

Machine Learning : characters recognition

Alexis Da Costa
Paul Ziolkowski

APP5 - INFO
2021-2022

Steps

Have a model of handwritten digits (Kaggle, MNIST)

Process and train this model

Have a model of handwritten characters (Kaggle, EMNIST)

Process and train the model

Algorithm to detect a sequence

Apply each fragment from the sequence, to recognise the good character

Digit recognition

For this part, we will use the following frameworks and libraries:

sklearn, for splitting datasets between training and testing sub datasets

numpy, to perform operations on arrays and matrices

tensorflow, for importing the MNIST dataset as well as the realization of machine learning and deep learning algorithms

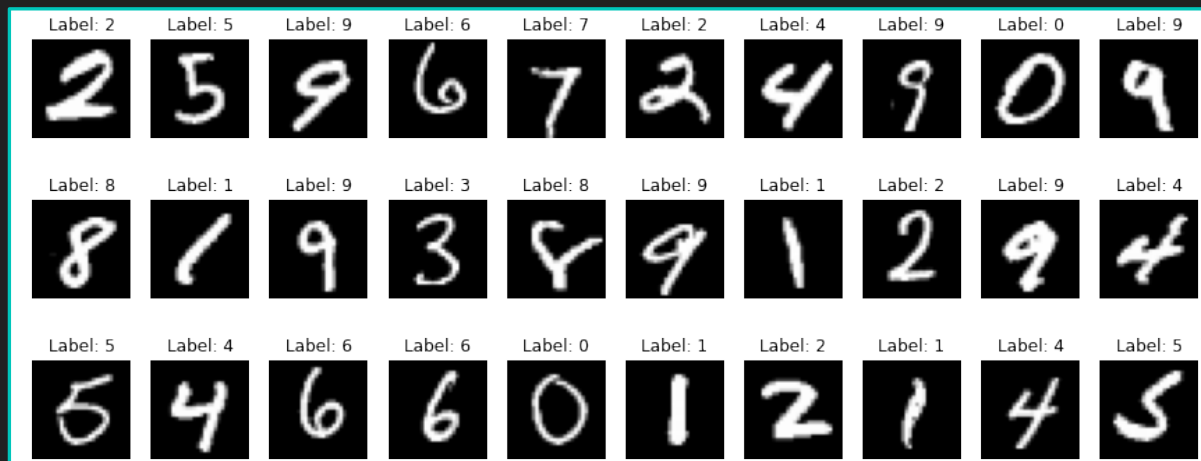
matplotlib, for displaying the dataset images

random, for the generation of random numbers

opencv2, for the processing of the dataset test images

Digit recognition

Use of the MNIST Digits Dataset



Digit recognition

We split the dataset into 2 sub-datasets

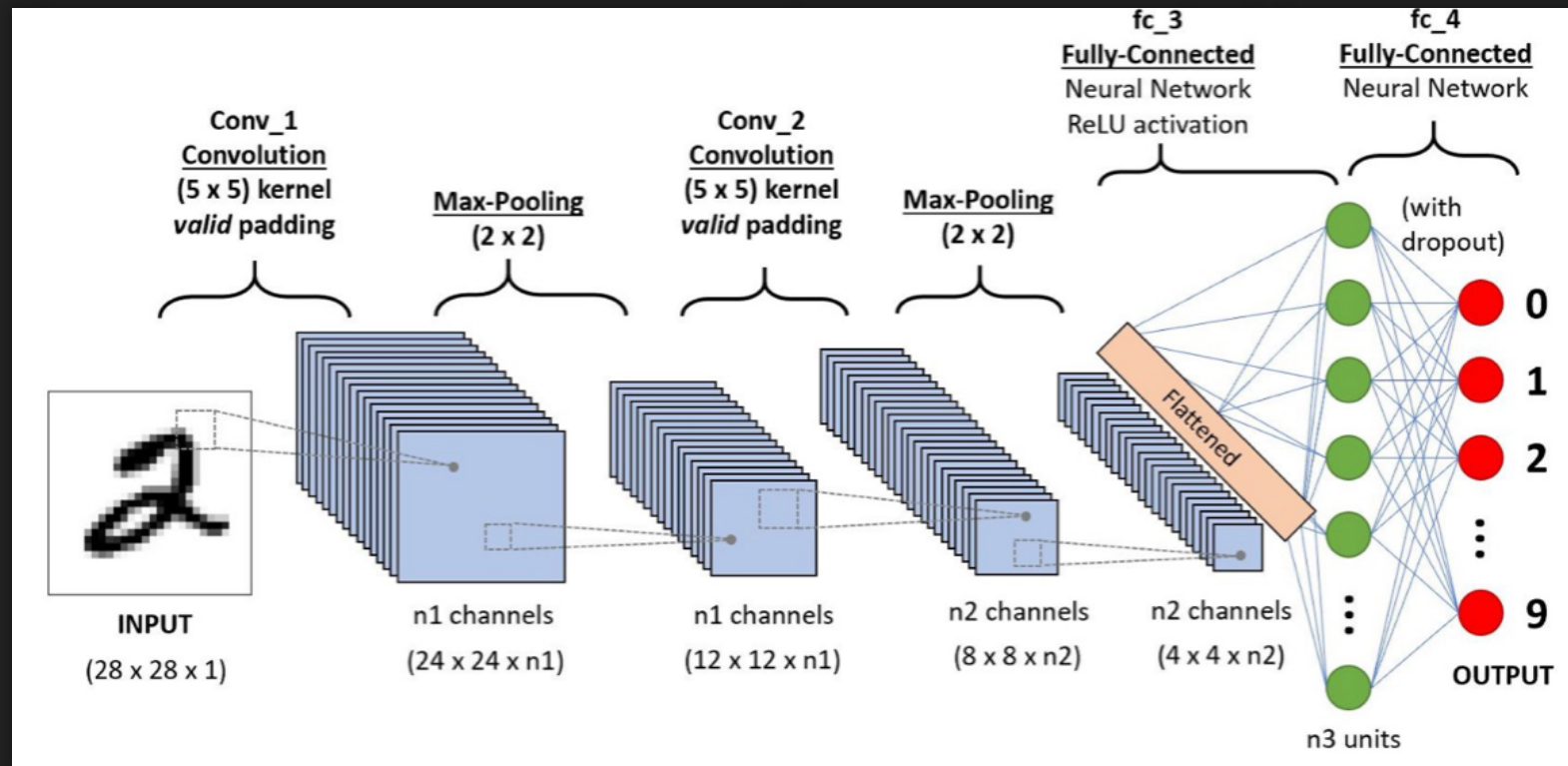
- A training one, containing 60000 images

- A other one, containing 10000 image, use for testing

We normalise all the datas, and reshape to the (28,28) shape dimension

Digit recognition

Using CNN (convolutional neuronal network) instead of SVM (Support Vector Machine)

































Digit recognition

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
conv2d_1 (Conv2D)	(None, 24, 24, 64)	18496
max_pooling2d (MaxPooling2D)	(None, 12, 12, 64)	0
dropout (Dropout)	(None, 12, 12, 64)	0
flatten (Flatten)	(None, 9216)	0
dense (Dense)	(None, 128)	1179776
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1290
Total params: 1,199,882		
Trainable params: 1,199,882		
Non-trainable params: 0		

Digit recognition

With the training, accuracy ~99% (10 epochs)

New training (sub-dataset)

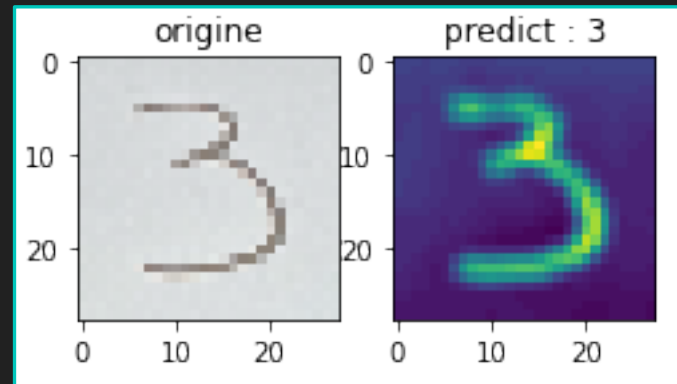
predicted: 4 actual: 4	predicted: 3 actual: 3	predicted: 3 actual: 3	predicted: 9 actual: 4	predicted: 5 actual: 5	predicted: 7 actual: 7	predicted: 2 actual: 2	predicted: 4 actual: 4	predicted: 0 actual: 0	predicted: 1 actual: 1
									
predicted: 8 actual: 8	predicted: 7 actual: 7	predicted: 5 actual: 5	predicted: 8 actual: 8	predicted: 6 actual: 6	predicted: 8 actual: 8	predicted: 3 actual: 3	predicted: 2 actual: 2	predicted: 1 actual: 1	predicted: 6 actual: 6
									
predicted: 1 actual: 1	predicted: 9 actual: 9	predicted: 1 actual: 1	predicted: 4 actual: 4	predicted: 2 actual: 2	predicted: 4 actual: 4	predicted: 8 actual: 8	predicted: 6 actual: 6	predicted: 5 actual: 5	predicted: 1 actual: 1
									

Digit recognition

Some issues to differentiate « 1 » and « 7 »

Train more ? More datas to differentiate

Very good accuracy (99%), so we can validate the process & the model








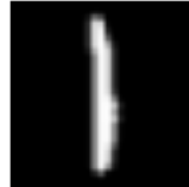
























Character recognition

Same as the previous part, using CNN

Use of 2 csv files (from Kaggle) with some mathematical transformation (model can fit the datas)

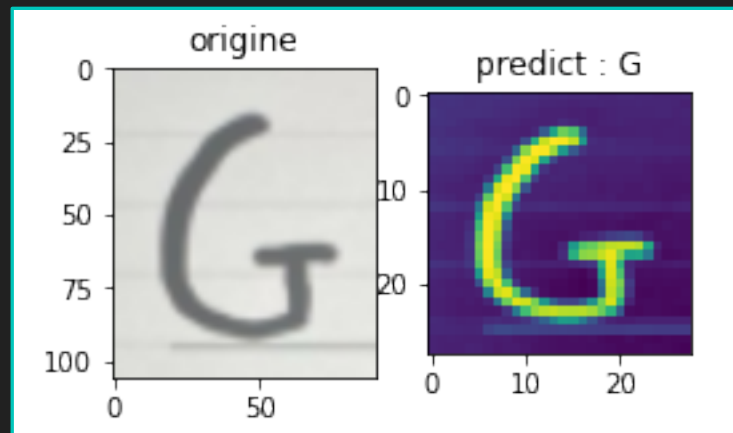
More epochs (30) ~30min to process

Character recognition

predicted: M actual: M	predicted: B actual: B	predicted: B actual: B	predicted: 8 actual: 8	predicted: N actual: N	predicted: 1 actual: 1	predicted: C actual: C	predicted: 6 actual: 6	predicted: U actual: U	predicted: 5 actual: 5
									
predicted: A actual: A	predicted: B actual: B	predicted: T actual: T	predicted: 3 actual: 3	predicted: 9 actual: 9	predicted: F actual: F	predicted: D actual: 0	predicted: a actual: a	predicted: P actual: P	predicted: 6 actual: 6
									
predicted: C actual: C	predicted: h actual: h	predicted: h actual: h	predicted: B actual: R	predicted: E actual: E	predicted: 4 actual: 4	predicted: r actual: r	predicted: Q actual: Q	predicted: t actual: t	predicted: 5 actual: 5
									

Character recognition

Accetable accuracy (87,8%), we can accept the model



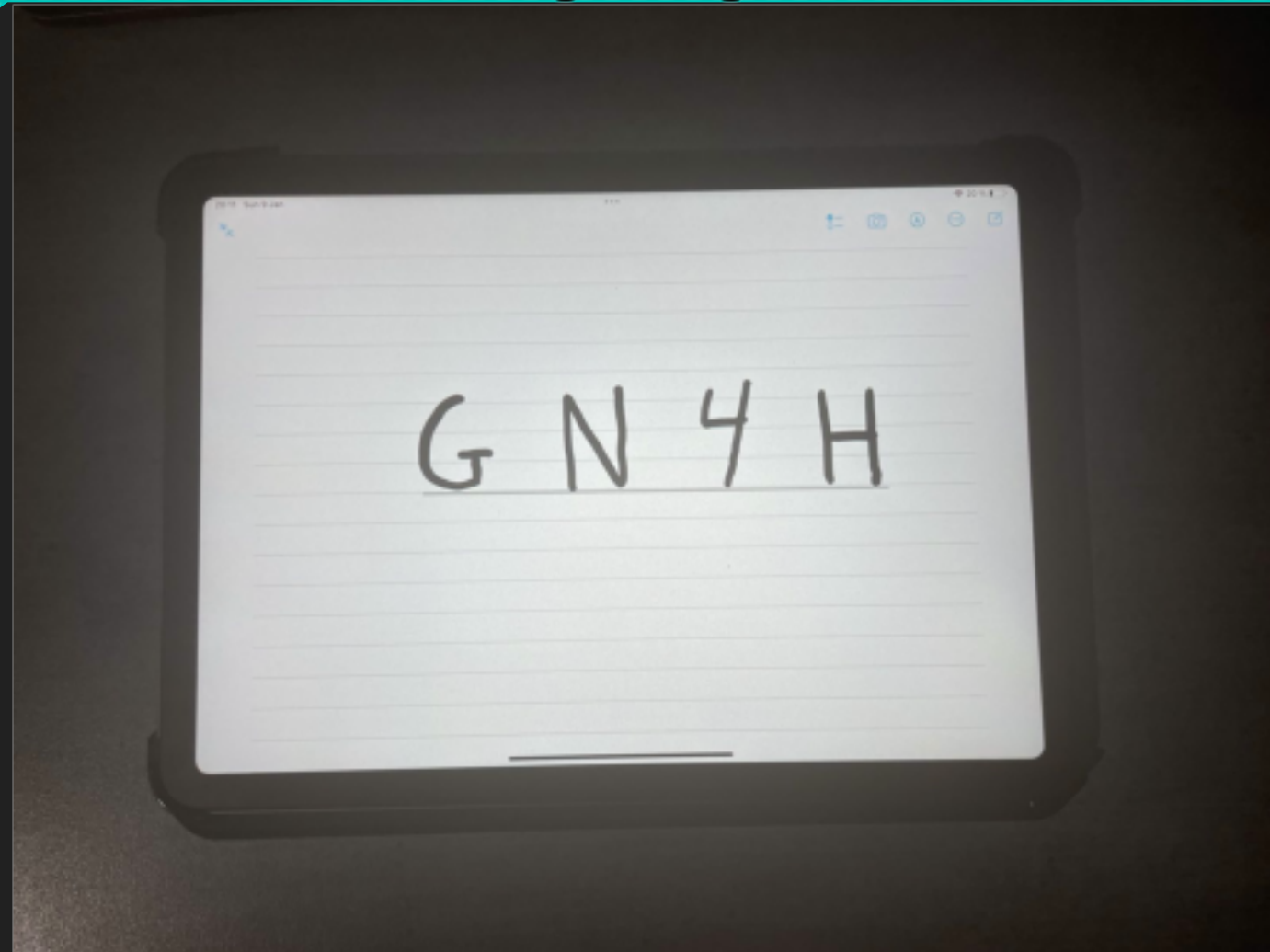
Sequence detection & character recognition

Preprocessing step : transform the image to data interpretable by our model, applying filters (to binarize it, removing angle) → New 28*28 image interpretable by our model

Train with previous model and datas

Sequence detection & character recognition

Image Original



Sequence detection & character recognition

image



gray



blur



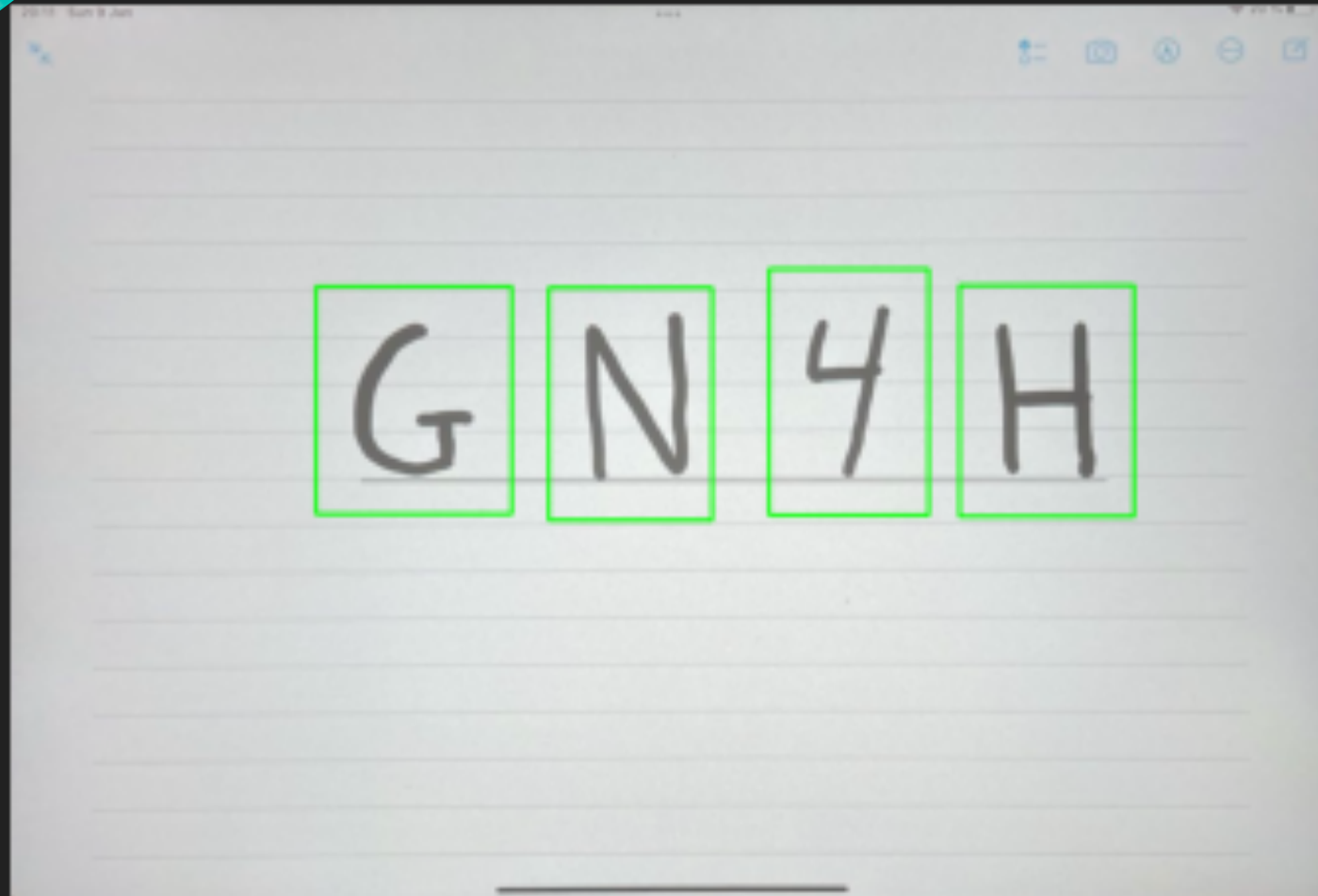
binary



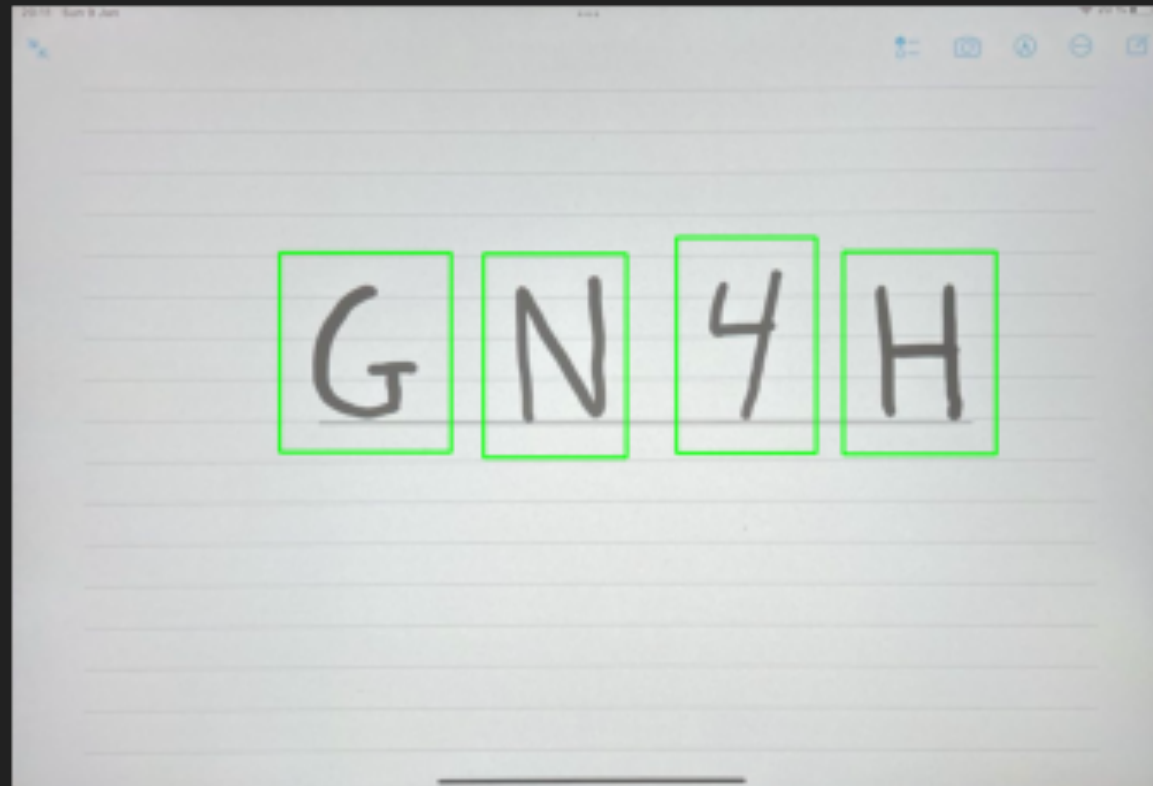
dilation



Sequence detection & character recognition



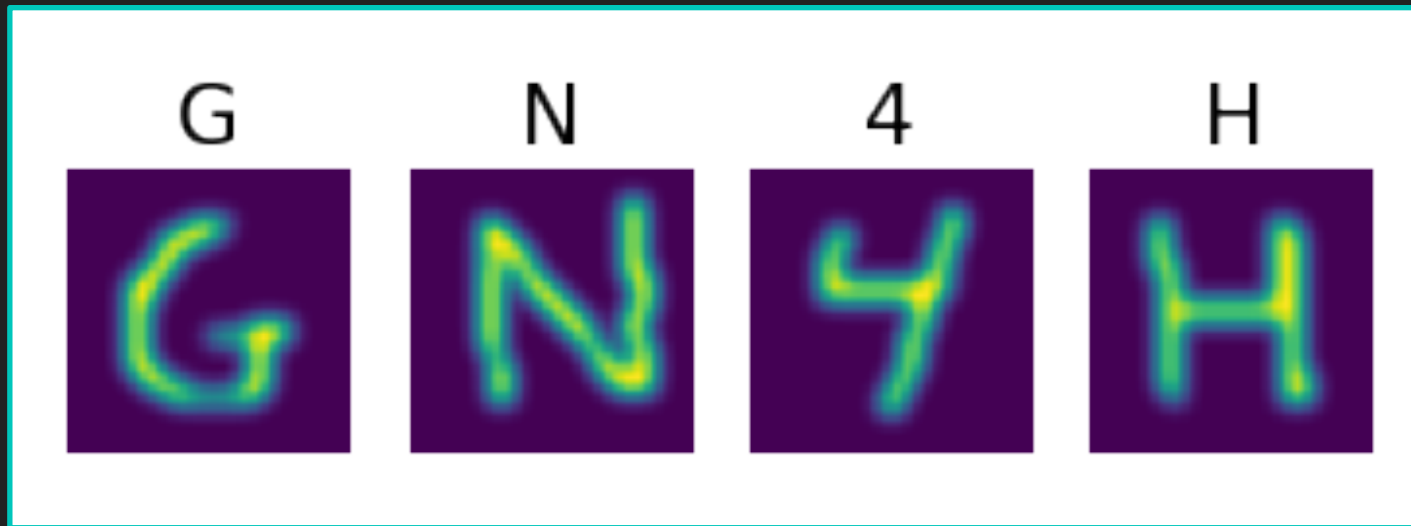
Sequence detection & character recognition



Sequence detection & character recognition

G N 4 H

Sequence detection & character recognition



Conclusion

Presentation of the application

Good accuracy with our model

To be improved for character recognition

Sources

Dataset : MNIST dataset and EMNIST dataset

Reference : <https://www.kaggle.com/c/digit-recognizer>