| ID | Name |
| --- | --- |
| 201900439 | عبدالرحمن محمود محمود |
| 201900419 | عبدالرحمن سيد فتحي |
| 201900384 | ضحي علي مصطفي عبدالرحمن |
| 201900162 | الاء محمدي عبدالعظيم |
| 201900433 | عبدالرحمن محمد عبدالرجال عيسي |
| 201900417 | عبدالرحمن خليفة حلمي خليفة |

# 1- Architecture used in the paper

- This research has highlighted the general architectural methodology that can use for Arabic handwritten Recognition based on deep learning thoroughly and criticized.

  Depending on the classifiers combination and the feature extraction.

- Deep learning for Arabic Handwriting Recognition

Computer vision (CV) refers to the study of the computer simulation of human visual science. Major task of CV is to collect images (or video) so that they could be used for analysis, gathering information, and making decisions or judgements.

(DL) dramatically improved the state-of-the-art in visual object recognition, object detection, handwritten recognition and many other domains.

Handwritten recognition technique is one of this tasks that targeted to extract the text from documents or another images type.

This paper highlighted the pre-processing and binarization methods that have been used in the literature along with proposed numerous directions for developing.

| Literature | Brief Description | Dataset | dataset Type | Results |
|---|---|---|---|---|
| Porwal, Zhou [1] | Deep Belief Networks which incrementally learns complex structure of the data | AMA Arabic PAW dataset | Arabic Handwriting Character | 75,05% |
| Tamen, Drias [2] | Combine several classifiers integrated at the decision level in classification phase | IFN/ENIT | Arabic Handwritten Words | 87,00 |
| Ashiquzzaman and Tushar [3] | CNN model uses several convolutional layers + Dropout used to reduce overfitting | CMATERDB 3.3.1 | Arabic Handwritten Digit | 97.4 |
| Rabi, Amrouch [4] | Hidden Markov Models (HMMs) | IFN/ENIT | Arabic Handwritten Words | 87.93%. |
| Elleuch, Mokni [6] | Deep learning model based on Support Vector Machine(DSVM) | HACDB | Arabic Handwriting Character | Error rate: 8.86% |
| Elleuch, Maalej [8] | CNN based-SVM model without and with dropout for training and recognizing Arabic characters | HACDB and IFN/ENIT | Arabic Handwriting Character | Error rate: HACDB : 5.83 IFN/ENIT :7.05 |
| Elleuch, Tagougui [9] | Deep Belief Network (DBN) + with dropout/dropconnect | Arabic handwritten script recognition | HACDB | Error classification rate of 2.73% using dropout. 2.27% using dropconnect |

## 2- Dataset Details

Train Data:

x_train = (13440, 1024)

y_train = (13440, 1)

Test Data:

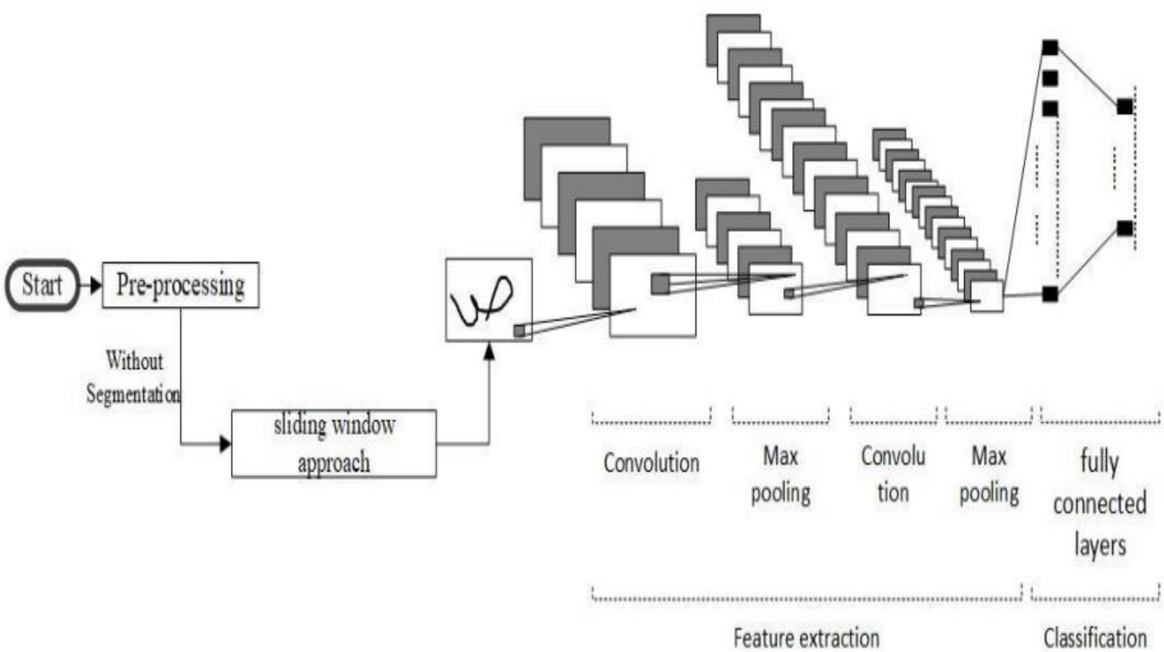
x_test = (3360, 1024)

y_test = (3360, 1)

## 3- Implementation Details

- Reshape for data to `(-1,32,32, 1)`
- Make `Conv2D(80,(5,5) activation='relu', input shape=(32,32,1)`
- Make `MaxPooling2D(2,2)`
- Make Conv2D(64, (5,5), activation='relu' )
- Make MaxPooling2D (2,2)
- Make Flatten() to returns a copy of the array in one dimensional
- Make CNN Dense (1024, activation='relu') with name 'featurs'
- Make CNN Dense(28 ,activation='softmax')
- Make compile for the model with optimizer= 'adam'

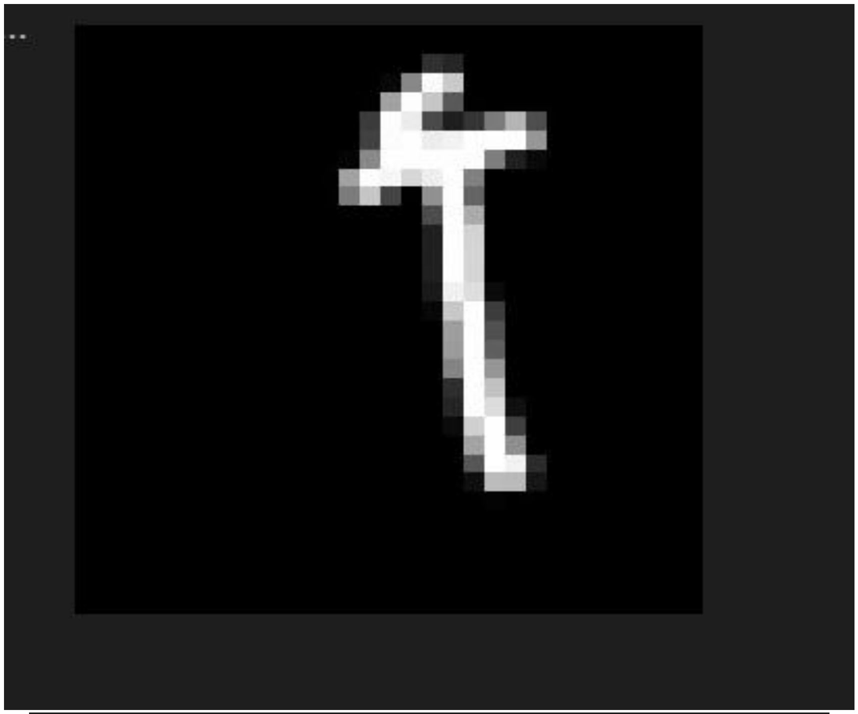
Figure 2 flowchart of Arabic Handwritten recognition based on CNN model

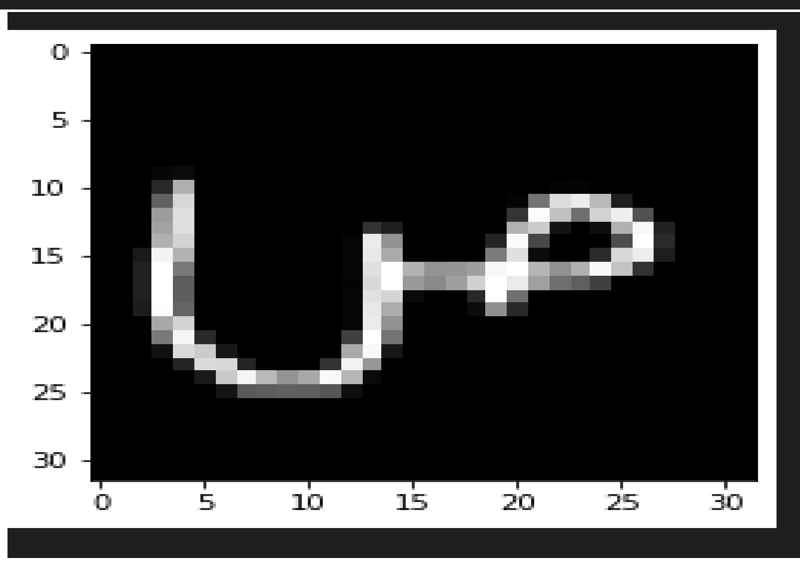


## 4- Results and Visualization

```
... Model: "sequential"
    _________________________________________________________________
     Layer (type)                Output Shape              Param #
    =================================================================
     conv2d (Conv2D)             (None, 28, 28, 80)        2080

     max_pooling2d (MaxPooling2D  (None, 14, 14, 80)       0
     )

     conv2d_1 (Conv2D)           (None, 10, 10, 64)        128064

     max_pooling2d_1 (MaxPooling  (None, 5, 5, 64)         0
     2D)

     flatten (Flatten)           (None, 1600)              0

     featurs (Dense)             (None, 1024)              1639424

     dense (Dense)               (None, 28)                28700

    =================================================================
    Total params: 1,798,268
    Trainable params: 1,798,268
    Non-trainable params: 0
    _________________________________________________________________
```

```
Epoch 1/50
209/210 [============================>.] - ETA: 0s - loss: 1.9950 - accuracy: 0.4418WARNING:tensorflow:Learning rate reduction is conditioned on metric `
210/210 [==============================] - 16s 23ms/step - loss: 1.9908 - accuracy: 0.4430 - val_loss: 0.8253 - val_accuracy: 0.7729 - lr: 0.0010
Epoch 2/50
209/210 [============================>.] - ETA: 0s - loss: 0.9953 - accuracy: 0.7190WARNING:tensorflow:Learning rate reduction is conditioned on metric `
210/210 [==============================] - 5s 22ms/step - loss: 0.9944 - accuracy: 0.7193 - val_loss: 0.5861 - val_accuracy: 0.8408 - lr: 0.0010
Epoch 3/50
210/210 [==============================] - ETA: 0s - loss: 0.7795 - accuracy: 0.7827WARNING:tensorflow:Learning rate reduction is conditioned on metric `
210/210 [==============================] - 5s 22ms/step - loss: 0.7795 - accuracy: 0.7827 - val_loss: 0.5135 - val_accuracy: 0.8491 - lr: 0.0010
Epoch 4/50
208/210 [============================>.] - ETA: 0s - loss: 0.6736 - accuracy: 0.8077WARNING:tensorflow:Learning rate reduction is conditioned on metric `
210/210 [==============================] - 5s 22ms/step - loss: 0.6734 - accuracy: 0.8078 - val_loss: 0.4087 - val_accuracy: 0.8982 - lr: 0.0010
Epoch 5/50
210/210 [==============================] - ETA: 0s - loss: 0.6067 - accuracy: 0.8284WARNING:tensorflow:Learning rate reduction is conditioned on metric `
210/210 [==============================] - 5s 22ms/step - loss: 0.6067 - accuracy: 0.8284 - val_loss: 0.3934 - val_accuracy: 0.9104 - lr: 0.0010
Epoch 6/50
208/210 [============================>.] - ETA: 0s - loss: 0.5601 - accuracy: 0.8455WARNING:tensorflow:Learning rate reduction is conditioned on metric `
210/210 [==============================] - 5s 22ms/step - loss: 0.5596 - accuracy: 0.8456 - val_loss: 0.3195 - val_accuracy: 0.9256 - lr: 0.0010
Epoch 7/50
210/210 [==============================] - ETA: 0s - loss: 0.5124 - accuracy: 0.8574WARNING:tensorflow:Learning rate reduction is conditioned on metric `
210/210 [==============================] - 5s 22ms/step - loss: 0.5124 - accuracy: 0.8574 - val_loss: 0.3168 - val_accuracy: 0.9238 - lr: 0.0010
Epoch 8/50
210/210 [==============================] - ETA: 0s - loss: 0.4724 - accuracy: 0.8712WARNING:tensorflow:Learning rate reduction is conditioned on metric `
210/210 [==============================] - 5s 22ms/step - loss: 0.4724 - accuracy: 0.8712 - val_loss: 0.3125 - val_accuracy: 0.9199 - lr: 0.0010
Epoch 9/50

show more (open the raw output data in a text editor) ...

209/210 [============================>.] - ETA: 0s - loss: 0.2370 - accuracy: 0.9422WARNING:tensorflow:Learning rate reduction is conditioned on
metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy,lr
210/210 [==============================] - 5s 22ms/step - loss: 0.2368 - accuracy: 0.9423 - val_loss: 0.1746 - val_accuracy: 0.9652 - lr: 0.0010
Epoch 32/50
208/210 [============================>.] - ETA: 0s - loss: 0.2412 - accuracy: 0.9404WARNING:tensorflow:Learning rate reduction is conditioned on
metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy,lr
210/210 [==============================] - 5s 22ms/step - loss: 0.2413 - accuracy: 0.9403 - val_loss: 0.1829 - val_accuracy: 0.9655 - lr: 0.0010
```




0
5
10
15
20
25
30
0
5
10
15
20
25
30