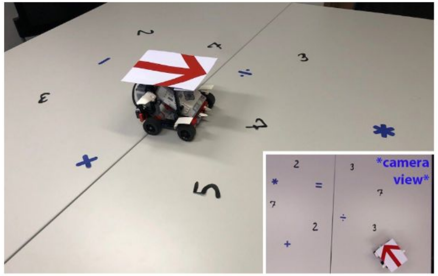
Image Analysis & Pattern Recognition - Special Project

EE-451, Prof. J-P. Thiran

Frank Dupont
Maximilien Gomart

Goal

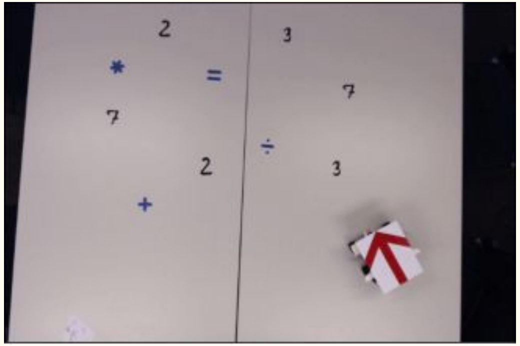


Implementation milestones:

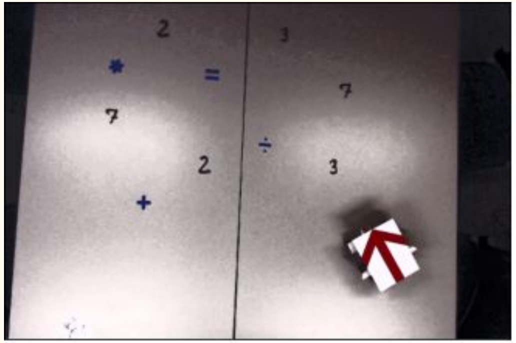
- Detect the positions of the ***digits***, the ***operators*** and the ***red arrow*** and label them
- Track the position of the ***arrow*** while the robot is moving
- Detect when the robot is above a ***digit*** or an ***operator***
- Store the equation as the robot moves
- When the robot reaches the "=" sign, solve the equation and display it

Preprocessing: Enhancing and Masking

- **Enhancing:** Histogram equalization and Image Thresholding



raw image



histogram equalization

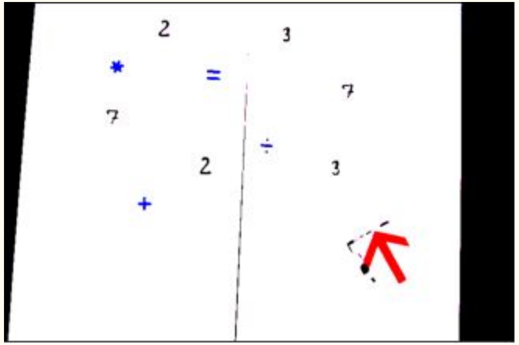
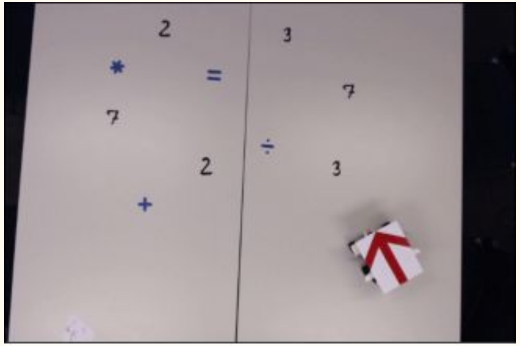


image threshold

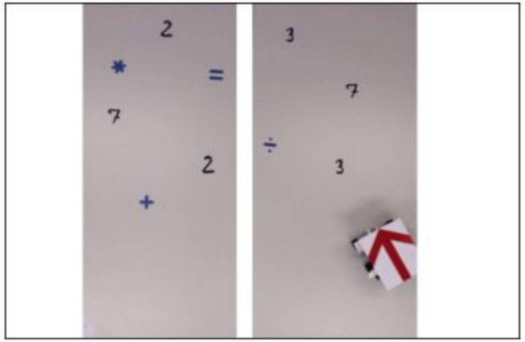
histogram equalization: helps with highlighting the ***blue operators*** and the ***red arrow***
image threshold: helps highlighting black objects, i.e. ***black digits***

Preprocessing: Enhancing and Masking

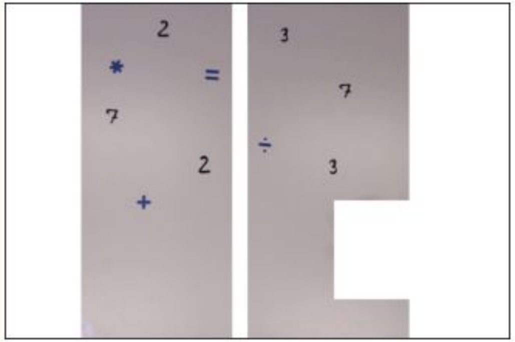
- **Masking :** black borders and robot masks



no masking



border masks

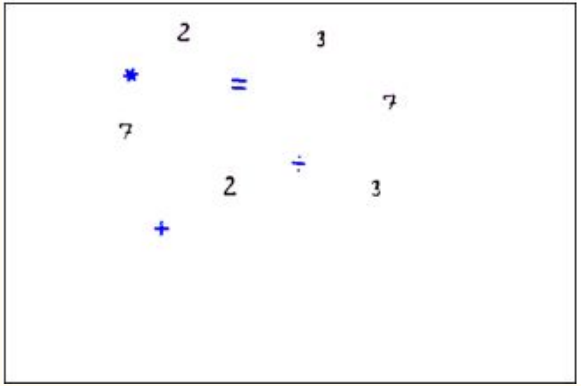


robot mask

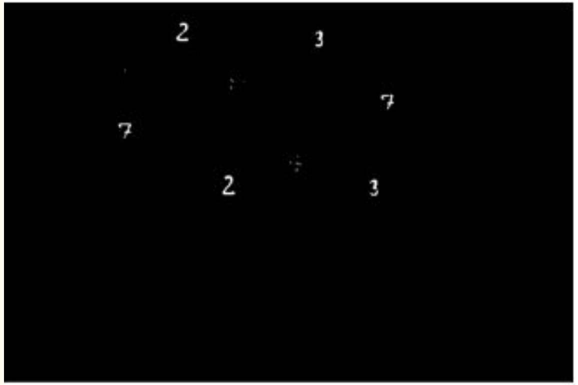
masking: use of OpenCV2 **rectangle** tool
crucial to perform robust detection of the objects

Digit detection

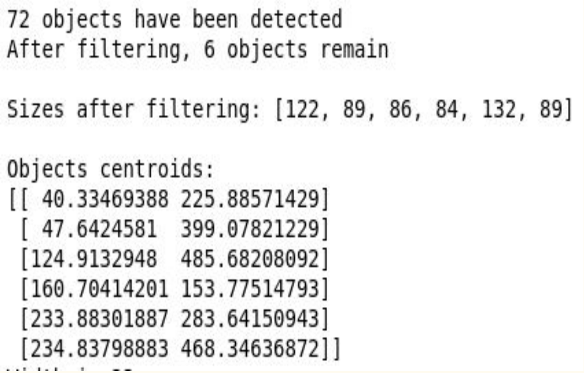
- **Pre-processing, region growing and filtering**



masking + image threshold



“black only” mask



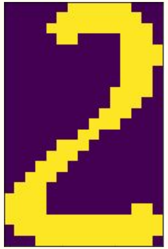
```
72 objects have been detected
After filtering, 6 objects remain

Sizes after filtering: [122, 89, 86, 84, 132, 89]

Objects centroids:
[[ 40.33469388 225.88571429]
 [ 47.6424581  399.07821229]
 [124.9132948  485.68208092]
 [160.70414201 153.77514793]
 [233.88301887 283.64150943]
 [234.83798883 468.34636872]]
```

region growing with size calculation + filtering by size

“black only” mask: use of OpenCV2 *inRange* interval color detection (using HSV colors)
region growing with size: detects objects as lists of points coordinates, also stores their size
if their size is not within a specific interval (typical size of digit [70-130]), then discarded.
centroids of ***digits*** are calculated (gravity centre of points)

Digit detection (continued)

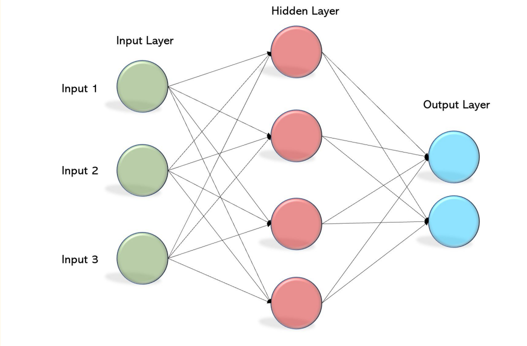
- Processing



detected digit

digit with added borders

rescale to 28x28 images (input for MLP Classifier)



MLP classifier trained on MNIST:

INPUT: 28x28 images, reshaped to 784x1

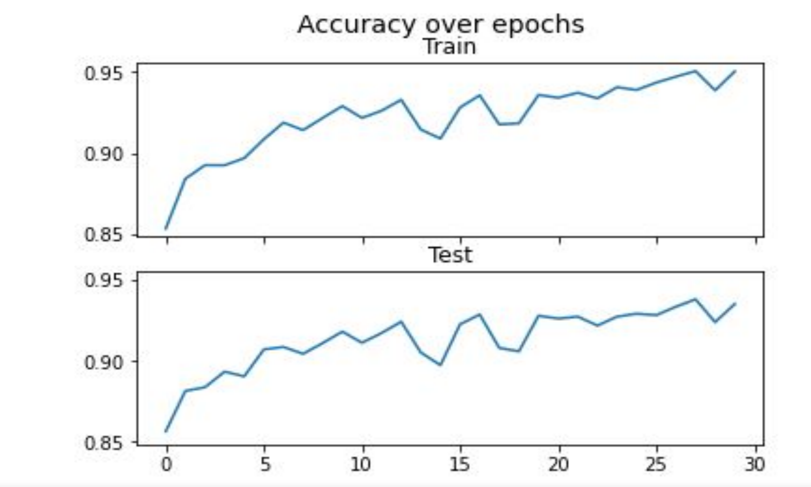
HIDDEN LAYER SIZE: 100

OUTPUT: digit between 0 & 9

Digit detection (continued)

- MLP Classifier training and testing

Training with 30 epochs, batch size of 256



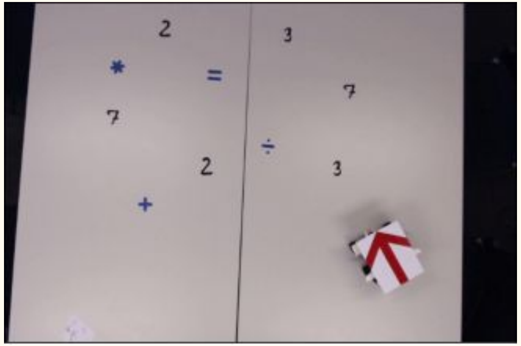
Training accuracy ~ 95%

Testing accuracy ~ 93%

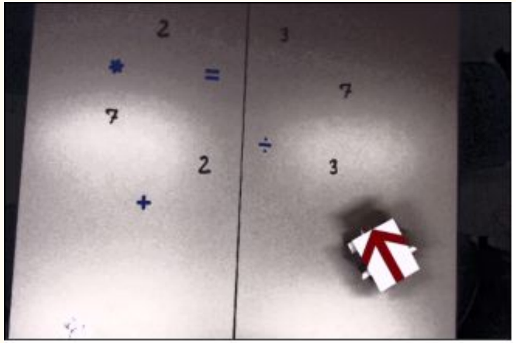
RESULTS: 5/6 digits correctly classified in average

Operators detection

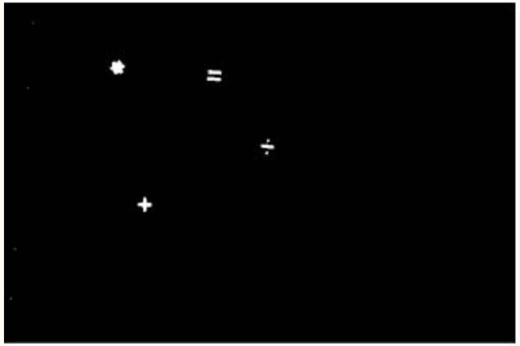
- **Processing, region growing and morphology analysis**



raw image



histogram equalization



“blue” masking + region growing

To detect “÷“ : composed of **3 objects** (1 bar and 2 dots). Neighbor analysis

To detect “=“ : composed of **2 objects** (1 bar and 2 dots). Neighbor analysis

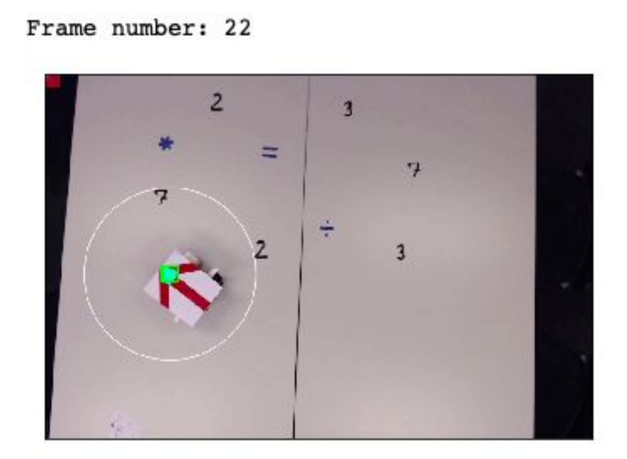
To detect “*”: in the remaining operators, it’s the one with the biggest area

To detect “+“ : has a smaller area than “*”

To detect “**-**“ : has the smallest area in the remaining operators

Red arrow detection & tracking

- **Use of a particle filter for the tracking:**
    - Input: the frame to be tracked
    - Places particles in the image and determines best region that corresponds to the frame
    - Robust to rotation and translation
    - Returns a series of points corresponding to the position of the head of the arrow for each frame

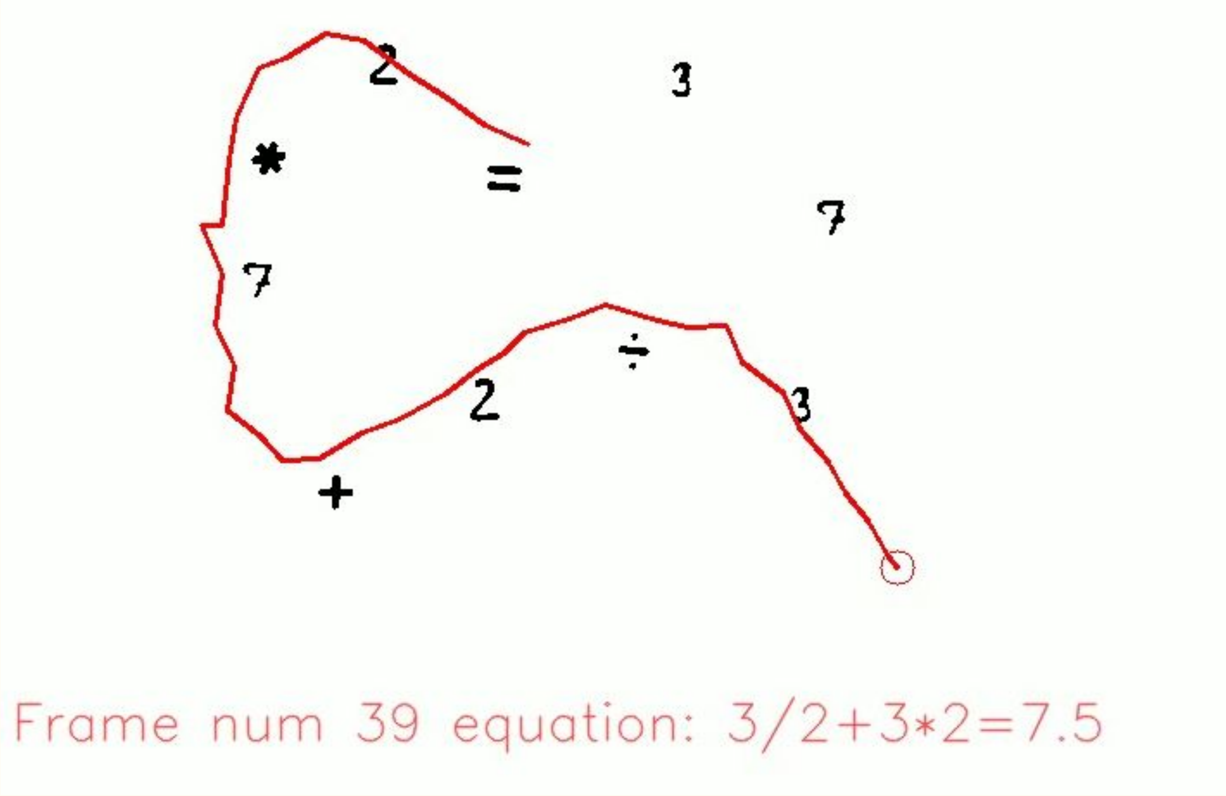


*particles are in **green** and target the head of the red arrow*

Equation maker

- Uses the coordinates of each ***digit***, ***operand*** and a array of points for the robot for each frame
- Returns for each frame a list containing the equation at time t
- For each frame:
    - Computes the distance between the robot and each operands/digits
    - Determines the closest and add the associated element to the equation
    - Once the “=” is detected -> determines the result of the equation
- Then assigns to each frame the corresponding equation
- Background of the output video: Initial image with the borders/vertical lines/arrow masked

Demonstration !



Frame num 39 equation: 3/2+3*2=7.5

Discussion

- Versatility & robustness: improvements in ***operators*** and ***digits*** detection
  - Improvement in the digit classifier model (MLP ?), **only 5/6 were correctly classified**
  - Classify operators by **size intervals:** ($size$(*) $>$ $size$(+) $>$ $size$(-))
  - Use of a handwritten symbols dataset and train a model on it?

- Versatility on the tracking
  - Increase the performance of the tracker with a better frame initialization
  - Test on other trackers