Arabic Handwritten Recognition Using CNN

Bootcamp T5 Project

By Anwar Aljohani

Introducation

Why handwritten Arabic OCR?

- The Arabic alphabet is also used to script other languages

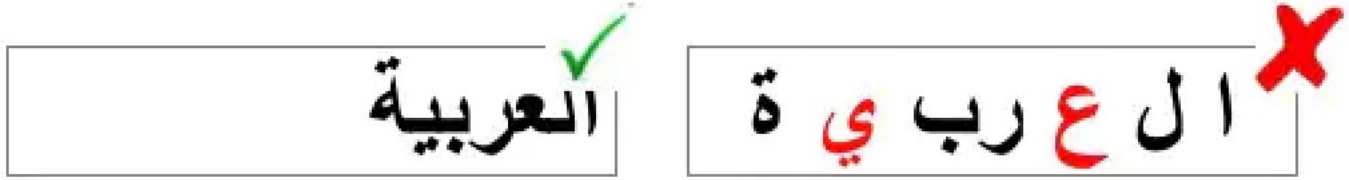
such as Farsi, Kurd, Persian, Urdu, etc.

- Little research has been addresses in the Arabic OCR.

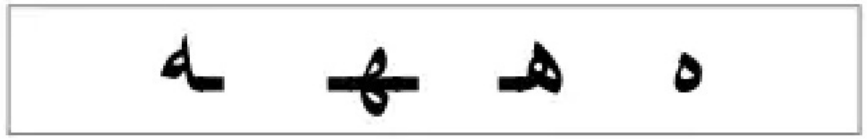
- Handwritten OCR has a wide range of applications: invoice receipt processing, subscription collection,

Characteristics of the Arabic language

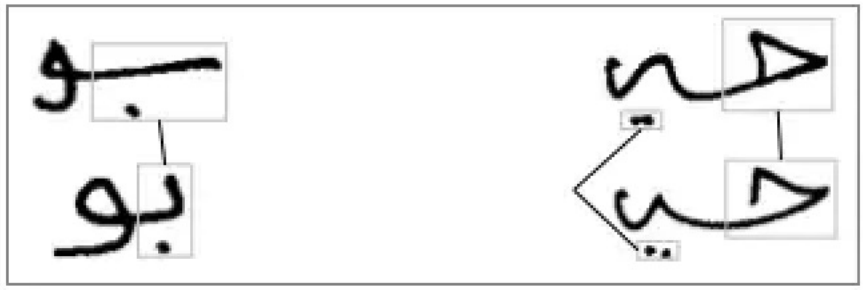
Arabic is cursive

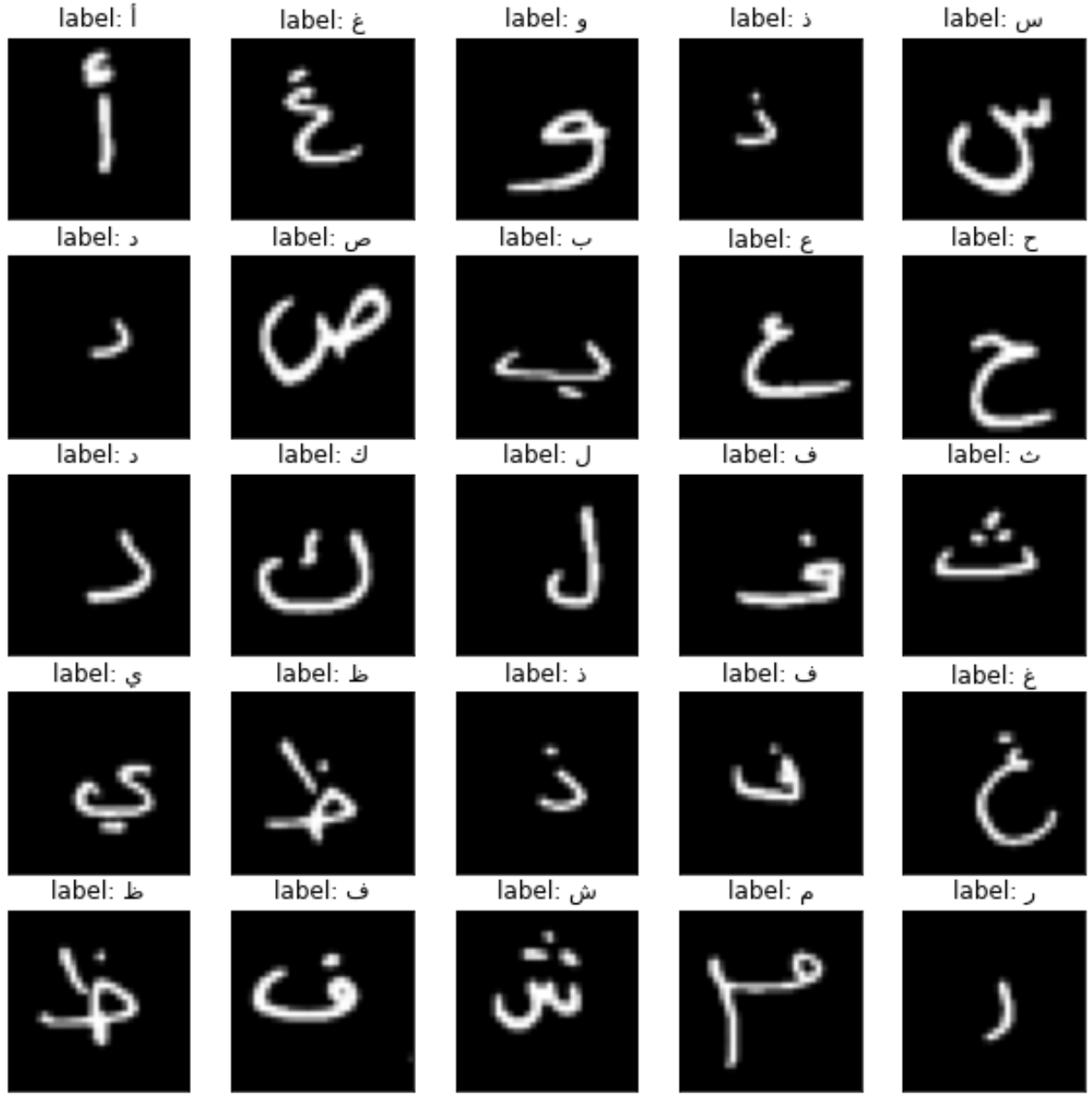


Arabic letter shapes are context dependent



Variability of letter shapes (in handwritings)




label: أ
label: غ
label: و
label: ذ
label: س
label: د
label: ص
label: ب
label: ع
label: ح
label: د
label: ك
label: ل
label: ف
label: ث
label: ي
label: ظ
label: ذ
label: ف
label: غ
label: ظ
label: ف
label: ش
label: م
label: ر

Convolutional Neural Network (CNN)

- Very Popular:

  -Toolboxes: tensorflow, cuda-convnet and caffe (user friendlier)

A high performance Classifier (multi-class)

- Successful in object recognition, handwritten optical character OCR recognition, image noise removal etc.
- Easy to implementation

  -Slow in learning

  -Fast in classification

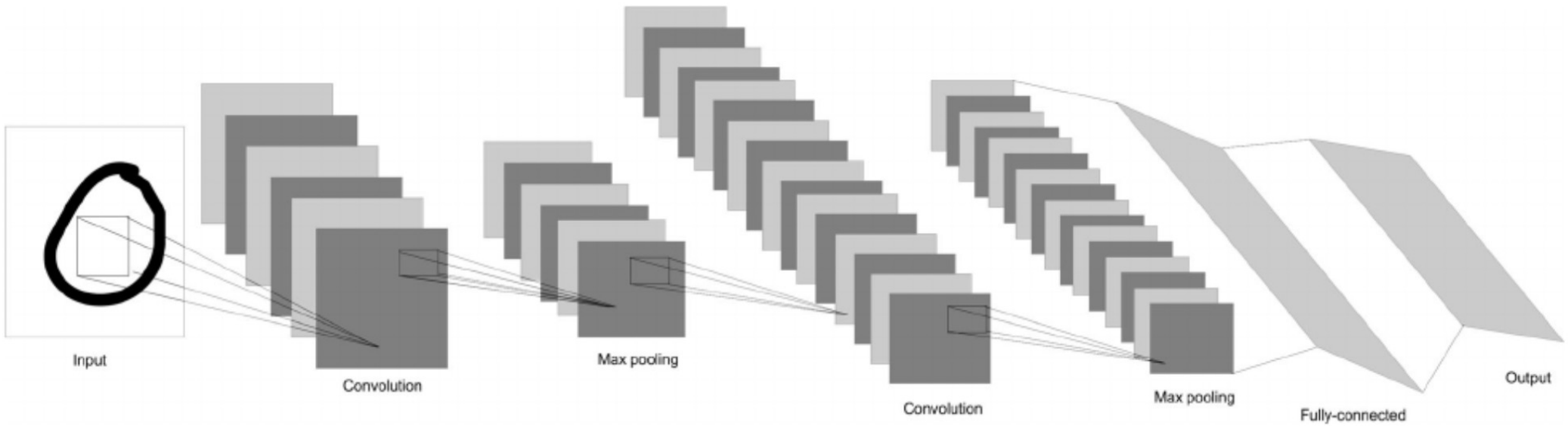
Why CNN

Traditional optical character recognition (OCR) systems lacked efficiency, as character features were hard-coded and used to match characters. On the other hand, neural networks are able to learn features by analysing a dataset, alleviating the need for manual hard coding of features. At the end of the training process, the neural network learns parameters, making it more flexible when handwriting styles are changed. The creation of the Modified National

Data

Arabic Letters Dataset is composed of 16,800 characters written by 60 participants, and 90% of participants are right-hand. Each participant wrote each character (from ’alef’ to ’yeh’) ten times.

CNN Workflow



Methodology

1- Data Preprocessing

- Image Normalization : between 0–1 without loss of information.

  Data normalization is an important step which ensures that each input parameter (pixel, in this case) has a similar data distribution. This makes convergence faster while training the network.

- Reshape the data to ensure that the input data to the model is in the correct shape.

  Keras CNNs require a 4D array as input, with shape (nb_samples,rows,columns,channels)

Methodology

2- Encoding Categorical Labels

- From the labels csv files we can see that labels are categorical values and it is a multi-class classification problem.

- One-hot encoding transforms integer to a binary matrix where the array contains only one ‘1’ and the rest elements are ‘0’.

Methodology

- 3- Model Architecture

  - Model Summary And Visualization
  - Parameters Tuning
  - Training the Model
  - Test the Model

Methodology

- Optimizer:

RMSprop ,Adam

Adagrad , Nadam

- kernel_initializer

Normal,uniform

- Activation

Relu ,Linear, tanh

24 of different possible parameters

Results

- **Optimizer: RMSprop , kernel_initializer: uniform , activation: linear**

  Loss: 0.7705

  Accuracy: 0.7382

  val_loss: 0.3556

  val_accuracy: 0.8833

- **Optimizer: Adam, kernel_initializer: uniform , activation: relu**

  loss: 0.7203

  Accuracy: 0.7578

  val_loss: 0.4107

  val_accuracy: 0.8705

Results

- **Optimizer: Adagrad , kernel_initializer: uniform , activation: tanh**

  loss: 3.3758

  accuracy: 0.1000

  val_loss: 2.7361

  val_accuracy: 0.1852

- **Optimizer: Nadam, kernel_initializer: normal, activation: relu**

  loss: 0.7634

  accuracy: 0.7404

  val_loss: 0.4217

  val_accuracy: 0.8577

Results

Optimizer: RMSprop
Kernel_initializer: uniform
Activation: linear

Accuracy: 91%