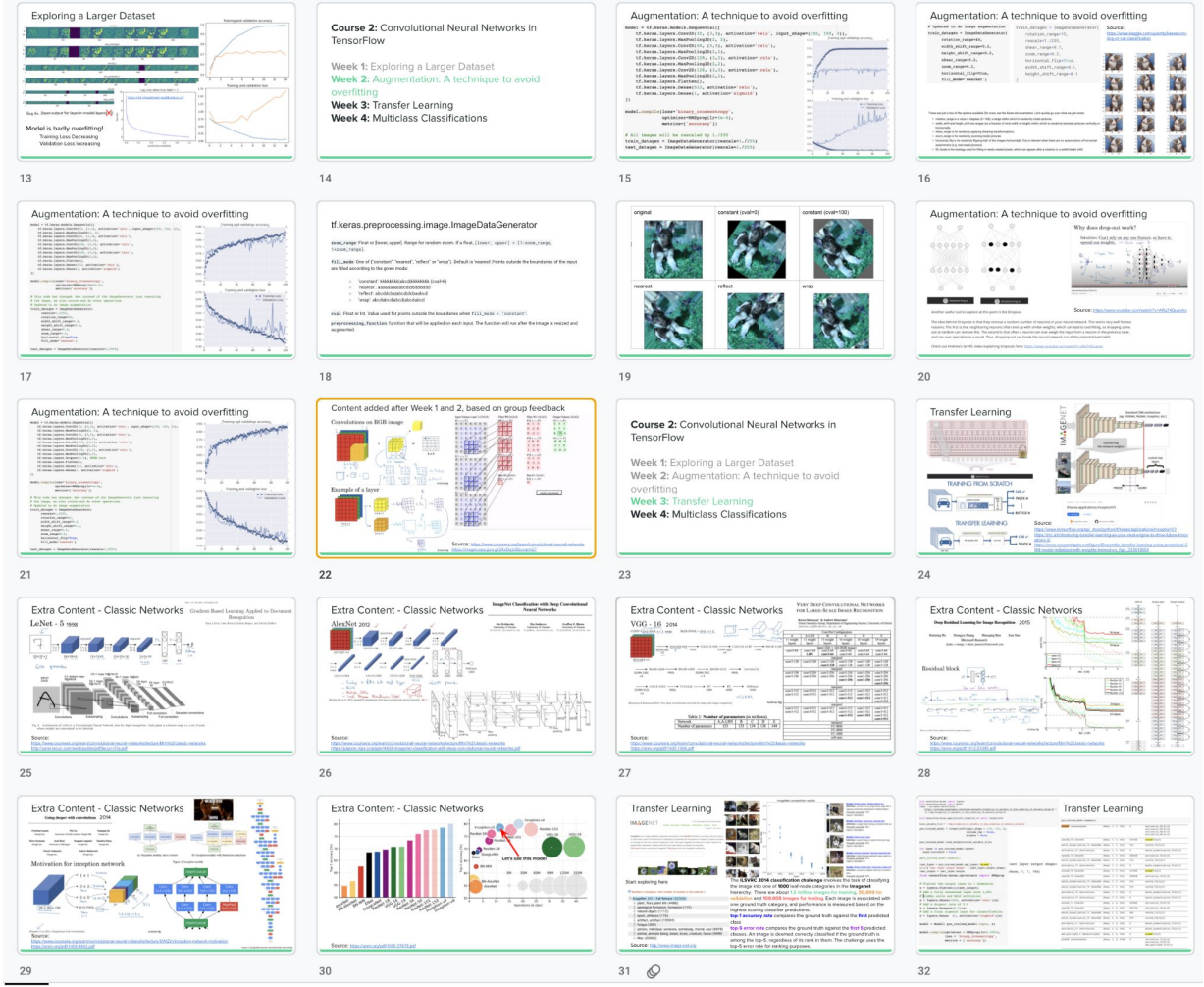
George Zoto


Kaggle Challenge, Data Augmentation and Dropouts

George Zoto


Transfer Learning, Multiclass Classifications and Overfitting

George Zoto




[Deep-Learning-Adventures-Tensorflow-In-Practice-Presentation-3]

http://bit.ly/dla-youtube

ILSVRC challenge

*Table 14-1. LeNet-5 architecture*

| Layer | Type | Maps | Size | Kernel size | Stride | Activation |
|---|---|---|---|---|---|---|
| Out | Fully connected | – | 10 | – | – | RBF |
| F6 | Fully connected | – | 84 | – | – | tanh |
| C5 | Convolution | 120 | 1 × 1 | 5 × 5 | 1 | tanh |
| S4 | Avg pooling | 16 | 5 × 5 | 2 × 2 | 2 | tanh |
| C3 | Convolution | 16 | 10 × 10 | 5 × 5 | 1 | tanh |
| S2 | Avg pooling | 6 | 14 × 14 | 2 × 2 | 2 | tanh |
| C1 | Convolution | 6 | 28 × 28 | 5 × 5 | 1 | tanh |
| In | Input | 1 | 32 × 32 | – | – | – |

*Table 14-2. AlexNet architecture*

| Layer | Type | Maps | Size | Kernel size | Stride | Padding | Activation |
|---|---|---|---|---|---|---|---|
| Out | Fully connected | – | 1,000 | – | – | – | Softmax |
| F10 | Fully connected | – | 4,096 | – | – | – | ReLU |
| F9 | Fully connected | – | 4,096 | – | – | – | ReLU |
| S8 | Max pooling | 256 | 6 × 6 | 3 × 3 | 2 | `valid` | – |
| C7 | Convolution | 256 | 13 × 13 | 3 × 3 | 1 | `same` | ReLU |
| C6 | Convolution | 384 | 13 × 13 | 3 × 3 | 1 | `same` | ReLU |
| C5 | Convolution | 384 | 13 × 13 | 3 × 3 | 1 | `same` | ReLU |
| S4 | Max pooling | 256 | 13 × 13 | 3 × 3 | 2 | `valid` | – |
| C3 | Convolution | 256 | 27 × 27 | 5 × 5 | 1 | `same` | ReLU |
| S2 | Max pooling | 96 | 27 × 27 | 3 × 3 | 2 | `valid` | – |
| C1 | Convolution | 96 | 55 × 55 | 11 × 11 | 4 | `valid` | ReLU |
| In | Input | 3 (RGB) | 227 × 227 | – | – | – | – |

ILSVRC challenge

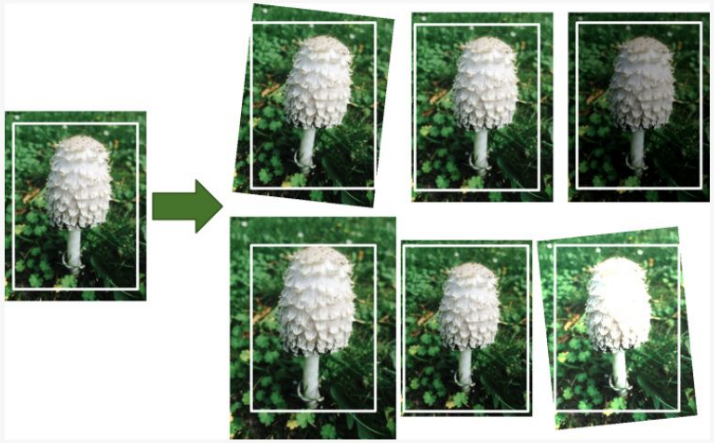


*Figure 14-12. Generating new training instances from existing ones*

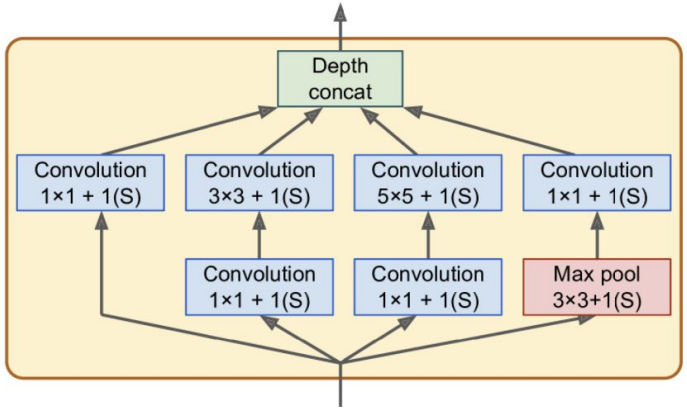



*Figure 14-13. Inception module*

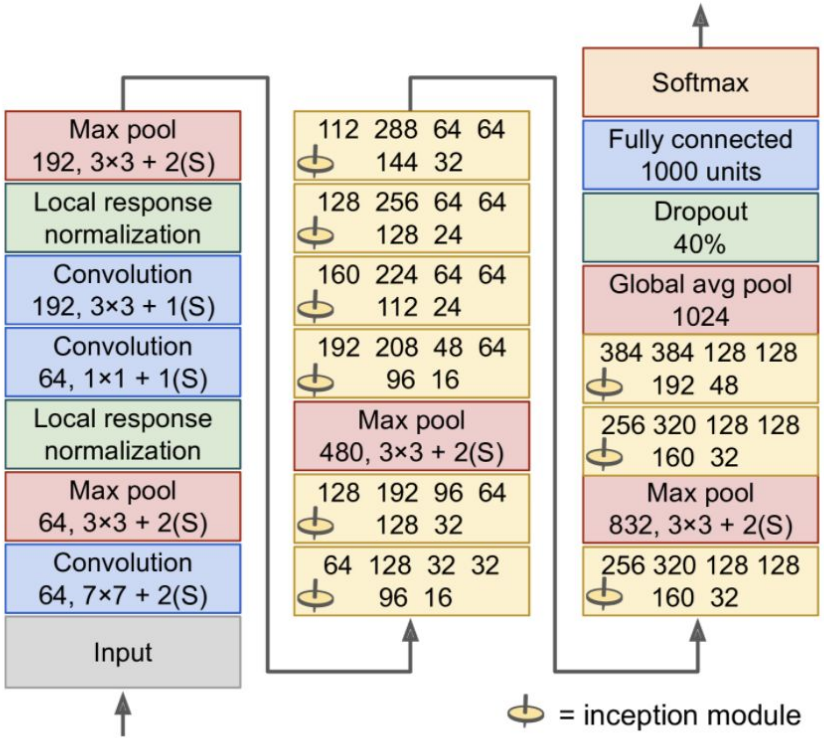



*Figure 14-14. GoogLeNet architecture*

ILSVRC challenge

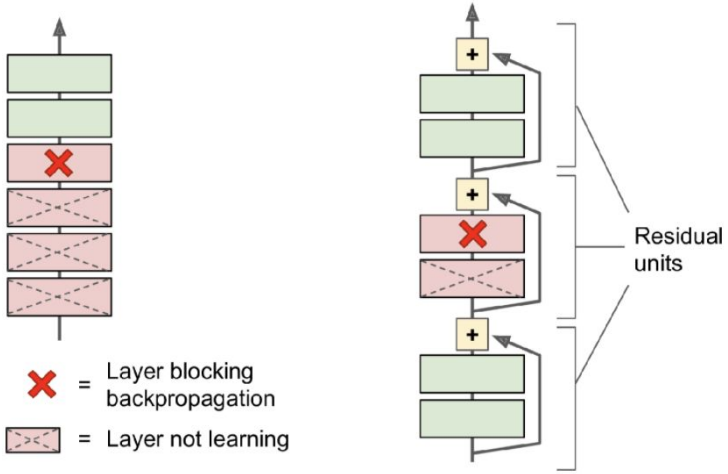



*Figure 14-16. Regular deep neural network (left) and deep residual network (right)*

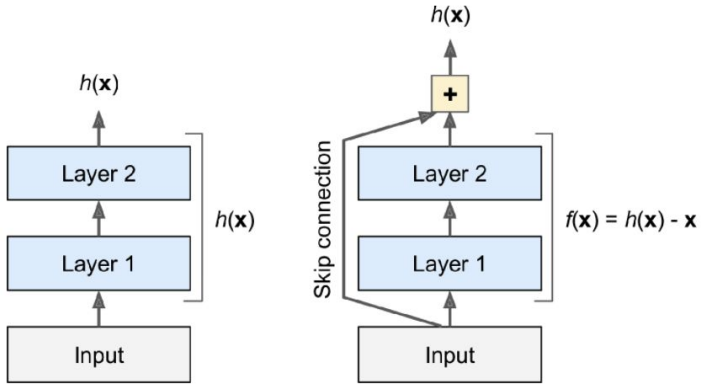



*Figure 14-15. Residual learning*

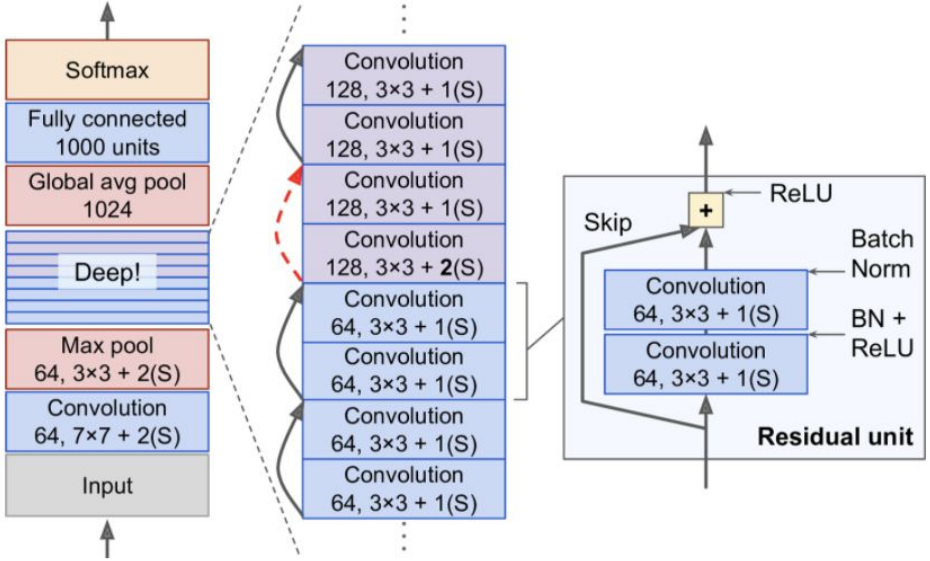



*Figure 14-17. ResNet architecture*

ILSVRC challenge

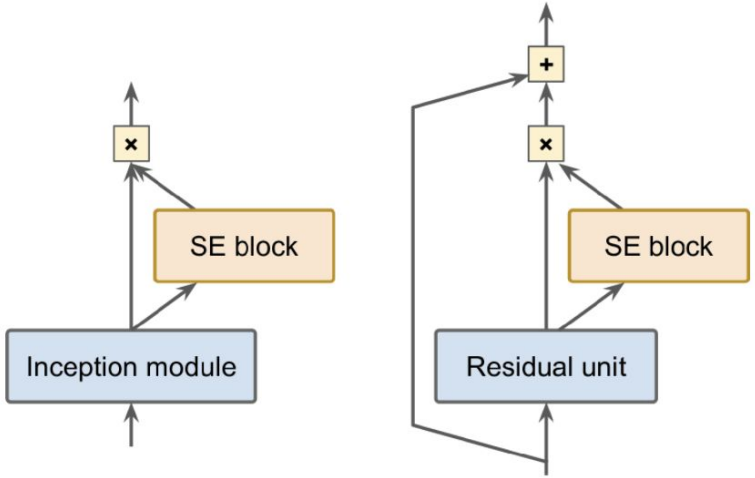



*Figure 14-20. SE-Inception module (left) and SE-ResNet unit (right)*

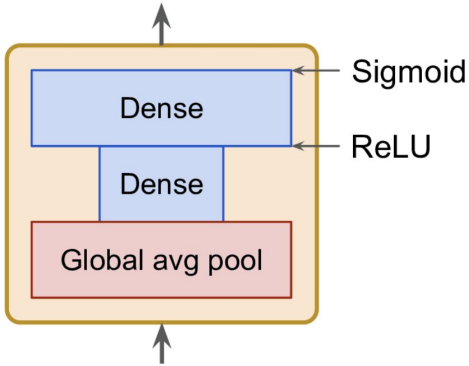



*Figure 14-22. SE block architecture*

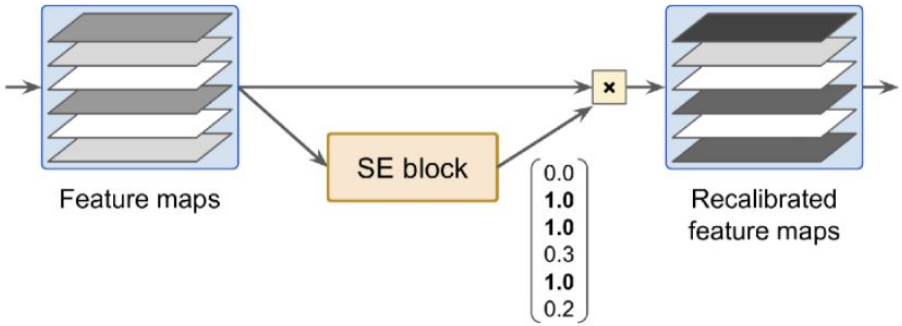



*Figure 14-21. An SE block performs feature map recalibration*

```
model = keras.applications.resnet50.ResNet50(weights="imagenet")

base_model = keras.applications.xception.Xception(weights="imagenet",
                                                  include_top=False)
avg = keras.layers.GlobalAveragePooling2D()(base_model.output)
output = keras.layers.Dense(n_classes, activation="softmax")(avg)
model = keras.Model(inputs=base_model.input, outputs=output)


optimizer = keras.optimizers.SGD(lr=0.2, momentum=0.9, decay=0.01)
model.compile(loss="sparse_categorical_crossentropy", optimizer=optimizer,
              metrics=["accuracy"])
history = model.fit(train_set, epochs=5, validation_data=valid_set)
```

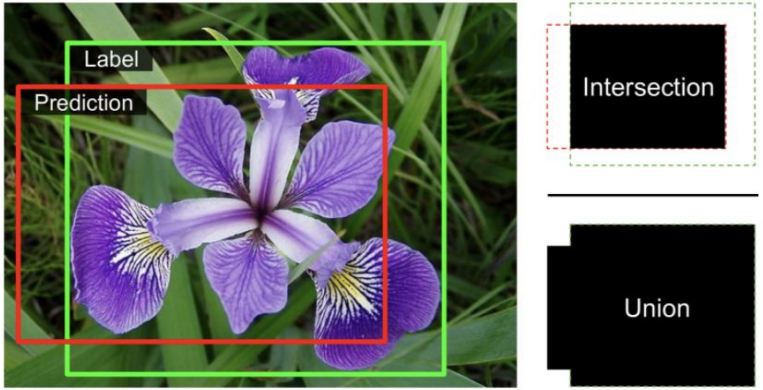



*Figure 14-23. Intersection over Union (IoU) metric for bounding boxes*

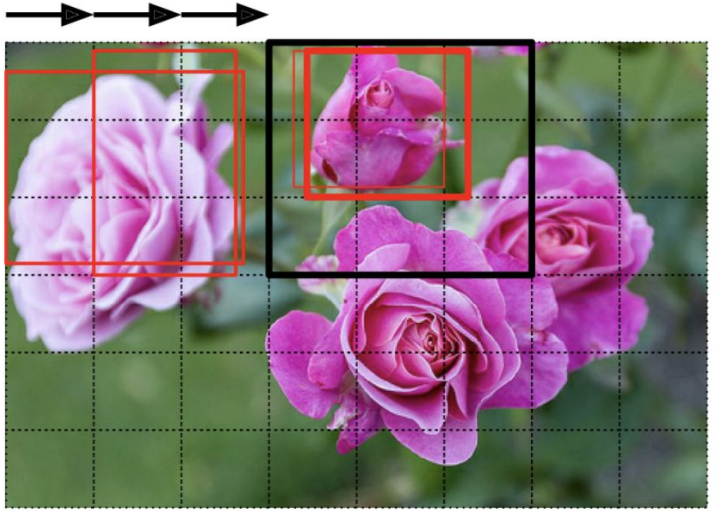


*Figure 14-24. Detecting multiple objects by sliding a CNN across the image*

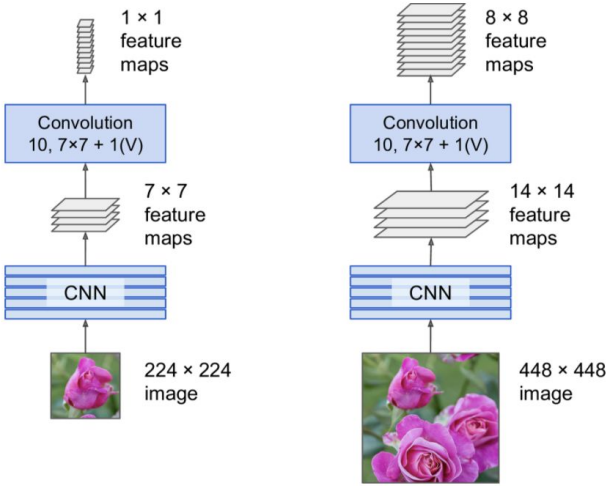
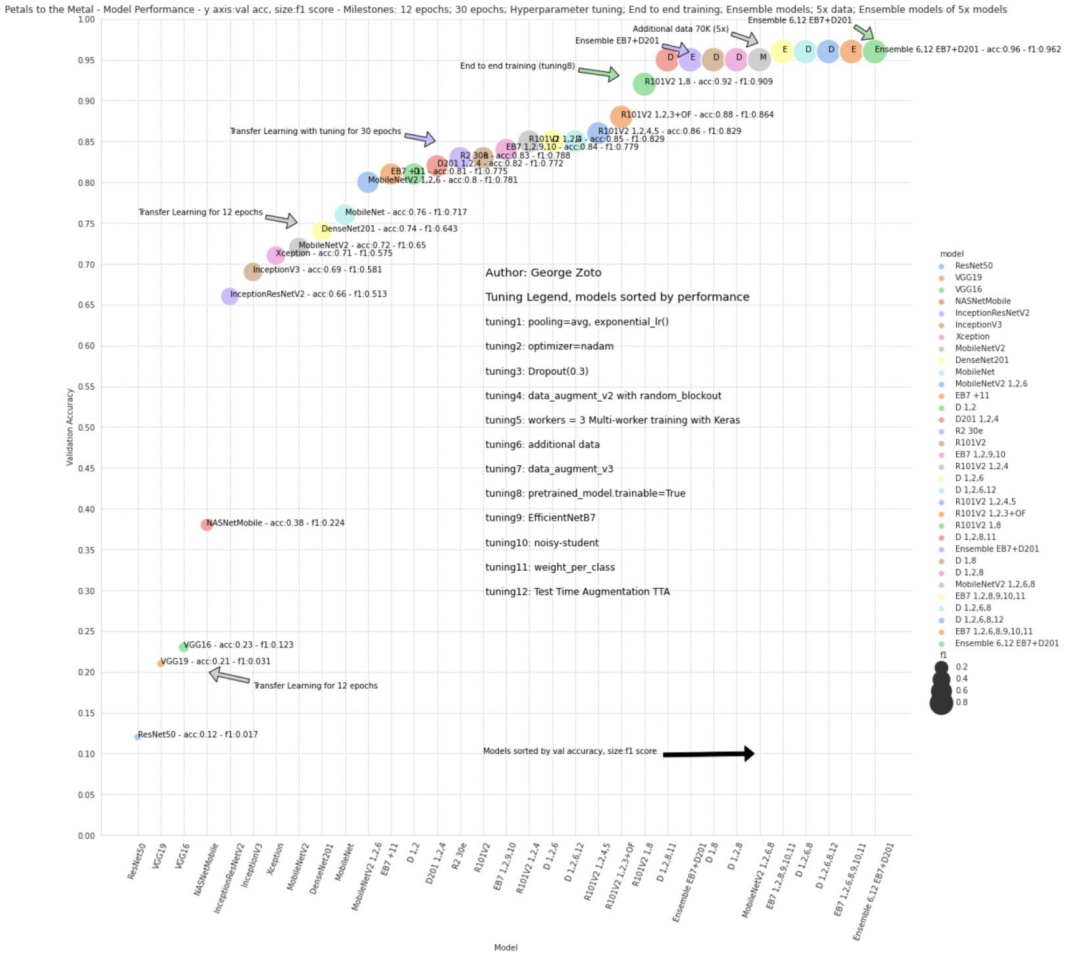
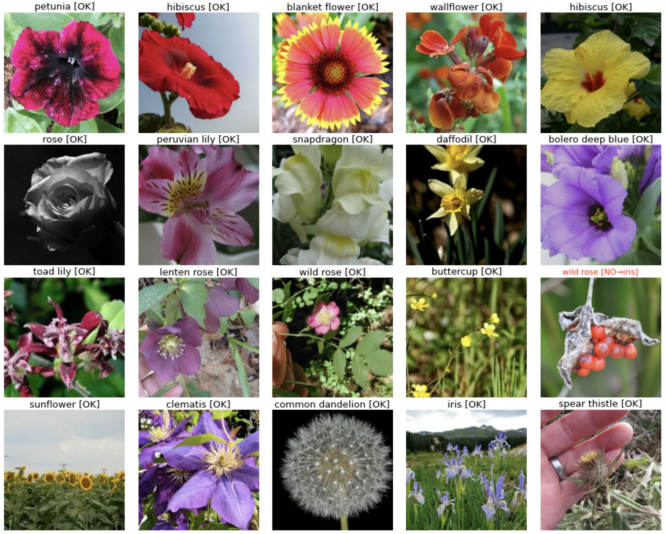



*Figure 14-25. The same fully convolutional network processing a small image (left) and a large one (right)*


*Figure 14-26. Semantic segmentation*



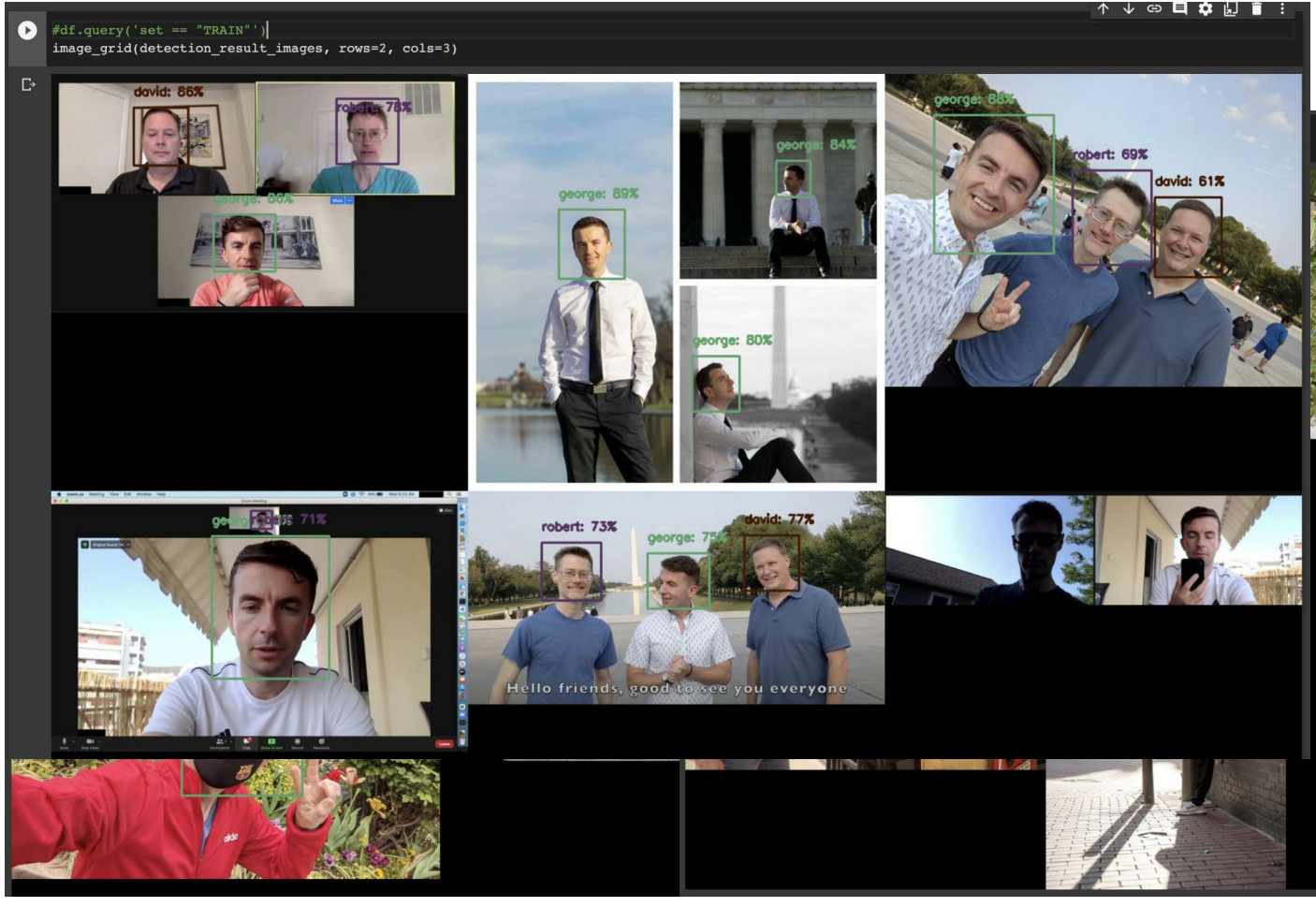

[Computer Vision - Petals to the Metal 🌻🌸🌹]



# Colab Demo

Deep Learning Adventures

Animated - Best viewed as a presentation



# Android App Demo

Deep Learning Adventures

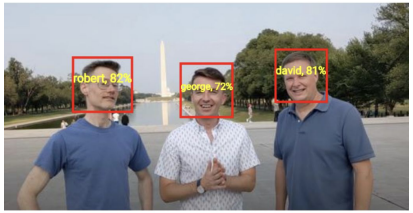
Deep Learning Adventures

Custom Object Detection App

Select a preset image or take a new photo

Take photo

Deep Learning Adventures




Select a preset image or take a new photo

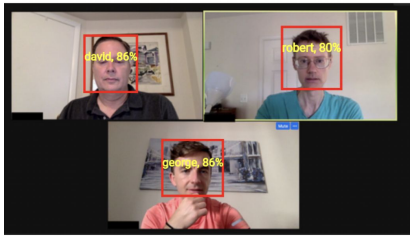
Take photo

Deep Learning Adventures

Select a preset image or take a new photo

Take photo

Deep Learning Adventures




Select a preset image or take a new photo

Take photo

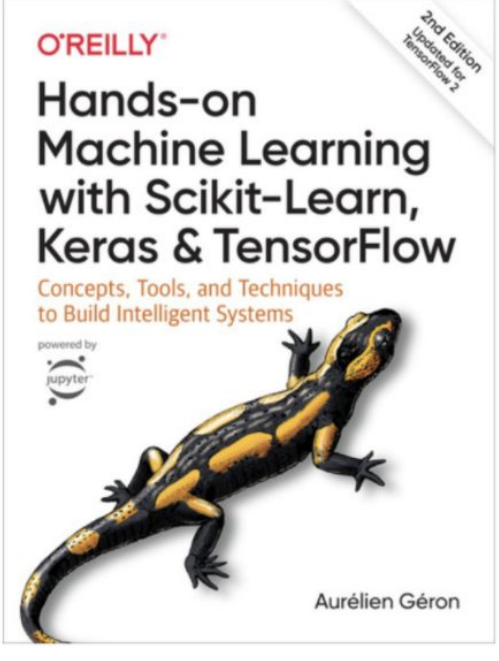
# Useful Resources

- Long list coming up, see live presentation for now

# Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition

★★★★★ 92 reviews

By Aurélien Géron




**TIME TO COMPLETE:**
24h 18m

**TOPICS:**
Machine Learning

**PUBLISHED BY:**
O'Reilly Media, Inc.

**PUBLICATION DATE:**
September 2019

**PRINT LENGTH:**
848 pages

| Chapter | Title |
|---|---|
| 1 | The Machine Learning Landscape |
| 2 | End-to-End Machine Learning Project |
| 3 | Classification |
| 4 | Training Models |
| 5 | Support Vector Machines |
| 6 | Decision Trees |
| 7 | Ensemble Learning and Random Forests |
| 8 | Dimensionality Reduction |
| 9a | Unsupervised Learning Techniques: Clustering |
| 9b | Unsupervised Learning Techniques: Gaussian Mixtures |
| 10 | Introduction to Artificial Neural Networks with Keras |
| 11 | Training Deep Neural Networks |
| 12 | Custom Models and Training with TensorFlow |
| 13 | Loading and Preprocessing Data with TensorFlow |
| 14 | Deep Computer Vision Using Convolutional Neural Networks |
| 15 | Processing Sequences Using RNNs and CNNs |
| 16 | Natural Language Processing with RNNs and Attention |
| 17 | Representation Learning and Generative Learning Using Autoencoders and GANs |
| 18 | Reinforcement Learning |
| 19 | Training and Deploying TensorFlow Models at Scale |

Chapter 14 : Deep Computer Vision Using Convolutional Neural Networks
Discussion led by George Zoto