TEXT DETECTION AND RECOGNITION IN NATURAL IMAGES

Yamid Espinel, Avinash Narayana

CONTENTS

PROBLEM STATEMENT AND GOAL

- Text in natural images is generally more difficult to detect because of differences in illumination, noise and text orientation

- **GOAL:** Obtain a chain of characters in string format out of a picture.

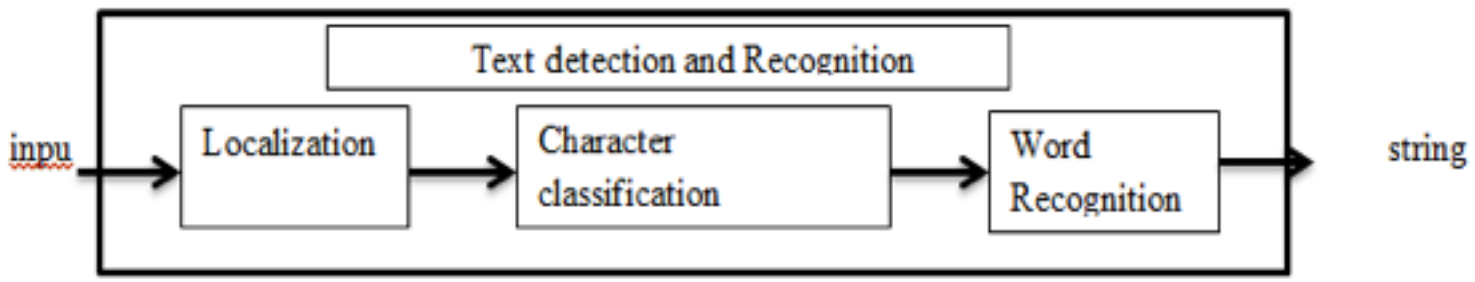
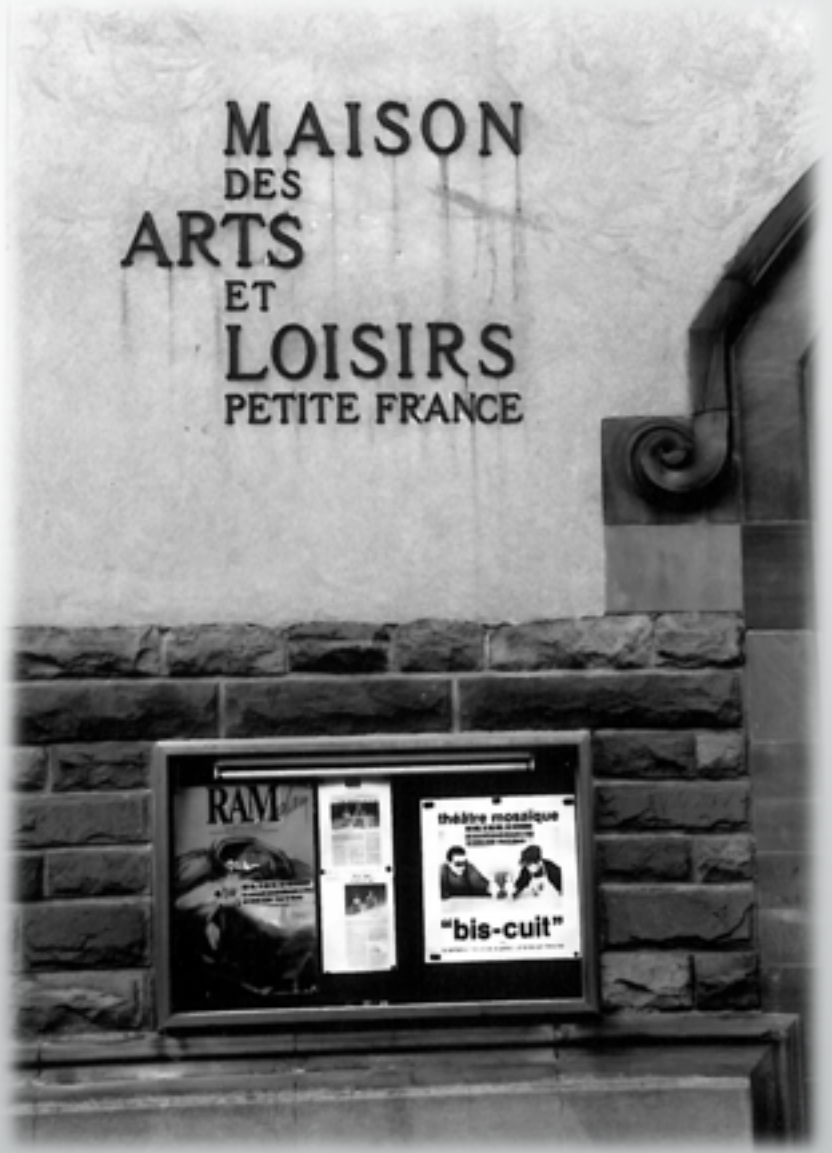


Figure 1 Frame work for commonly used text detection and recognition Methodologies



Text localization using MSER

MSER: Maximally Stable Extremal Regions

- 1) Detect regions with pixel density between a certain range and a defined difference in threshold

Text localization using MSER

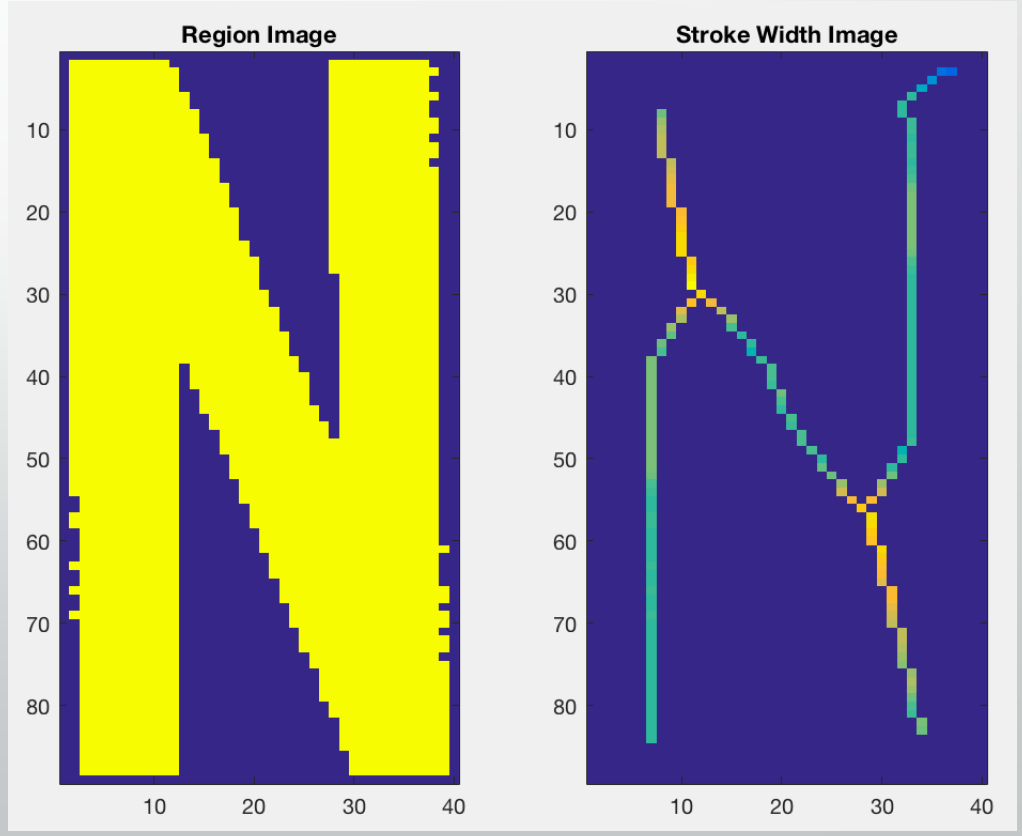
MSER: Maximally Stable Extremal Regions

- 2) Remove Non-Text Regions Based On Basic Geometric Properties

Text localization using MSER

MSER: Maximally Stable Extremal Regions

- 4) Remove Non-Text Regions Based On Stroke Width Variation



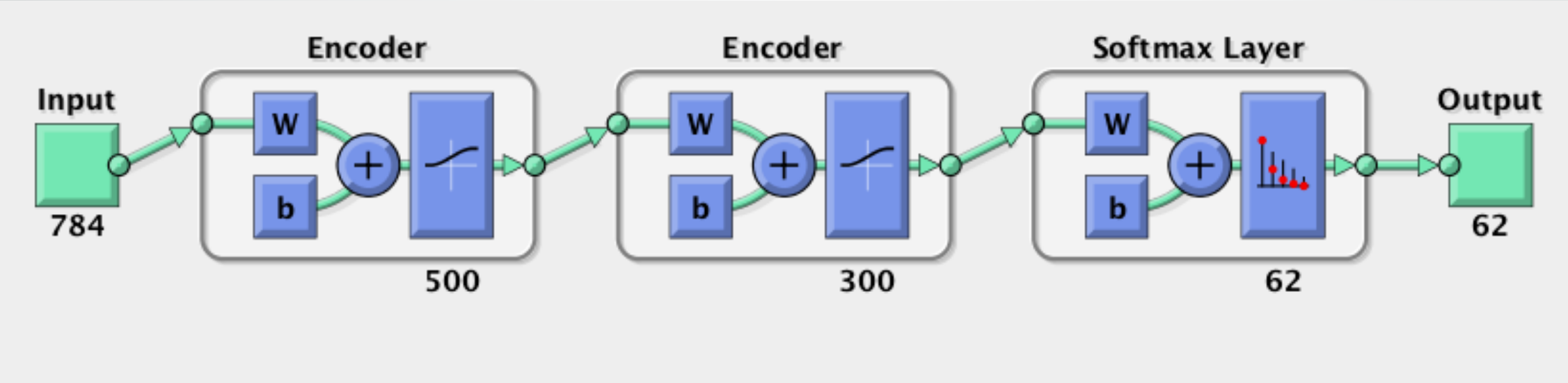
Text localization using MSER

MSER: Maximally Stable Extremal Regions

- 4) Enclose characters using bounding boxes and extract images

Character recognition with Deep Neural Network

- 1) Type of neural network used:



Character recognition with Deep Neural Network

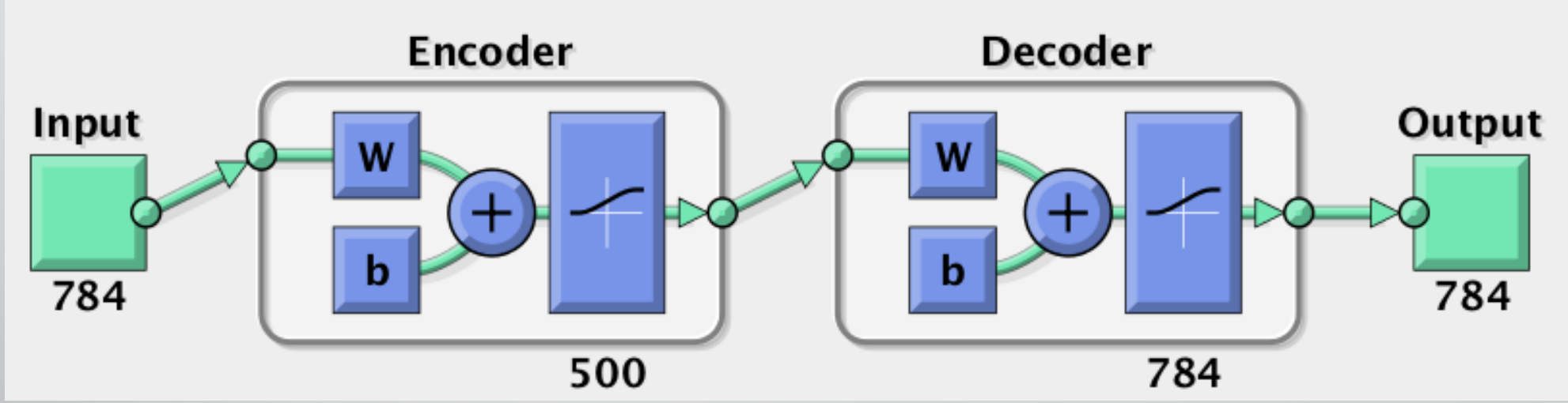
- Dataset: **Chars74K dataset** from T. de Campos:
- *74k images*
- *62 classes (0-9, A-Z, a-z)*

Character recognition with Deep Neural Network

- 2) Character preprocessing:
- *RGB-grayscale conversion*
- *Histogram equalization*
- *Resizinf (28x28)*
- *Image binarization*

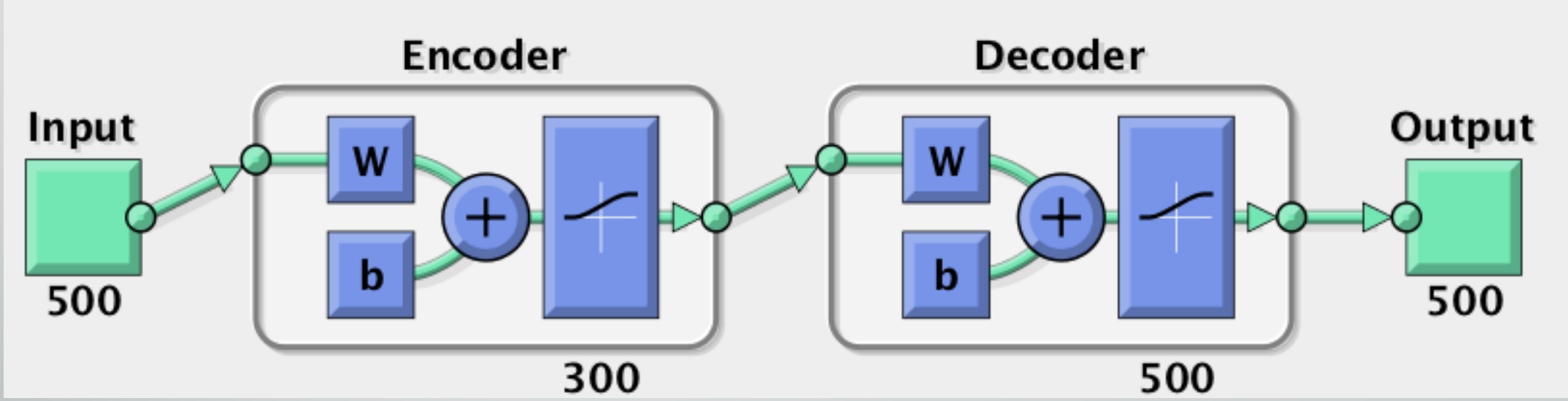
Character recognition with Deep Neural Network

- 3) Training of first hidden layer:



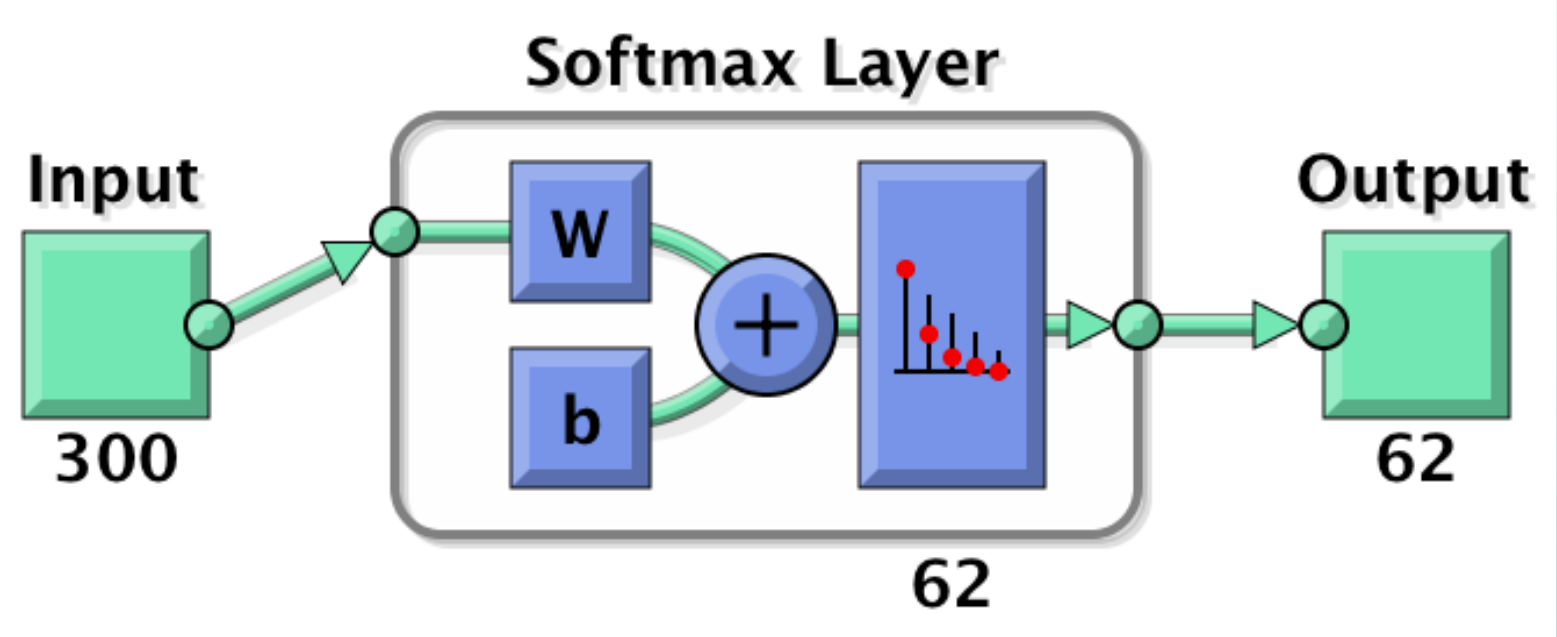
Character recognition with Deep Neural Network

- 4) Training of second hidden layer:



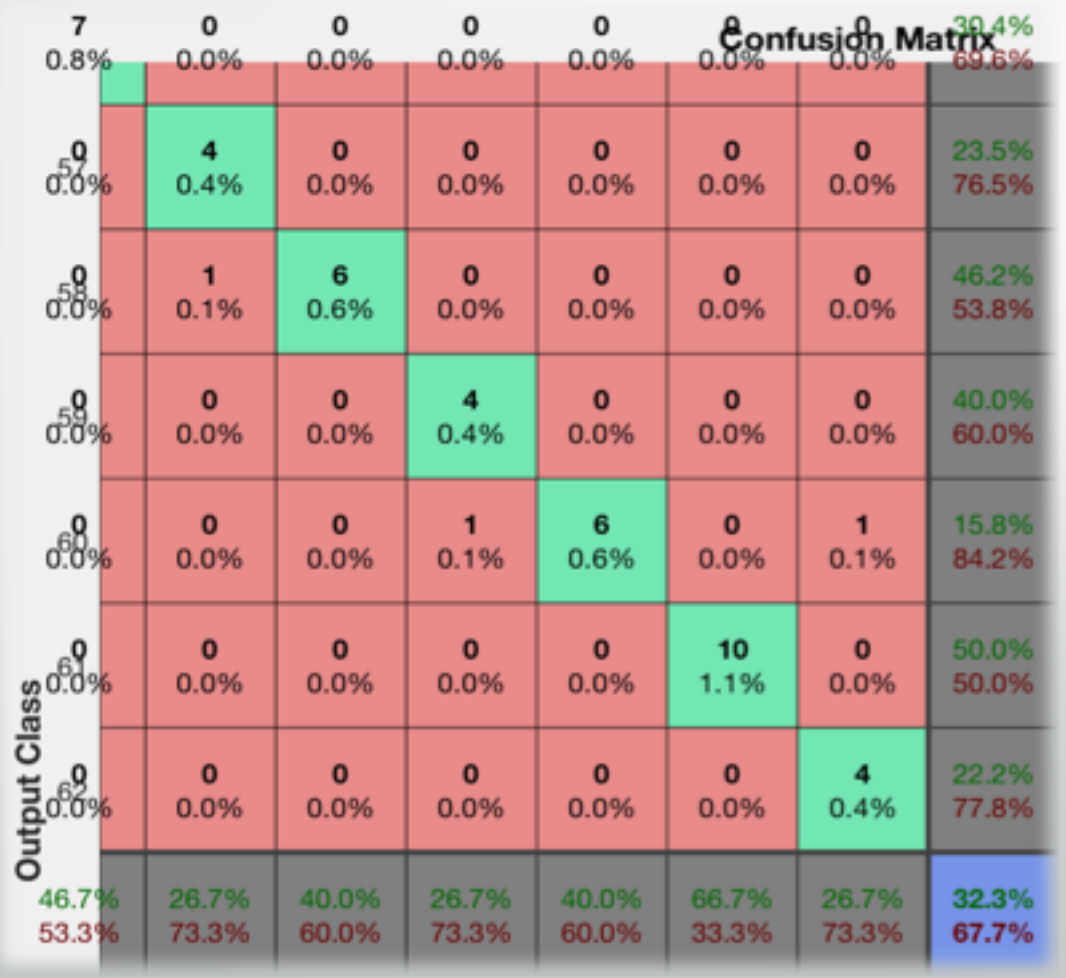
Character recognition with Deep Neural Network

- 5) Training of softmax layer:



Character recognition with Deep Neural Network

- 6) Testing and performance results:

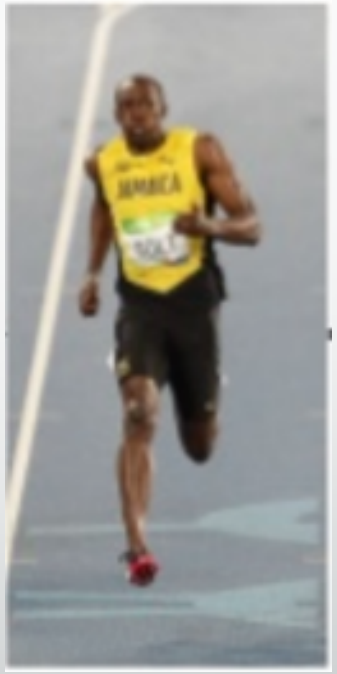


Character recognition using HOG features and SVM

- 1) Feature detection of training data using HOG (histogram of oriented features):

-1	0	1

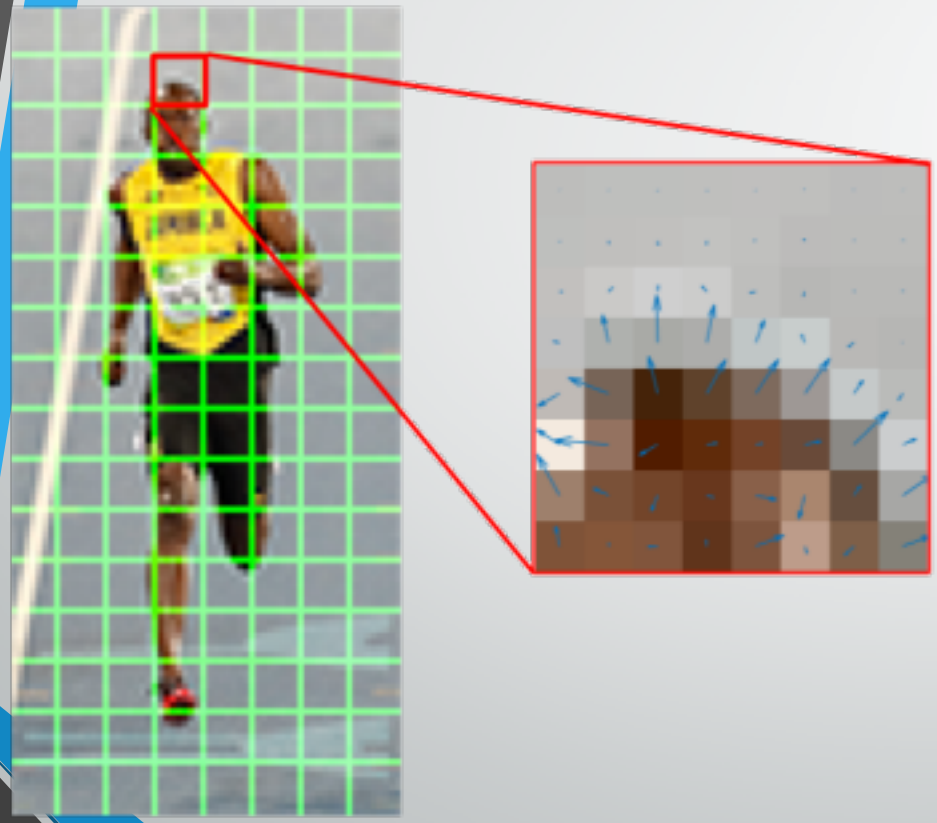
-1
0
1



Gradient computation

Character recognition using HOG features and SVM

- 1) Feature detection of training data using HOG (histogram of oriented features):



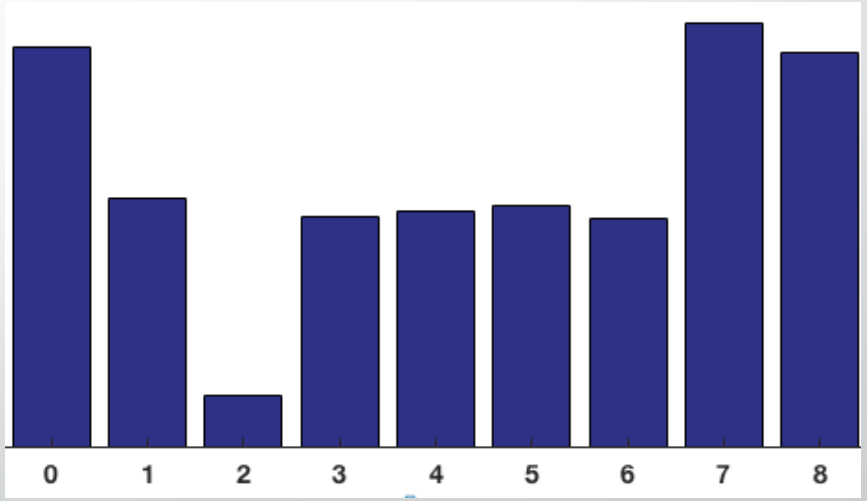
Division in 8x8 patches

2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction



Histogram from an 8x8 patch

Character recognition using HOG features and SVM

- 2) Training of the classifier:
- This is done by using an error-correcting output codes (ECOC) multiclass model with SVM binary learners.

- 3) Testing and performance of the classifier:

- **Accuracy:** 52%

RESULTS


```
finalstring =

  Columns 1 through 8

    'S'    'E'    'u'    'D'    'O'    '1'    'M'    's'

  Columns 9 through 16

    'E'    'E'    'N'    'R'    'O'    'U'    'A'    'J'

  Columns 17 through 19

    'D'    'B'    'y'
```

Things to improve

- Create an automatic method to estimate correct MSER parameters.
- Use a wider set of character training examples.
- Better preprocessing of the training and test samples.
- Train the classifier using non-text images, adding support for text and non-text classification.
- Correction of text orientation for a more precise recognition.

CONCLUSIONS

- MSER combined with other filter methods (such as stroke width variation) let us to segment text easily thanks to its inherent properties (such as color stability and constant stroke width).
- The classification stage can take care of the noise that comes from the MSER stage, if it's properly trained with non-text images too.
- A proper preprocessing of the training and testing samples is critical for a succesful character classification.