Object Moving by Digit Recognition with Open Manipulator - Group 10

Anton Mauritzon, Alexandre Laborie, Montserrat Olivares, Richard Kuhlmann

Problem Formulation

- **Many sorting problems:**
  - Often sort by number is enough
  - E.g. *warehouse, post office*
  - Number recognition enough (object recognition not needed)
  - Robot most efficient resource
- **Our Goal:**
  - Robot can **move objects autonomous**
    - For detailed information on robot: view "Robot description"
  - Recognizes label (=number) on object
  - -> achieved with neuronal network
  - Object is moved according to label
  - -> sample solution for a warehouse

Related Work

Studies for similar tasks:

- Surgeon assistant robotic arm, looks for surgeon's tools (S. Yin, A.S. Yuschenko, 2020)
- Robotic arm able to identify and pick a specific leaf from plants (K. Ahlin et all., 2016)

Why is CNN good for label recognition?

- The CNN´s **convolutional layers** reduce the high dimensionality of pictures without them losing information.
- A network with huge number of parameters would likely run into **overfitting**
- Large number of parameters make the network stop paying attention to image details, that would be lost but are required for precision

Pseudo Code

1. **Scan object area with camera.**
   a. *If NN detects number→ proceed.*
      Process picture
      Feed into CNN.
      Get number output
   b. *Else scan again in 10 seconds*
2. **Reposition object withOpenMANIPULATOR.**
   a. gripper **grabs object**
   b. Fetches the number's placement position in memory. Computes path to the number's position.
   c. **Move to position** and release the object.
   d. Reset to starting position.

Conclusion

Challenges we faced:

- **Controlling robot (with MoveIT):**
  - No full tutorial -> figure out how on our own
  - Unexpected errors, missing libraries, etc…
  - Many sources outdated -> hard to find what needed
  - Some info on robotis official website are only available for ROS Noetic (Ubuntu 18)
- **Number recognition in real life:**
  - Harder cause of disruptive elements in photos -> developing NN totally different than in assignment

Did we reach our goals?

- Not everything (we defined goals step by step)
- AI part could have had better results without robotic problems (cause we could have spend more time on AI)
- Basic goals were achieved

Flash to get to our github page!

Neural Network 1

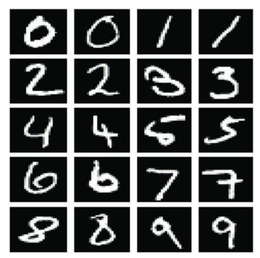
- **Purpose:** Check area for digit
  - Without a digit: No need for recognition -> Nothing to move
- **Structure**: Linear Neuronal Network/MLP
  - ***Input***: 28x28 black and white picture
  - First: Batch normalization
  - Second: Dropout layer
  - ***4 linear layers***, reLu functions
  - ***Output***: 2x1 vector (2 for no digit/digit)
- **Dataset**: "Half MNIST"
  - ***Self-created, only 450 pictures***
  - Half MNIST, half pictures without number
- **Performance**:
  - No validation data -> no exact value
  - Worked ***almost every time*** we tested it

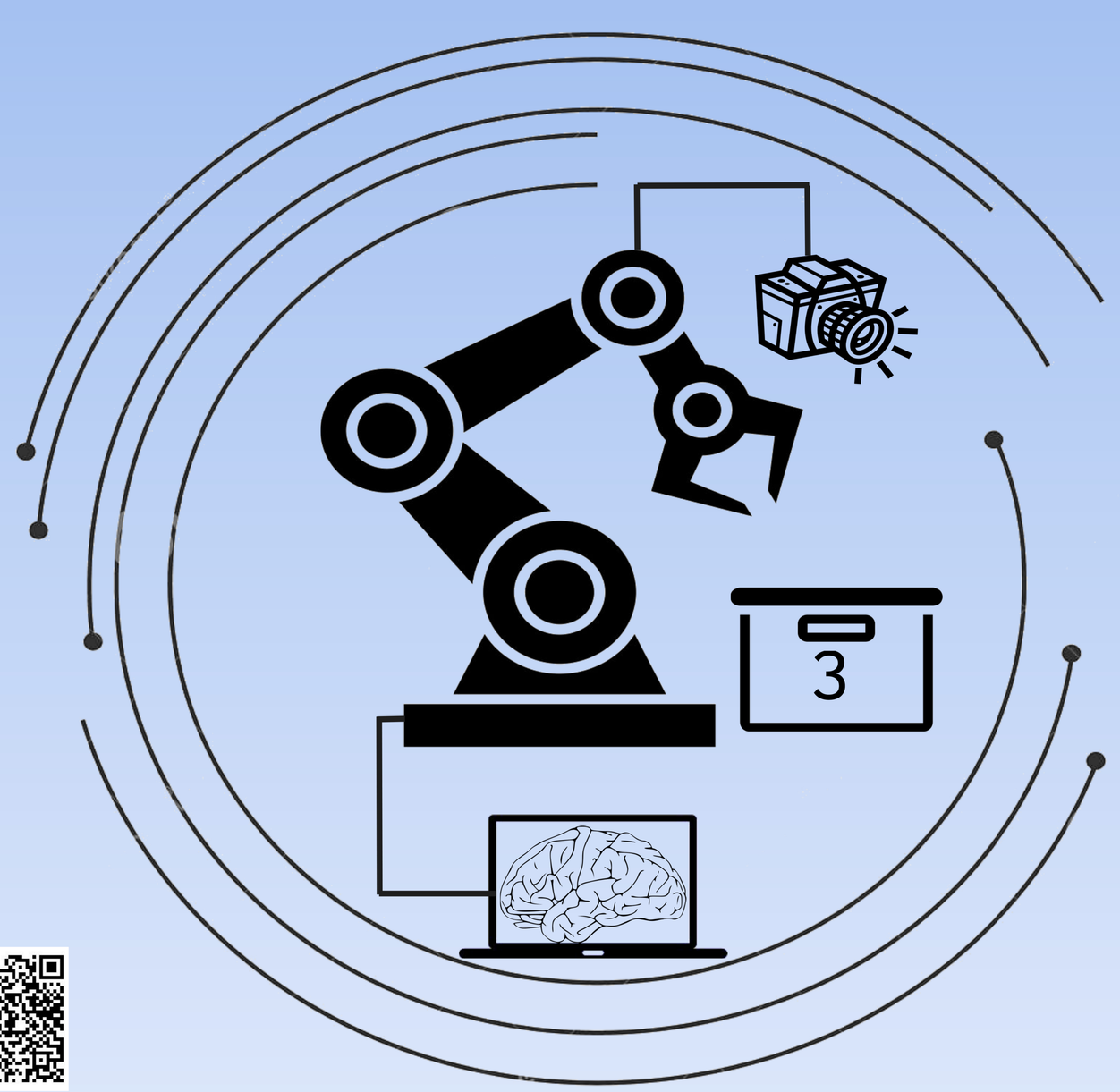
vs

(example pictures)

Neural Network 2

- **Purpose:** Recognize digit:
  - Robot moves object according to labeled box
- **Structure:** CNN
  - **Input:** 28x28 black and white picture
  - **2 convolutional layers, 1 maxPooling**
  - **Batch normalization**
  - **Dropout layer**
  - **3 linear layers**
  - **Output:** 10x1 vector (one position/number)
- **Dataset:** MNIST
  - Data augmentation required (for self created numbers)
- **Performance:**
  - **>99%** (validation accuracy)
  - Also worked for **self created numbers**

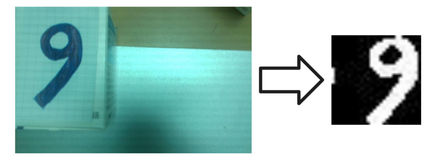




Sources: Scan this code!

Camera & Image Processing

- **Camera:** INTEL Realsense 3D Camera:
  - Takes 3D pictures
  - Requires specific libraries
  - Takes **high-resolution coloured pictures**
    - Network needs **28x28 greyscale**
    - -> processing necessary
- **Image Processing:**
  - 1. Convert to **grayscale**
  - **Find** part with **number**
  - **Downscale**
  - Normalize
  - Convert to **tensor**



Robot description

- **Model**: OpenMANIPULATOR-X
- **Features**:
  - **4 controllable joints**
  - 1 gripper
  - INTEL Realsense **Camera**
    - Mounted on robot
    - Technically not part of it
- **Software**: ROS + MoveIT
  - ROS1 for communication (Laptop-Robot)
  - MoveIT for moving robot
  - pyrealsense2 library for the camera
- **Object**: labeled box