Google Developer Groups
On Campus • Telkom University Bandung

Deep Learning using Tensorflow

Ardavaa ardava-barus @rdavaa_

Muhammad Karov Ardava Barus
Machine Learning Mentor @GDGoC Tel-U,
Data Science Student | Telkom University

Presentation Link

ristek.link/ML-slides-6

Google Developer Groups
On Campus • Telkom University Bandung

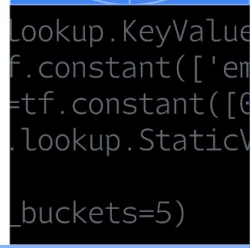
Today's Topic

Core Concepts

- What is Deep Learning?
- Tensorflow & Keras
- Neural Network
- Challenges

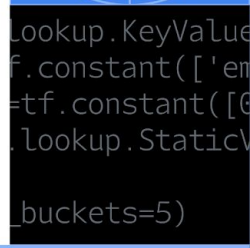
Computer Vision

- Definition
- Image Processing
- Image Classification

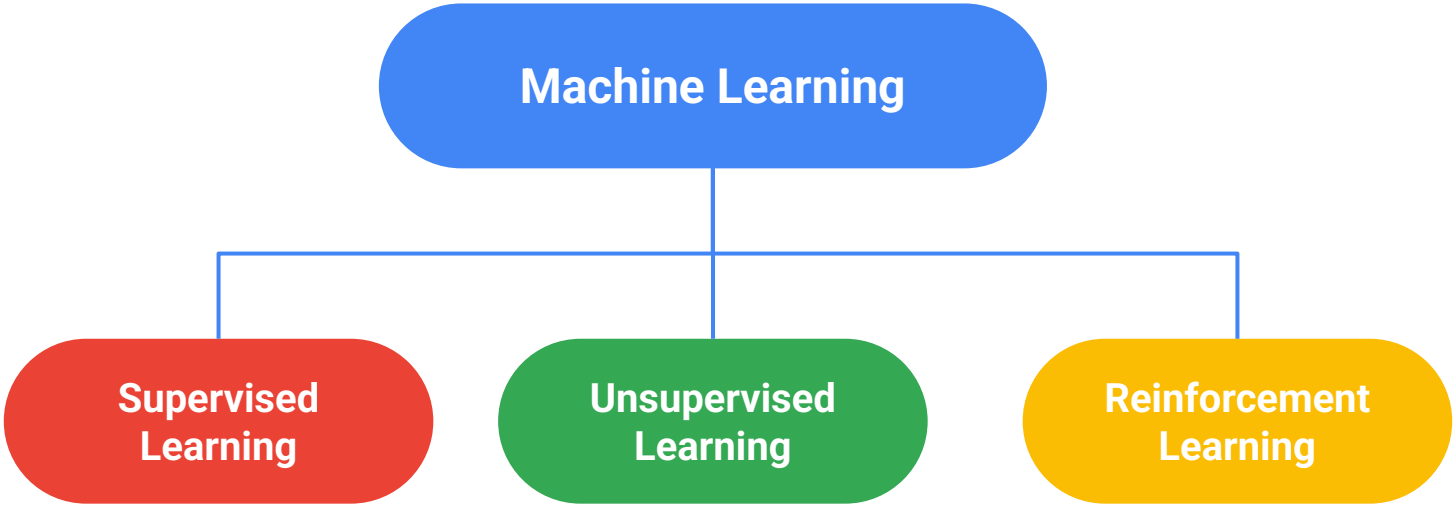


Google Developer Groups
On Campus • Telkom University Bandung

Let’s recap what we’ve learned about Machine Learning



Recap: Machine Learning



Recap: Machine Learning

Regression

In Machine Learning, regression is a method to predict the continuous value of a dependent variable based on the relationship with independent variables.

Recap: Machine Learning

Classification

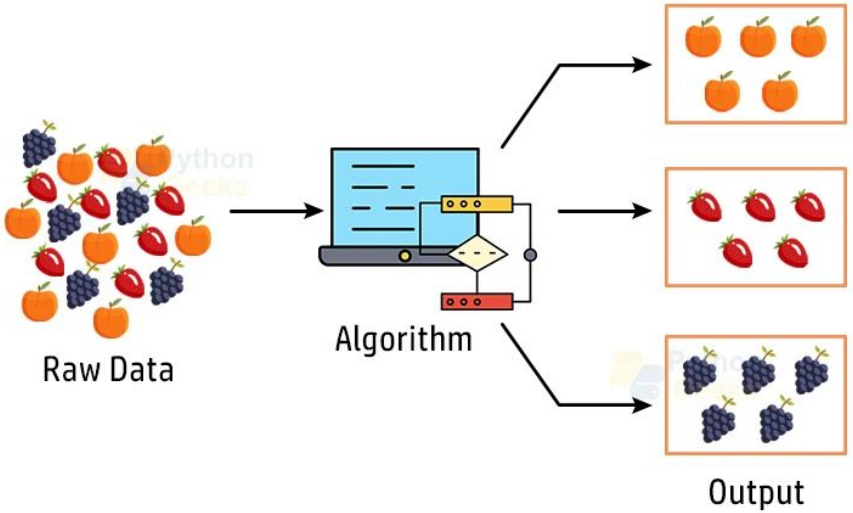
In Machine Learning, classification is a method to predict the category or class of a dependent variable based on the relationship with independent variables.

Classification: a Machine Learning technique to identify the category of new observations based on training data.

Recap: Machine Learning

Clustering

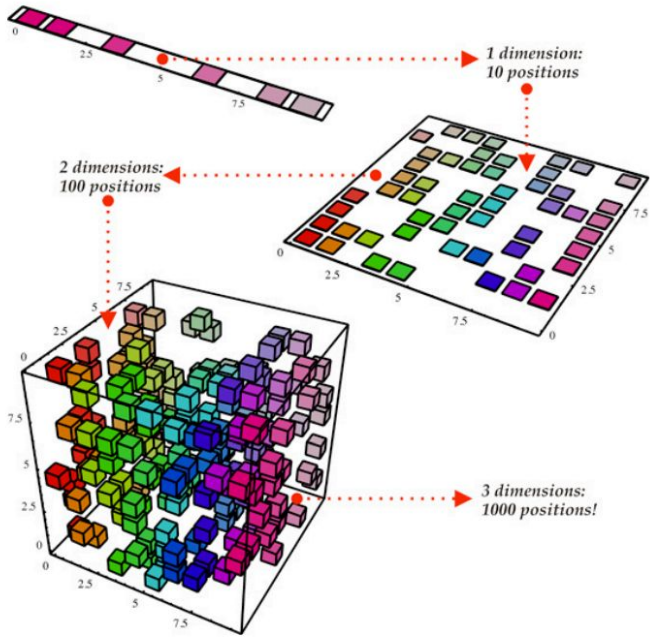
In Machine Learning, clustering is a method to group unlabeled data by finding patterns or similar characteristics among the data.



Recap: Machine Learning

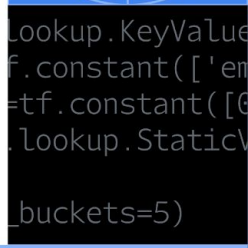
Dimensionality Reduction

In Machine Learning, Dimensionality Reduction is a method to reduce the number of features in data while retaining important information, in order to improve efficiency and reduce overfitting.

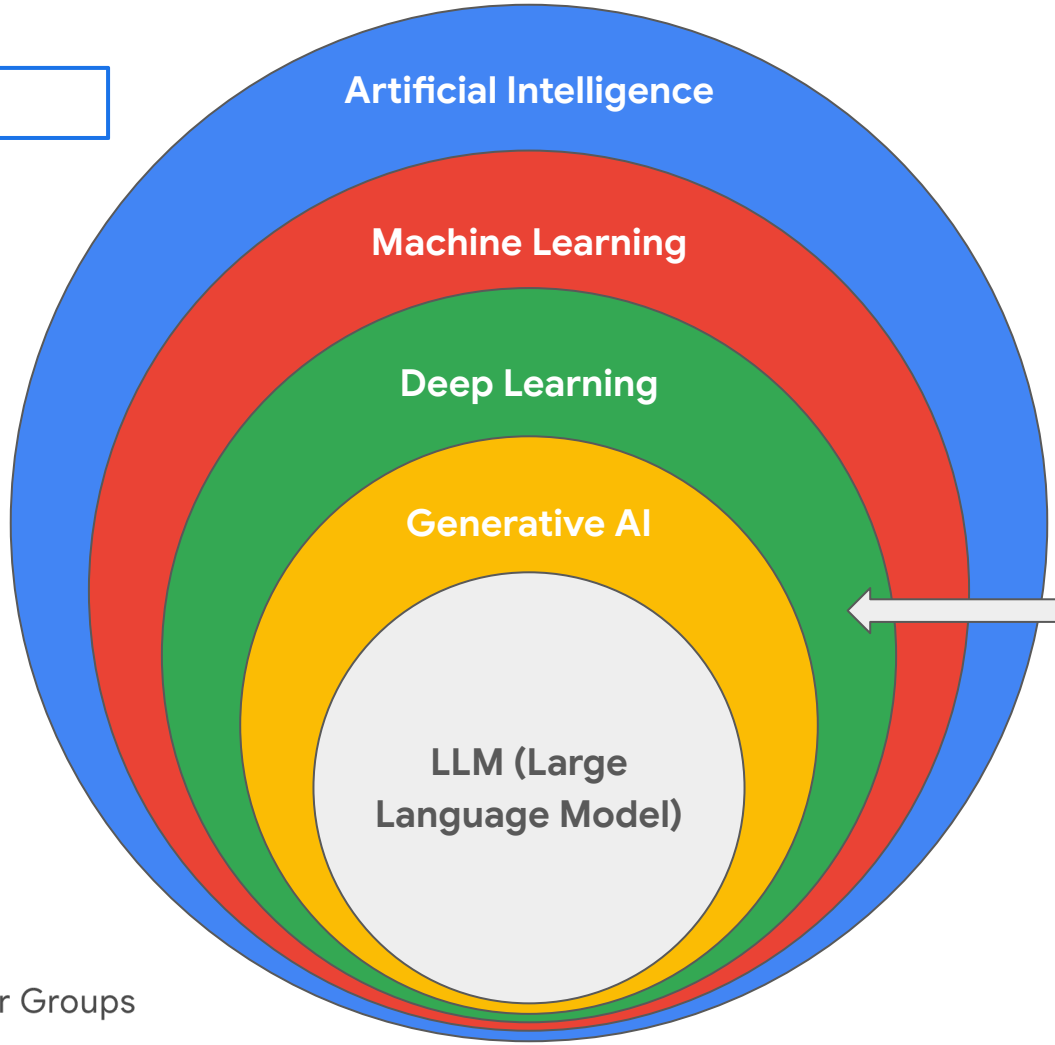


Google Developer Groups
On Campus • Telkom University Bandung

Core Concepts



Core Concepts



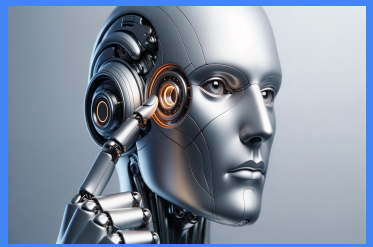
We're in Deep Learning phase now

Core Concepts

What is Deep Learning

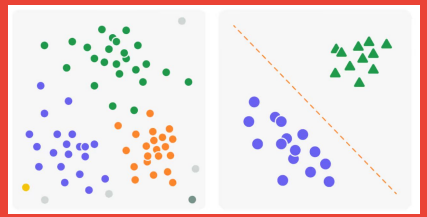
Artificial Intelligence

Any technique that enables computers or machines to mimic human intelligence



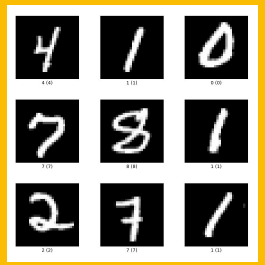
Machine Learning

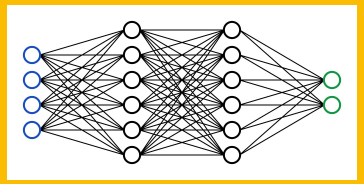
A technique by which a computer learn patterns from data and make data-driven predictions



Deep Learning

Extract patterns from data using neural networks





Core Concepts

1952 Stochastic Gradient Descent

1958 Perceptron
- Learnable Weights

•
•
•

1986 Backpropagation
- Multi-layer Perceptron

1995 Deep Convolutional NN
- Digit Recognition

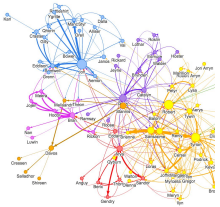
Why Now?

Neural Networks date back decades, so why the dominance?

I. Big Data

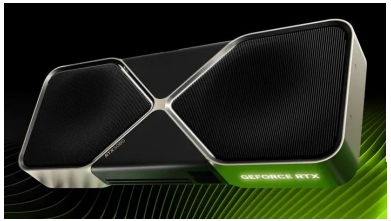
- Larger Datasets
- Easier Collection & Storage

IMAGENET



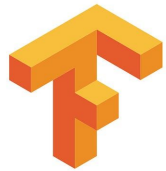
II. Hardware

- Graphics Processing Units (GPU)
- Massively Parallelizable



III. Software

- Improved Techniques
- New Models
- Toolboxes

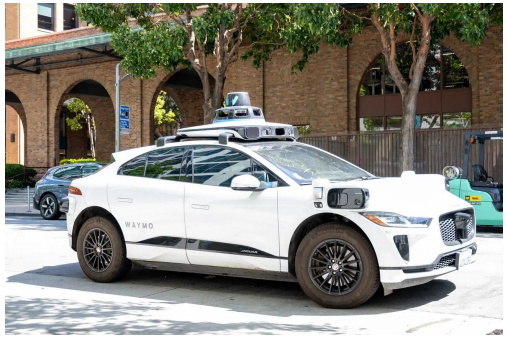


Core Concepts

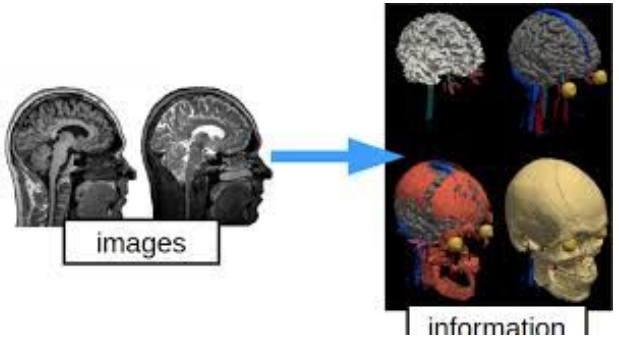
Real Life Applications

Gemini Claude

Generative AI



Autonomous vehicles



Medical Image Computing

and more...!

Core Concepts

Tensorflow

Tensorflow is an open-source machine learning framework developed by **Google Brain**. It's widely used for deep learning, neural networks and large machine learning tasks

Key features

- Supports both CPUs and GPUs (and even TPUs) for training and inference.
- Uses dataflow graphs for efficient computation
- Used in applications like image recognition, natural language processing (NLP), and many others!

Core Concepts

Keras

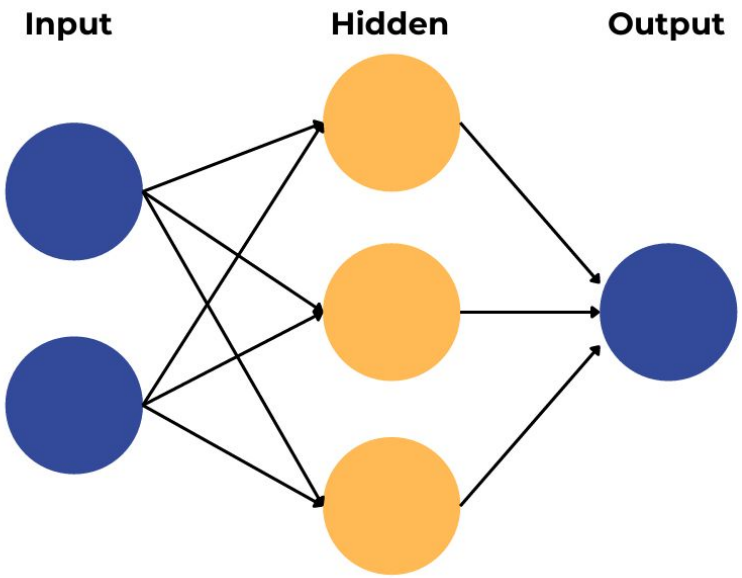
Keras is an open-source deep learning API written in Python. It acts as a high-level interface for TensorFlow, making it easier to build and train neural networks.

Key features

- User-friendly
- Runs on Tensorflow
- Pretrained Models: VGG, ResNet, and MobileNet

Core Concepts

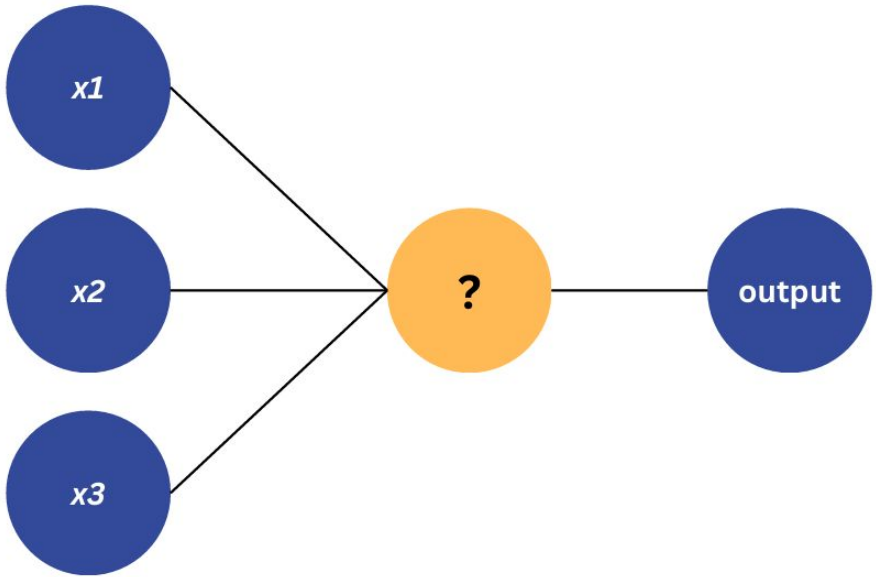
Neural Network Architecture



Core Concepts

Neural Network

Simplified Artificial Neuron

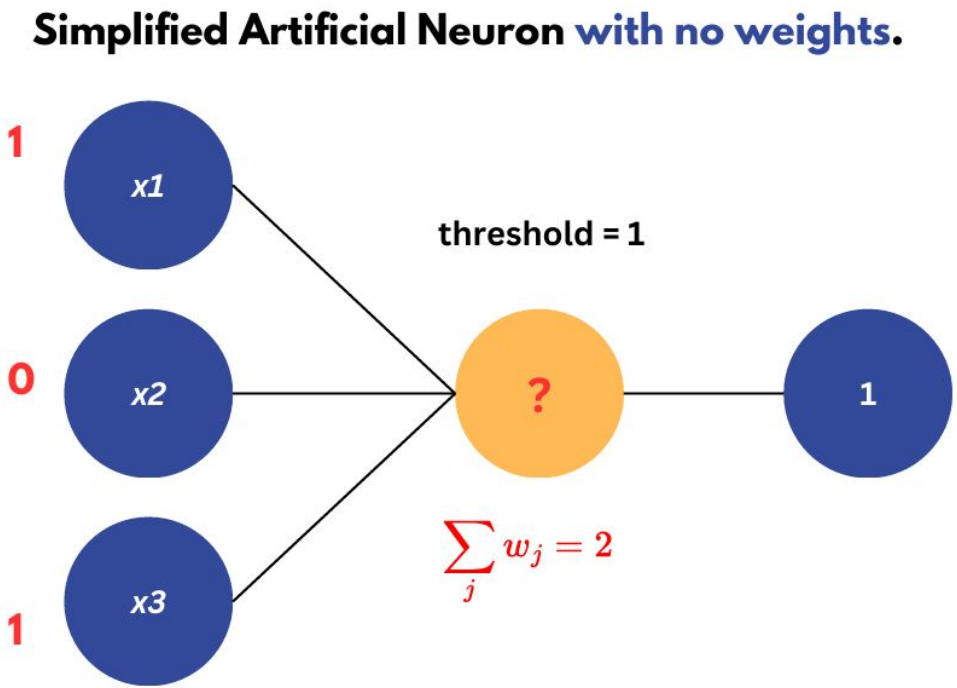


Will I be cooking at home?

- *x1 -> Do I have ingredients?*
- *x2 -> Do I have a full set of cooking utensils?*
- *x3 -> Do I have time to cook?*

Core Concepts

Neural Network

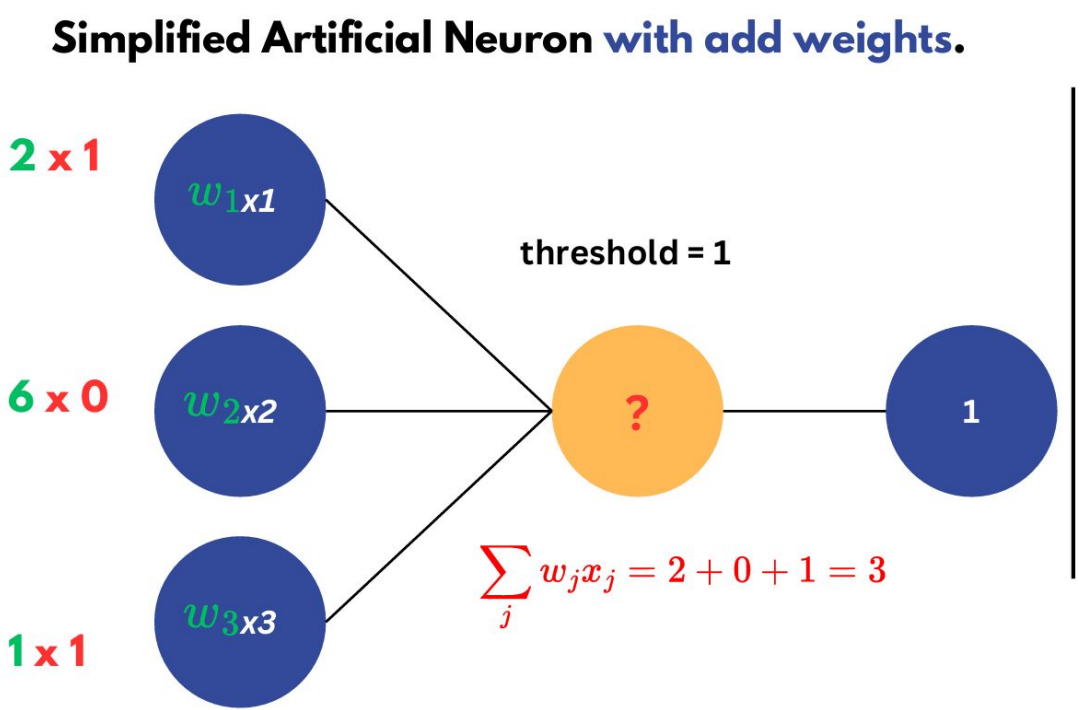


Will I be cooking at home?

- *x1 -> 1 (Have enough ingredients)*
- *x2 -> 0 (Have incomplete cooking utensils)*
- *x3 -> 1 (Have time to cook)*

Core Concepts

Neural Network



Will I be cooking at home?

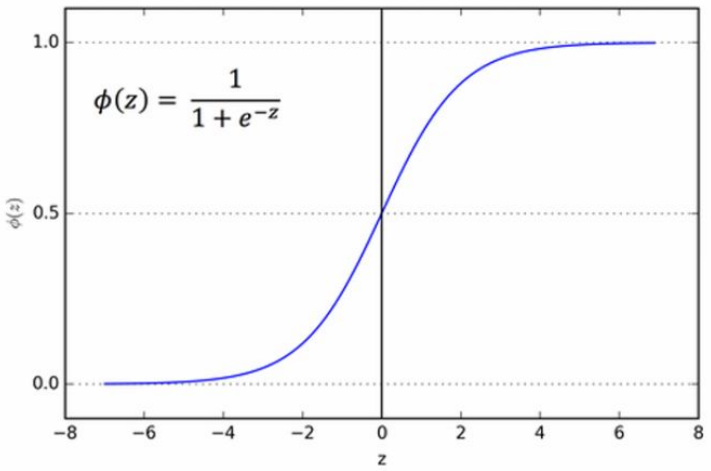
- *x1 -> Do I have ingredients?*
  w1=2
- *x2 -> Do I have a full set of cooking utensils?*
  w2=6
- *x3 -> Do I have time to cook?*
  w3=1

Core Concepts

Activation Functions

Sigmoid

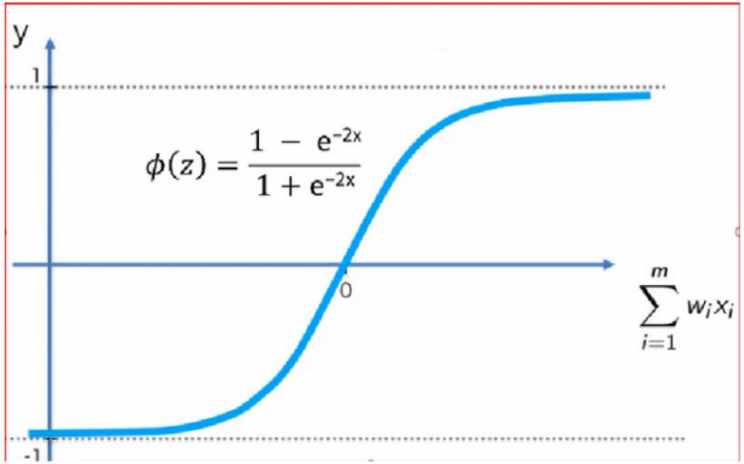
- Use for binary classification **output layers** (e.g., yes/no, 0/1 problems).
- **Avoid in hidden layers** (vanishing gradient issue).



Activation Functions

tanh

- Use in hidden layers for small neural networks (better than sigmoid).
- Avoid in deep networks (still has vanishing gradient issues).



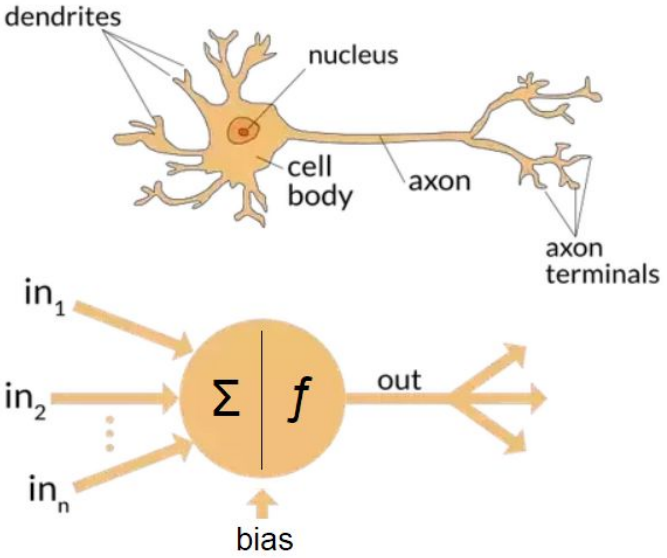
Activation Functions

👉 General Rule

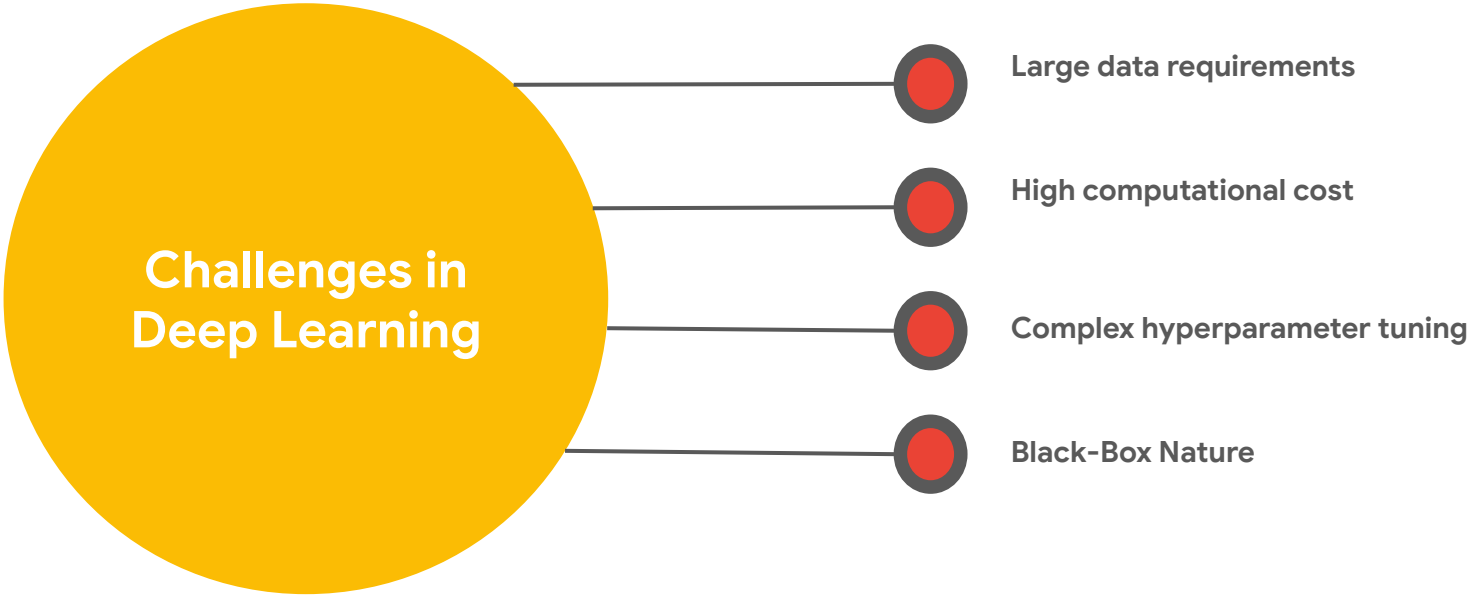
- Use **ReLU** in hidden layers for most deep learning tasks.
- Use **Sigmoid/Tanh** only when necessary (e.g., tanh in small networks, sigmoid for probability outputs).

Core Concepts

Neural Network is inspired by the human brains!



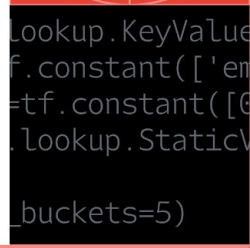
Core Concepts



Google Developer Groups
On Campus • Telkom University Bandung

Computer Vision

Definitions



What is Computer Vision?

- **Computer vision** is a branch of artificial intelligence (AI) that focuses on developing systems capable of understanding and processing visual data, like images and videos.
- It combines **deep learning**, **image processing**, and **machine learning** to extract meaningful insights from visual content.

Source: https://towardsai.net/p/l/machine-learning-7

Computer Vision: Definition

Key Applications of Computer Vision

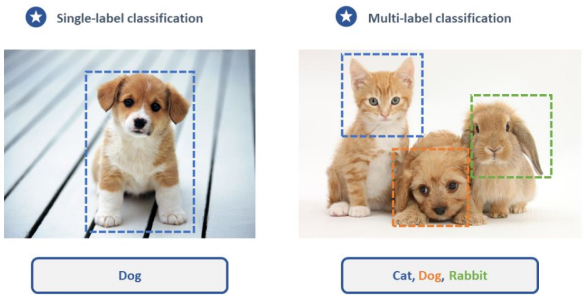
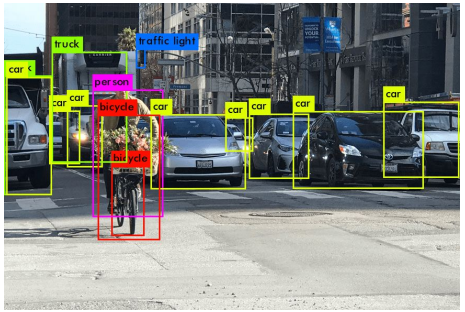
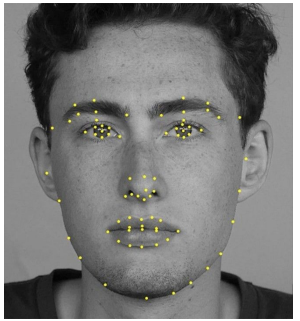


Image Classification



Object Detection



Facial Recognition

and more...!

Google Developer Groups
On Campus • Telkom University Bandung

Computer Vision

Image Processing

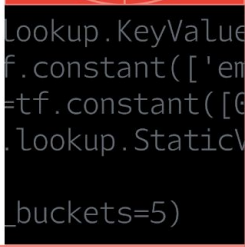
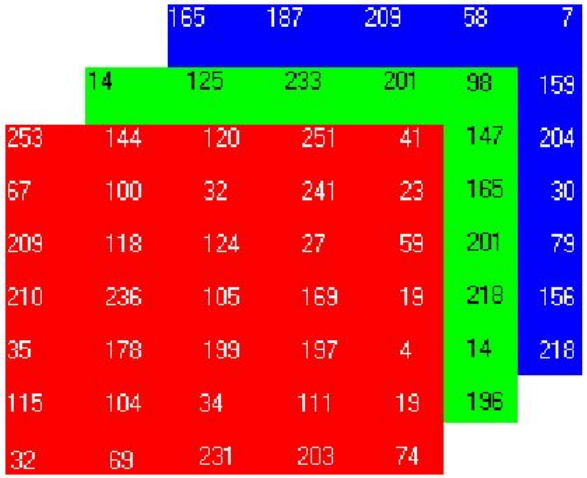
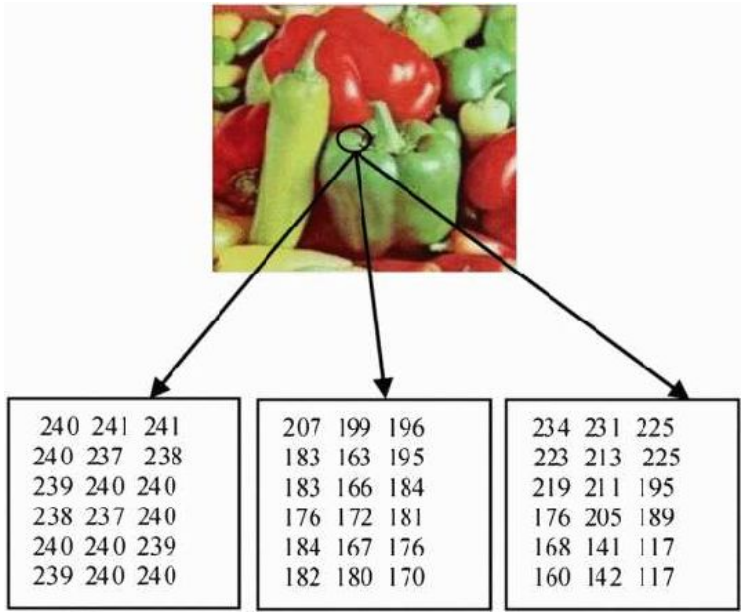


Image is just a bunch of numbers!

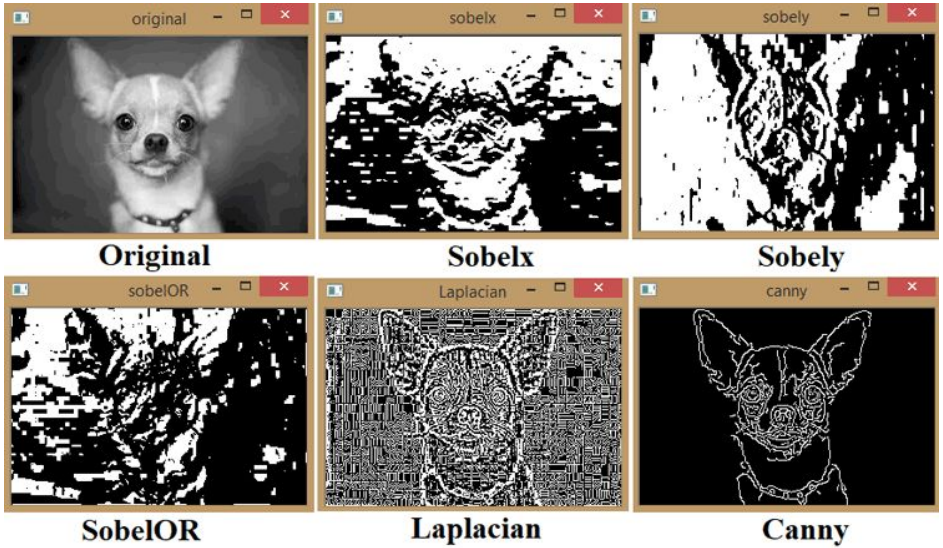




Matrix representation

What is Image Processing?

- **Image Processing** is a technique used to manipulate and analyze digital images to enhance their quality or extract useful information
- Usual tasks:
  - Image resizing,
  - Grayscale conversion
  - Normalization
  - Data Augmentation (rotate, flip, zoom)



Source: https://circuitdigest.com/tutorial/image-manipulation-in-python-opencv-part2

Image Resizing

Original Image - 500x500

Resized Image - 250x250

- **Image Resizing** is the process of changing the dimensions (width & height) of an image while maintaining its aspect ratio or adjusting it to fit a specific size.

  It's a crucial step in image preprocessing for machine learning, deep learning, and computer vision applications.

- **When to use?**
  - Before feeding images into CNNs (VGG16, ResNet, MobileNet)
  - To standardize different image sizes in a dataset
  - To improve training speed and reduce memory usage

Image Resizing

```python
import tensorflow as tf

# Load and resize an image
image = tf.image.resize(image, (128, 128))
```

Resizing Images for CNN Input (e.g., 128x128)

```python
datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale=1./255)

dataset = datagen.flow_from_directory("dataset/", target_size=(128, 128))
```

Resizing Dataset Images Automatically

Grayscale Conversion

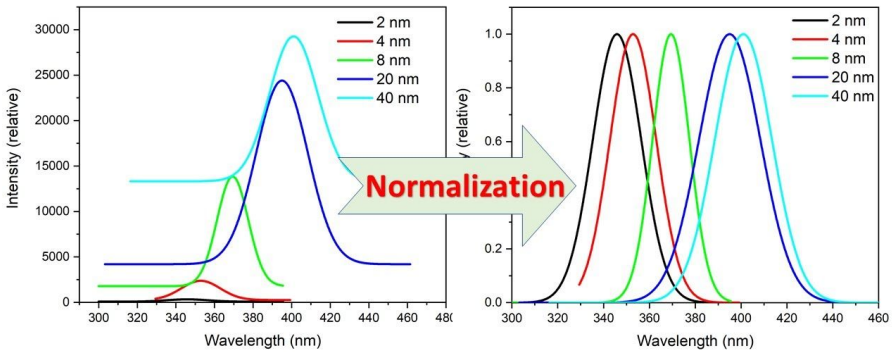
- **Grayscale conversion** is the process of transforming a **color image (RGB)** into **a black-and-white (grayscale) image** by reducing the three color channels (**Red**, **Green**, **Blue**) into a single intensity channel.
- **When to use?**
  - Face & Object Classification or Detection
  - Medical Imaging
  - Edge Detection

Grayscale Conversion

```python
import tensorflow as tf

# Load an image
image = tf.io.read_file("image.jpg")
image = tf.image.decode_jpeg(image)

# Convert to grayscale
gray_image = tf.image.rgb_to_grayscale(image)
```

Grayscale Conversion in Tensorflow

Normalization

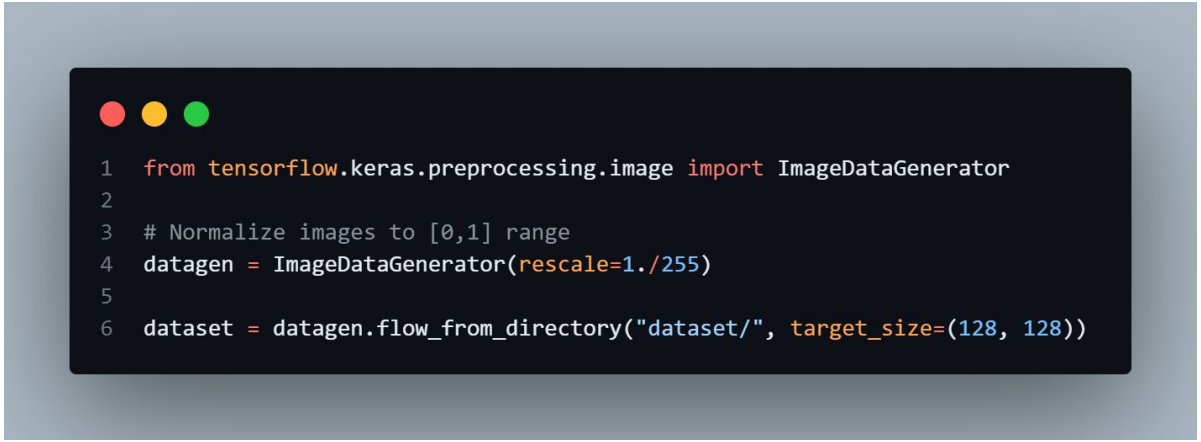


- **Image Normalization** is the process of adjusting pixel values in an image to a standard range, such as **[0,1]** or **[-1,1]**.

  This step is crucial in deep learning and computer vision tasks to improve model performance and training stability.

- **When to use?**
  - Before training CNNs (VGG16, ResNet, MobileNet)
  - When pixel values vary significantly across a dataset
  - To improve gradient flow in deep learning models

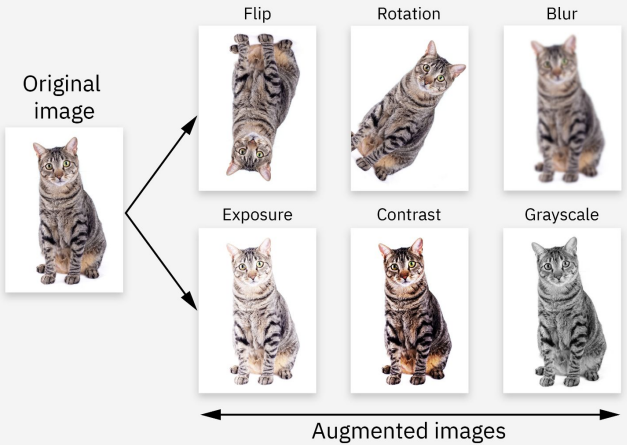
Image Normalization



```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Normalize images to [0,1] range
datagen = ImageDataGenerator(rescale=1./255)

dataset = datagen.flow_from_directory("dataset/", target_size=(128, 128))
```

Image Normalization in Tensorflow

Data Augmentation



- **Data Augmentation** is a technique used in deep learning and computer vision to **artificially expand a dataset by applying transformations** to existing images.

  It helps improve model generalization, prevents overfitting, and makes models more robust to variations in real-world data.

- **When to use?**
  - When you have limited training data
  - To make models robust against variations in real-world images
  - To prevent overfitting in deep learning models

Data Augmentation

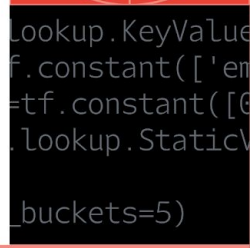
```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Create an ImageDataGenerator object with various augmentation settings
datagen = ImageDataGenerator(
    rotation_range=30,        # Randomly rotate images by up to 30 degrees
    width_shift_range=0.2,    # Shift image horizontally by up to 20% of its width
    height_shift_range=0.2,   # Shift image vertically by up to 20% of its height
    horizontal_flip=True,     # Randomly flip images horizontally
    brightness_range=[0.5, 1.5]  # Adjust brightness randomly between 50% and 150%
)

# Load dataset from a directory, applying the above augmentations on-the-fly
dataset = datagen.flow_from_directory(
    "dataset/",        # Path to dataset folder
    target_size=(128, 128)  # Resize all images to 128x128 pixels
)
```

Data Augmentation in Tensorflow

Google Developer Groups
On Campus • Telkom University Bandung

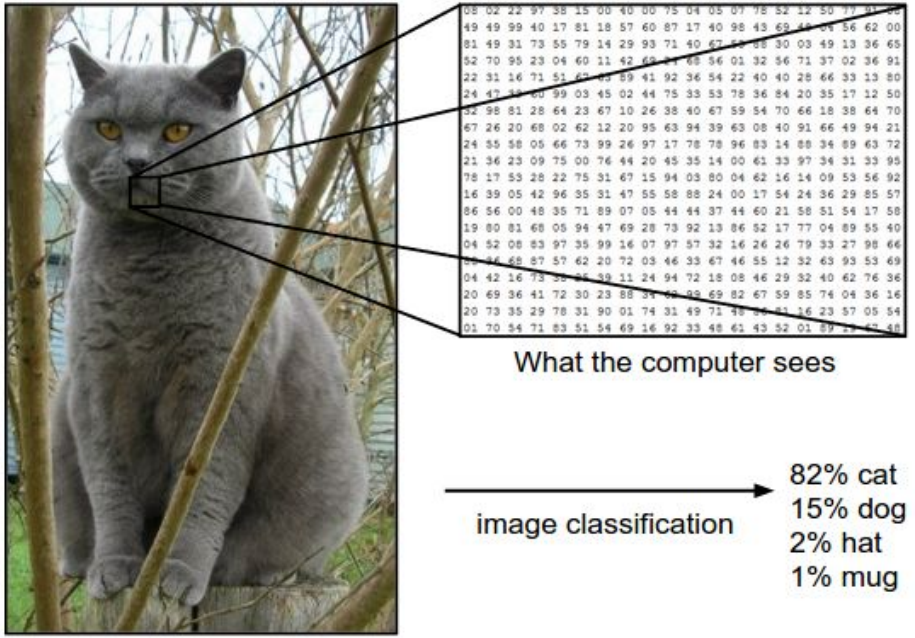
Computer Vision

Image Classification



What is Image Classification?

- **Image Classification** is the task of assigning a label or category to an image based on its content.
- It is a fundamental problem in computer vision, where the goal is to teach a machine to recognize and categorize images into predefined classes.



Source: https://cs231n.github.io/classification/

Image Classification Models

Convolutional Based Models

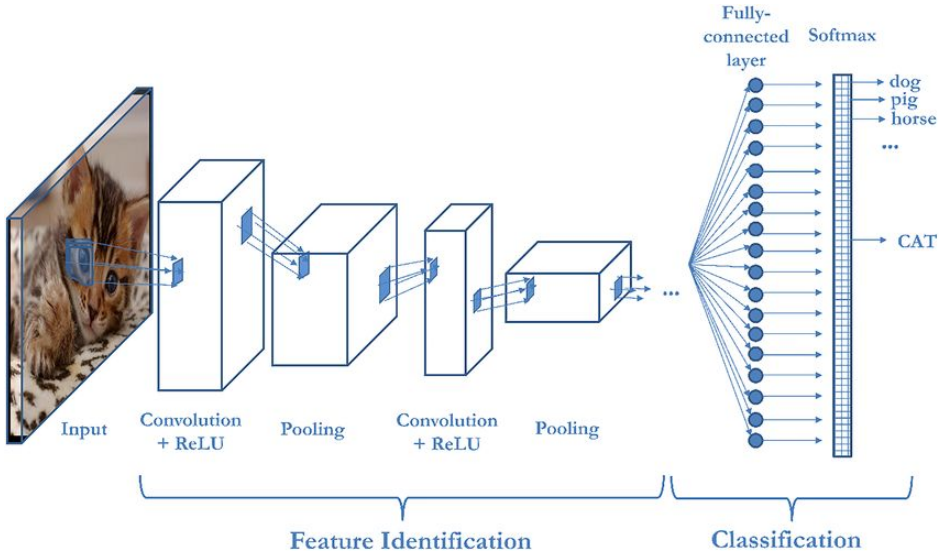
- Original CNN
- VGG16, VGG19
- ResNet, ResNet V2
- Inception, Xception
- And more!

Non-Convolutional Based Models

- Vision Transformer
- Swin Transformer
- Fully Connected Neural Networks (FNN)
- And more!

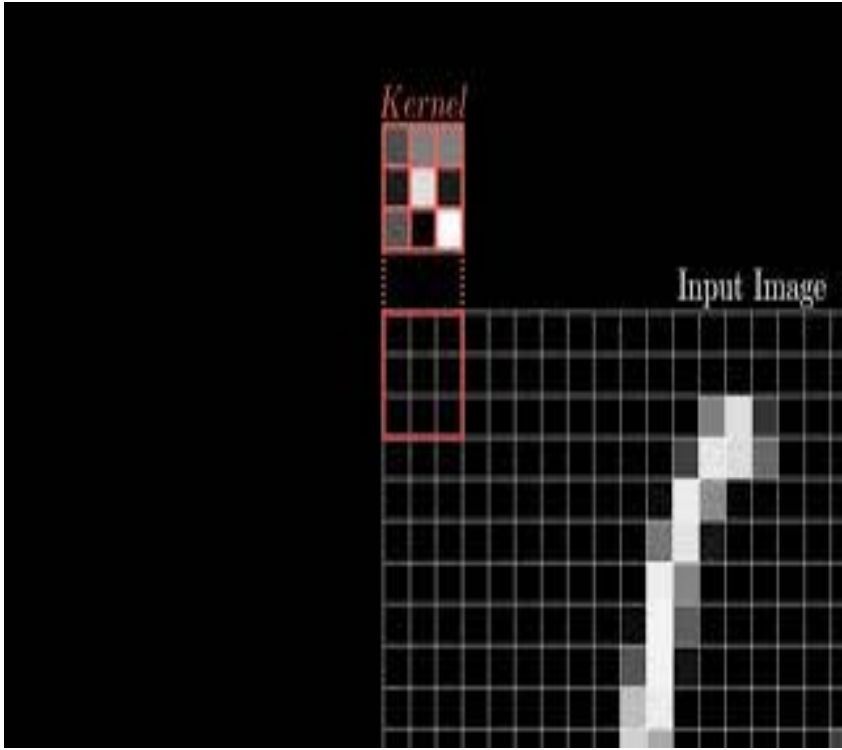
Convolution-Based Models

- **Convolutional-based models** refer to deep learning architectures that use convolutional layers to process and analyze visual data, primarily images.
- These models are Convolutional Neural Networks (CNNs) and their variations, designed to automatically extract spatial and hierarchical features from images.



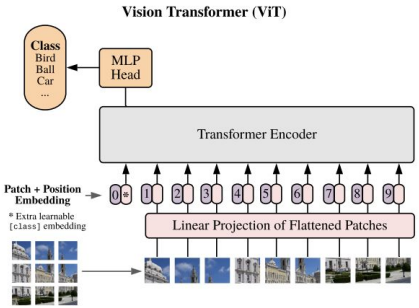
How does Convolution work?

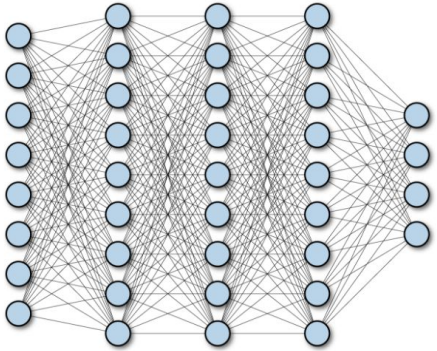
```
    Input: Input_activation, Weight
    Output: Output_activation)
 0: // VLOAD (SRC1, DST1, LENGTH), VSOTRE (DST1, LENGTH)
 1: // COMPUTE_MAC (SRC1, SRC2), COMPUTE_ACT ()
 2: // SET_MAT (M, N, K), SET_TMAT (tm, tn), SET_ACT (Mode)
 3:
 4: // Set matrix/tiled matrix (SET_MAT/SET_TMAT) parameter, activation function
 5: SET_MAT (M, N, K), SET_TMAT (tm, tn), SET_ACT (0) // ReLU (0), ReLU6 (1)
 6:
 7: // Compute copy size (input, weight, output)
 8: // Copy data considering buffer size, matrix/tiled matrix parameter
 9: offsetm=floor(MAXinput/tm/K), offsetm=floor(MAXweight/tn/K)
10: sizeinput=tm*k*Offsetm, sizeweight=tn*k*Offsetn, sizeoutput=tm*tn*Offsetm*Offsetn
11:
12: for(io=0; io<M; io=io+tm*offsetm) // Activation/weight_Buffer (0x00/0x01)
13:  VLOAD (&Input_activation[io*sizeinput], 0x00, sizeinput) // Load activation
14:  for(jo=0; jo<N; jo=jo+tn*offsetn)
15:    VLOAD (&Weight[jo*sizeweight], 0x01, sizeweight) // Load weight
16:
17:    // Matrix multiplication and Activation function
18:    COMPUTE_MAC (0x00, 0x01), COMPUTE_ACT ()
19:
20:    // Store output from accumulation buffer to DRAM
21:    VSTORE (&Output_activation[io*sizeoutput], sizeoutput)
22: end for(io, jo)
```



Non-Convolution-Based Models

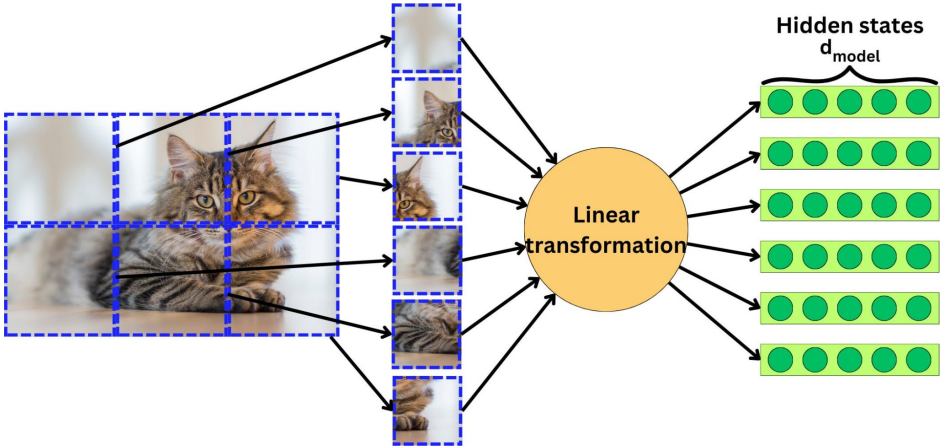
- **Non-convolutional-based models** refer to deep learning architectures that do not use convolutional layers for feature extraction.
- Instead, they rely on other mechanisms such as fully connected layers (MLP Based), transformers, or recurrent architectures to process images.





Vision Transformer

- **Vision Transformer (ViT)** is a deep learning model that applies transformer-based architectures (originally designed for natural language processing) to image recognition tasks.
- Unlike Convolutional Neural Networks (CNNs), ViTs rely on self-attention mechanisms to capture relationships between different parts of an image.



Computer Vision: Image Classification

Vision Transformer

Based On “**Attention Is All You Need**” Paper

Attention Is All You Need

Ashish Vaswani[*]
Google Brain
avaswani@google.com

Noam Shazeer[*]
Google Brain
noam@google.com

Niki Parmar[*]
Google Research
nikip@google.com

Jakob Uszkoreit[*]
Google Research
usz@google.com

Llion Jones[*]
Google Research
llion@google.com

Aidan N. Gomez[*] [†]
University of Toronto
aidan@cs.toronto.edu

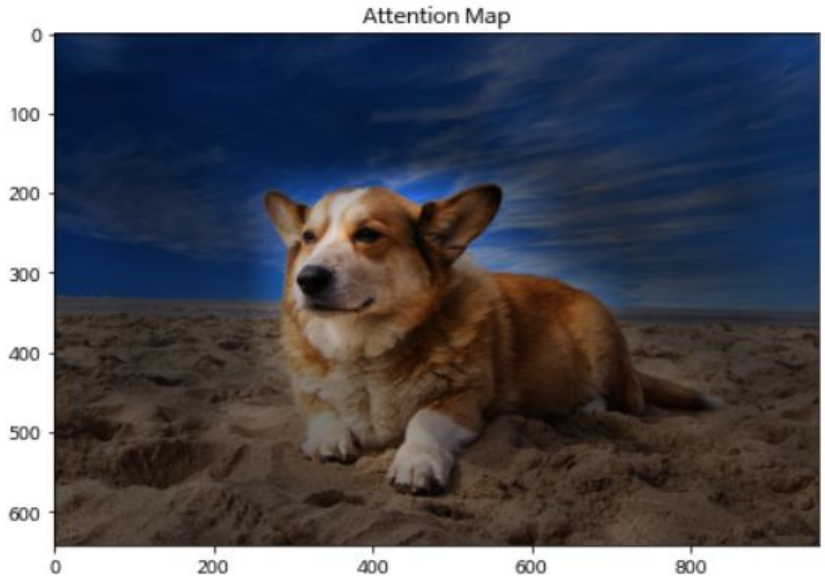
Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

Illia Polosukhin[*] [‡]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

Vision Transformer



Notice how the model only focuses on the dog's head.

Guide to use Vision Transformers

Source: https://keras.io/examples/vision/image_classification_with_vision_transformer/

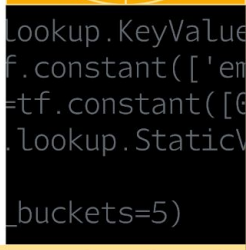
Google Developer Groups

Google Developer Groups
On Campus • Telkom University Bandung

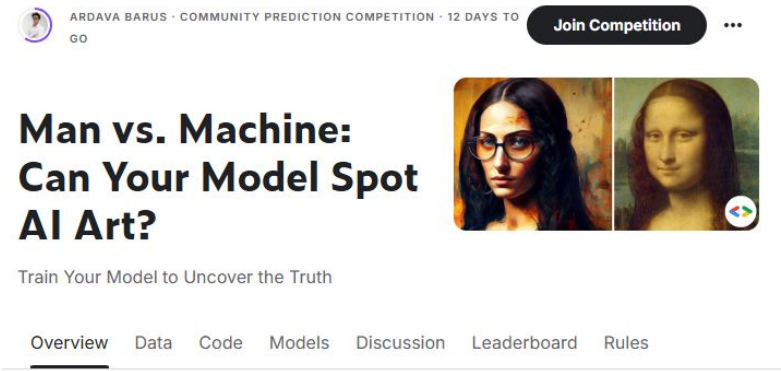
Hands On

Dataset

ristek.link/covid19-img-data



Assignment



Kaggle Competition! 👋
Submit your best deep learning predictions and compete with other participants!

- Can only submit 4 submissions in a day!

  After reaching this limit, participants will need to wait until the next day

Competition Source: ristek.link/GDGoC_Tel-U_MLCompetition_2

Google Developer Groups

Any Question ?

Google Developer Groups

Let’s Connect!

- **Instagram: @rdavaa_**
- **LinkedIn:** www.linkedin.com/in/ardava-barus