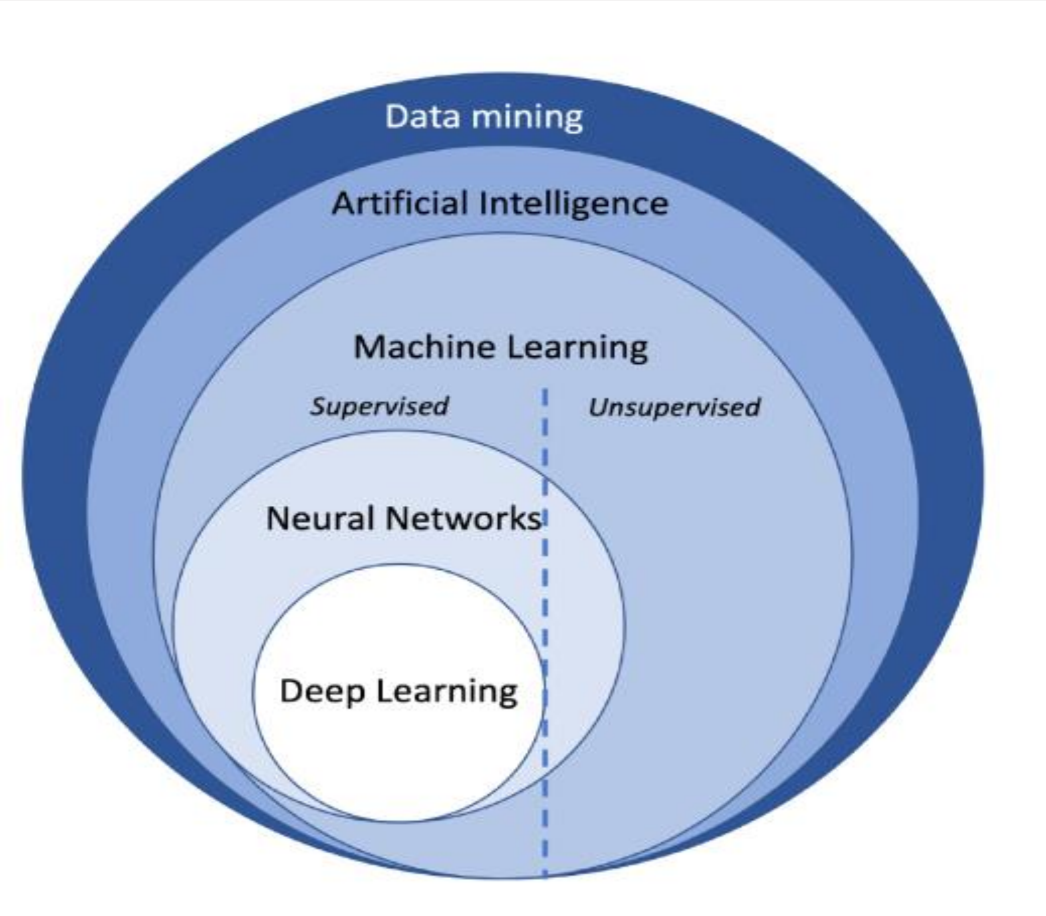
Topic Covered

- Introduction to Artificial Intelligence and Machine Learning
- Deep Learning
- Various Deep Learning Architectures
- Feedforward Neural Networks (FNNs)
- Convolutional Neural Networks (CNNs)
- Training and Optimization of CNNs
- CNN Applications in Computer Vision
- Recurrent Neural Networks (RNNs)
- Generative Adversarial Networks (GANs)
- Transfer Learning

Artificial Intelligence (AI)

- Artificial intelligence (AI) is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence.



Machine Learning (ML)

- Machine Learning is a subset of artificial intelligence (AI) that involves the use of algorithms and statistical models to enable computers to perform a task without explicit instructions.
- ML focuses on the development of computer programs or models that can access data and use it to learn for themselves. Some of the possible tasks that can be possible using ML:

1. **Classification:** Assignment of a given instance to two or more previously labelled classes
2. **Regression:** Classification using a continuous output rather than discrete labels
3. **Clustering:** Grouping of inputs without prior given classes
4. **Density Estimation:** Outputs the spatial distribution of inputs
5. **Dimensionality Reduction:** Maps instances to a lower-dimensional space

Types of Machine Learning (ML)

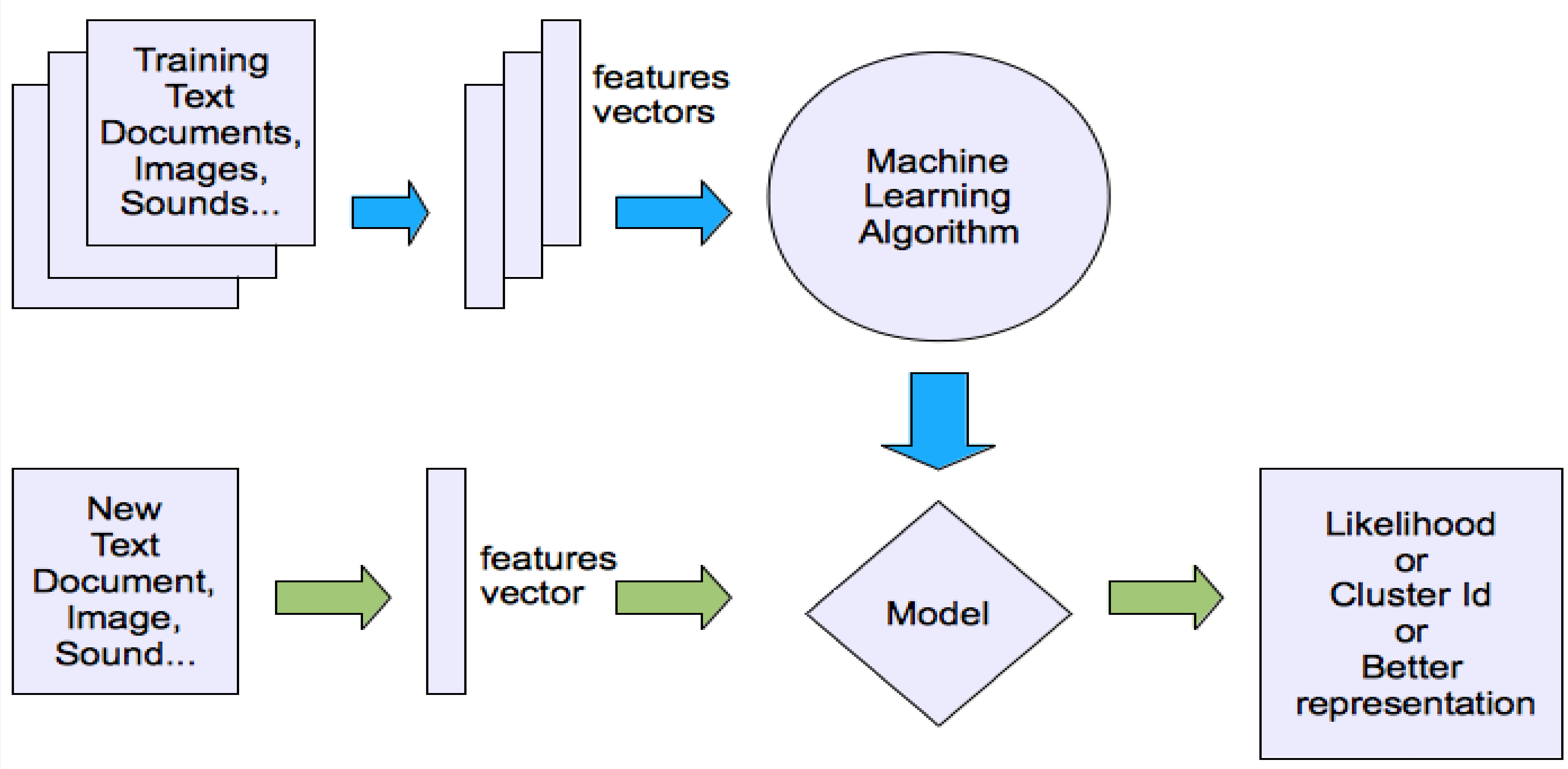
- Unsupervised Learning
- Supervised Learning
- Reinforcement Learning
- Semi Supervised Learning
- Transfer Learning

Unsupervised Learning

- Unsupervised learning is a type of machine learning that involves finding patterns in data without using pre-existing labels. In unsupervised learning, algorithms attempt to identify underlying structures or relationships in the data based on inherent similarities, patterns, or differences without external guidance.
- **Key Aspects of Unsupervised Learning:**

1. **No Labels:** The primary distinction from supervised learning is the absence of target labels in the training data. The model isn't trying to predict an outcome; rather, it's trying to learn the underlying structure.
2. **Exploratory:** Unsupervised learning is often used for exploratory data analysis since it can reveal hidden patterns or groupings that might not be evident or known a priori.

Working Flow of Unsupervised Learning



Unsupervised Learning Techniques

- **Clustering**: Grouping data points that are similar to each other.
  - **K-Means:** Partitions data into K distinct, non-overlapping clusters.
  - **Hierarchical Clustering:** Builds a tree of nested clusters.
  - **DBSCAN:** Groups data into dense clusters and identifies points that don't belong to any.
- **Dimensionality Reduction:** Reducing the number of features in data while preserving its essential characteristics.
  - **Principal Component Analysis (PCA):** Transforms data into a new coordinate system, capturing the most significant variances.
  - **t-Distributed Stochastic Neighbor Embedding (t-SNE):** Useful especially for visualizing high-dimensional data in 2D or 3D space.
  - **Autoencoders:** Neural networks that are trained to replicate their input data during output, thus learning a compressed representation in the process.

Unsupervised Learning

- **Advantages of Unsupervised Learning:**
  - **Handling Unknown Data:** Can work with data even when the outcomes or groupings aren't known ahead of time.
  - **Feature Learning:** Can help in understanding significant features or patterns in the data.
  - **Scalability:** Can often handle large datasets, given there's no need for labeling.
- **Challenges in Unsupervised Learning:**
  - **Interpretability:** Results (like clusters) might not always be straightforward to interpret.
  - **Validation:** It's harder to measure the accuracy of a model without labeled data, making it challenging to determine its effectiveness.
  - **Sensitivity:** Some algorithms, like K-means, can be sensitive to initialization, outliers, or the scaling of data.

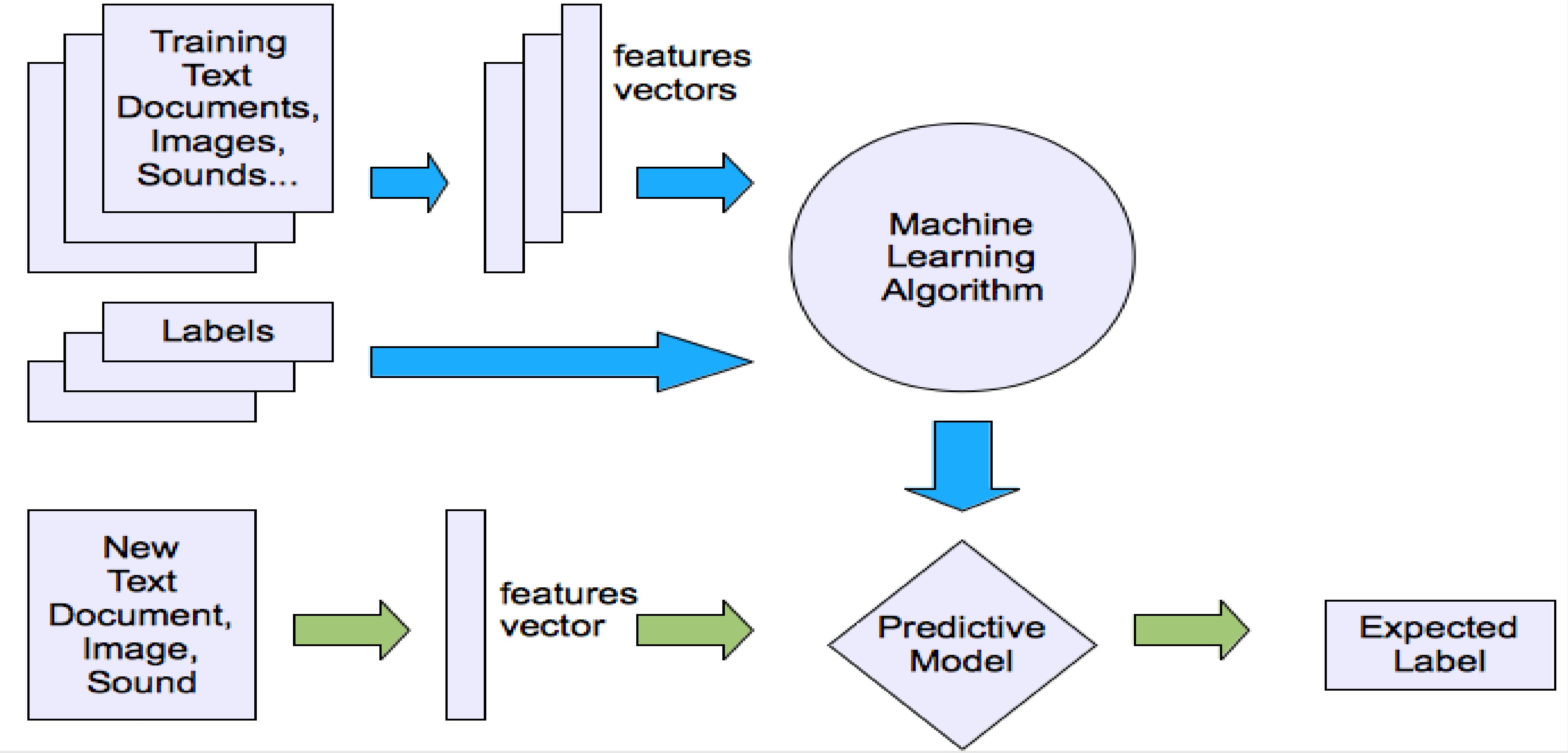
Applications of Unsupervised Learning

- **Market Segmentation:** Businesses can group customers based on purchasing behavior.
- **Anomaly Detection:** Identifying rare events, like fraud detection in credit card transactions.
- **Recommendation Systems:** Recommending items based on similarities.
- **Topic Modeling:** Identifying topics in a large text corpus.
- **Feature Learning:** Can help in understanding significant features or patterns in the data.

Supervised Learning

- Supervised learning is a type of machine learning where the model is trained on labeled data. The "supervision" consists of the model making predictions and then being corrected by the actual label whenever it's wrong.
- **Key Aspects of Supervised Learning:**
  - **Labeled Data:** Data is considered 'labeled' if each example comes with a corresponding output (or "label"). For instance, for an image classification task, labeled data would consist of images paired with their correct categories.
  - **Learning from Feedback:** The algorithm iteratively makes predictions on the training data and is corrected by the training labels, allowing it to learn over time.
  - **Goal:** The primary objective is to learn a mapping from inputs (features) to outputs (labels) based on the training data, so as to make accurate predictions on unseen data.

Working Flow of Supervised Learning



Supervised Learning

- **Types of Supervised Learning Problems:**

1. **Classification:** The output variable is a category, e.g., "spam" or "not spam", "cat", "dog", or "horse".
2. **Regression:** The output variable is a real or continuous value, e.g., "weight" or "price".

Supervised Learning

- **Key Algorithms & Techniques:**
  - **Linear Regression:** Used for regression problems, it finds the best linear relationship between input and output.
  - **Logistic Regression:** Despite its name, it's used for binary classification problems.
  - **Decision Trees:** Splits data into subsets based on the value of input features.
  - **Random Forest:** A collection (or "forest") of decision trees that aggregates their outputs.
  - **Support Vector Machines (SVM):** Finds the "best" margin (or hyperplane) that separates classes.
  - **Neural Networks:** Comprised of layers of interconnected nodes (or "neurons"). Especially powerful for complex tasks like image and voice recognition.
  - **Naive Bayes:** Based on Bayes' theorem, it's particularly suited for high-dimensional datasets.
  - **K-Nearest Neighbors (KNN):** Classifies a data point based on how its neighbors are classified.

Applications of Supervised Learning

1. **Medical Diagnoses:** Predicting diseases based on symptoms or medical test results.
2. **Financial Forecasting:** Predicting stock prices or currency exchange rates.
3. **Email Filtering:** Identifying spam or important emails.
4. **Image Recognition:** Identifying and categorizing objects within images.

Reinforcement Learning

- Reinforcement learning (RL) is a type of machine learning where an agent learns to behave in an environment by taking certain actions and receiving rewards or penalties in return. It's inspired by behavioral psychology and involves agents who take actions in an environment to maximize some notion of cumulative reward.
- **Key Components of RL:**
  - **Agent:** The decision-maker or learner.
  - **Environment:** Everything the agent interacts with.
  - **State:** A snapshot of the environment at a certain point in time.
  - **Action:** The moves made by the agent.
  - **Reward:** Feedback from the environment. It can be positive (a reward) or negative (a penalty).

Reinforcement Learning

- **The RL Process:**
  - The agent observes the state of the environment.
  - Based on this observation, the agent takes an action.
  - The environment transitions to a new state based on the action taken.
  - The agent receives a reward or penalty based on the action's outcome.

Reinforcement Learning

- **Learning Methods in RL:**
  - **Model-free methods:** The agent learns the value function or policy directly from interaction with the environment without knowing its dynamics. Examples include Q-learning and Deep Q Network (DQN).
  - **Model-based methods:** The agent first learns a model of the environment and then uses this model to compute the value function or policy.
- **Applications of RL:**
  - **Gaming:** AlphaGo by DeepMind used RL to defeat the world champion in the game of Go.
  - **Robotics:** For tasks like walking and jumping.
  - **Finance:** For portfolio management and trading strategies.
  - **Healthcare:** For treatment planning.
  - **Autonomous Vehicles:** For driving strategy optimization.

Semi-Supervised Learning

- Semi-supervised learning is a machine learning paradigm where the model is trained using a combination of a small amount of labeled data and a large amount of unlabeled data. The underlying assumption is that the distribution of the labeled data provides some information about the distribution of the unlabeled data.

1. **Labeled Data is Expensive.**
2. **Unlabeled Data is Plentiful.**
3. **Improving Learning Accuracy.**

Techniques in Semi-Supervised Learning

1. **Self-training:** The model is initially trained with the labeled data. The model then predicts labels for the unlabeled data.
2. **Multi-view Learning:** Assumes that we have multiple independent views of the data. Each view can provide different, complementary information about the underlying structure.
3. **Pseudo-labelling:** Train on labeled data, predict on unlabeled data. Mix pseudo-labeled data with true labeled data and retrain.
4. **Consistency Training:** The model is encouraged to be consistent between its predictions on perturbed and unperturbed versions of the unlabeled data.
5. **Graph-based Methods:** Constructs a graph where nodes are data points (both labeled and unlabeled) and edges represent similarity between data points.
6. **Generative Models:** Models like Variational Autoencoders (VAEs) or Generative Adversarial Networks (GANs) can be used in a semi-supervised setting.

Semi-Supervised Learning

- **Challenges in Semi-Supervised Learning:**
  - **Assumption Risks:** If the underlying assumptions (like the distribution of labeled data representing the distribution of unlabeled data) do not hold, SSL can give worse performance than supervised learning.
  - **Quality of Pseudo-labels:** Incorrect pseudo-labels can harm the model's performance.
  - **Algorithm Complexity:** Some SSL methods can be computationally intensive or complex.
- **Applications of SSL:**
  - **Text Classification:** When you have a few labeled documents and many unlabeled ones.
  - **Image Classification:** When labeling images is costly, and you have access to a lot of unlabeled images.
  - **Bioinformatics:** E.g., for predicting protein functions where labeled data is scarce.

Deep Learning (DL)

- Deep learning is a subfield of AI that focuses on training neural networks to make intelligent decisions.
- Deep learning models can automatically learn and improve from experience, without explicit programming.
- Deep learning architectures, such as deep neural networks, use multiple layers to process complex patterns and large datasets.
- Deep learning is used in various applications, including image recognition, natural language processing, and speech recognition.
- Deep learning models achieve high accuracy and can perform complex tasks compared to traditional machine learning methods.

Key Components of Deep Learning

- **Neural Networks:** Composed of layers of interconnected nodes (neurons). Each connection has a weight, and each neuron processes input data, applies an activation function, and passes it forward.
- **Layers:**
  - **Input Layer:** Takes in the data features.
  - **Hidden Layers:** Intermediate layers between input and output, where the actual processing happens.
  - **Output Layer:** Produces the final prediction or classification.
- **Activation Functions:** Introduce non-linearities to the model. Common examples include ReLU (Rectified Linear Unit), Sigmoid, and Tanh.

Advantages of Deep Learning

- **Hierarchical Feature Learning:** Can automatically learn features from raw data in a hierarchical manner. Early layers might learn simple patterns, while deeper layers learn more complex features.
- **End-to-End Learning:** Can directly map inputs to outputs without needing explicit feature engineering.
- **Versatility:** Has achieved state-of-the-art results in many domains, including image recognition, natural language processing, and game playing.

Challenges of Deep Learning

- **Data Demand:** Typically requires vast amounts of labeled data to train effectively.
- **Computational Intensity:** Requires high computational power, often necessitating GPUs or TPUs.
- **Interpretability:** Deep models, with their numerous parameters, can act as "black boxes", making it hard to understand their decision-making process.
- **Overfitting:** Due to their complexity, deep models can easily be overfit to the training data, which may reduce their generalization ability on unseen data.

Applications of Deep Learning

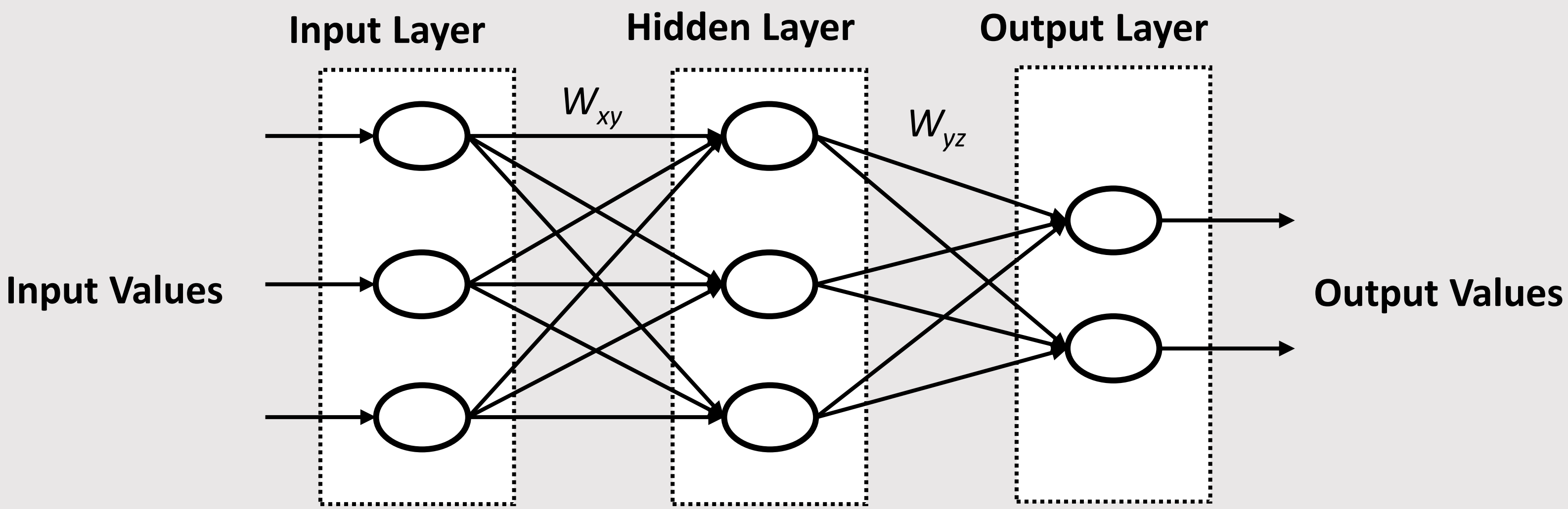
- Deep learning is important because it has revolutionized various fields, including computer vision, natural language processing, and speech recognition.
- In computer vision, deep learning has significantly improved image and video analysis tasks such as object detection, image classification, and image segmentation.
- Deep learning has also been extensively used in natural language processing tasks like sentiment analysis, machine translation, and question answering systems.
- Deep learning has played a crucial role in advancing speech recognition systems, enabling accurate transcription, voice assistants, and voice-controlled devices.
- Deep learning has also found applications in healthcare, finance, autonomous vehicles, fraud detection, recommendation systems, and many other domains, making it a powerful tool for solving complex problems in different industries.

Various Deep Learning (DL) Architectures

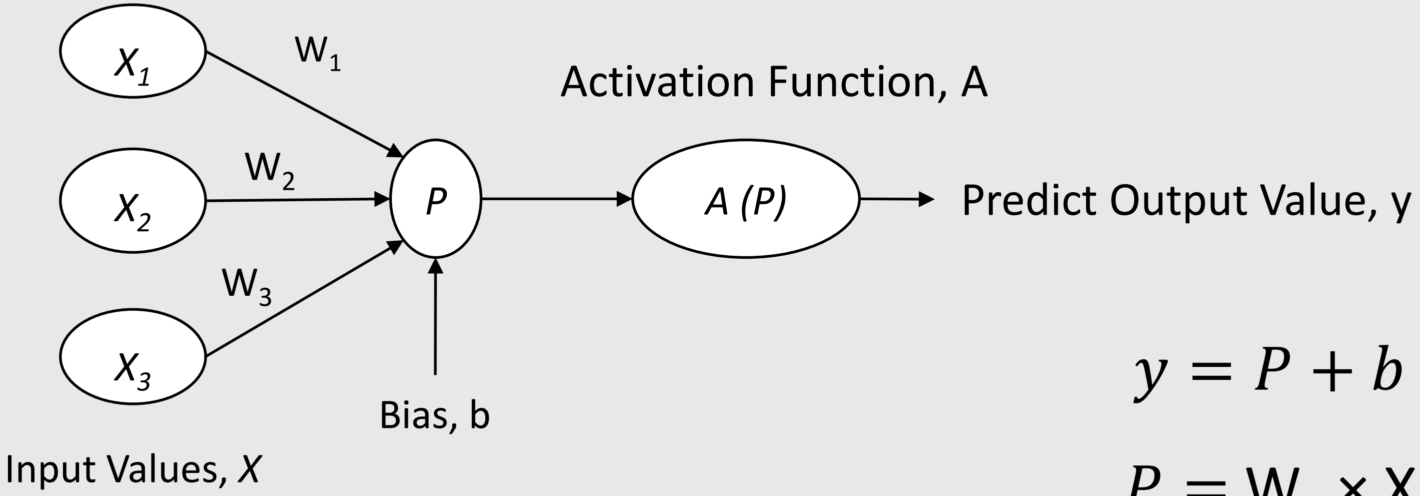
- **Feedforward Neural Networks (FNNs)**
- **Convolutional Neural Networks (CNNs)**
- **Recurrent Neural Networks (RNNs)**
- **Generative Adversarial Networks (GANs)**

Feedforward Neural Networks (FNNs)

- Also known as multi-layer perceptron (MLP).
- Consists of an input layer, one or more hidden layers, and an output layer.
- Information flows only in one direction, from input to output.
- Widely used for tasks like image classification, pattern recognition, and regression.



Feedforward Neural Networks (FNNs)



$$y = P + b$$

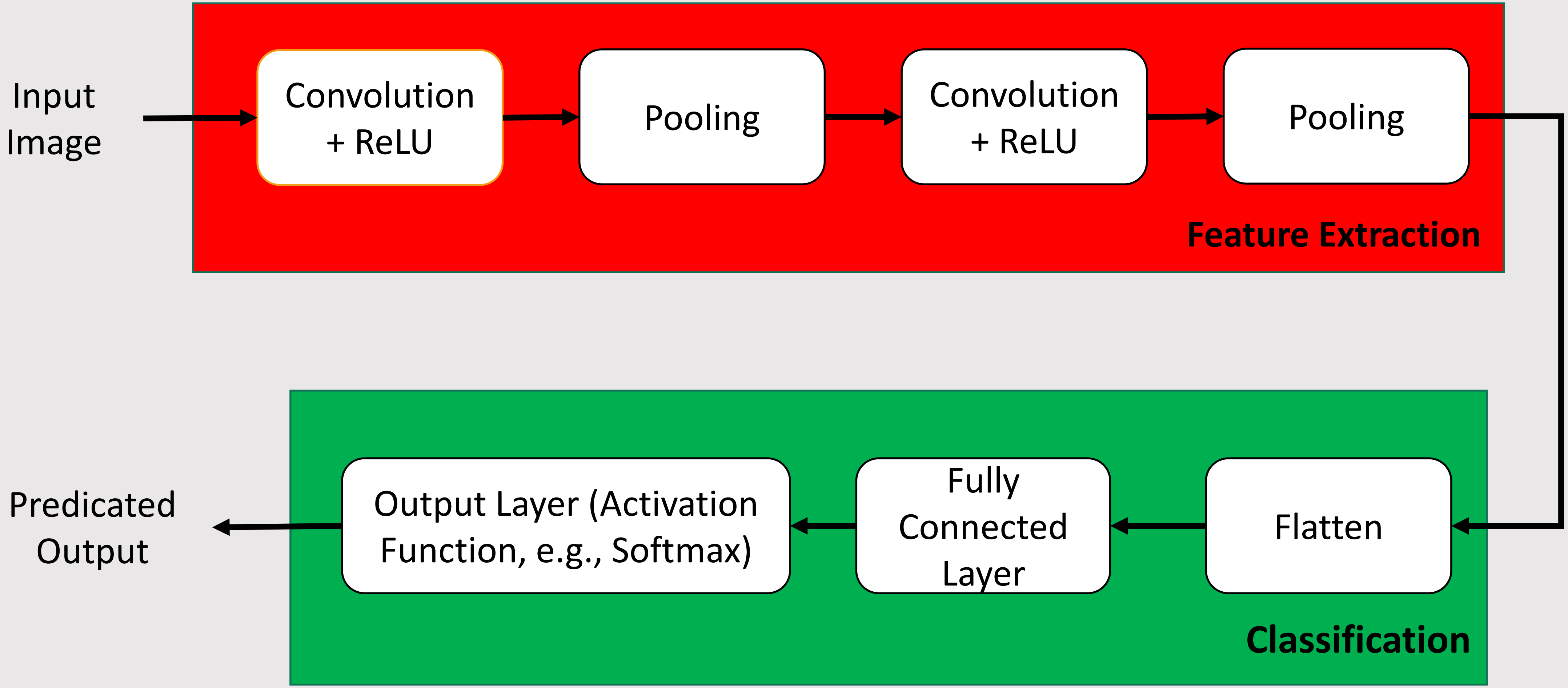
$$P = W_1 \times X_1 + W_2 \times X_2 + W_3 \times X_3$$

An activation function is a function that is added into a neural network in order to help the network learn complex patterns in the data.

Convolutional Neural Networks (CNNs)

- Specifically designed for analyzing visual data, such as images or videos.
- Utilizes convolutional layers to extract spatial hierarchies of features.
- Employs pooling layers to downsample the extracted features.
- Often used in tasks like image classification, object detection, and image segmentation.

Convolutional Neural Networks (CNNs)



Convolution Operation

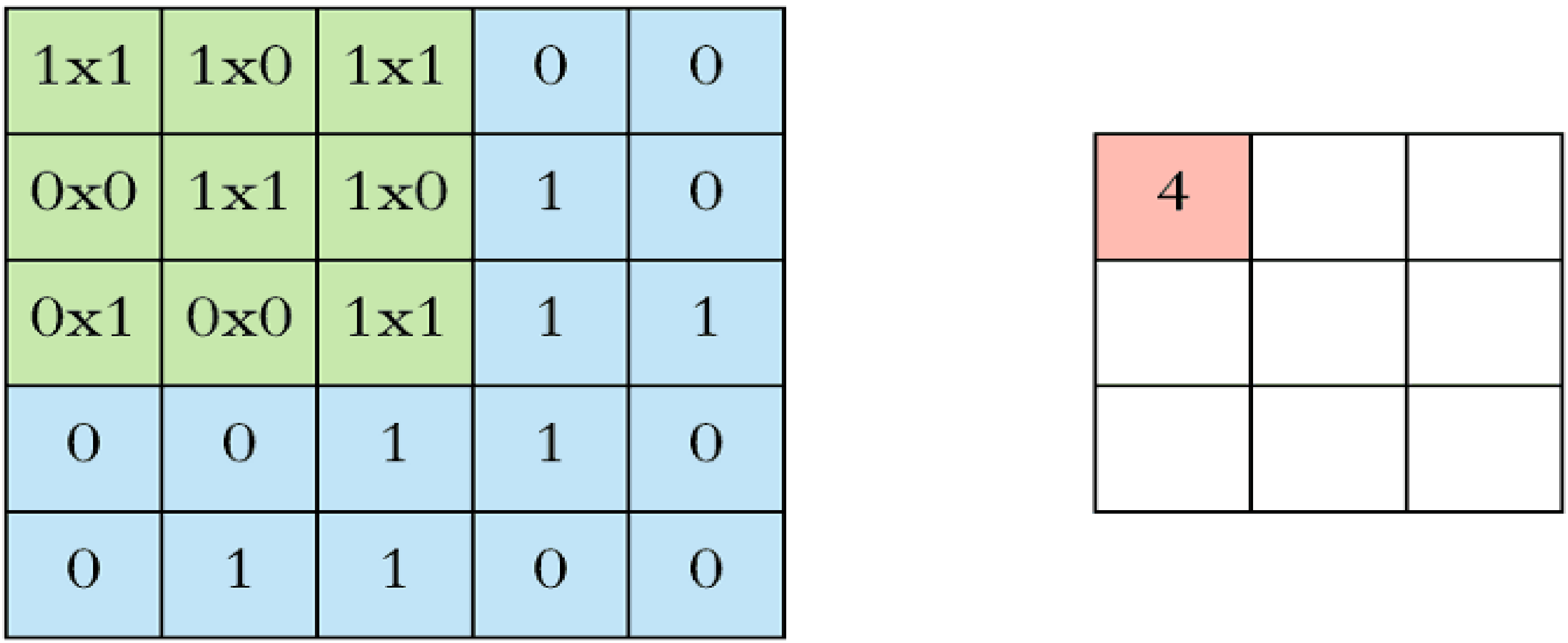
1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Image Matrix

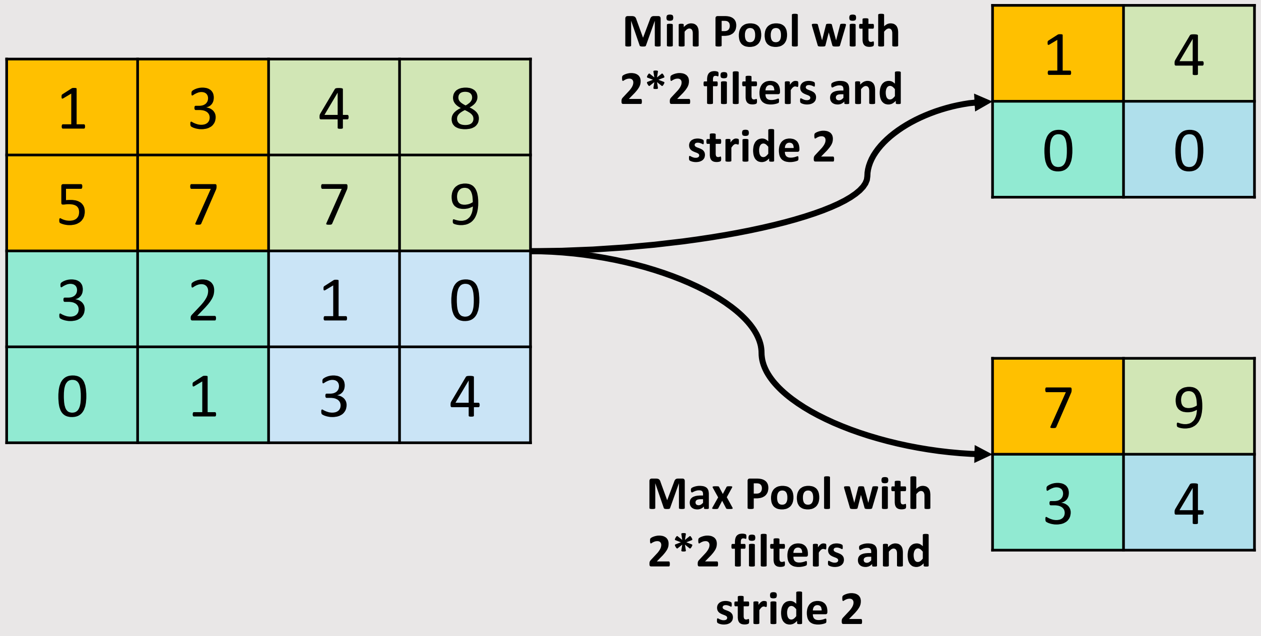
1	0	1
0	1	0
1	0	1

Filter Matrix

Convolution Operation



Pooling Operation



Classification Operation

- This operation consists of three different operations such as flatten, prediction of features and activation function.
- The flatten is flattening out extracted features from the input image into vector.
- This vector is feed to a fully connected network such as a neural network for prediction of the input feature vector.
- Finally, an activation function such as softmax or sigmoid is used to classify the predicted value of the output of the neural network.

Training and Optimization of CNNs

- **Backpropagation Algorithm:**
  - **Forward Pass:** The input data is fed into the neural network, and it propagates forward through the layers. Each neuron calculates its weighted sum of inputs and applies an activation function to produce an output.
  - **Loss Calculation:** The output of the neural network is compared with the expected output using a loss function. The loss function quantifies the error between the predicted output and the actual output.
  - **Backward Pass:** The error is propagated backward through the network to update the weights and biases. The algorithm starts from the output layer and calculates the gradient of the loss with respect to the parameters (weights and biases). This is done using the chain rule of calculus, where gradients are computed layer by layer.

Training and Optimization of CNNs

- **Gradient Descent Optimization:** The gradients are used to update the weights and biases of the network. The update is performed in the opposite direction of the gradient (moving towards minimizing the loss). The learning rate determines the step size of the update, balancing between fast convergence and overshooting.
- **Regularization Techniques** (L1/L2 regularization, Dropout, Early Stopping):
    1. L1/L2 techniques penalize large weight values, encouraging the model to distribute its importance across different features.
    2. Dropout randomly sets a fraction of the neurons' outputs to zero during training. By doing this, dropout prevents complex co-adaptations between neurons, forcing the network to learn more robust representations.
    3. Early stopping monitors the model's performance on a validation set during training. Training is stopped when the validation loss starts increasing or stops improving for a certain number of epochs.

Tuning of Hyper Parameters

- Convolution: Filter Size, Number of filters, Padding, Stride
- Pooling: Window size, Stride
- Fully Connected: Number of neurons

Tuning of Hyper Parameters

- No of hidden layers -Trial and error
- No of neurons in each hidden layer -Trial and error
- Learning rate. (The most important parameter) –0.1, 0.01, 0.001, 0.0001, .00001
- Activation functions –Sigmoid, Relu, Leaky Relu, Tanh
- Number of iteration –Trial and error
- Epoch –Trial and error
- Batch size –4, 8, 16, 32, 64, 128… (Power of 2)
- Regularization –Dropout, L1, L2
- Regularization Rate and Dropout lambda
- Normalizing inputs –Min-max, mean, Z-Score
- Weights and Bias Initialization –Zero, Random
- optimization algorithm
- Learning rate decay, Momentum

CNN Applications in Computer Vision

- **Image Classification:** Determining the category of an object present in an image.
- **Object Detection:** Identifying multiple objects in an image and providing a bounding box around each one. Frameworks like Faster R-CNN, YOLO (You Only Look Once), and SSD (Single Shot MultiBox Detector) are popular choices.
- **Image Segmentation:**
  - Semantic Segmentation: Assigning a class label to each pixel of an image such that pixels with the same label belong to the same object category.
  - Instance Segmentation: Extending semantic segmentation to differentiate individual object instances, i.e., distinguishing between two cars in an image. Mask R-CNN is a well-known model for this task.
- **Face Recognition:** Identifying or verifying a person based on their face. DeepFace and FaceNet are examples of architectures that excel in this domain.

CNN Applications in Computer Vision

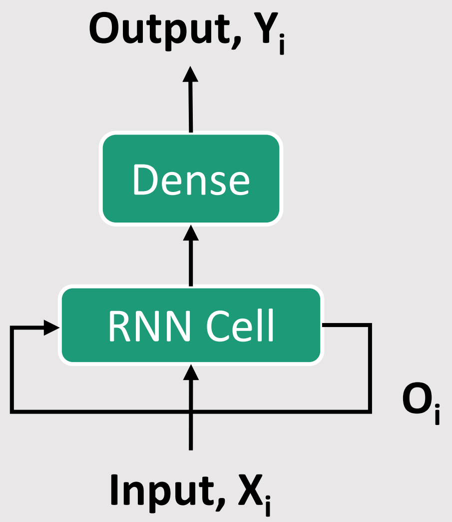
- **Gesture Recognition:** Recognizing gestures from image sequences to facilitate interactions with devices, often seen in AR and VR setups.
- **Video Analysis:**
  - **Action Recognition:** Identifying the primary activity in a video sequence.
  - **Anomaly Detection:** Spotting abnormal behaviors in surveillance videos.
- **Scene Parsing:** Understanding and labeling all components of a scene, such as identifying roads, buildings, trees, and cars in a street view image.
- **Image Captioning:** Generating a natural language description of an image's content. This involves a combination of CNNs (for feature extraction) and RNNs (for sequence generation).
- **Style Transfer:** Combining the content of one image with the style of another, often seen in applications that "paint" photos in famous art styles.

CNN Applications in Computer Vision

- **Super-resolution:** Increasing the resolution of images, making them sharper. SRCNN and EDSR are popular models in this domain.
- **Visual Attention and Scene Understanding:** Models that can focus on specific parts of an image when processing information, similar to human attention mechanisms.
- **Medical Image Analysis**: Identifying, classifying, and even predicting diseases from medical images such as X-rays, MRIs, and CT scans.
- **Optical Character Recognition (OCR)**: Converting images of typed, handwritten, or printed text into machine-encoded text.
- **Image Generation**: Generating new images that are similar to a given set. Generative Adversarial Networks (GANs) with CNN architectures are commonly used for this.
- **3D Object Recognition**: Recognizing objects from 3D data as opposed to traditional 2D images.
- **Age and Gender Prediction**: Estimating the age and gender of a person from their picture.

Recurrent Neural Networks (RNNs)

- Designed to process sequential data, such as time series or text.
- Contains recurrent connections that allow information to persist in the network.
- Can handle variable-length inputs and capture temporal dependencies.
- Frequently used in tasks like language modeling, speech recognition, and machine translation.



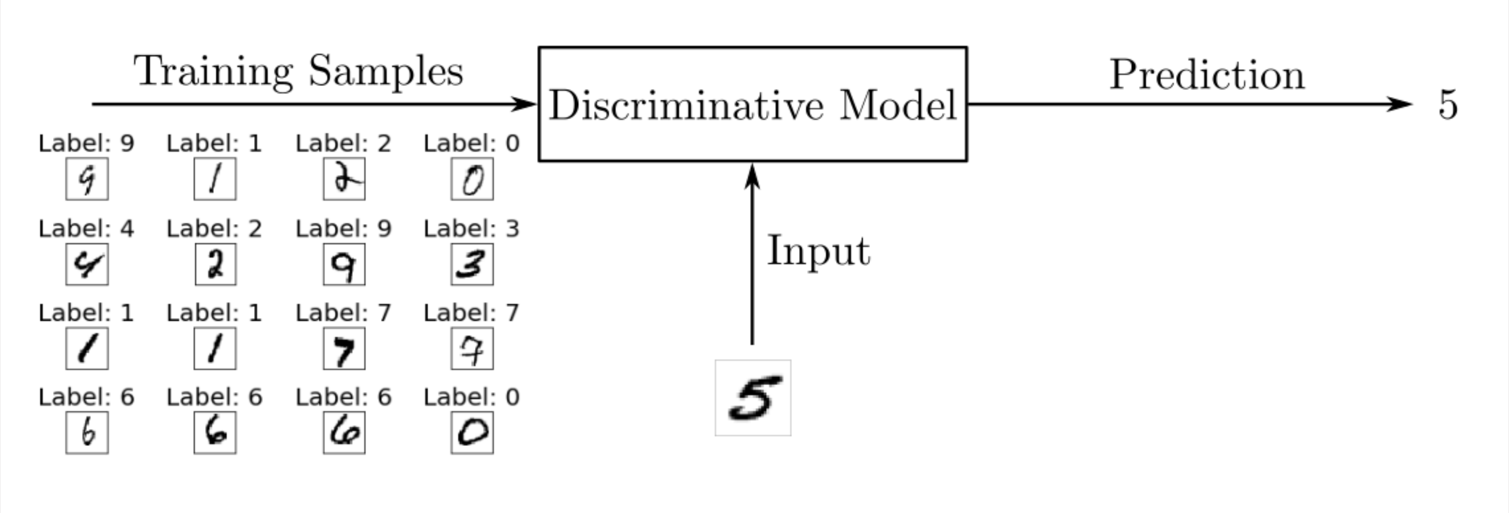
Long Short-Term Memory (LSTM)

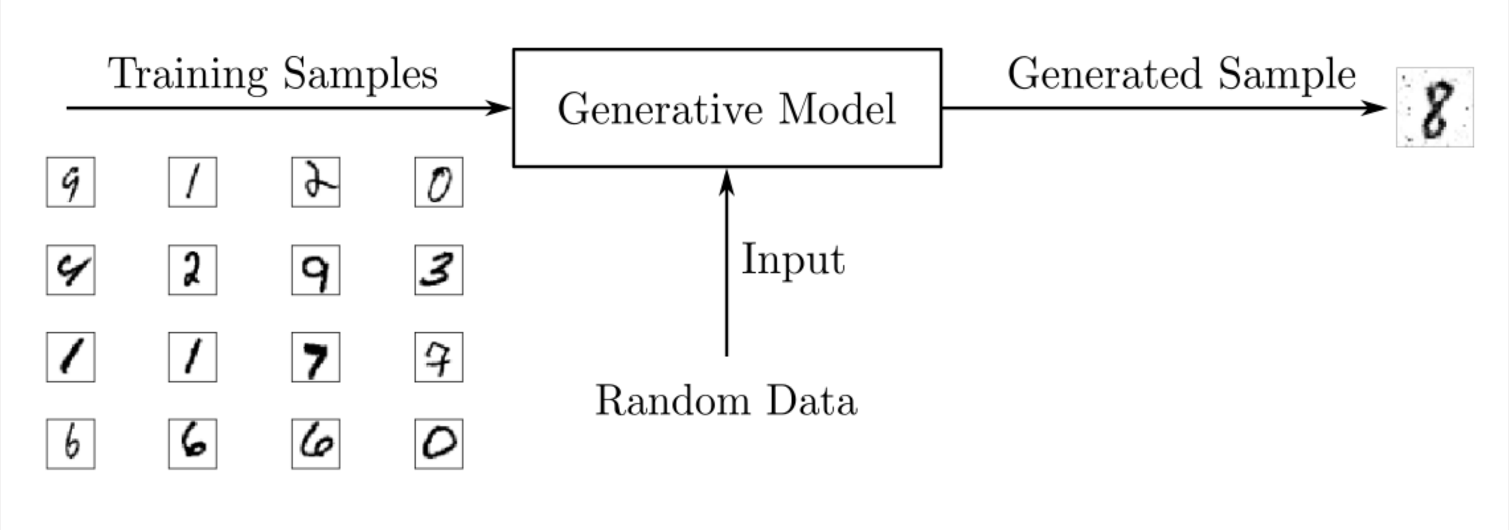
- A type of RNN that addresses the vanishing gradient problem.
- Utilizes memory cells with gating mechanisms to selectively retain or forget information.
- Suitable for modeling long-term dependencies in sequential data.
- Commonly used in tasks like speech recognition, sentiment analysis, and language translation.

Generative Adversarial Networks (GANs)

- Comprises two components: a generator and a discriminator.
- Generator creates synthetic samples while the discriminator tries to distinguish between real and fake samples.
- Trained in an adversarial setting, where the generator and discriminator learn from each other.
- Widely used for generating realistic images, video synthesis, and data augmentation.

Generative Adversarial Networks (GANs)





Transfer Learning

- Transfer learning is a learning method where a model developed for a task is reused for a second task.
- Popular approach in deep learning.
- Reduced vast computing and time resources required to develop DL model.
- Speed up training and improve the performance of DL model.

Transfer Learning Workflow in Python

- Load base model with pre-trained weights.

```
base_model = keras.applications.Xception(
                 weights='imagenet', # Load weights pre-trained on ImageNet.
                input_shape=(150, 150, 3), include_top=False) # Do not include the ImageNet classifier at the top.
```

- Freeze all layers in the base model by setting ***trainable = False***.

```
base_model.trainable = False
```

Transfer Learning Workflow in Python

- Create a new model on top of the output of one (or several) layers from the base model.

```
inputs = keras.Input(shape=(150, 150, 3)) # We make sure that the base_model is running in inference mode here,
# by passing `training=False`. This is important for fine-tuning, as you will
# learn in a few paragraphs.
x = base_model(inputs, training=False) # Convert features of shape `base_model.output_shape[1:]` to vectors
x = keras.layers.GlobalAveragePooling2D()(x) # A Dense classifier with a single unit (binary classification)
outputs = keras.layers.Dense(1)(x)
model = keras.Model(inputs, outputs)
```

Transfer Learning Workflow in Python

- Train new model on new dataset.

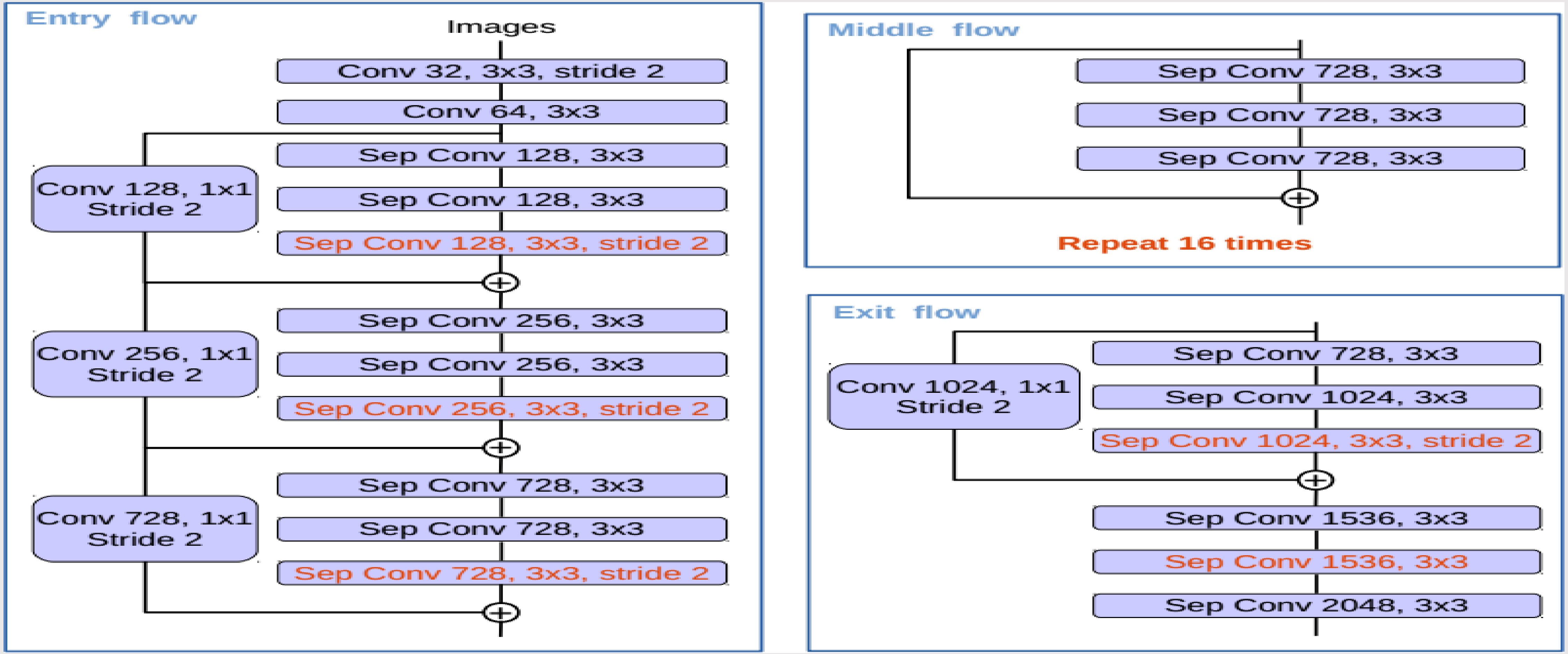
```
model.compile(optimizer = keras.optimizers.Adam(),
              loss = keras.losses.BinaryCrossentropy(from_logits=True),
              metrics=[keras.metrics.BinaryAccuracy()])
model.fit(new_dataset, epochs=20, callbacks=..., validation_data=...)
```

Various CNN Architectures used in Transfer Learning

- **Xception**
- **VGG (VGG16, VGG19)**
- **ResNet** (ResNet101, ResNet152, ResNet50)
- **InceptionV3**, InceptionResNetV2
- MobileNet, MobileNetV2
- DenseNet121, DenseNet169, DenseNet201

Xception Model

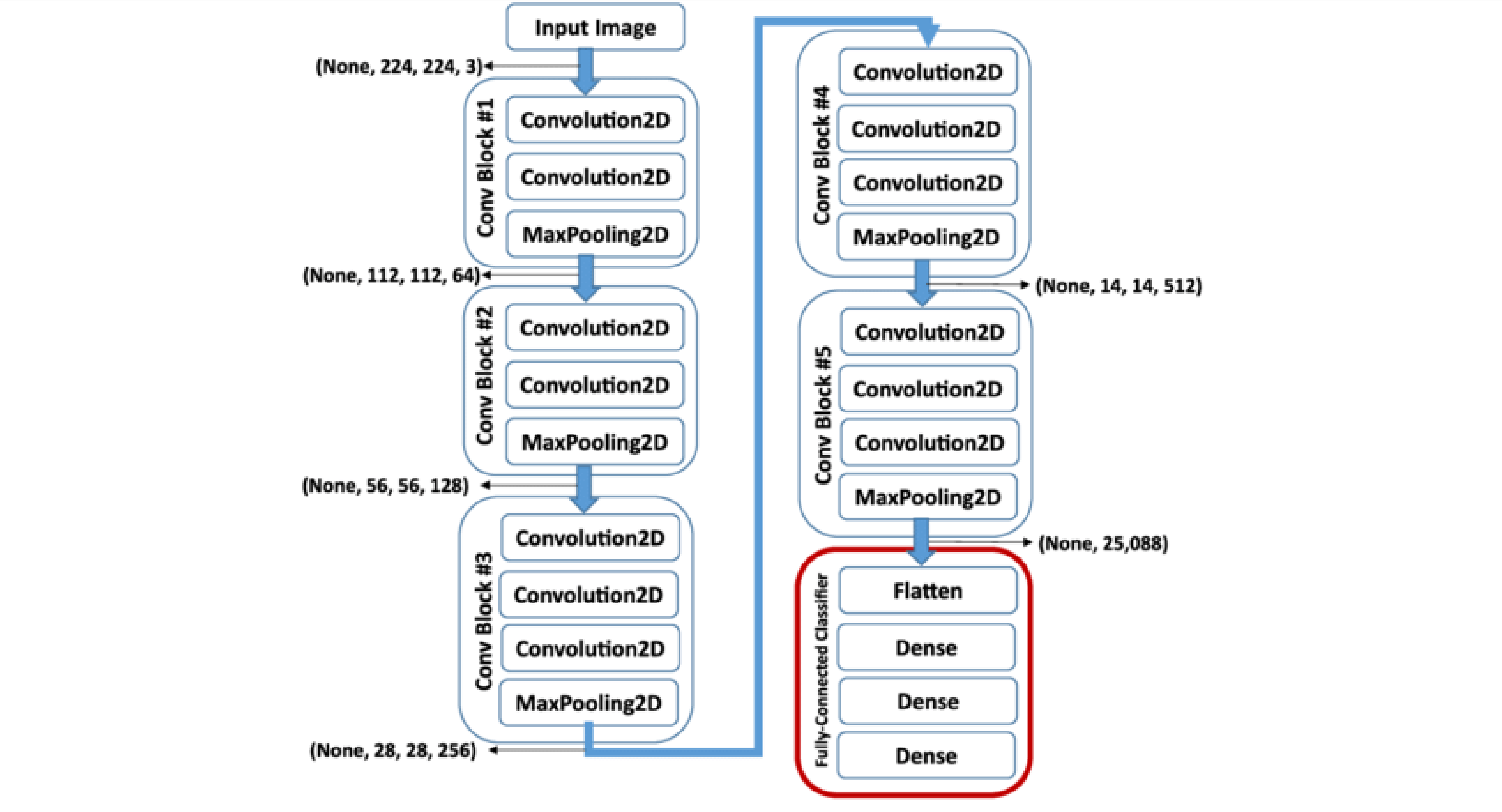
- Proposed by Francois Chollet



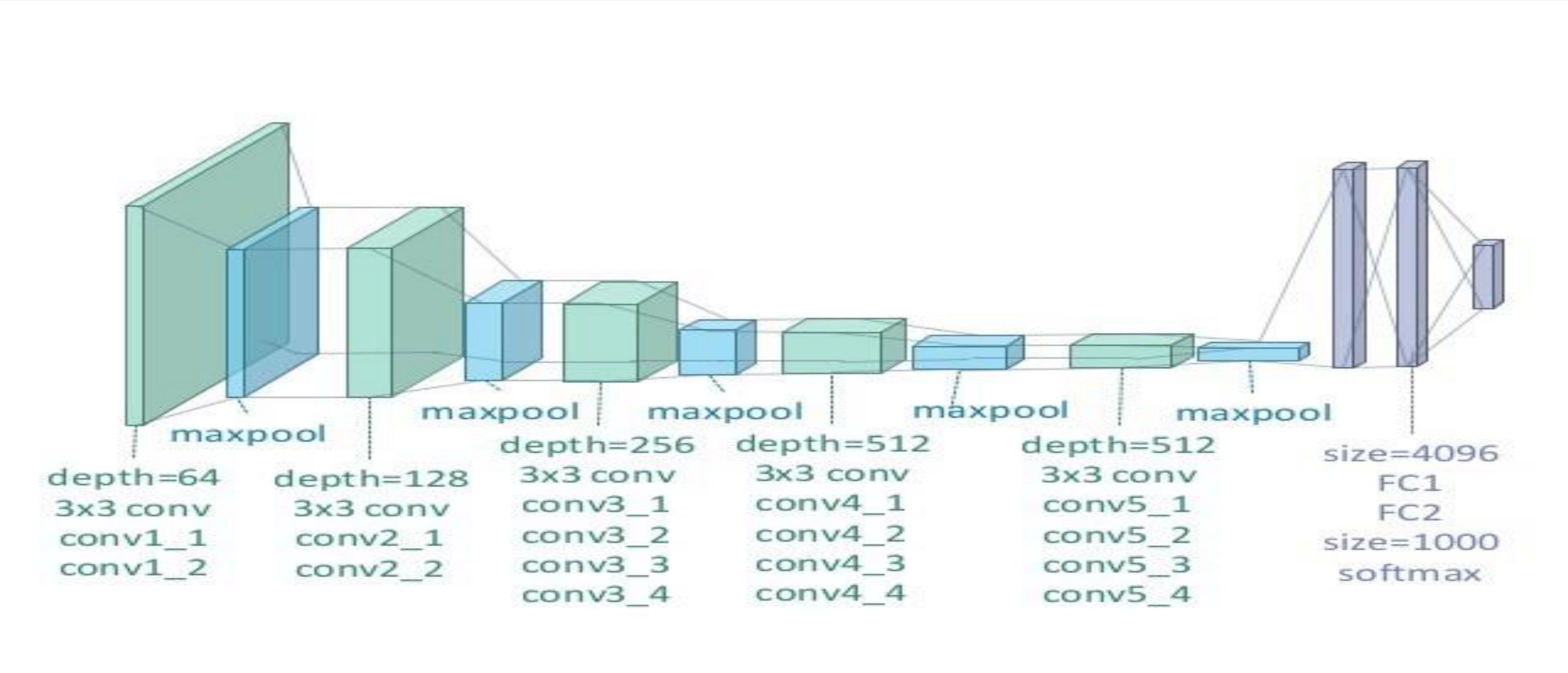
VGG Model

- Proposed by Karen Simonvan and Anderw Zisserman of Oxford Robotics Institute in the year 2014.
- Purpose of proposed this model is that to understand how the depth of convolutional networks affects the accuracy of large-scale image classification and recognition.

VGG16 Model


Input Image
(None, 224, 224, 3)
Conv Block #1
Convolution2D
Convolution2D
MaxPooling2D
(None, 112, 112, 64)
Conv Block #2
Convolution2D
Convolution2D
MaxPooling2D
(None, 56, 56, 128)
Conv Block #3
Convolution2D
Convolution2D
Convolution2D
MaxPooling2D
(None, 28, 28, 256)
Conv Block #4
Convolution2D
Convolution2D
Convolution2D
MaxPooling2D
(None, 14, 14, 512)
Conv Block #5
Convolution2D
Convolution2D
Convolution2D
MaxPooling2D
(None, 25,088)
Fully-Connected Classifier
Flatten
Dense
Dense
Dense

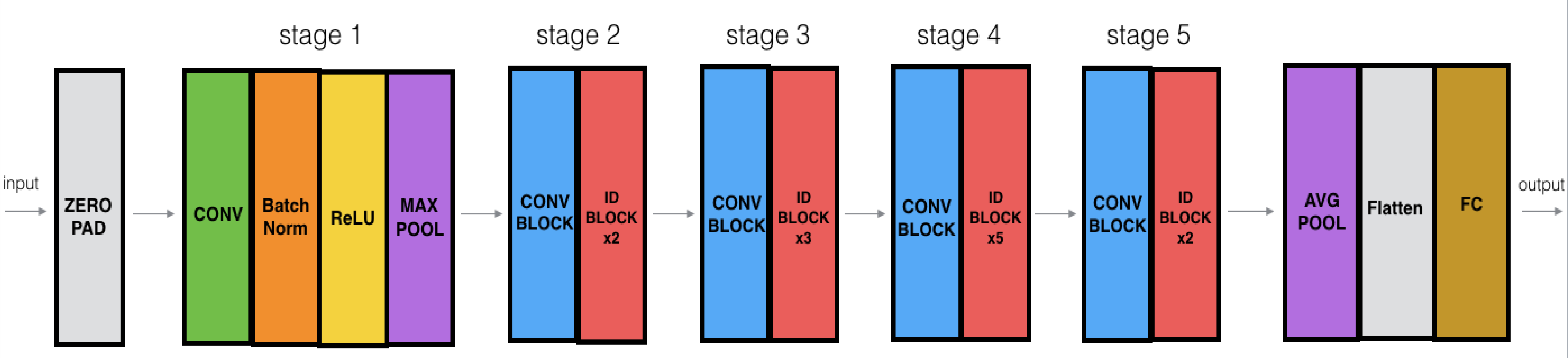
VGG19 Model



maxpool
maxpool
maxpool
maxpool
maxpool
depth=64
3x3 conv
conv1_1
conv1_2
depth=128
3x3 conv
conv2_1
conv2_2
depth=256
3x3 conv
conv3_1
conv3_2
conv3_3
conv3_4
depth=512
3x3 conv
conv4_1
conv4_2
conv4_3
conv4_4
depth=512
3x3 conv
conv5_1
conv5_2
conv5_3
conv5_4
size=4096
FC1
FC2
size=1000
softmax

ResNet Model

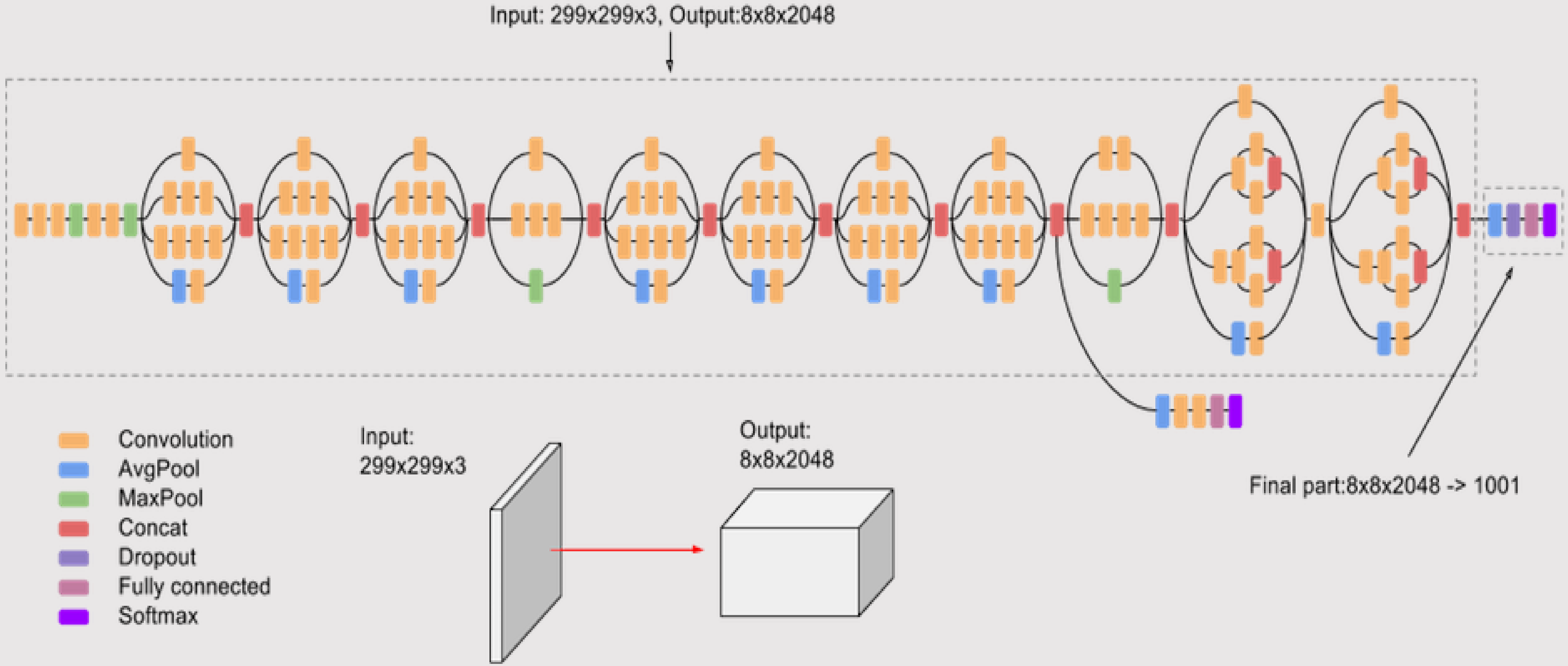
- It is a classic neural network.
- Used as a backbone for many computer vision applications.
- Extremely deep neural networks with 150+layer successfully.



Inception V3 Model

- Used for Image Classification.
- It is a superior version of the basic model Inception V1 which was introduced as GoogLeNet in 2014.
- It was developed by a team at Google.

Inception V3 Model



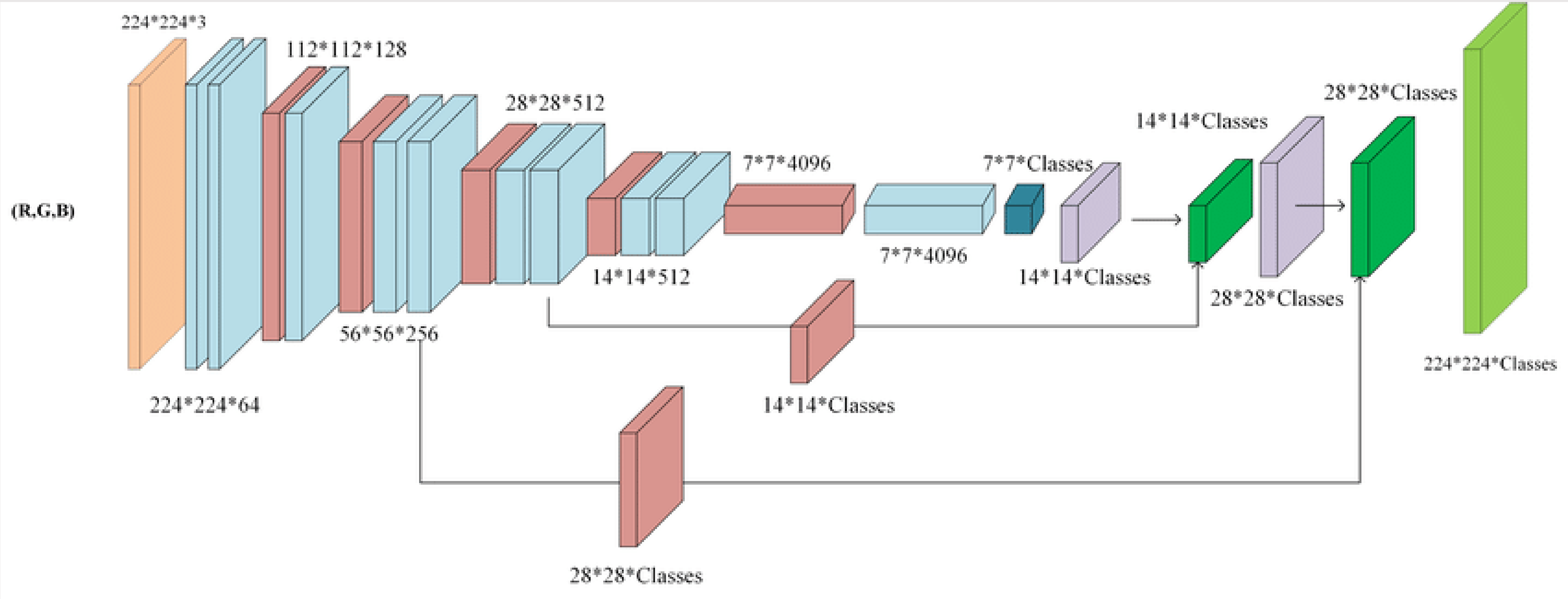
Deep Learning Model for Image Segmentation

- **Fully Convolutional Networks (FCN):** The first end-to-end trainable model for semantic segmentation. Uses transposed convolution (deconvolution) to upscale the feature map to the original image size.
- **U-Net:** Popular for biomedical image segmentation. Has an encoder-decoder structure where the encoder captures context, and the decoder enables precise localization. Skip connections are added between layers of the same size in the encoder and decoder, which helps in better localization.
- **Segmentation Refined Mechanism (DeepLab)**
  - DeepLabv3 and DeepLabv3+ are the most recent versions.
  - Uses atrous (dilated) convolution to increase the field of view without reducing the resolution.
  - Combines semantic information at multiple scales using atrous spatial pyramid pooling.
  - DeepLabv3+ also incorporates encoder-decoder structure for better object boundaries.

Deep Learning Model for Image Segmentation

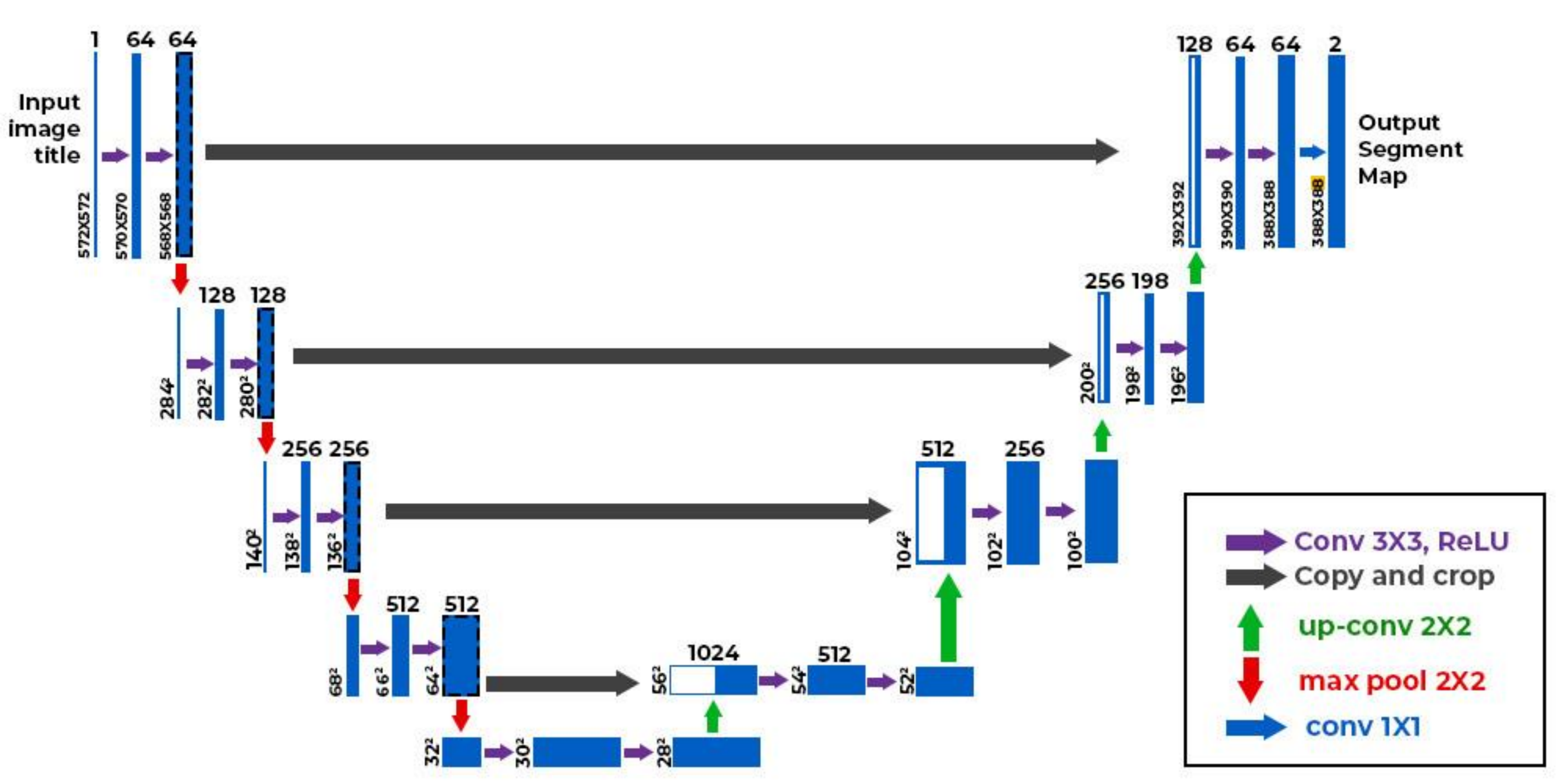
- **Mask R-CNN:** An extension of Faster R-CNN for object detection. Outputs a binary mask for each object in addition to the bounding box and class label. Uses a Region of Interest Align to maintain the spatial resolution.
- **PixelLink:** For instance, segmentation, particularly suited for detecting and segmenting text in images. Connects pixels in the output map to separate instances.
- **PSPNet (Pyramid Scene Parsing Network)**: Uses pyramid pooling at multiple grid scales. Captures different sub-region representations, beneficial for complex scenes.
- **HRNet (High-Resolution Network):** Maintains a high-resolution representation through the network rather than down-sampling and then up-sampling. Produces high-resolution output, making it suitable for tasks requiring finer details like semantic segmentation.

FCN Model



(R,G,B)
224*224*3
224*224*64
112*112*128
56*56*256
28*28*512
14*14*512
7*7*4096
7*7*4096
7*7*Classes
14*14*Classes
14*14*Classes
14*14*Classes
28*28*Classes
28*28*Classes
28*28*Classes
224*224*Classes

U-Net Model



Input image title
1
64
64
572X572
570X570
568X568
128
128
284²
282²
280²
256 256
140²
138²
136²
512
512
68²
66²
64²
32²
30²
28²
1024
56²
512
54²
52²
512
256
104²
102²
100²
256 198
200²
198²
196²
128
64
64
2
392X392
390X390
388X388
388X388
Output Segment Map
Conv 3X3, ReLU
Copy and crop
up-conv 2X2
max pool 2X2
conv 1X1