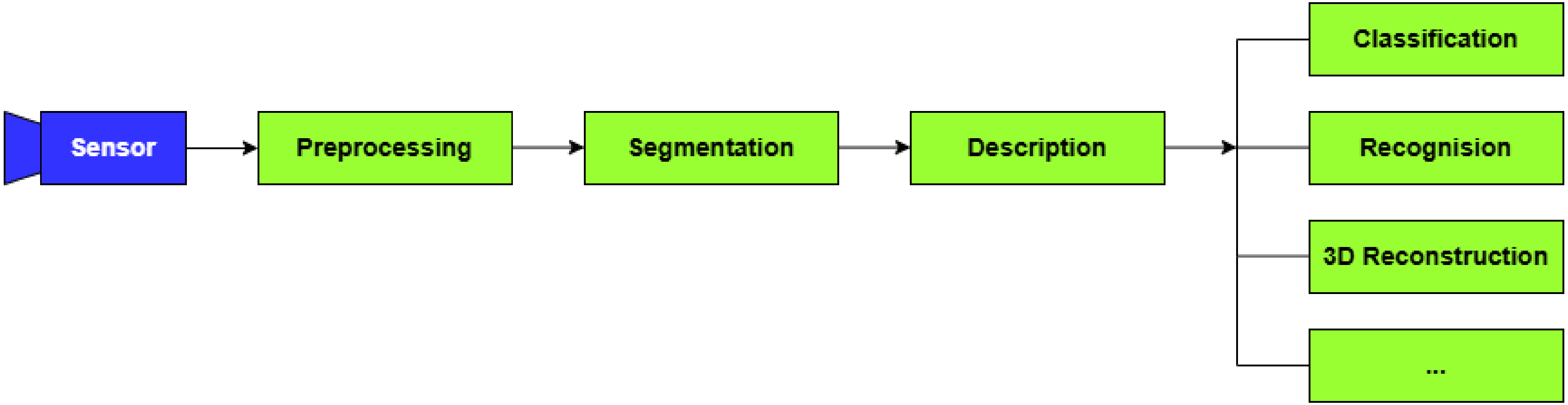
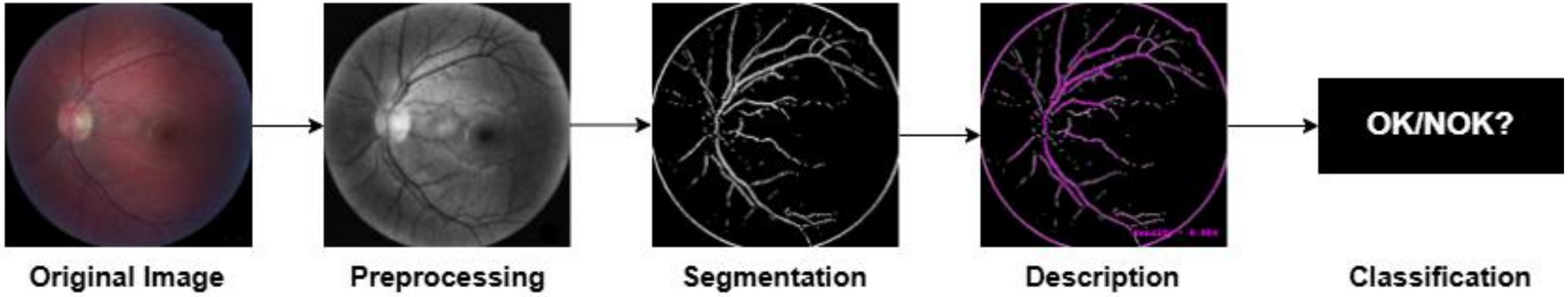
11 – Advanced Computer Vision for Robotics

Robotics and Computer Vision
BPC-PRP

Ing. Petr Šopák
Brno University of Technology
2025


Sensor
Preprocessing
Segmentation
Description
Classification
Recognision
3D Reconstruction
...

Example:

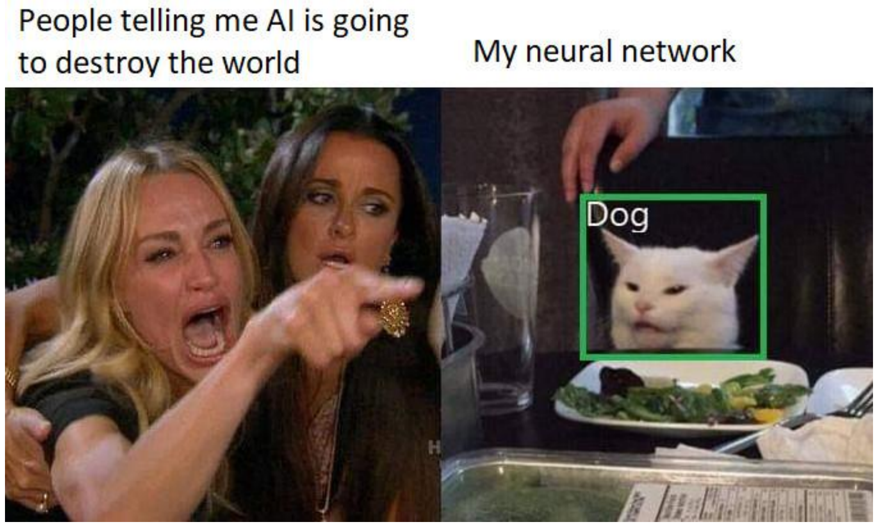

OK/NOK?
Original Image
Preprocessing
Segmentation
Description
Classification

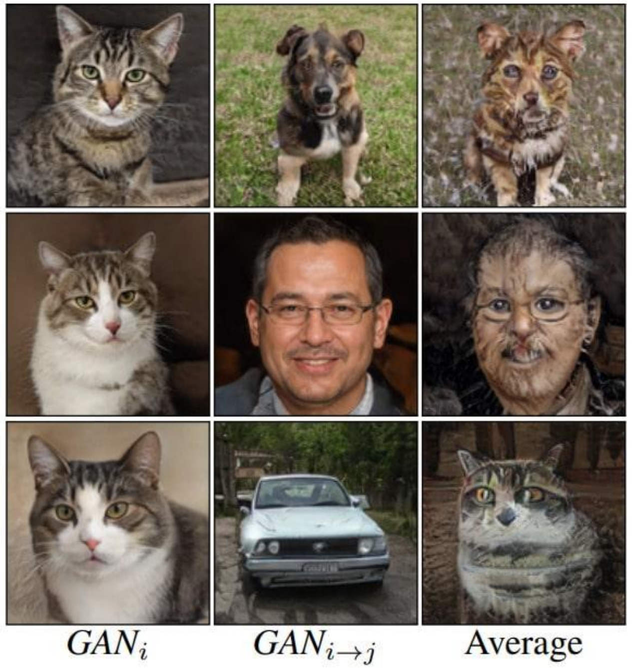
- Real-world robotics demands perception that is reliable, adaptive, and fast.

- **Disadvantages:**
  - Manual Feature engineering
  - Fragility (lighting, perspective, occlusions and noise)
  - Poor generalization
  - Scalability
  - Limited Robustness
  - Real-time Constraints

What will we learn today?

- **Convolutional Neural Networks**
- **CNN – Basic and Extended Architectures**
- **CNN – Uses in Robotics and other fields**
- **Beyond CNN: Advanced CV Techniques**
  - **Visual SLAM**
  - **GAN, Diffusion Models**
  - **3D reconstruction (Structure-from-Motion)**
  - **Multi-modal Vision Systems**





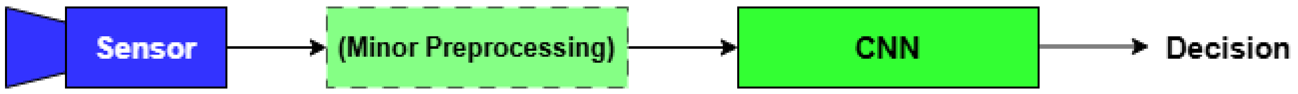
What is Advanced Computer Vision?

Introduction to Advanced Computer Vision and CNN applications.

What is Advanced Computer Vision?

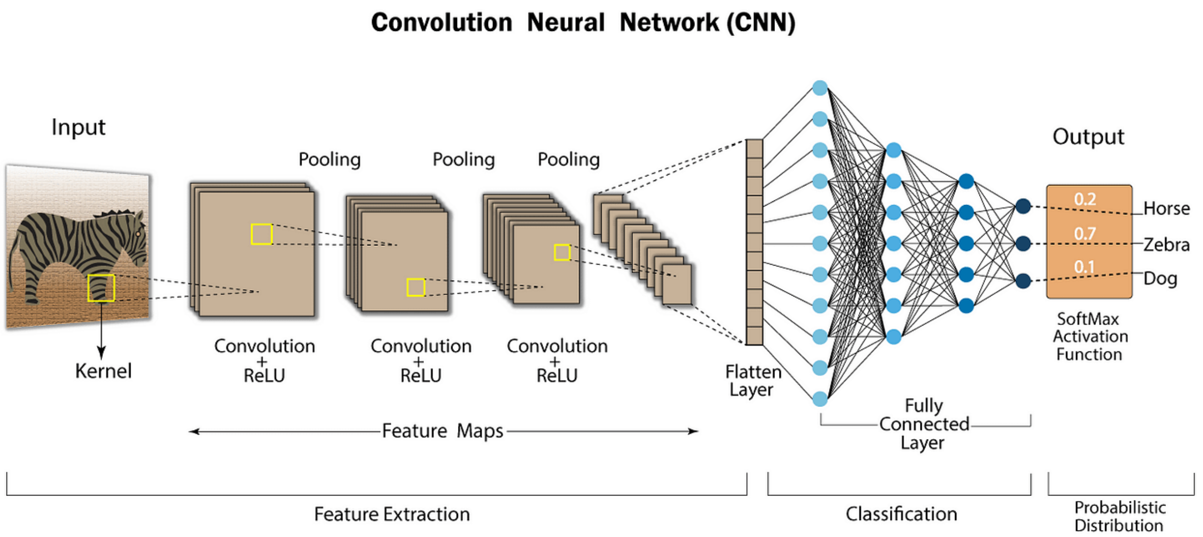
- Using **Deep Learning**
- Autonomously extract relevant features from data
- Capable of generalizing to new scenarios

- **Advantages:**
  - Automatic Feature learning
  - Higher Performance
  - Adaptivity
  - Complex tasks
- **Disadvantages:**
  - Large training dataset
  - Sensitivity to Bias, Overfitting, Difficult Interpretability



Convolutional Neural Networks

- Optimized **for image processing**
- Leverages image properties
  - Local dependencies
  - Translational invariance
- **Lower Layers** detects Simple Features (Edge, Colors)
- **Higher Layers** combines features into more complex structures (entire object or parts)
- Automatically learns which features are important



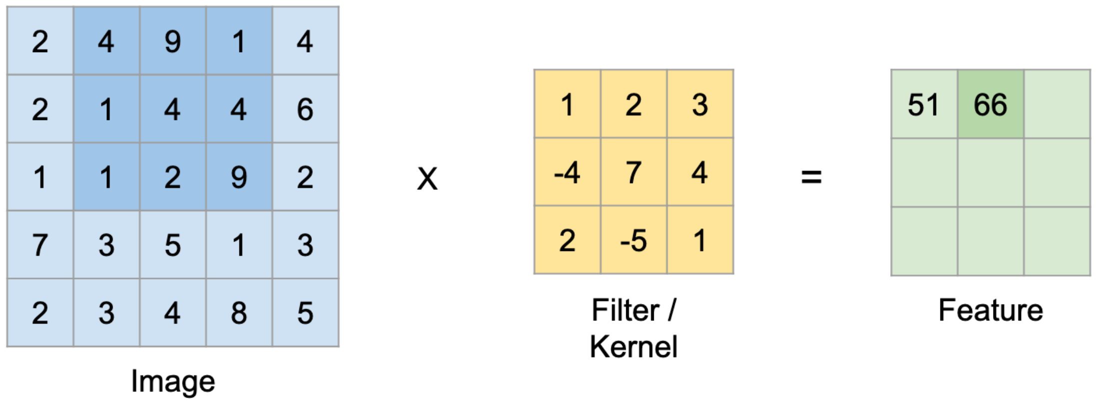
1. Shivam. „ Convolutional Neural Networks : Understand the Basics“. Analytics Vidhya, 2021. [online]. Available: Convolutional Neural Networks | Understand the Basics of CNN

CNN - architecture

- **Convolutional Layers:**
  - Applying small filters (e.g. 3x3) to input data
  - Detects basic patterns like edges, corners, textures
  - Using **2D Convolution operation:**

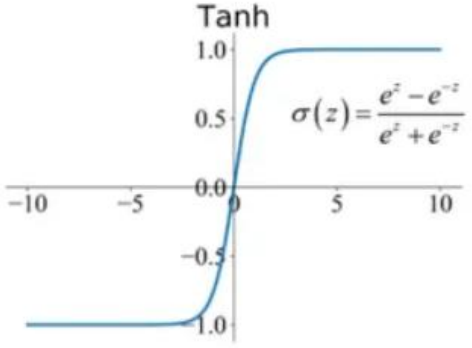
$$Y(i,j) = \sum_{m}\sum_{n} K(m,n) \times X(i+m, j+n)$$

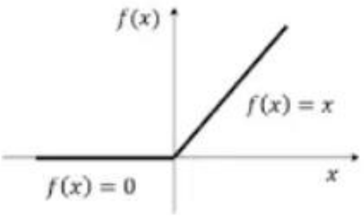
  - **Parameters:**
    - **Stride** – How many pixels the filter moves
    - **Padding** – Adding extra pixels around the input
    - **Number of Filters** – Detect different types of features



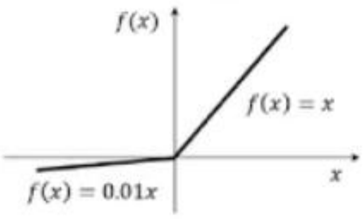
1. Shivam. „ Convolutional Neural Networks : Understand the Basics“. Analytics Vidhya, 2021. [online]. Available: Convolutional Neural Networks | Understand the Basics of CNN

- **Activation Function Layers:**
  - Applying a **non-linear function** to the output of the convolutional layer
  - Allows CNN to learn complex patterns
  - Linear layers can only model straight lines or planes – not complex decision boundaries
  - Activation layers bend the feature space
  - **Common Activation functions:**
    - ReLU
    - Sigmoid
    - Tanh
    - Leaky ReLU





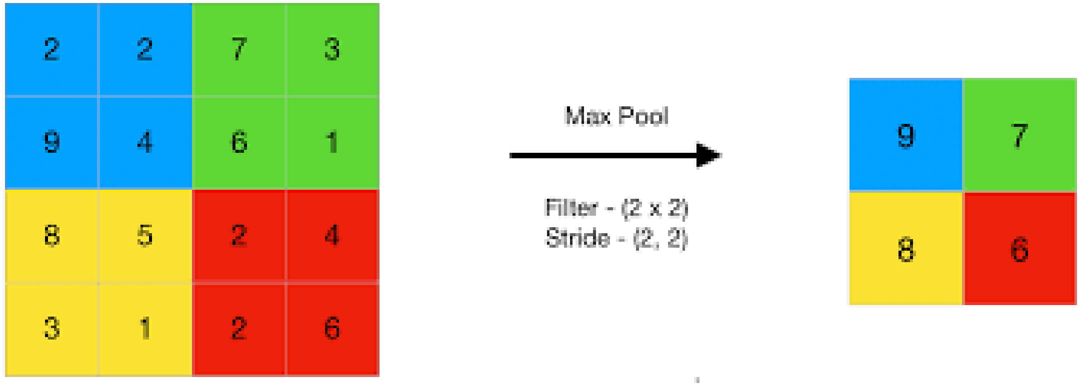
ReLU activation function



LeakyReLU activation function

1. Vishnuam. „Activation Functions in Convolutional Neural Networks (CNN)“. Medium, 2024. [online]. Available: Activation Functions in Convolutional Neural Networks (CNN) | by Vishnuam | Medium

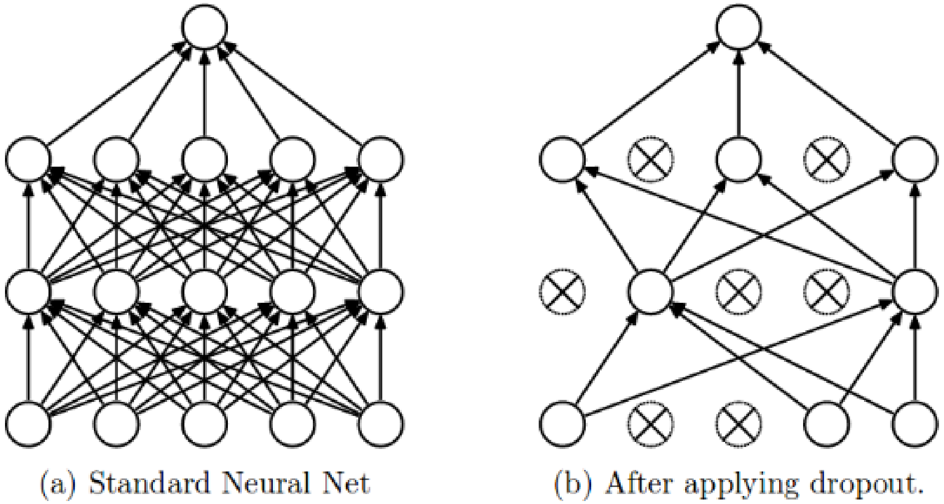
- **Pooling Layers:**
  - Reduces the spatial dimensions (width and height) of feature maps
  - Helps decrease the number of parameters and computations
  - Provides translation invariance by summarizing feature responses in local neighborhoods
  - Prevents overfitting by reducing the sensitivity to small shifts and distortions
- **Common Pooling operations:**
  - Max Pooling
  - Average Pooling
  - Global Average Pooling

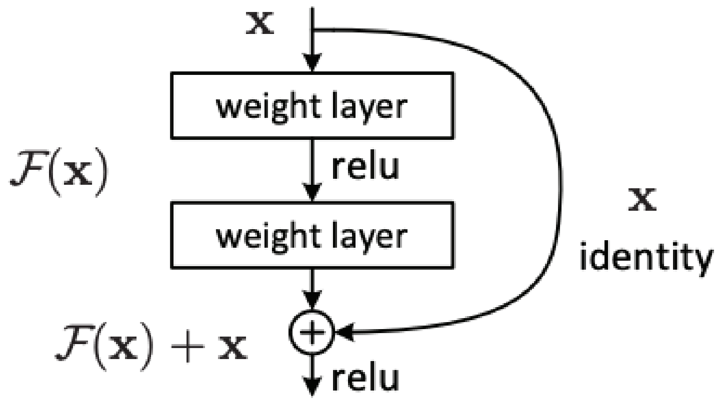


1. Shivam. „ Convolutional Neural Networks : Understand the Basics“. Analytics Vidhya, 2021. [online]. Available: Convolutional Neural Networks | Understand the Basics of CNN

- **Fully Connected Layers:**
  - Each neuron is connected to every neuron in the previous layer
  - Combines extracted features to make final decisions
  - Typically used at the end of CNN to map features into output classes
- **Softmax Layer:**
  - Converts the raw outputs into probability distribution over classes
  - Helps interpret the model's output as class probabilities
  - **Common Use:** Last layer in classification - choose the most probable class

- **Extended Layers:**
  - **Batch Normalization Layers**
    - Normalizes the activation (neuron outputs) within a mini-batch during training
    - Stabilizes and speeds up training
    - Reduces the sensitivity to initialization
  - **Dropout Layer**
    - Randomly sets some neuron outputs to zero
    - Reduces overfitting and increases the robustness
  - **Residual Connections**
    - Directly add the input of a layer to its output
    - Train deeper networks easily
  - **Global Average Pooling (GAP)**
    - Reduces the number of parameters compared to FC layer

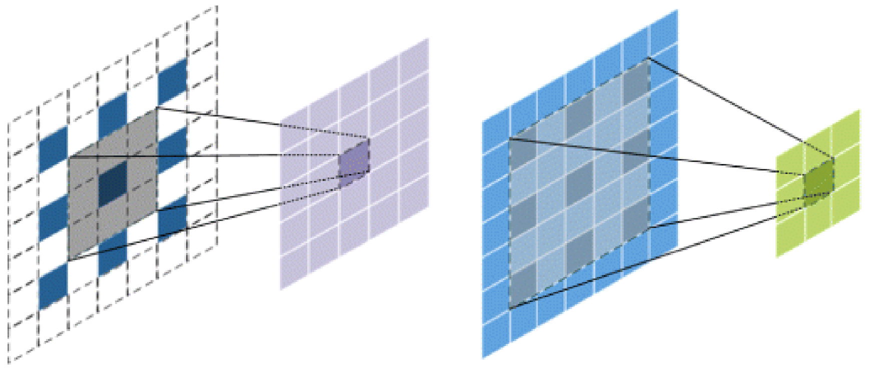




1. Yadav, H. „Dropout in Neutral Networks“. Towards, 2022. [online]. Available: Dropout in Neural Networks | Towards Data Science
2. Yadav, H. „Residual Blocks in Neutral Networks“. Towards, 2022. [online]. Available: Dropout in Neural Networks | Towards Data Science

- **Extended Layers:**
  - **Dilated (Atrous) Convolution**
    - Increases the receptive field without increasing computation
    - Inserts gaps (zeros) between filter elements
    - Commonly used in segmentation and dense prediction

- **Attention Mechanism**
  - Dynamically focuses on relevant parts of the input
  - Captures global relationships between any elements
  - Computes weighted combinations of the input features

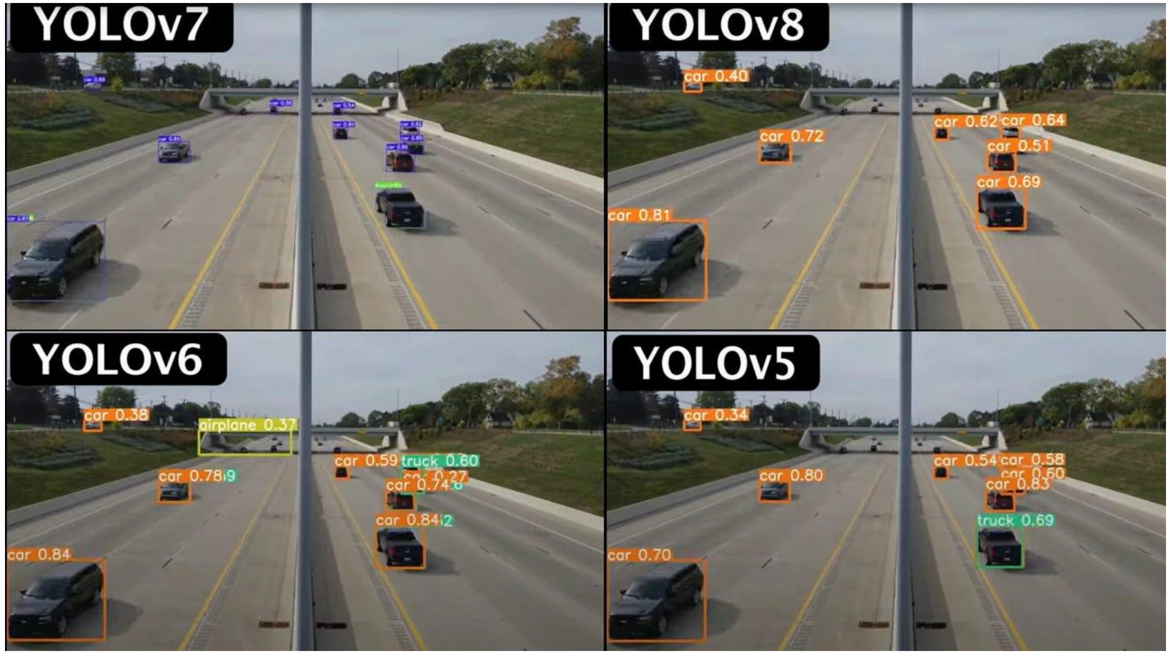


(a) Deconvolution operation

(b) Dilated convolution operation

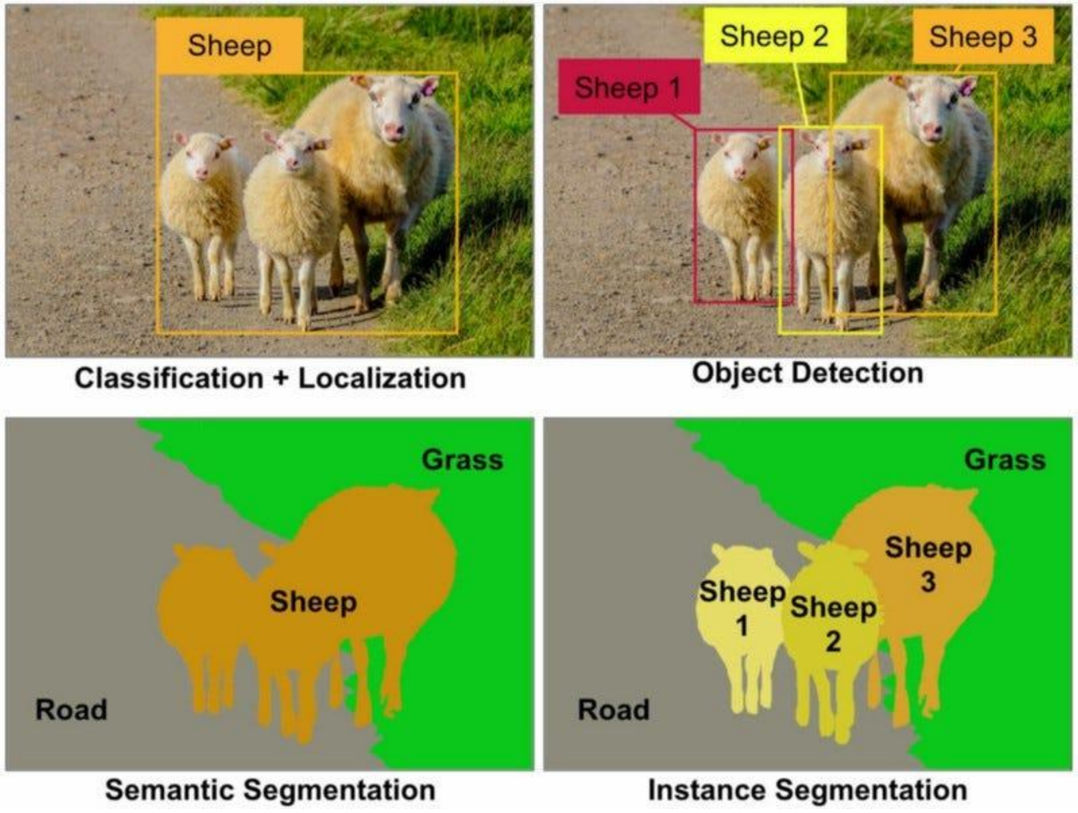
1. Qin, P, ed. „Research on improved algorithm of object detection based on feature pyramid". Springers, 2019. [online]. Available: 10.1007/s11042-018-5870-3

- **Producing where** the objects are located (bounding box) and **what** they are (label)
- Essential for scene understanding, obstacle detection, grasping objects, autonomous navigation and more
- **Common Models:**
  - **YOLOv8** – CNN-based real-time detector
    - Divides the images into a grid and predicts bounding boxes and classes
  - **DETR** – transformer-based end-to-end object detector
    - Object detection as a set prediction problem
  - **Sparse R-CNN**
    - Small set of learnable queries to predict with sparse supervision



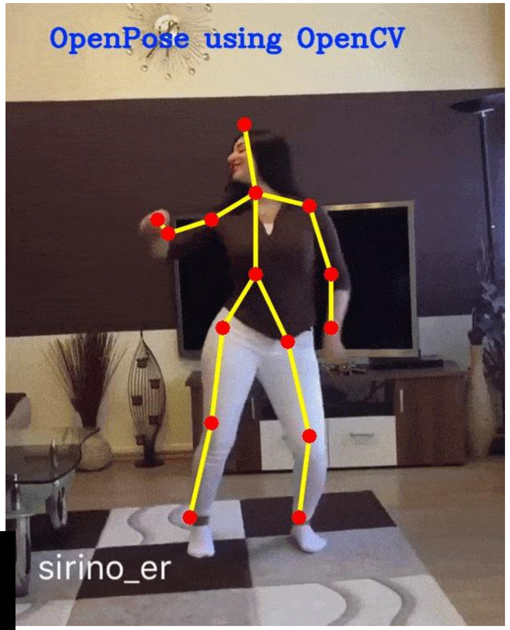
1. Darmadi, D, ed. „Traffic Counting using YOLO Version-8“. ASTONJADRO, 2024. [online]. Available: 10.32832/astonjadro.v13i1.14489
2. Rath, R. S. „Train DETR on Custom Dataset“. DEBUGGER CAFE, 2023. [online]. Available: Train DETR on Custom Dataset

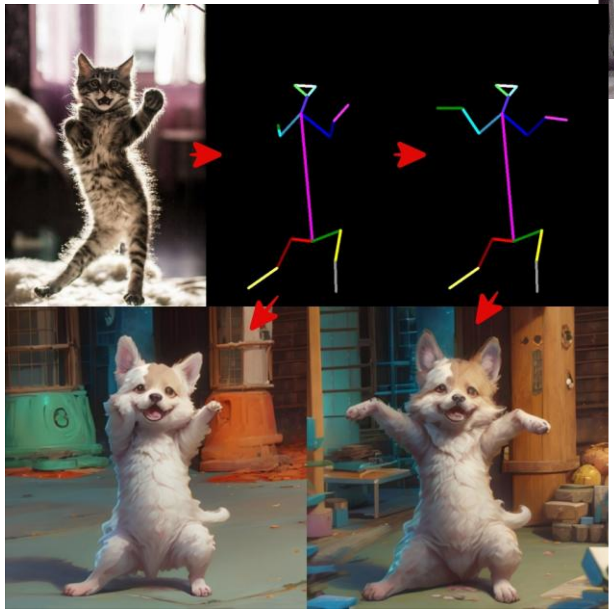
- **Semantic segmentation:** Assigns a class label to every pixel
- **Instance segmentation:** Separates different objects of the same class
- **Common Models:**
  - **Mask2Former** – transformer-based universal model
  - **DeepLabV3** – CNN with Atrous (dilated) convolution for multi-scale semantic segmentation
  - **SAM** – prompt-based segmentation



1. Murali, N. „Image Classification vs Semantic Segmentation vs Instance Segmentation“. Medium, 2021. [online]. Available: [Image Classification vs Semantic Segmentation vs Instance Segmentation | by Nirmala Murali | Medium]

- **Detect keypoints** of objects (usually human joints)
- Estimate body or hand poses in 2D or 3D images
- **Commonly used:** Human-robot interaction, sport analysis
- **Common Models:**
  - **OpenPose** – Open-source system – keypoint detection, using confidence maps and affinity fields
  - **MediaPipe Pose** – real-time pose estimator optimized for mobile devices





1. Aiditor. „Animal Openpose | AI 그림은 동물도 춤추게 한다?“. AIPOQUE, 2024. [online]. Available: Animal Openpose | AI 그림은 동물도 춤추게 한다? - AIPOQUE

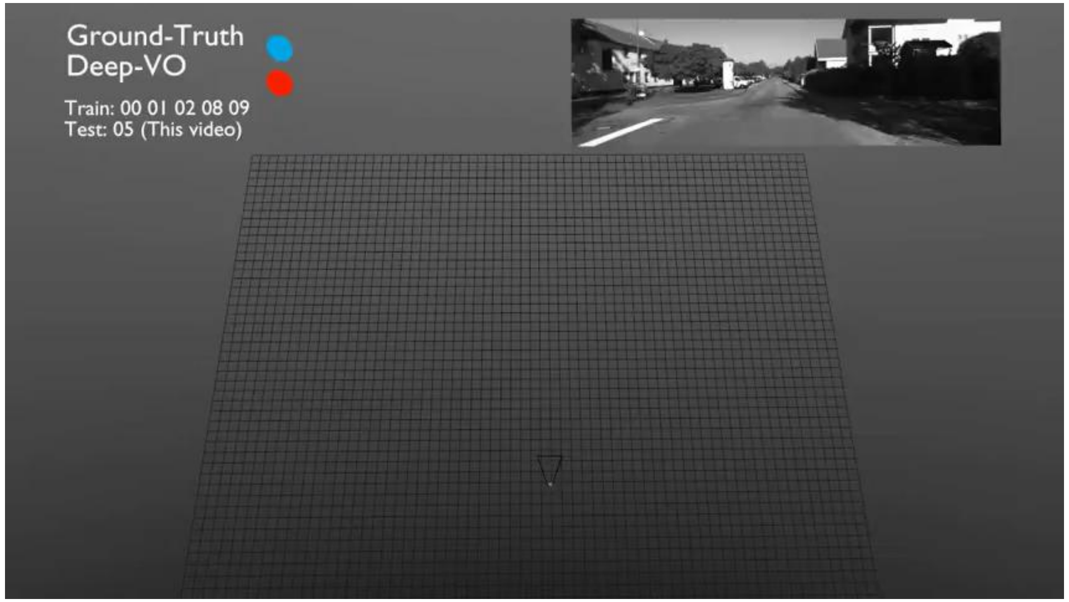
CNN – Depth Estimation

- Predict a depth value (distance to the camera) for every pixel.
- Generate relative or absolute depth maps from monocular images.
- **Commonly used:** 3D obstacle avoidance for robots, scene reconstruction, AR/VR depth sensing
- **Common Models:**
  - **MiDaS** Trained on diverse datasets to generalize monocular depth estimation
  - **DPT (Dense Prediction Transformer)** – prediction tasks like depth and segmentation

1. Intel IDS. „MiDaS models for computing relative depth from a single image. ". AIPOQUE. [online]. Available: [MiDaS | PyTorch]

CNN – Visual Odometry

- Estimate camera movement based on consecutive image frames
- Track relative pose changes without external localization like GPS
- **Commonly used:** Robot navigation, drone flight stabilization, autonomous driving
- **Common Models:**
  - **DeepVO** - CNN + RNN to directly predict ego-motion from image sequences
  - **DeepTAM** – Combines learned feature maps with classical tracking and mapping ideas

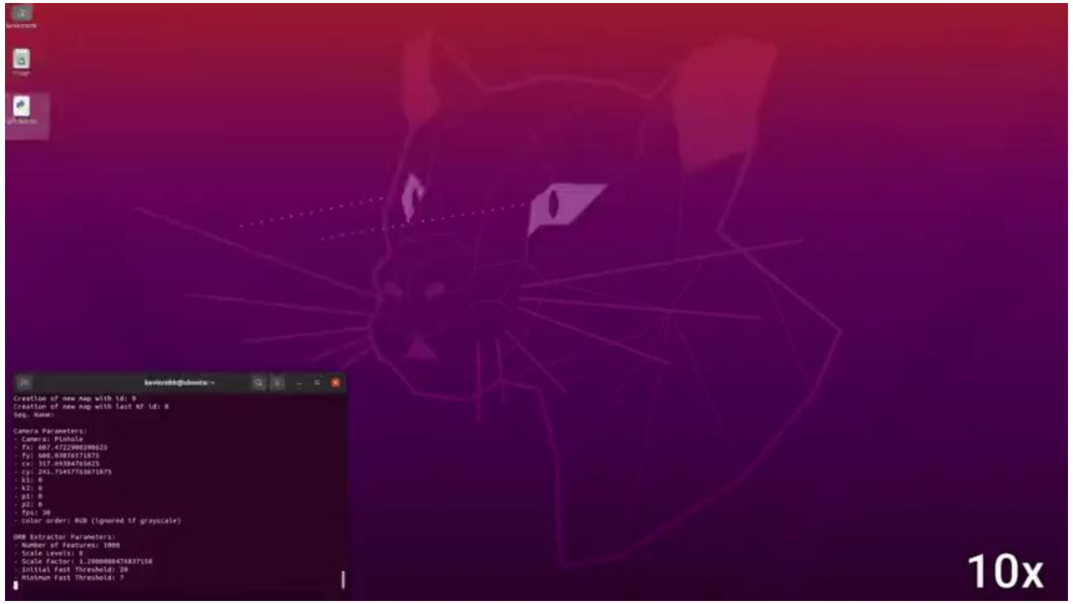


1. Cajamarca. M. „DeepVO Odometry Example Clip. “. Youtube, 2018. [online]. Available: DeepVO Odometry Example Clip

Is Advanced CV only CNN and Deep Learning?

Exploring SLAM, 3D Reconstruction, GANs, and more.

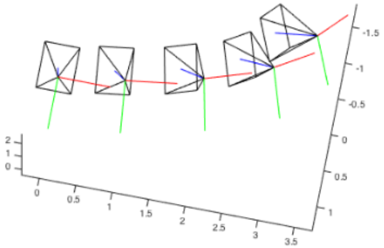
- Builds a map of an unknown environment while simultaneously estimating the robot’s location.
- **Uses:** Autonomous robot navigation, AR/VR tracking, drone mapping
- **Implementation Steps:**
  - Feature extraction (e.g., ORB)
  - Feature matching between frames
  - Motion estimation (pose)
  - Map update (3D landmarks)
  - Loop closure detection and optimization
- **Common Models:**
  - **ORB-SLAM3** (feature-based), **LSD-SLAM** (direct), **DROID-SLAM** (deep-learning + direct)

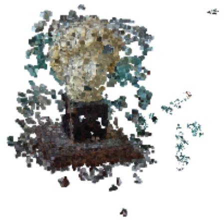


1. Kevin Robbs Designs. „ORB SLAM 3 demo with RealSense 435 monocular camera. “. Youtube, 2022. [online]. Available: (4) ORB SLAM 3 demo with RealSense 435 monocular camera - YouTube

Structure-from-Motion (SfM)

- Recovering the 3D structure of a scene and camera poses from multiple 2D images
- **Uses:** 3D scene reconstruction, photogrammetry
- **Implementation Steps:**
  - Detect and match features between images
  - Estimate relative camera poses
  - Triangulate 3D points to build a sparse 3D structure
  - Perform global optimization (bundle adjustment)
- **Common Models:**
  - **COLMAP** (feature-based), **OpenMVG**

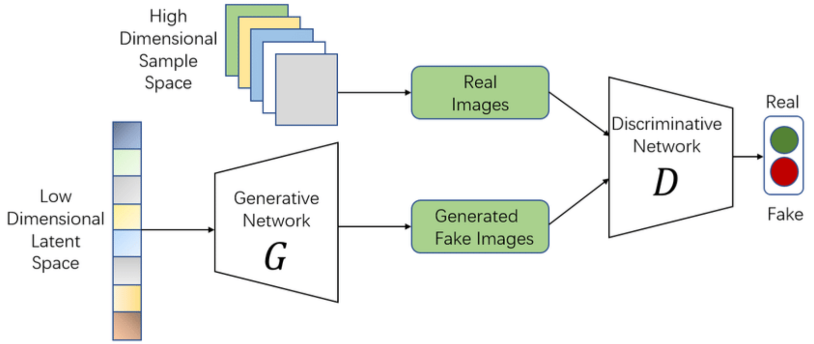


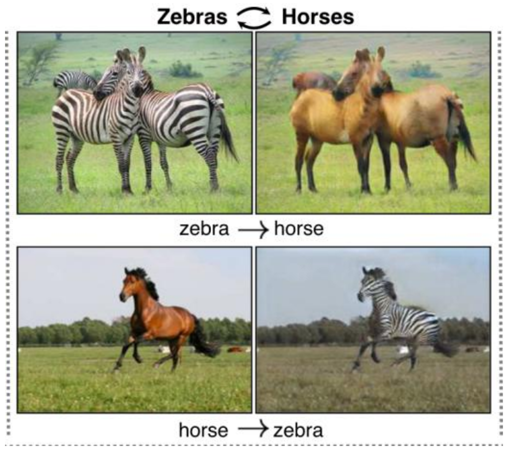


1. Wu. K. „Structure from Motion Tutorial“. Github, 2024. [online]. Available: Kai Wu | Structure from Motion Tutorial

Generative Adversarial Networks (GANs)

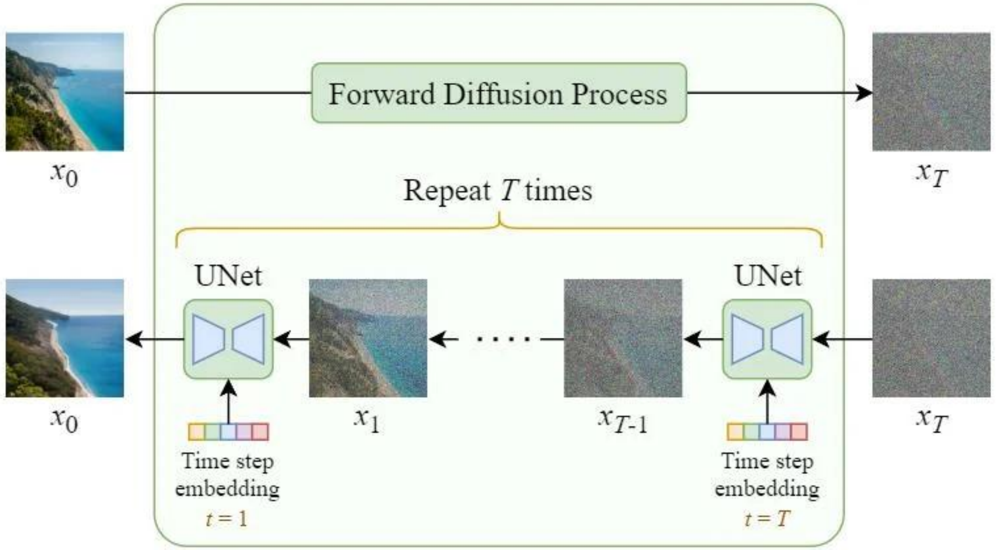
- Two networks (generator and discriminator) compete, resulting in realistic synthetic image generation
- The generator tries to fool the discriminator by producing fake images; the discriminator tries to detect fakes
- **Uses:** Data augmentation, Image-to-image translation
- **Implementation Steps:**
  - Train the generator to create realistic images
  - Train the discriminator to distinguish real from fake
  - Alternate optimization (adversarial learning)
- **Common Models:**
  - **CycleGAN** (unpaired image-to-image translation), **Pix2Pix** (paired image-to-image translation)

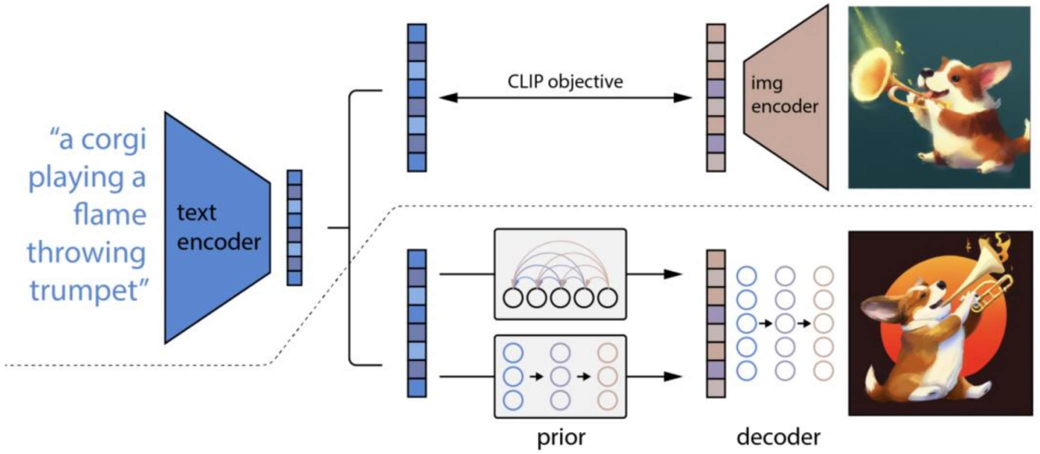




1. Junyaz „CycleGAN". Github, 2017. [online]. Available: junyanz/CycleGAN: Software that can generate photos from paintings, turn horses into zebras, perform style transfer, and more.

- Generate high-quality images by gradually denoising random noise, conditioned on inputs like text
- Noise Addition (**Forward Process**) x Noise Removal (**Reverse Process**)
- **Uses:** Text-to-image generation, visual content creation for AR/VR
- **Implementation Steps:**
  - Encode the prompt (optional for conditional generation)
  - Start from random noise
  - Perform multiple denoising steps according to the learned model
- **Common Models:**
  - **Stable Diffusion**, **DALL-E 2**

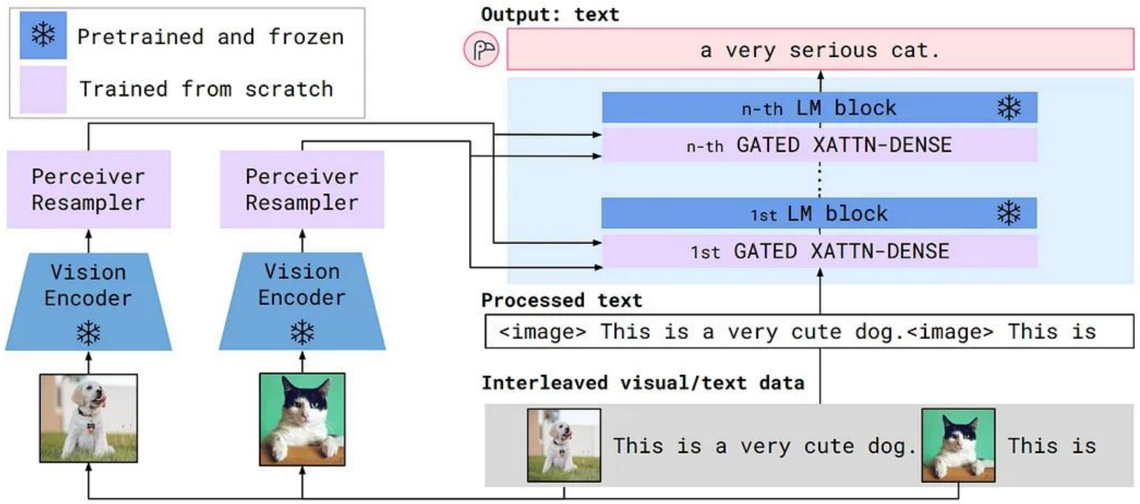


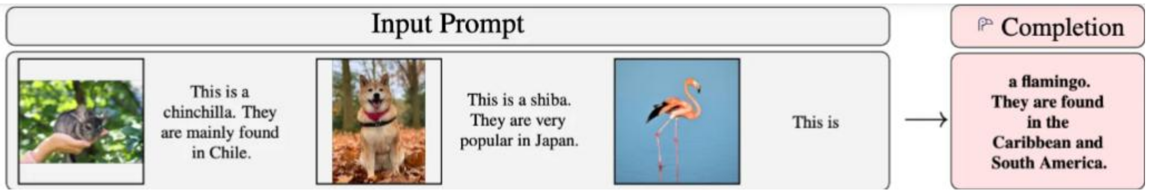


1. Steins. „Stable Diffusion Clearly Explained!". DORAVEN, 2023. [online]. Available: Stable Diffusion Clearly Explained! - CodoRaven.
2. Spektor, I. „From DALL·E to Stable Diffusion: how do text-to-image generation models work?!". tryolabs, 2022. [online]. Available: From DALL·E to Stable Diffusion: how do text-to-image generation models work? | Tryolabs

Multi-Modal Vision models

- Models that process and combine **visual** (images, videos) and **textual** (language) information
- **Uses:** Robotic perception + language understanding
- **Implementation Steps:**
  - Use a vision encoder to process the image
  - Use a language encoder to process the text
  - Train both encoders so that matching image-text pairs are close together in the embedding space
  - Use learned embeddings for retrieval, classification, or generation.
- **Common Models:**
  - **CLIP** (Contrastive Learning of Image and Text), **Flamingo** (Few-shot vision-language model)





1. Palucha, S. „Understanding DeepMind's Flamingo Visual Language Models“. Medium, 2024. [online]. Available: Understanding DeepMind's Flamingo Visual Language Models | by Szymon Palucha | Medium

Summary

- **Advanced Computer vision – its challenges and applications**
- Convolutional Neural Networks
  - Architecture - layers
  - Applications
- Further Techniques in Advanced Computer Vision
  - Visual SLAM
  - 3D reconstruction – Structure-from-Motion
  - GAN
  - Diffusion Models
  - Multi-Modal Vision Systems

Summary

- Advanced Computer vision – its challenges and applications
- **Convolutional Neural Networks**
  - **Architecture - layers**
  - **Applications**
- Further Techniques in Advanced Computer Vision
  - Visual SLAM
  - 3D reconstruction – Structure-from-Motion
  - GAN
  - Diffusion Models
  - Multi-Modal Vision Systems

Summary

- Advanced Computer vision – its challenges and applications
- Convolutional Neural Networks
  - Architecture - layers
  - Applications
- **Further Techniques in Advanced Computer Vision**
  - **Visual SLAM**
  - **3D reconstruction – Structure-from-Motion**
  - **GAN**
  - **Diffusion Models**
  - **Multi-Modal Vision Systems**

Ing. Petr Šopák
xsopak00@vutbr.cz

Brno University of Technology
Faculty of Electrical Engineering and Communication
Department of Control and Instrumentation

Robotics and AI Research Group