LAB 4 - GOING DEEPER INTO CONVOLUTIONAL NEURAL NETWORKS

Dr. Pandiyaraju V
Shravan Venkatraman

August 9, 2024

TABLE OF CONTENTS

01 INTRODUCTION

02 MLP RECAP

03 MINI BATCH GRADIENT DESCENT

04 HYPERPARAMETER TUNING

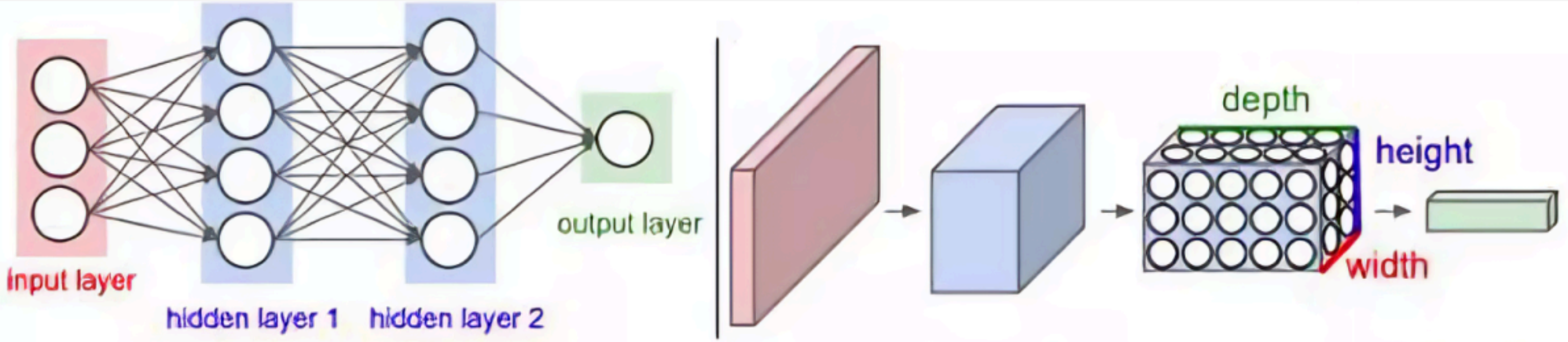
05 TUNING TECHNIQUES

CNN RECAP

Human Vision

Computer Vision

WHY CNNS?

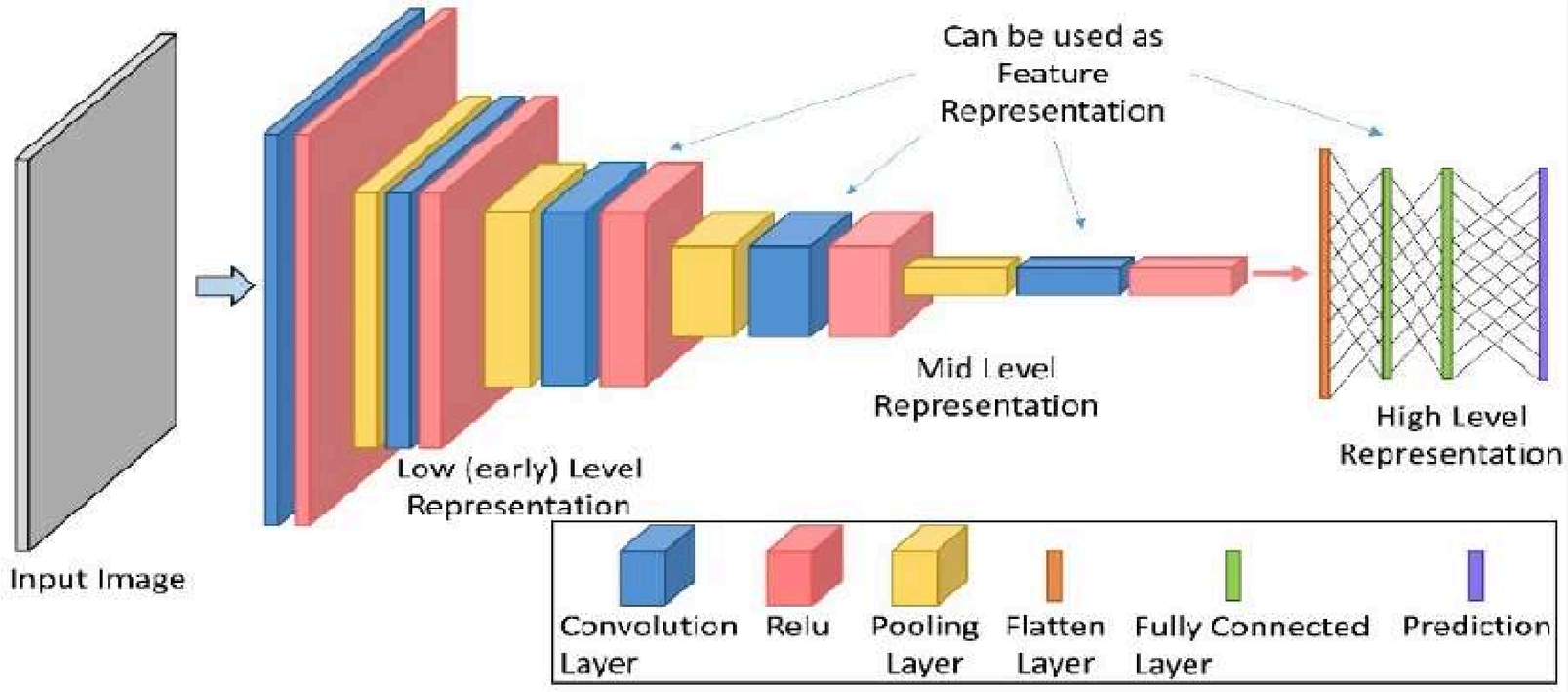


Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

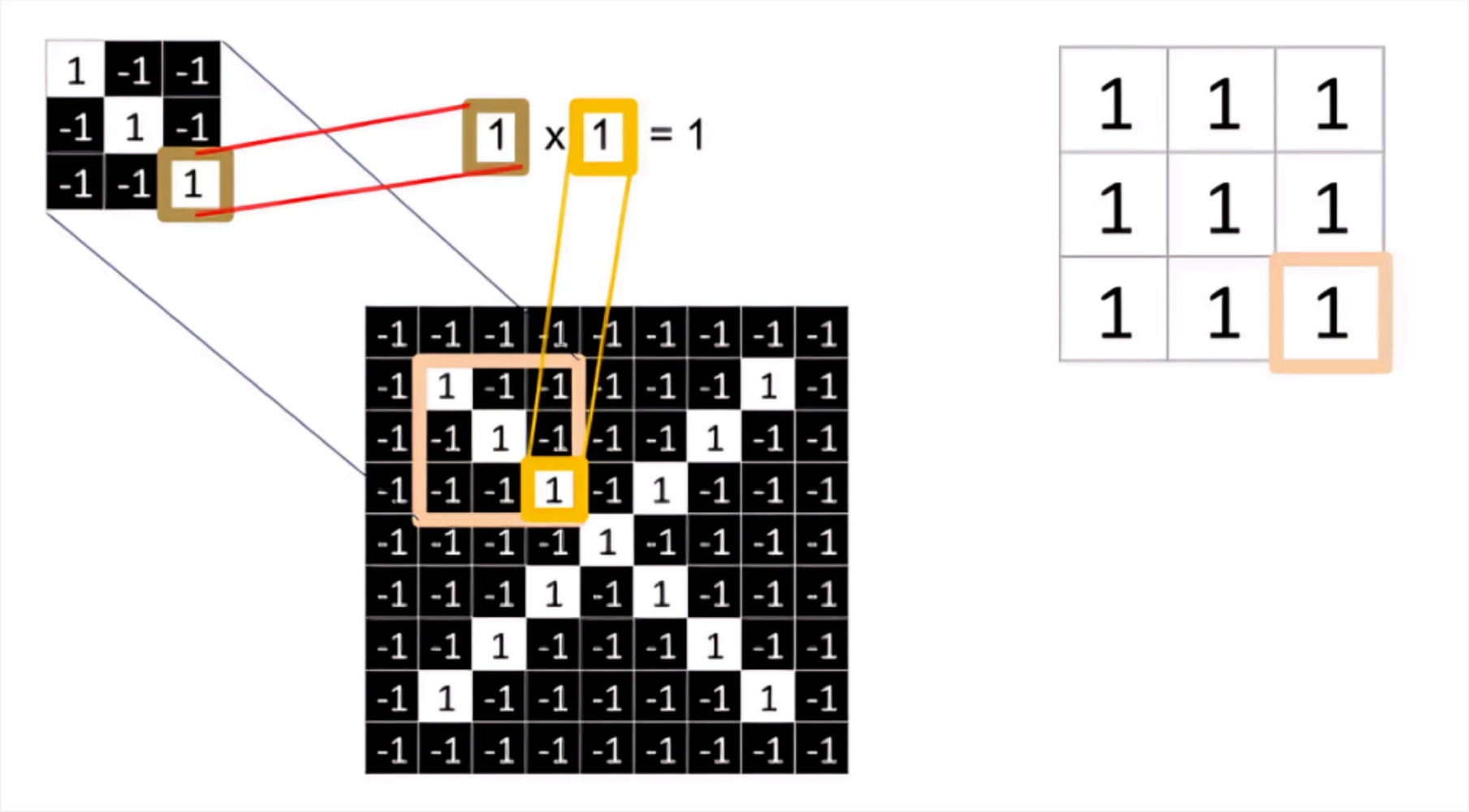
COMPONENTS OF A CNN

- **Hidden Layers / Feature Extraction Part: the part where feature extraction happens, via a series of convolutions and pooling operations.**

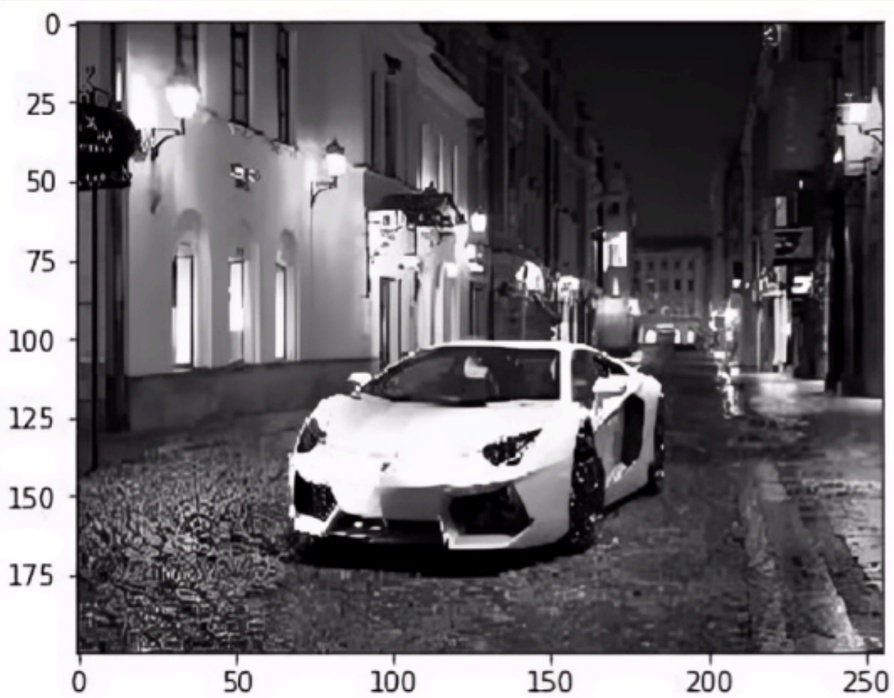
- **The classification part:** the fully connected layers will serve as the classifier on top of these extracted features.



WHAT HAPPENS WHEN YOU DO A CONVOLUTION?



WHAT HAPPENS WHEN YOU DO A CONVOLUTION?

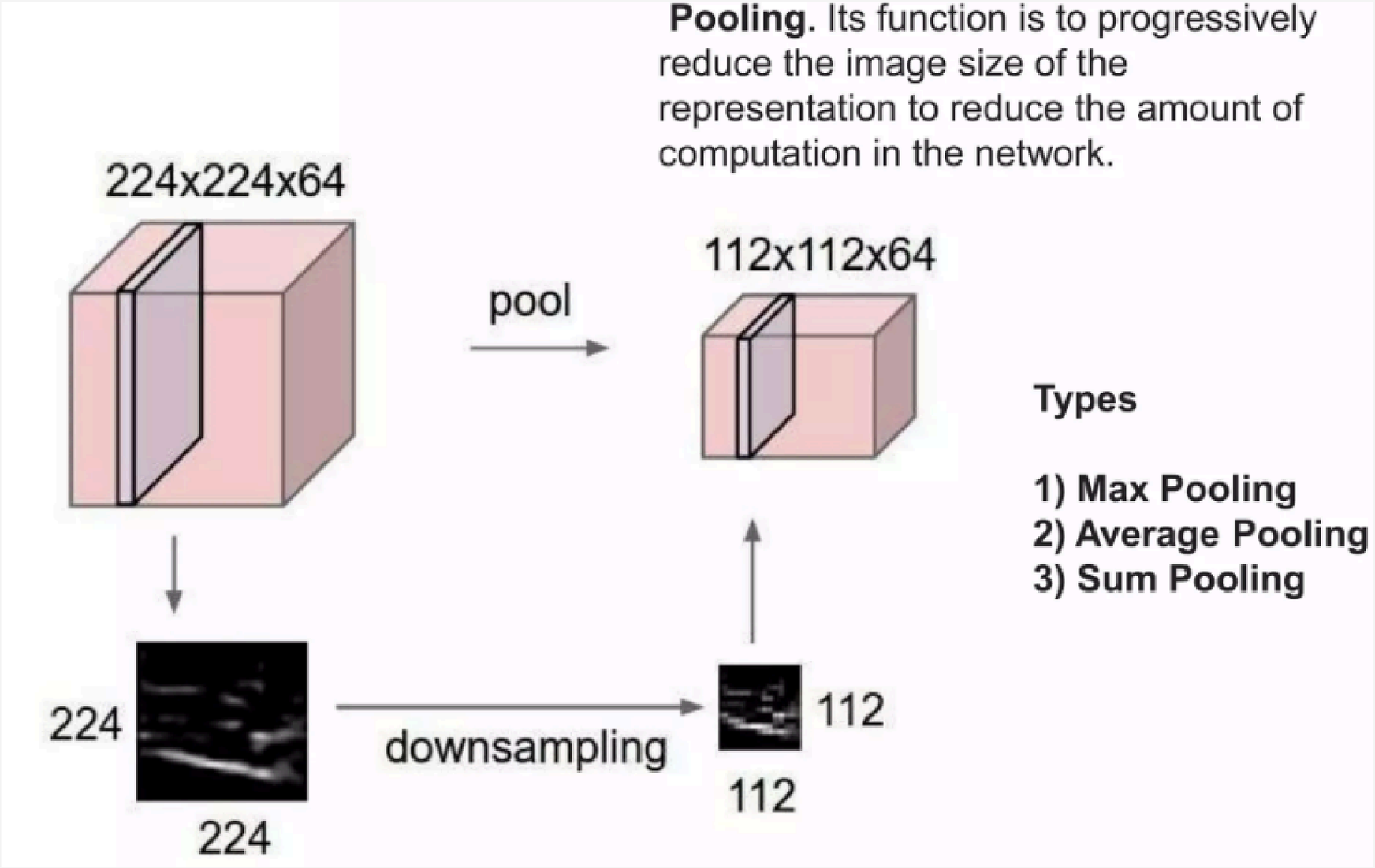


Original Image

Convoluted image

WHAT HAPPENS WHEN YOU MAXPOOL?

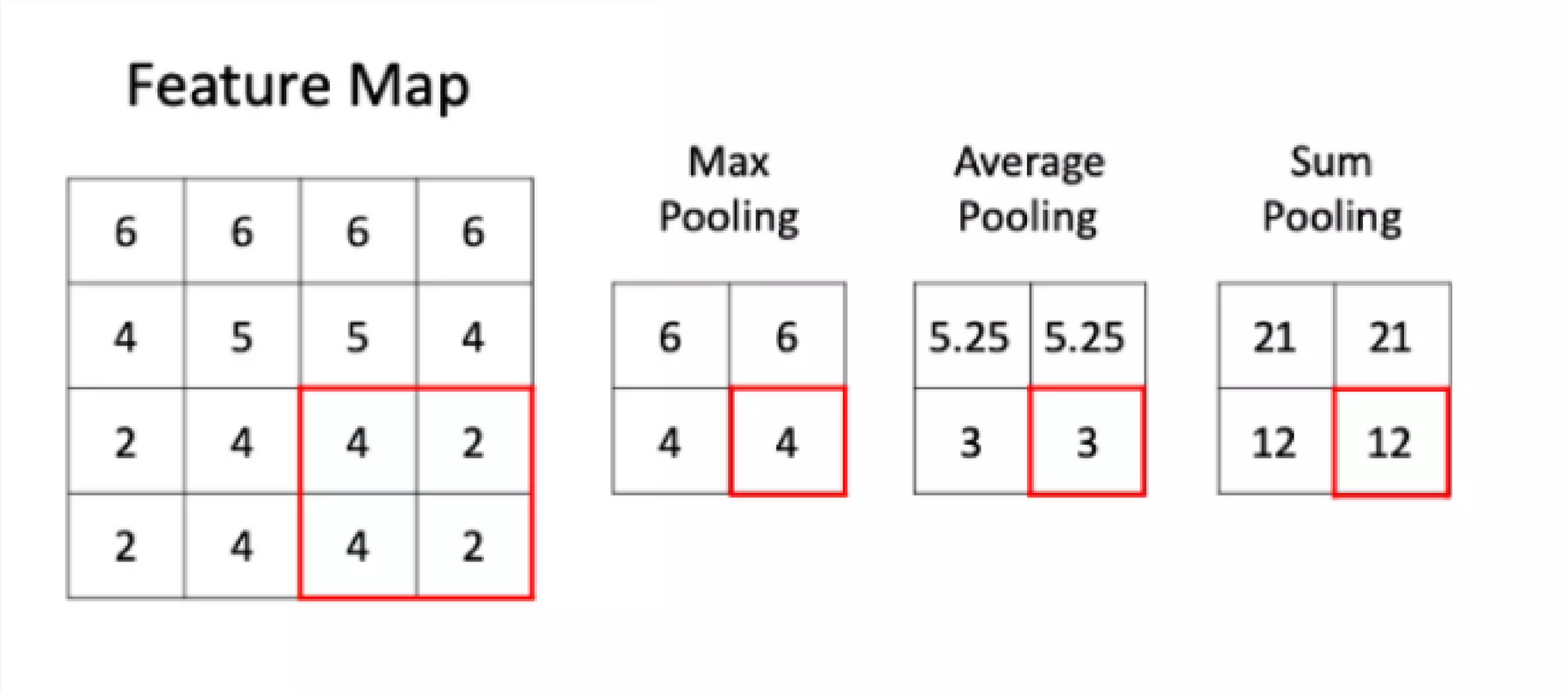
Pooling. Its function is to progressively reduce the image size of the representation to reduce the amount of computation in the network.



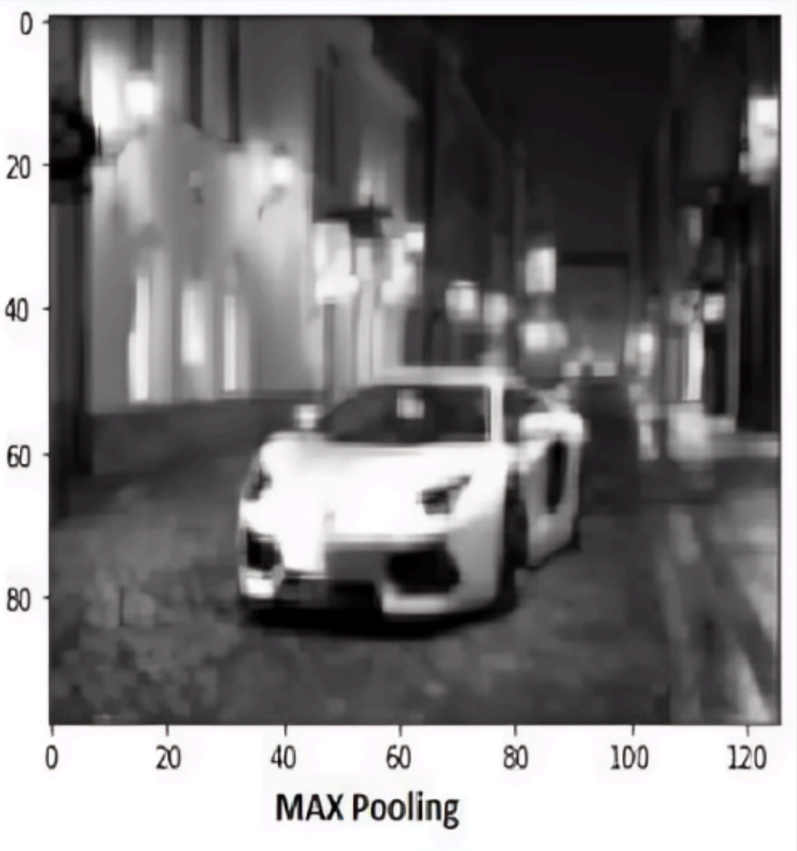
Types

1) Max Pooling
2) Average Pooling
3) Sum Pooling

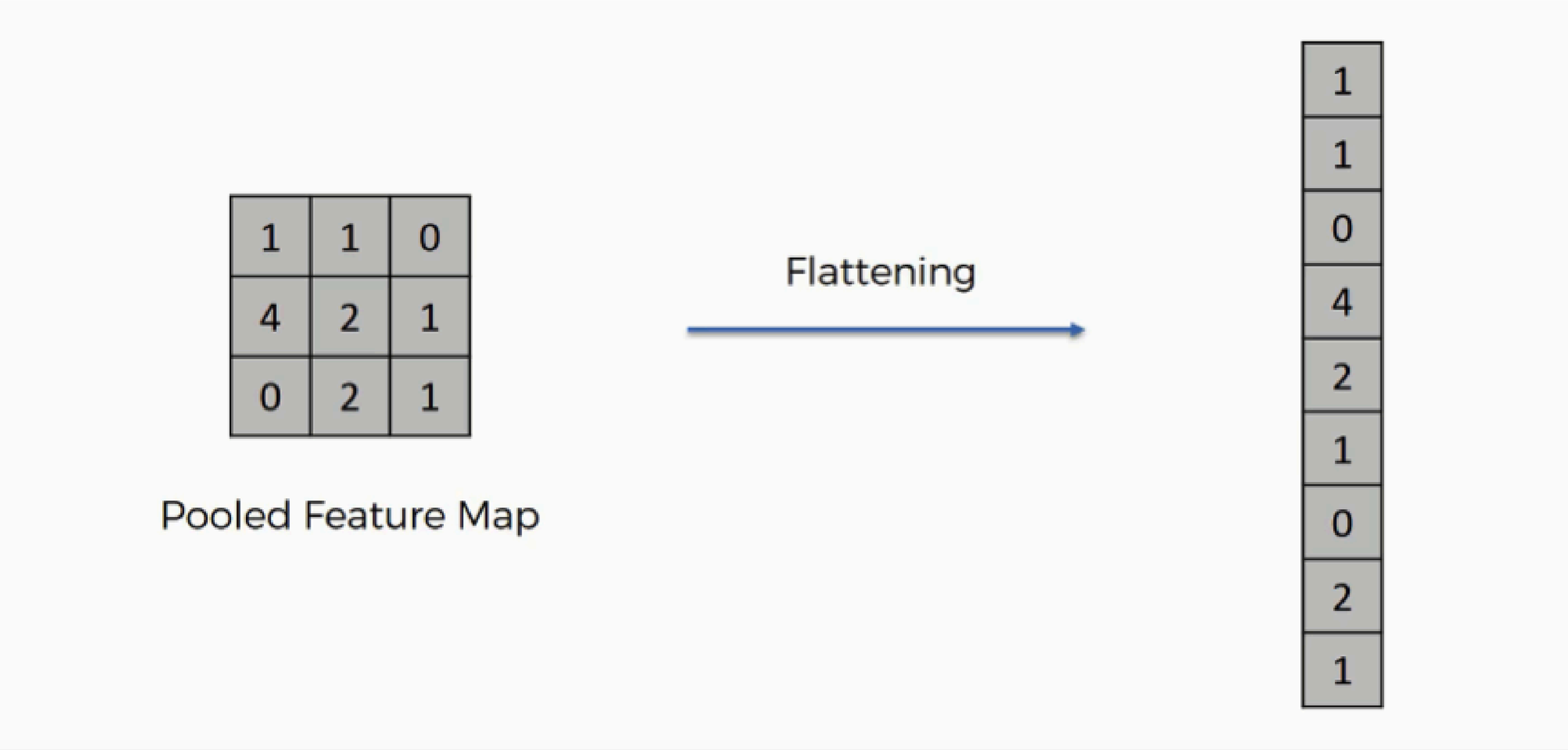
WHAT HAPPENS WHEN YOU MAXPOOL?



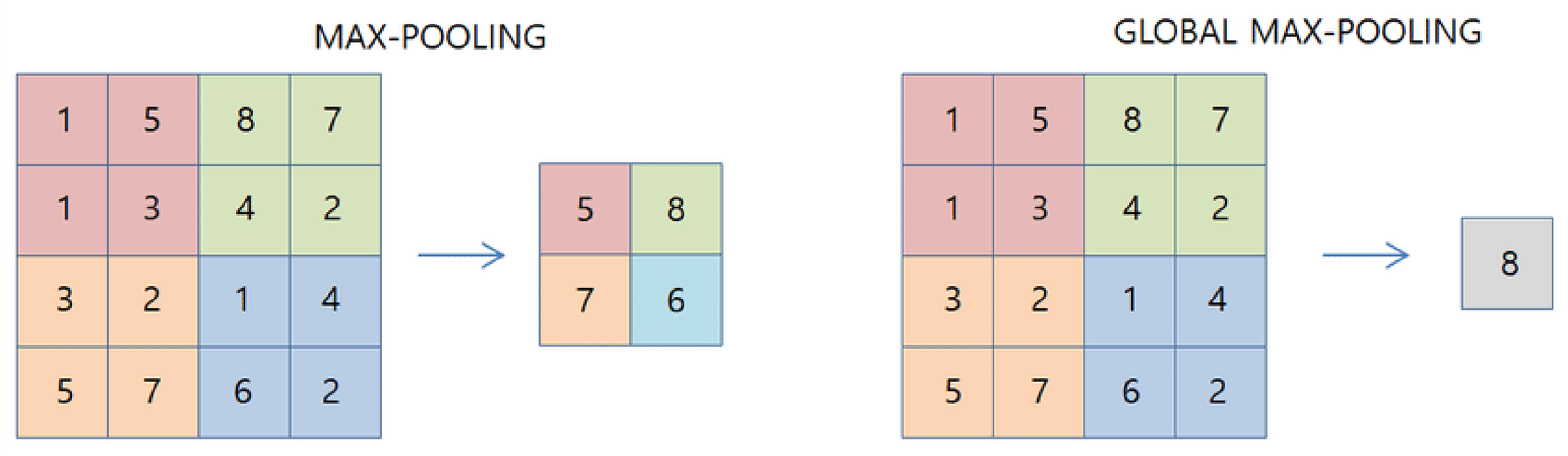
WHAT HAPPENS WHEN YOU MAXPOOL?



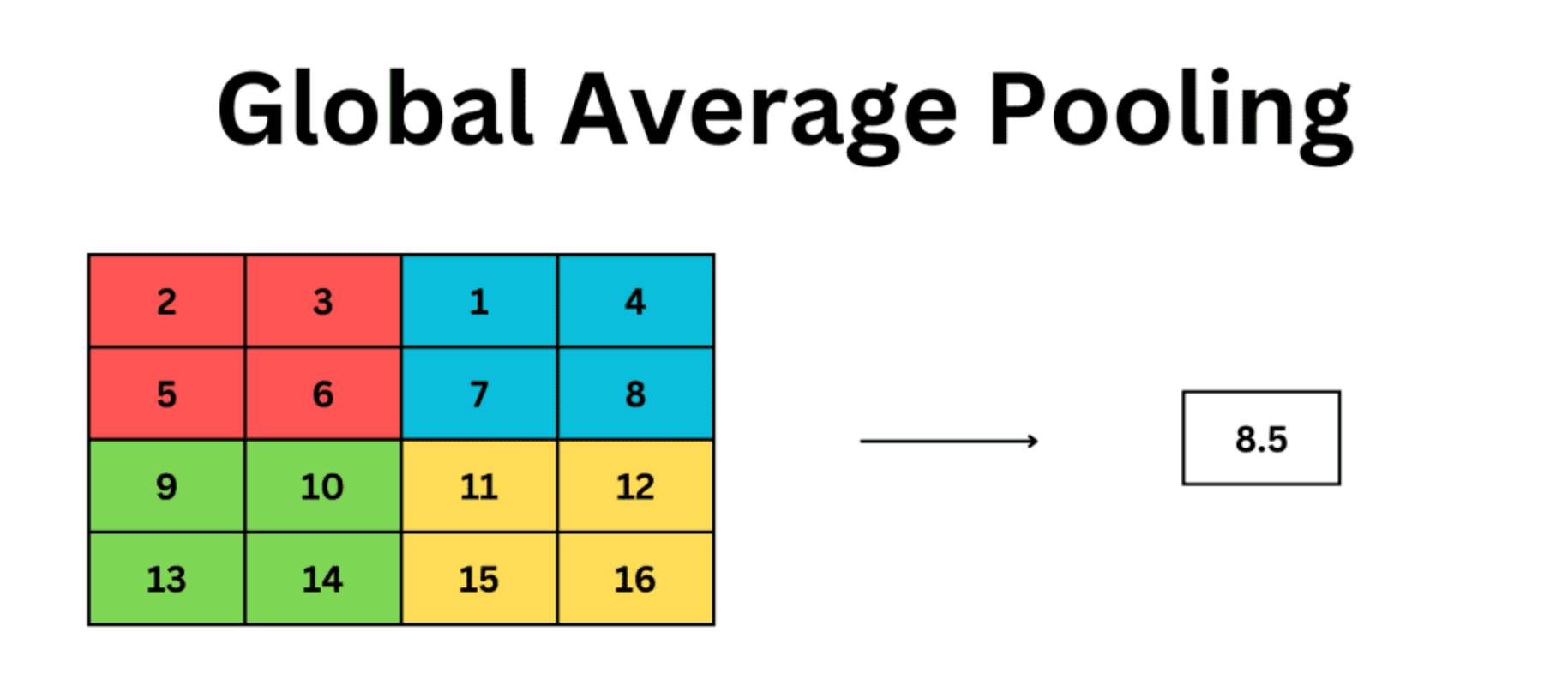
WAYS TO CONVERT FEATURE MAP TO 1D



WAYS TO CONVERT FEATURE MAP TO 1D



WAYS TO CONVERT FEATURE MAP TO 1D



WHERE ARE CNNS PRIMARILY USED

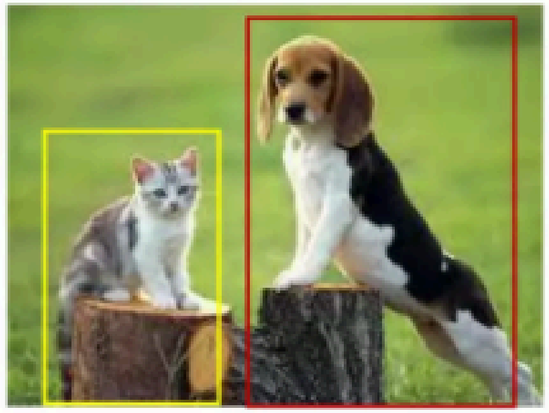
- Image Classification (Disease Detection, Face Recognition, etc.)
- Object Detection (Spot Lesions from Medical Images, Potholes from live surveillance cameras, etc.)
- Semantic Segmentation (Tumor Segmentation from Medical Images)
- Instance Segmentation (Segment individual objects in crowded scenes, Identify Overlapping Nuclei, etc.)
- Image Generation (GANs, Style Transfer, etc.)

WHERE ARE CNNS PRIMARILY USED

Image Classification

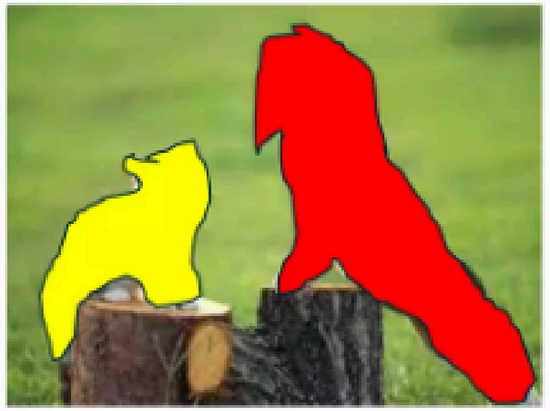
Cat or Dog? Entire Image

Object Detection



Classify Objects with Location

Segmentation

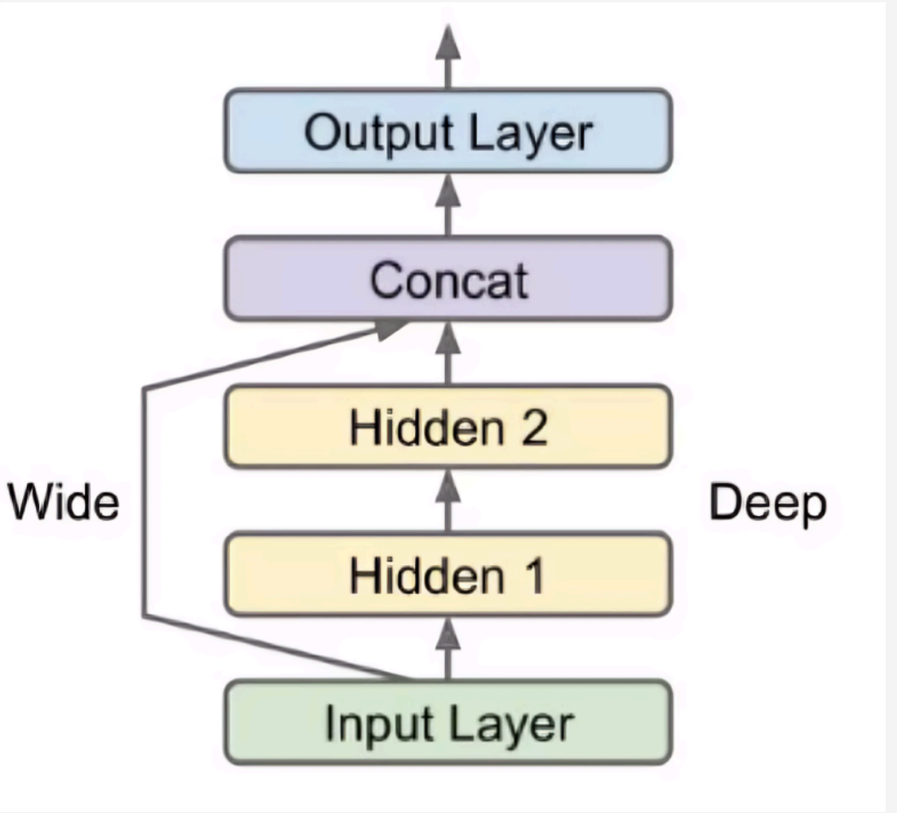


Classify each pixel to a class

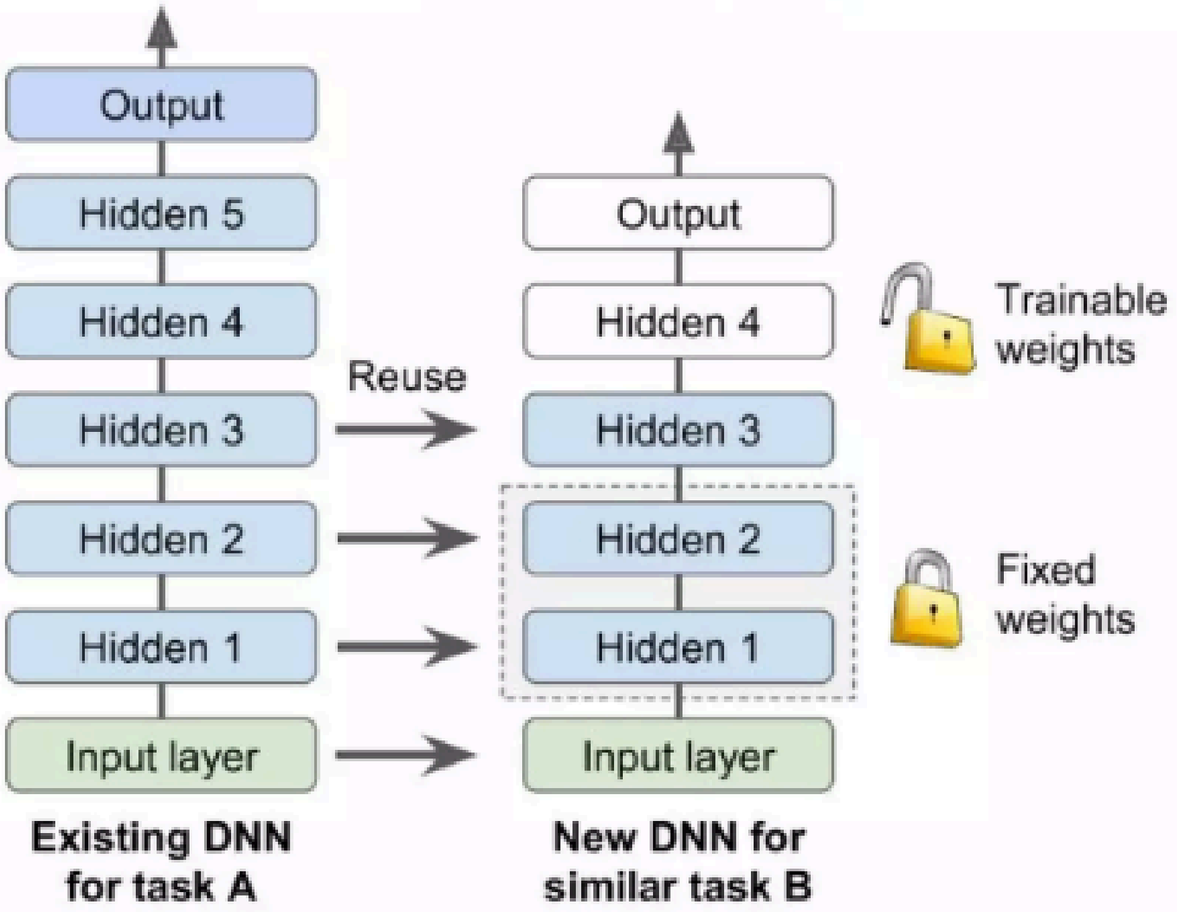
BUILDING COMPLEX MODELS - KERAS FUNCTIONAL API

FUNCTIONAL API

- The functional API is a way to create models that are more flexible than the Sequential API (tf.keras.Sequential)
- It can handle models with non-linear topology, models with shared layers, and models with multiple inputs or outputs.
- Enables us to build neural networks to learn both deep patterns (using the deep path) and simple rules (through the short path).



TRANSFER LEARNING



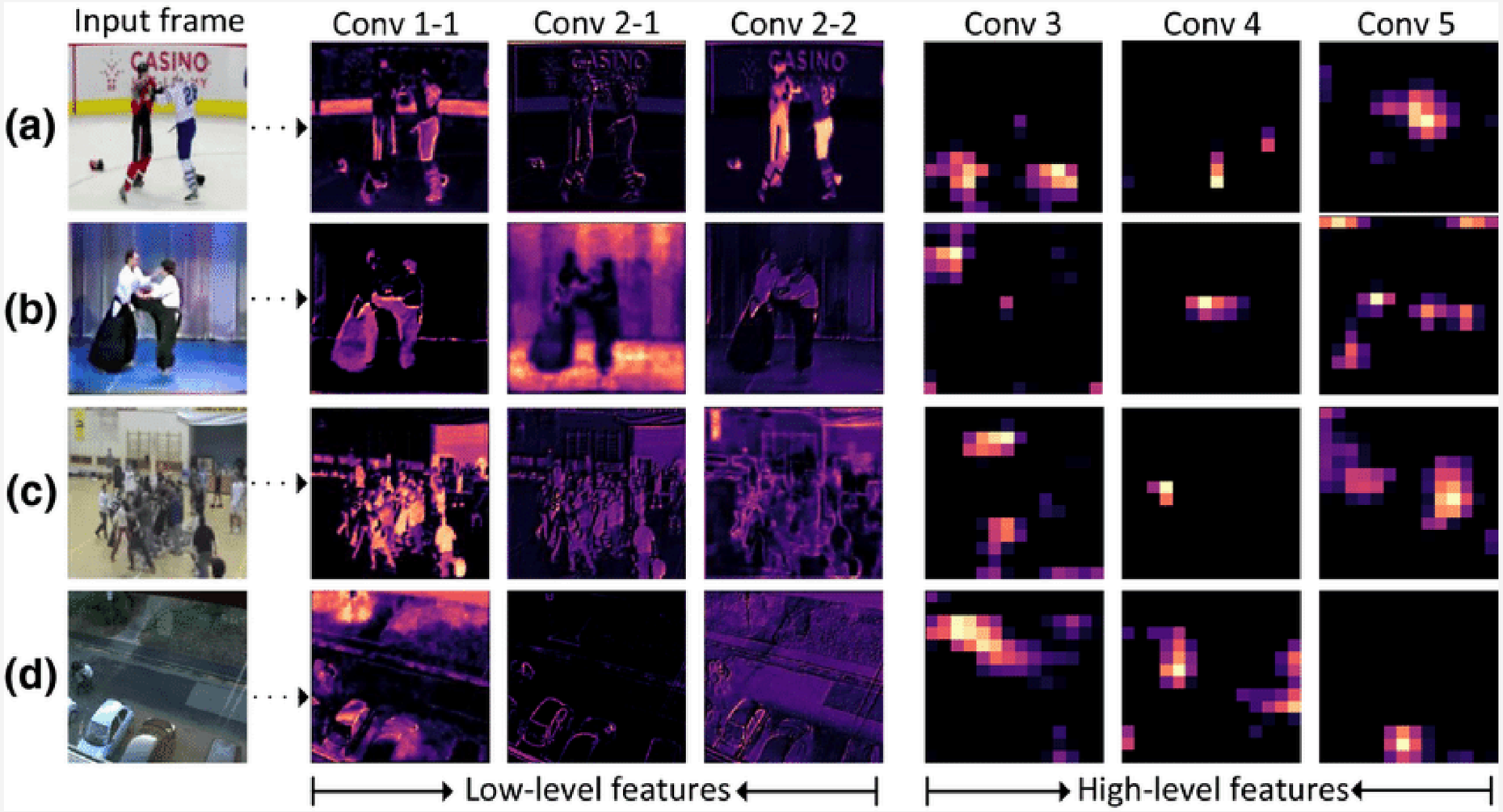
- ***Transfer learning will work best when the inputs have similar low-level features.***

LOW-LEVEL AND HIGH-LEVEL FEATURES

- **Low-Level Features**
  - These features are typically extracted by the early layers of a CNN and include:
    - Edges
    - Textures
    - Basic Geometric Shapes

- **High-Level Features**
  - These features are extracted by deeper layers of the CNN and represent more complex patterns or objects:
    - Object Parts (eyes, nose)
    - Complex Patterns (combination of certain features)

LOW-LEVEL AND HIGH-LEVEL FEATURES



TRANSFER LEARNING MODELS

When To Use Which Model?

(not a rule of thumb, but its good to know the reason behind why each model was designed)

Xception	• Suitable for high-resolution image datasets where the model needs to handle a large number of parameters efficiently while capturing complex features.
Inception	• Ideal when depth perception is crucial, such as when understanding the distance between the lens and the object.
VGG	• Effective when fine-grained details are important and where computational resources are less of a concern due to its deeper architecture and large number of parameters.

TRANSFER LEARNING MODELS

When To Use Which Model?

(not a rule of thumb, but its good to know the reason behind why each model was designed)

ResNet	• Beneficial for tasks where vanishing gradients are an issue: ◦ ResNet - residual connections that help gradients flow through the network. ◦ DenseNet - each layer's output is concatenated with the outputs of all preceding layers.
DenseNet	
EfficientNet	• Optimal for scenarios where computational efficiency and accuracy need to be balanced.
MobileNet	

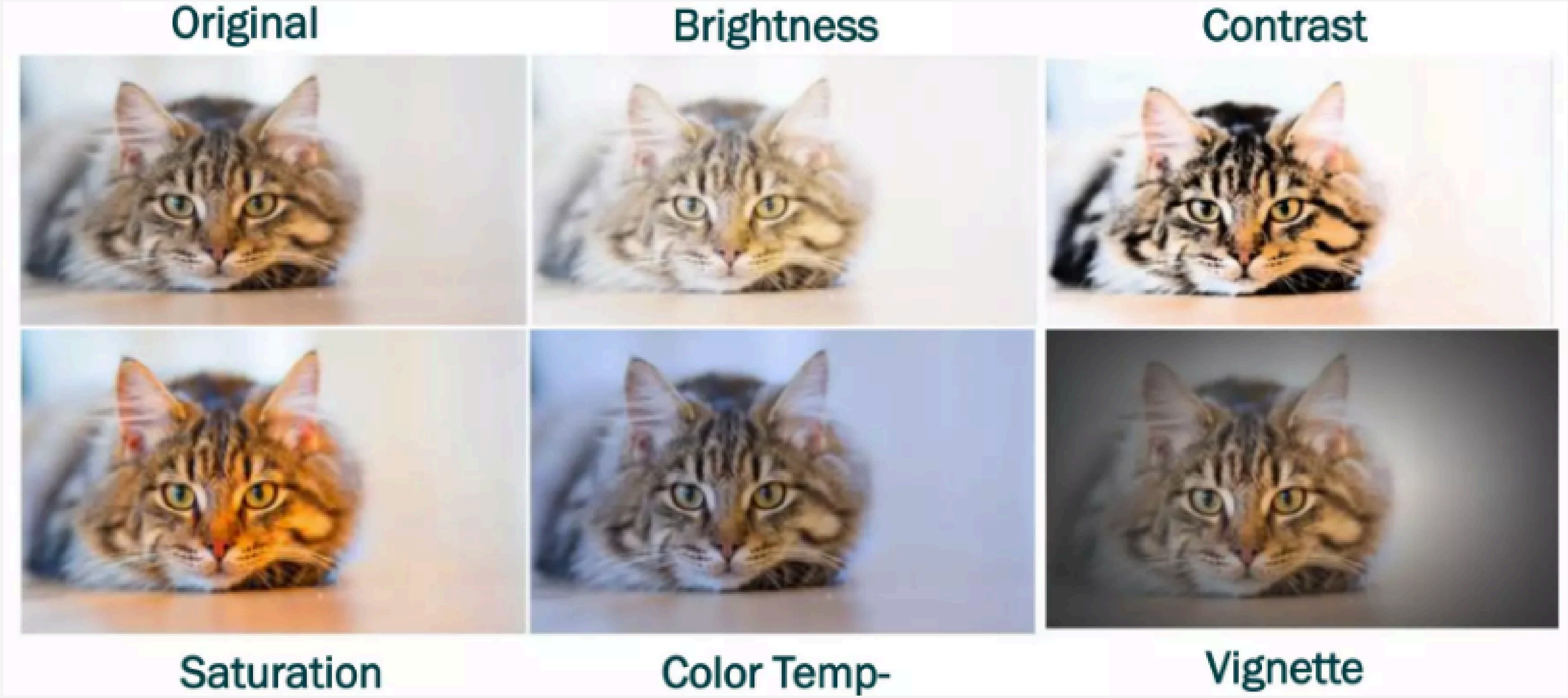
You can find the list of all the available transfer learning models under the TensorFlow Applications website: https://www.tensorflow.org/api_docs/python/tf/keras/applications

DATA AUGMENTATION

- **Enhance model robustness and generalization by artificially expanding the training dataset.**
- **Benefits:**
  - Increase Dataset Size
  - Prevent Overfitting
  - Improve Model Performance

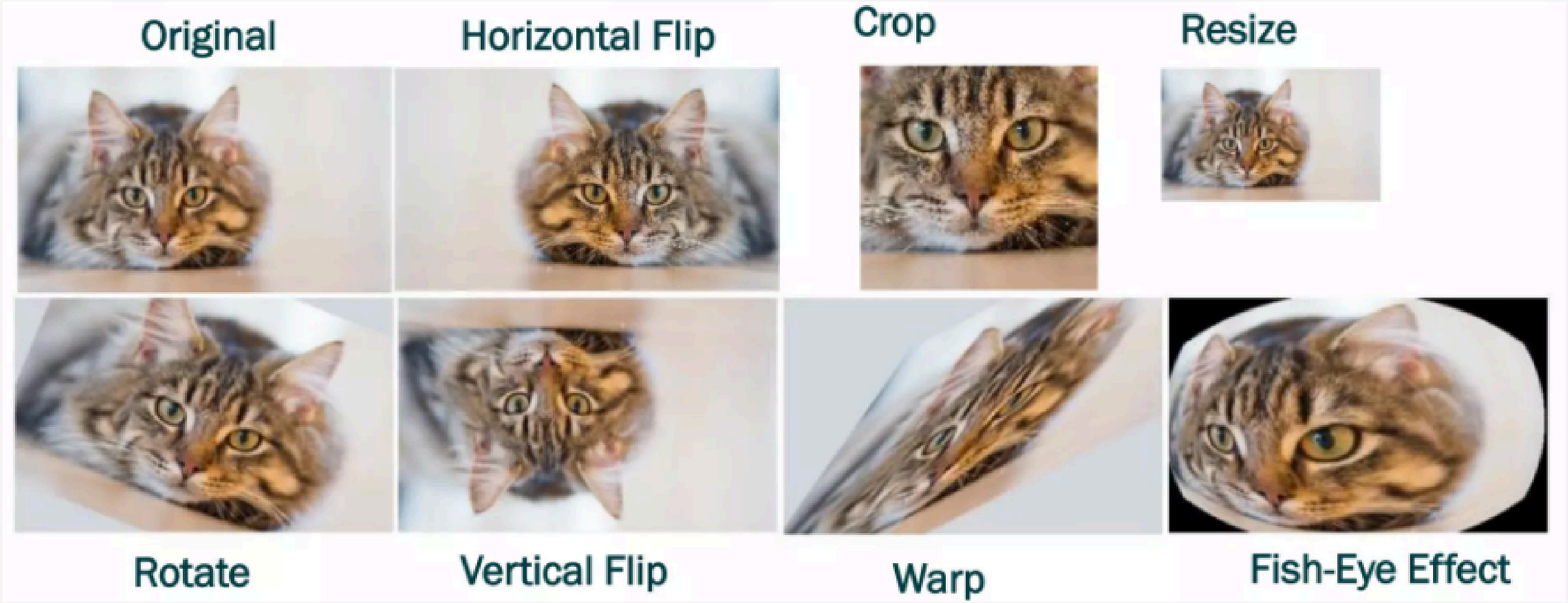
HOW IS IT DONE?

Color and Illumination Examples



HOW IS IT DONE?

Geometric and Displacement Distortion Examples

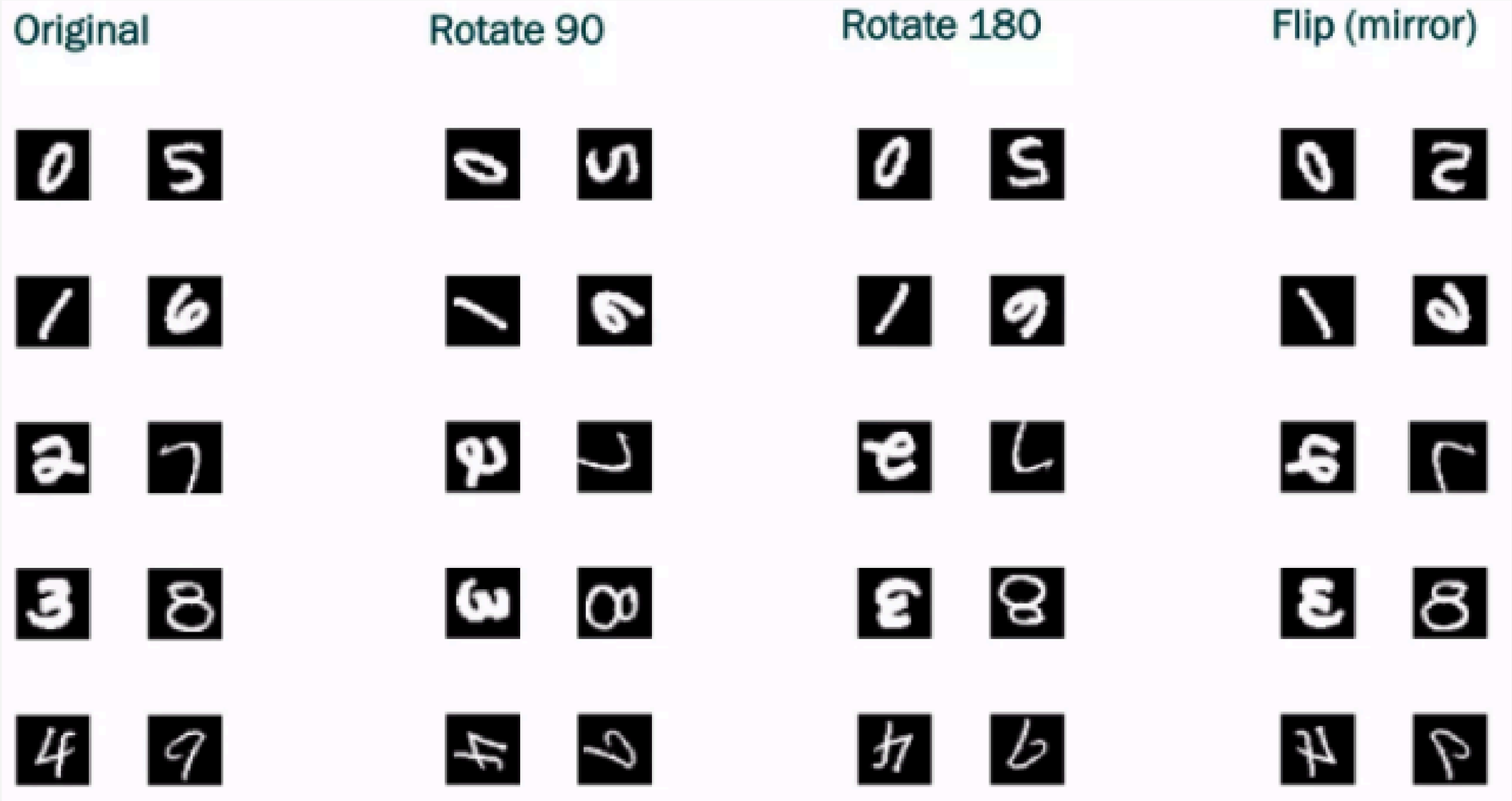


HOW IS IT DONE?

Mixing Augmentations - Disruptive

WHEN SHOULD YOU USE WHICH AUGMENTATION TECHNIQUE?

Not all augmentations apply to all datasets!



HANDS ON

Face Recognition using CNNs:
https://colab.research.google.com/drive/1YbOcNwU1IzLLaDA2z1cgdu-sitrzBGj8?usp=sharing

Ensembling using CNNs:
https://colab.research.google.com/drive/1mjUmNeoPJuibVc5z_Wr5rHLKNUIi0QaS?usp=sharing

CHALLENGE TASK

Task 1: Image Classification with CIFAR-10 Dataset

Train two artificial neural networks (ANNs) on the CIFAR-10 dataset using mini-batch gradient descent. Apply hyperparameter tuning to both models, using different regularization techniques for each. Evaluate the model's performance with visualizations of loss and accuracy, and display with reasoning as to which model performed better.

Task 2: Predicting House Prices with the Boston Housing Dataset

Implement an artificial neural network (ANN) for regression on the Boston Housing dataset, applying minibatch gradient descent, hyperparameter tuning, and various regularization techniques. Assess the model using Mean Squared Error and visualize training progress.
Dataset link: https://www.kaggle.com/code/prasadperera/the-boston-housing-dataset

You can load the datasets via TensorFlow/Kaggle API.

REFERENCES

- https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/
- https://www.geeksforgeeks.org/ml-mini-batch-gradient-descent-with-python/
- https://machinelearningmastery.com/gentle-introduction-mini-batch-gradient-descent-configure-batch-size/
- https://www.geeksforgeeks.org/regularization-in-machine-learning/
- https://theaisummer.com/regularization/
- https://medium.com/data-science-365/how-to-apply-l1-and-l2-regularization-techniques-to-keras-models-da6249d8a469