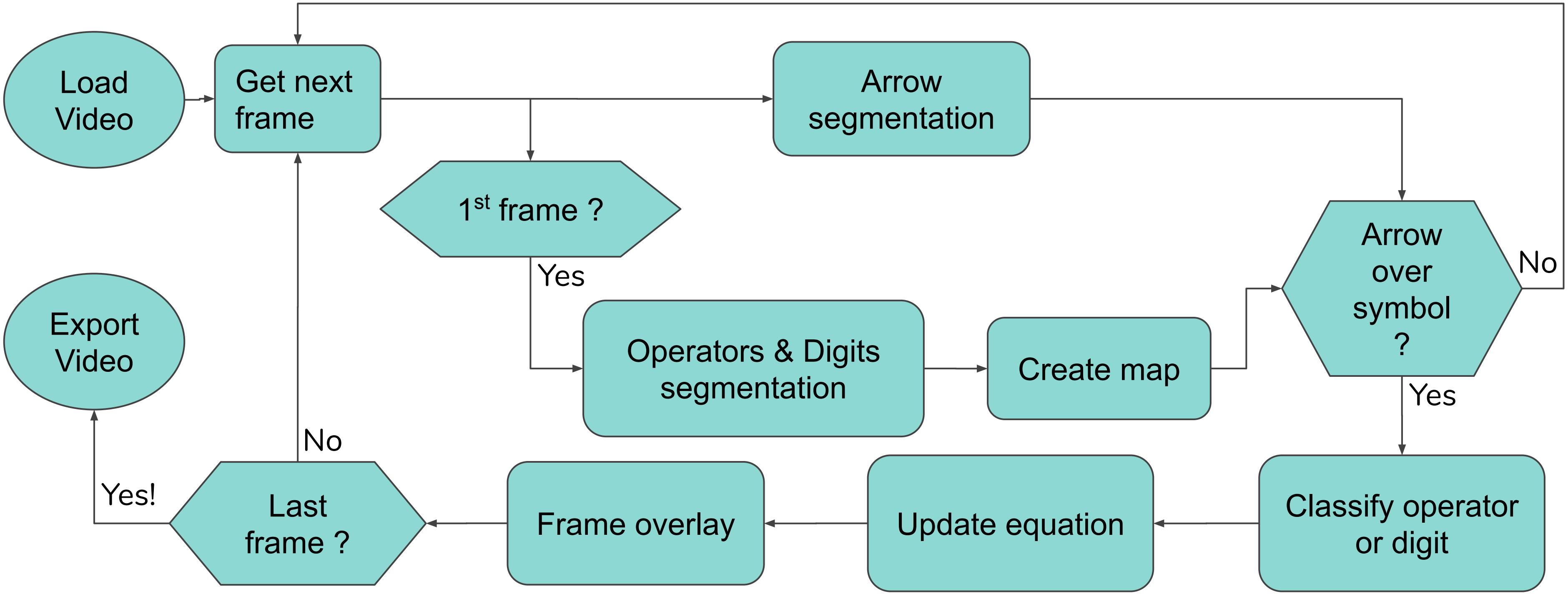
IAPR 2020 : Project presentation

Team 31 : Kilian Meier, Gabriel Tornare, Jérôme Savary

29.05.2020

General system overview

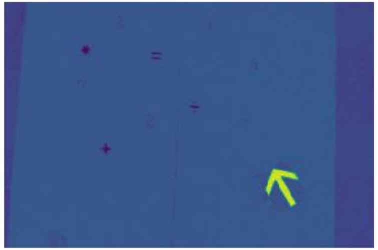


Arrow segmentation

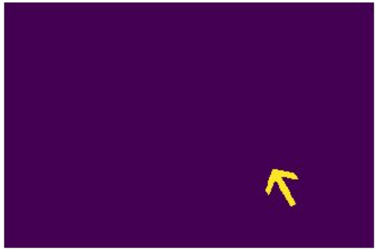
Goal: find the position of the arrow for each frame

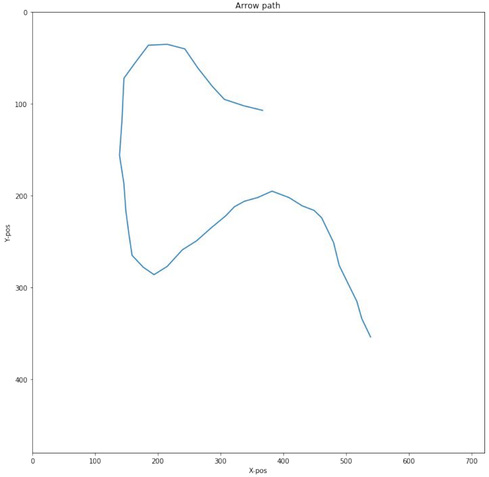
- Conversion from RGB to YUV format
  - Inspection of U channel: arrow shines above all
- Segmentation by Otsu thresholding with fail safe
  - Avoid sudden threshold decrease
- Find object mean position using moments
- Output arrow positions vector
- Plot arrow path (if you want to)

U-channel of 1st frame



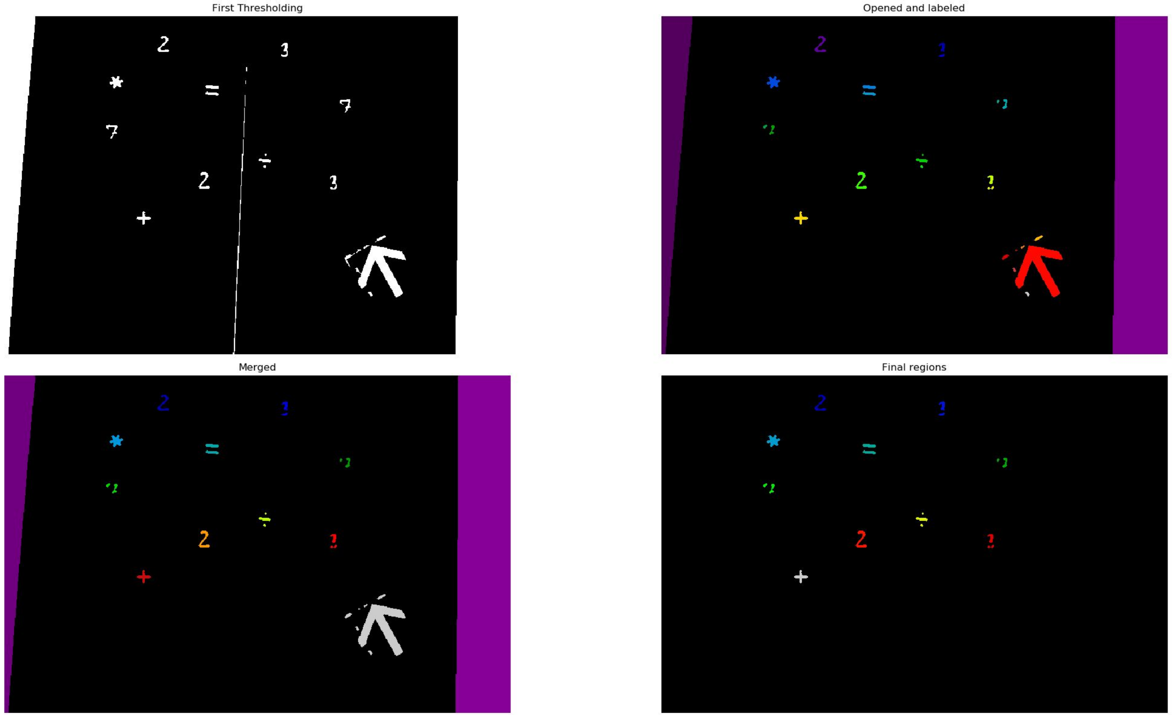
Binary image after thresholding





Operators and Digits segmentation

- Preprocessing (grayscale, equalization)
- Otsu thresholding (first approximation of objects)
- Opening (remove noise, table edge)
- Labeling regions
- Merging regions that are too close
- Discarding regions that are too big or too small, or around the arrow
- Compute center of regions
- Extracting sub-images around regions
- Otsu thresholding of sub-images



First Thresholding

Opened and labeled

Merged

Final regions

Operator classification

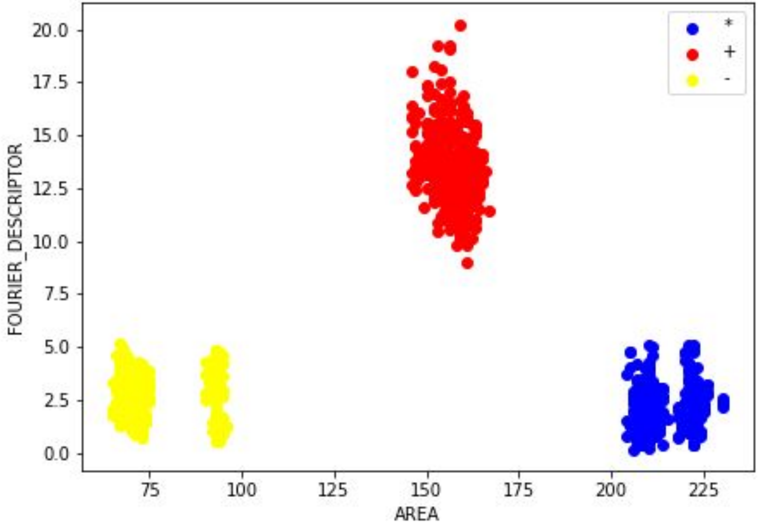
We used 3 features :

- The number of objects N inside each segmentation :
  - N = 3 → ÷
  - N = 2 → =
  - N = 1 → either + , ∗ , –
- The total Area
- The 5th Fourier Descriptor RTSI

We used data augmentation (with random translation and rotation) to generate 1200 pictures from :

- The operators given on Moodle (Originals)
- The operators extracted from the Test Video

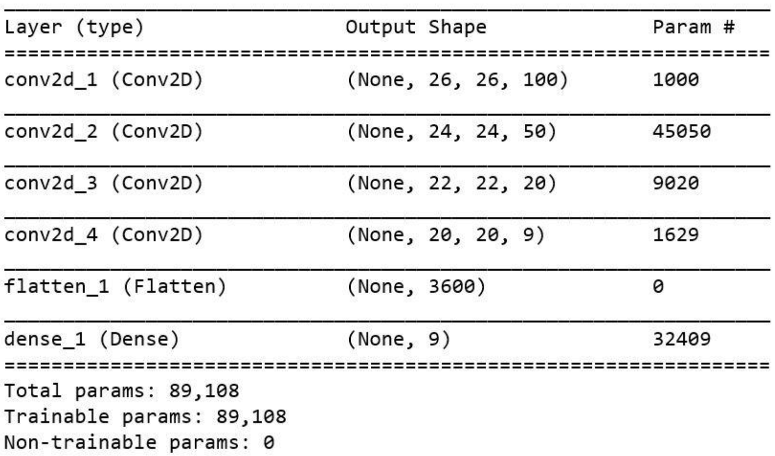
A simple LDA classifier was trained on the augmented dataset to separate + , ∗ , – operators :



Digit classification

- Data Augmentation :
  - *Training set* : 600’000 pictures were generated from the 60’000 Mnist training set with random rotations and translations.
  - *Video test set* : 1002 pictures were generated from the video (also with random rotations and translations)
- CNN (built with Keras) :
  - Optimizer : Adam
  - Batch size : 32
  - Nb epochs : 1
  - Loss : Categorical Crossentropy
  - Activation functions : Relu (and Softmax for last layer)
- Results :
  - Accuracy on test set : 93.5%
  - Accuracy on video test set : 93.1%
- Ameliorations :
  - Increase the CNN structures :
    - (784-40-80-500-1000-2000-10)
    - But it takes a lot of time to train

- CNN layer structure :



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 100)       1000
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 50)        45050
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 22, 22, 20)        9020
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 20, 20, 9)         1629
_________________________________________________________________
flatten_1 (Flatten)          (None, 3600)              0
_________________________________________________________________
dense_1 (Dense)              (None, 9)                 32409
=================================================================
Total params: 89,108
Trainable params: 89,108
Non-trainable params: 0
```