BEGINNER TO INTERMEDIATE

PYTHON

AND MACHINE LEARNING

WORKSHOP

WORKSHOP 3: BUILDING CLASSIFICATION MODELS WITH OPENVINO

BY JOSE RUIZ

intel. HACKER DOJO

SPONSORED BY

BEFORE WE START...

AGENDA
Introduction to Image Classification and its Applications
A Bit Of History About Deep Learning
Building an Image Classification Model using a Pre-Trained Model
Converting the Model to OpenVINO and Optimizing it for Different Architectures
Deploying the Optimized Model and Measuring its Performance

IMAGE CLASSIFICATION

What is it?

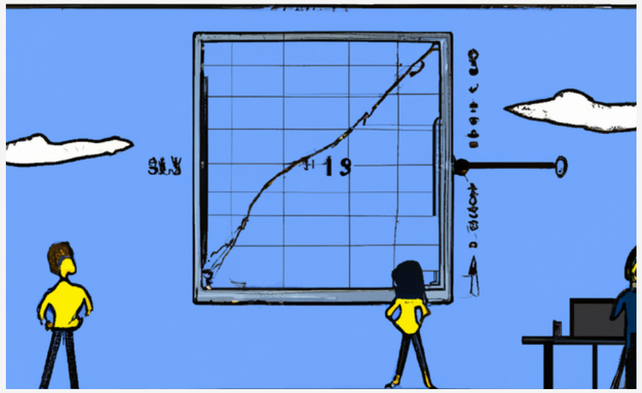
- Image classification is a computer vision task that involves assigning a label or a class to an input image.
- It is widely used in a variety of applications such as object recognition, face recognition, self-driving cars, and medical image analysis.
- Image classification can be done using traditional machine learning algorithms, but deep learning algorithms have shown superior performance in recent years.
- Deep learning models for image classification are typically trained using a large dataset of labeled images and require significant computational resources.

IMAGE CLASSIFICATION

Where is it used?

- Medical Diagnosis: Image classification is used in medical diagnosis to identify diseases, tumors, and other anomalies in medical images like X-rays, MRI scans, and CT scans.
- Agriculture: Image classification is used in agriculture to identify crop health and detect pests or diseases in crops.
- Artificial Intelligence: Image classification is used as a building block for developing more complex artificial intelligence systems that can perform tasks like facial recognition and speech recognition.

IMAGE CLASSIFICATION



What are the challenges?

- Need for very large and diverse datasets.
  - Overfitting: When a model learns the dataset it was trained on and results in poor generalization.
  - Variations in image properties: Diversity in scale, orientation, illumination, color and background on the dataset is needed for good generalization.
  - Class Imbalance: Imbalanced datasets can lead to a biased model.
- Model Complexity.
  - As models become more complex the need for computing resources to train and run increases as well.
- Ethical Considerations.
  - It is important to consider bias and privacy concerns when training a deploying a model.

A BIT OF HISTORY

What Brough Neural Networks back to life?

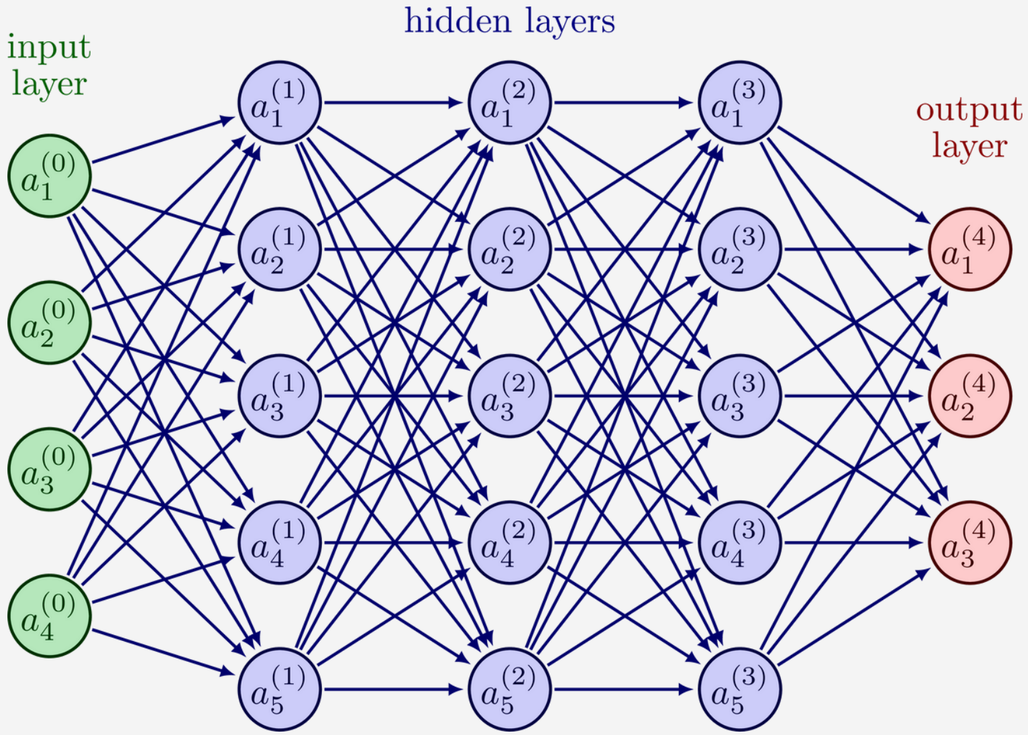
- In 1998, **LeNet-5** was introduced by Yann LeCun et al. as the first successful convolutional neural network for handwritten digit recognition.
- In 2012, **AlexNet** was introduced by Alex Krizhevsky et al. AlexNet was composed of eight layers, including five convolutional layers and three fully connected layers. It was the first deep learning model to incorporate ReLU activation functions and dropout regularization.
- In 2014, the **VGG** network was introduced by Simonyan and Zisserman. VGG was designed to have a simple architecture, consisting of multiple layers with a small receptive field and 3x3 convolutions, followed by max pooling layers. VGG was also notable for its depth, with up to 19 layers.
- In 2015, the **Inception** network (GoogLeNet) was introduced by Szegedy et al. using a series of parallel convolutional layers with different filter sizes and pooling operations to address
- In 2015, **ResNet** (Residual Network) was introduced by He et al. was designed to address the problem of vanishing gradients by introducing residual connections, which allowed for the training of very deep neural networks (up to 152 layers). ResNet also introduced using batch normalization to improve model performance and reduce overfitting.

CONVOLUTIONAL LAYERS

Why are they important?

- Convolutional layers are a fundamental component of many deep learning models used for image classification.
- They enable the model to learn relevant features from the images by applying filters to the input data.
- These filters identify specific patterns in the input images, such as edges, corners, and textures, and create feature maps as output.
- Feature maps are then passed through additional convolutional layers, enabling the model to learn increasingly complex features.
- By using convolutional layers, the model can identify features that are useful for classification, regardless of their position in the input image.
- This enables the model to classify images even when the objects of interest are not located in the center of the image or have varying sizes or orientations.

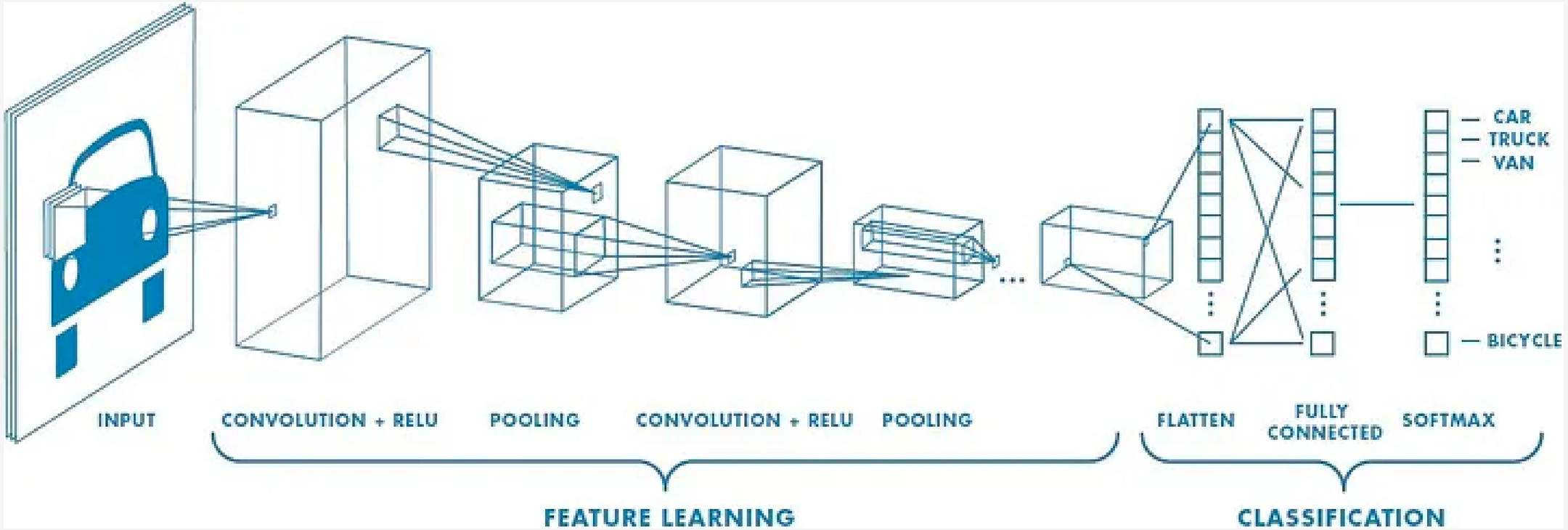
FULLY CONNECTED NETWORK



Why not just multiple layers of fully connected neurons?

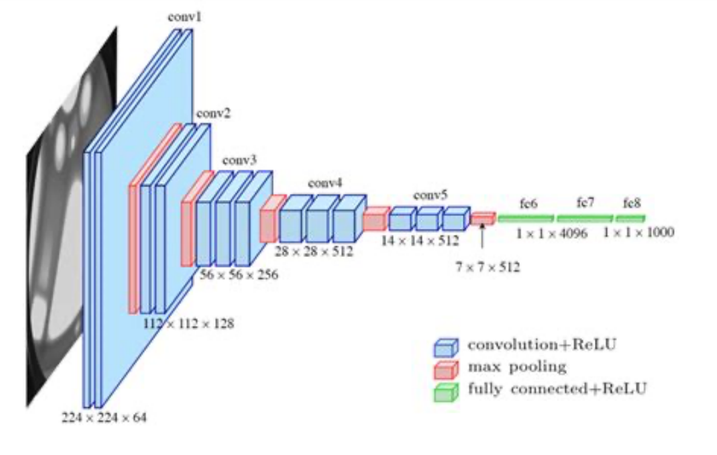
- The gradient descent algorithm is used to train neural networks by updating their weights.
- In deep neural networks, the gradients can become too large (exploding gradient) or too small (vanishing gradient) during backpropagation.
- This can make training difficult, slow or even impossible, as the network may fail to learn.

CONVOLUTIONAL NEURAL NETOWRK



- CNNs can better handle image data than fully connected networks because they take advantage of the spatial structure of images by using convolutional layers. This allows them to learn important features from smaller subsets of the image, reducing the number of parameters and improving performance.

BUILDING AN IMAGE CLASSIFICATION MODEL



What is a pre-trained model?

- A pre-trained model is a pre-built neural network that has been trained on a large dataset, typically on a specific image classification task.
- By using a pre-trained model, you can leverage the knowledge and experience gained by others who have spent countless hours training a model on a similar problem.
- Pre-trained models can be used for a variety of image classification tasks and can be easily fine-tuned for your specific use case.
- The most popular pre-trained models used for image classification include VGG, ResNet, Inception, and MobileNet.

IMPORTANT RESOURCES

Some very useful repositories:

- https://github.com/openvinotoolkit/openvino
- https://github.com/openvinotoolkit/openvino_notebooks
- https://github.com/openvinotoolkit/open_model_zoo

Workshop 3 lab:

- Example 301: Tensorflow Training (directly from link 2)
- To run:
  - Clone or update this repo:
  - **`git clone https://github.com/hackerdojo/python-ml`**
  - `(or just` **`git pull`** `)`
  - `cd python-ml/workshop_3/scripts`
  - `chmod +x run_docker.sh`
  - `./run_docker.sh`

THANKYOU!

JOSE RUIZ

intel HACKER DOJO

SPONSORED BY