Models and libraries in ocr for Arabic language in general and in Android :

Easy ocr :

-[CRAFT](https://github.com/clovaai/CRAFT-pytorch) for text detection and [CRNN](https://arxiv.org/abs/1507.05717) for Text Recognition: Trainable model with Resnet for Feature Extraction, LSTM and CTC for Decoding.

-for cpu :text extraction process takes 2 to 3 minutes.for gpu can take 13 sec .

-accepted error rate for low noisy images and high error rate for images with high noises

-accuracy can be increased by filtering the noise images.

-Robust word-based and character-based accuracy metrics for performance evaluation

-One thing to improve in Easy OCR, is the tweaking ability: although language selection is easy, changing models and retraining for different purposes.

-Easy OCR is around 95% accurate.

-One more thing is that while Easy OCR is better on characters,

[Tesseract](https://pypi.org/project/pytesseract/) :

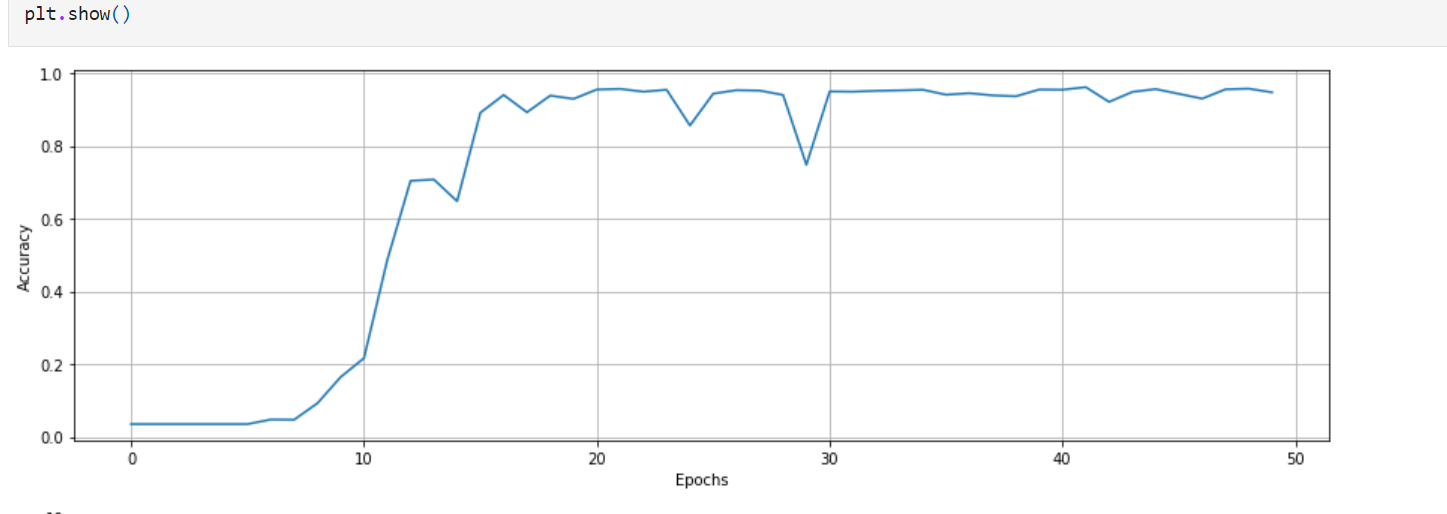
-Pytesseract package for python users.

-better NER extraction and conversion time with better accuracy but lack in the number of entities acquired.

-Results  :

-Patel and friends produced 70% accuracy using 20 sample images in 2012 . Kumar and friends produced 97% accuracy for small scanned bill documents and 83% accuracy for small scanned bill documents using Tesseract OCR on 25 scanned bills in 2020 . Akinbade and friends produced 81.9% character accuracy and 69.7% word accuracy on 11 sample images in 2020 .

model achieves test accuracy above 95% for [Arabic Handwritten Digits Dataset on Kaggle](https://www.kaggle.com/mloey1/ahdd1)



<https://github.com/elnomrosy66/ocr-arabic>

**Recognition Models :**

* [tessdata\_fast](https://github.com/tesseract-ocr/tessdata_fast): Tesseract is written in C++ and optimized for performance but if you further need to improve the performance, try using tessdata\_fast models which are 8-bit integer versions of the tessdata models.
* [tessdata\_best](https://github.com/tesseract-ocr/tessdata_best): Best trained models of tesseract OCR and acts as the base models for fine-tuning.

## **Limitations of Tesseract OCR :**

1. It does not work for low-quality images with DPI points less than a certain threshold.
2. If the image is noisy and the separation of foreground and background is not significant, it can generate errors.
3. It does not recognize handwritten text.
4. It introduces garbage characters in some cases.
5. It cannot analyze the text of languages that are out of its scope.

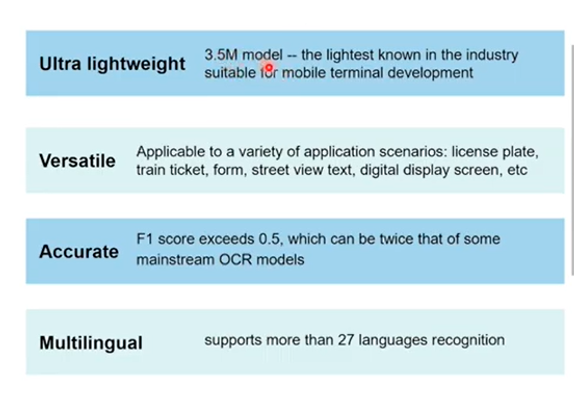
# PaddleOCR:

# The latest lightweight OCR system, provides an easy-to-use ultra lightweight OCR system.

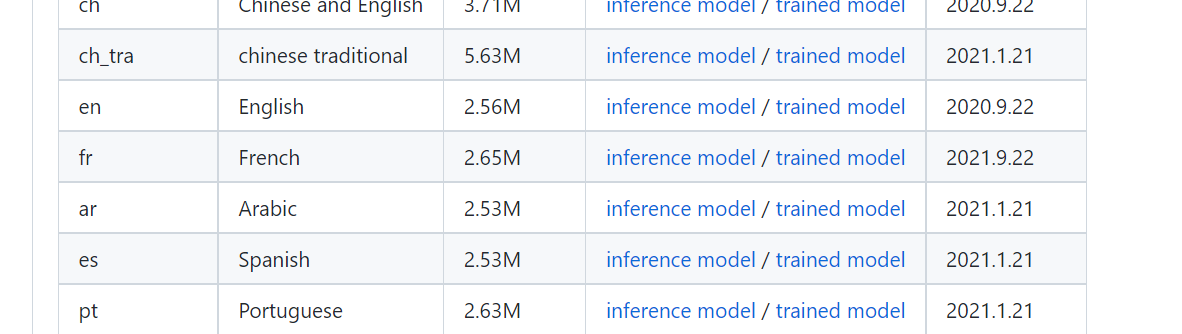
Pour paddleOCR le model de l ’arabe et nouveau en 2021,il est encore besoin de quelque amelioration,pour chinese et english il est performant

PaddleOCR For mobile

<https://medium.com/axinc-ai/paddleocr-the-latest-lightweight-ocr-system-a13171d7ea3e>

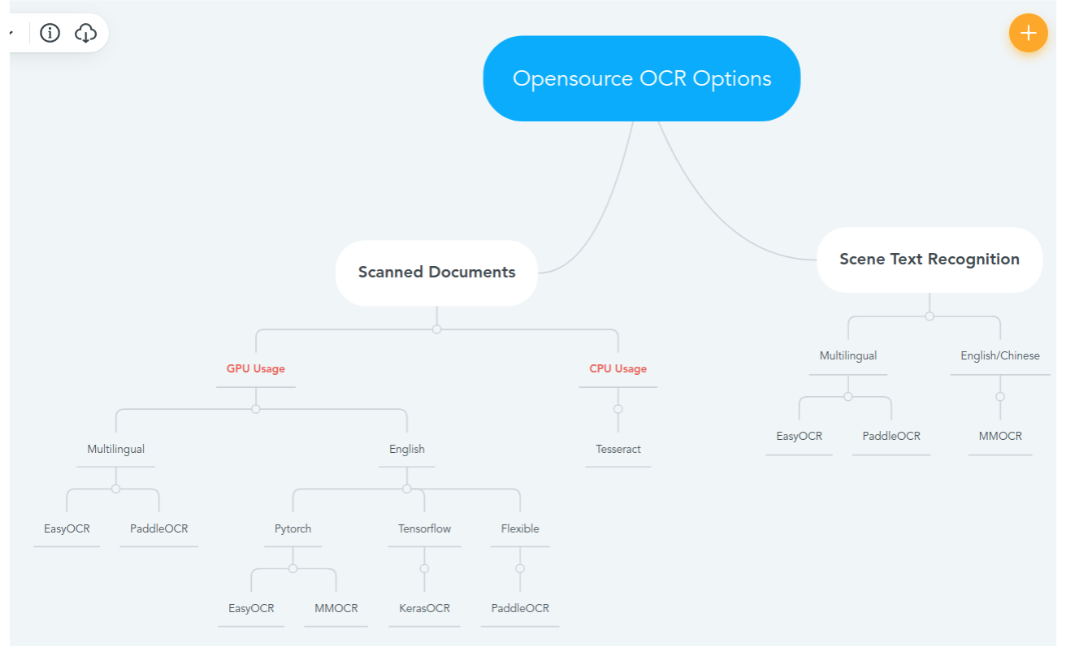


Paddle ocr model size :

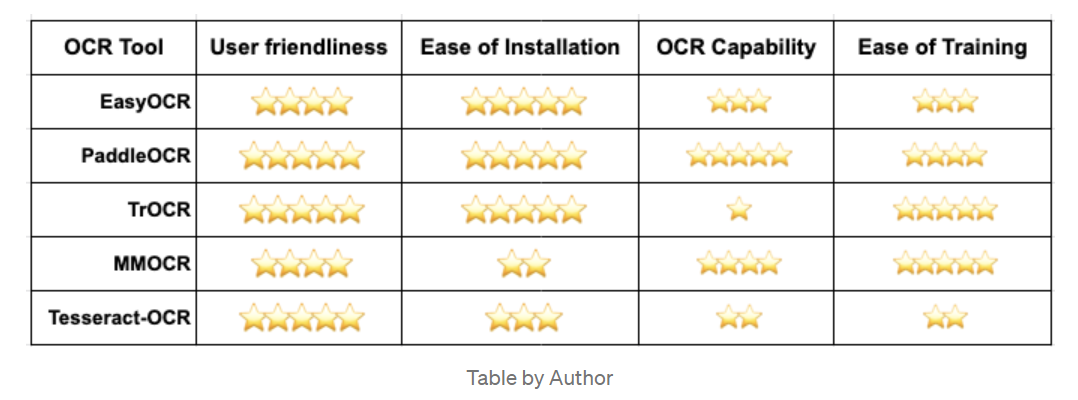


* + Ultra lightweight PP-OCRv2 series models: detection (3.1M) + direction classifier (1.4M) + recognition 8.5M) = 13.0M
  + Ultra lightweight PP-OCR mobile series models: detection (3.0M) + direction classifier (1.4M) + recognition (5.0M) = 9.4M
  + General PP-OCR server series models: detection (47.1M) + direction classifier (1.4M) + recognition (94.9M) = 143.4M
  + Support multi-lingual recognition: about 80 languages

Choose your OpenSource OCR :



comparaison entre quelque opensource ocr engines :



<https://towardsdatascience.com/5-open-source-tools-you-can-use-to-train-and-deploy-an-ocr-project-8f204dec862b>

ArabicOcr LIBRARY  :

# ArabicOcr Package to convert any Arabic image text to text by ocr techniques

Library in python .

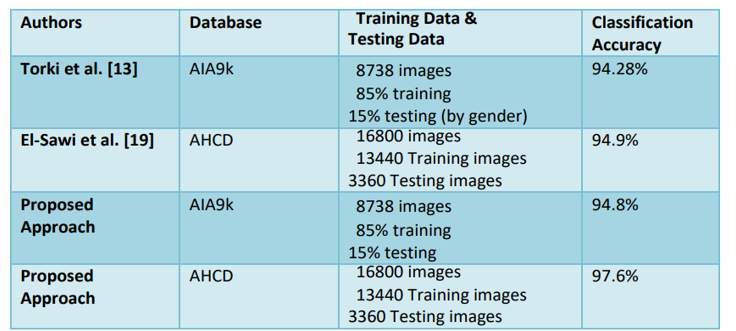
Pip install ArabicOcr

Apres implementer le model sur les images qui contient l arabe

Voir le notebook envoyer ..

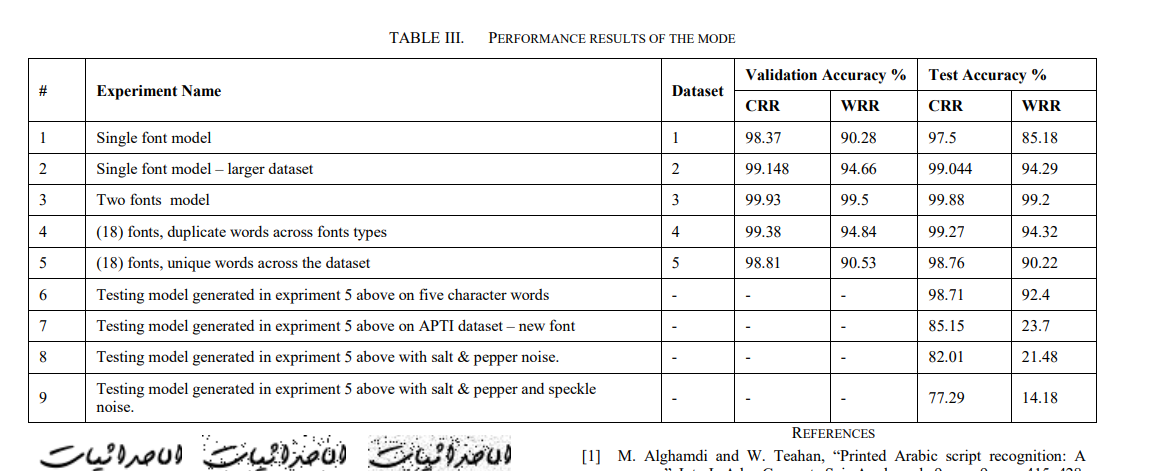
CNN :

capable of classifying Arabic handwritten characters with a stateof-the-art classification accuracy of 94.8% and 97.6% on the AIA9k and AHDC datasets :



Evaluating CNN model:

To evaluate the model, CRR and Words Recognition Rate (WRR) measures were employed :



The dataset used :

from kaggle kernels which include **[Arabic Digits](https://www.kaggle.com/mloey1/ahdd1)** and **[Arabic Letters](https://www.kaggle.com/mloey1/ahcd1)**.

* **Arabic Digits Dataset represents MADBase  which contains 60,000 training images, and 10,000 test images.**
* **Arabic Letters Dataset is composed of 16,800 characters written by 60 participants**

**Phases and results :**

Data Preprocessing

Image Normalization

### Encoding Categorical Labels(one hote encoding)

## Designing Model Architecture

As results we can see that best parameters are:

* Optimizer: Adam
* Kernel\_initializer: uniform
* Activation: relu

## training and fitting Model

Result :

**Test accuracy is improved from 98.286% to 98.862% As we train the model on 20 more epochs. so the CNN model is tested on more than 13000 image with all possible classes and got very high accuracy of 98.86%.**

CNN and RNN :

model takes an input image and generates feature sequences through a CNN. These sequences are transferred to a bidirectional RNN to obtain feature sequences in order.

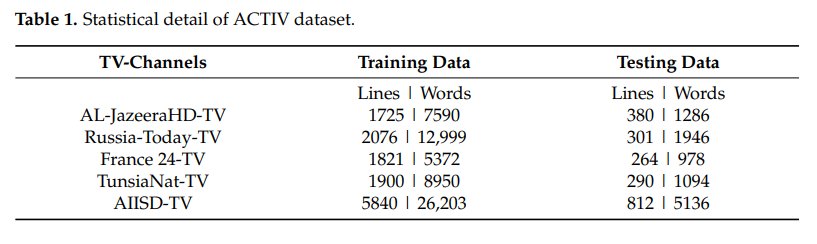
Datasets used :

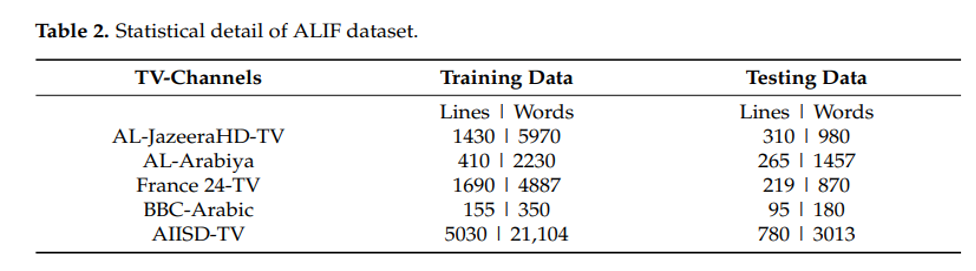
ACTIV (21520 images)and ALIF(6532 text lines images); they are firstly preprocessed, then fed to the convolutional layer.

VGG architecture used ;

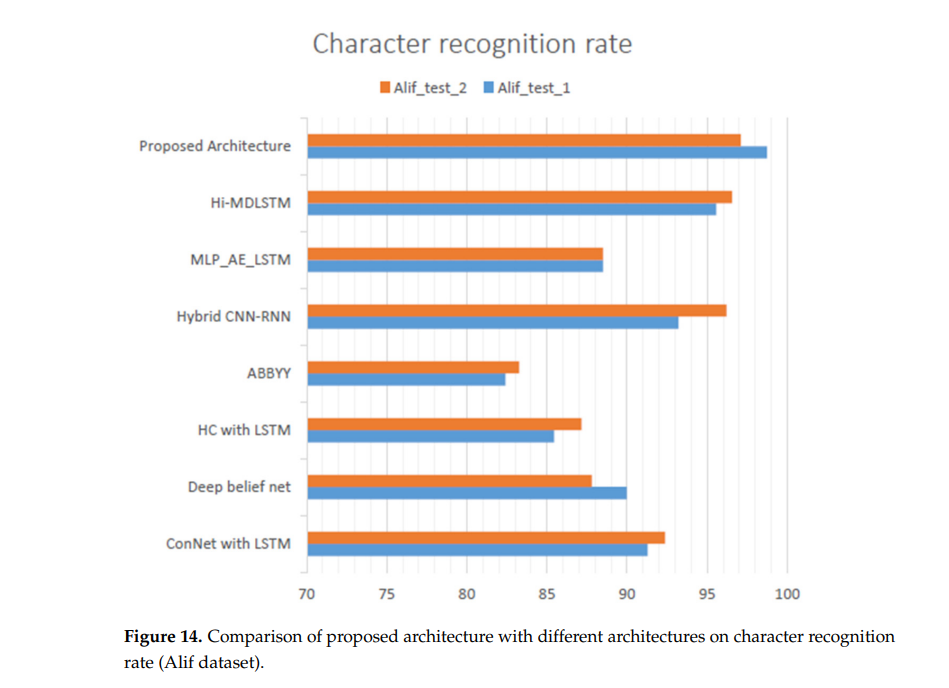
Article de 2021 :

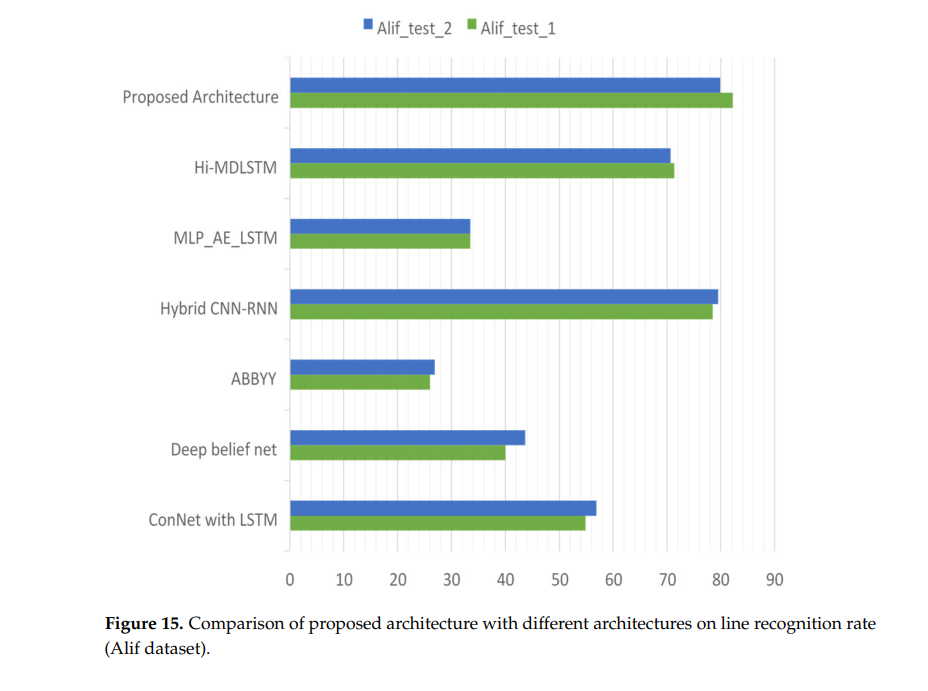
file:///C:/Users/pc/Downloads/forecasting-03-00033-v2%20(1).pdf

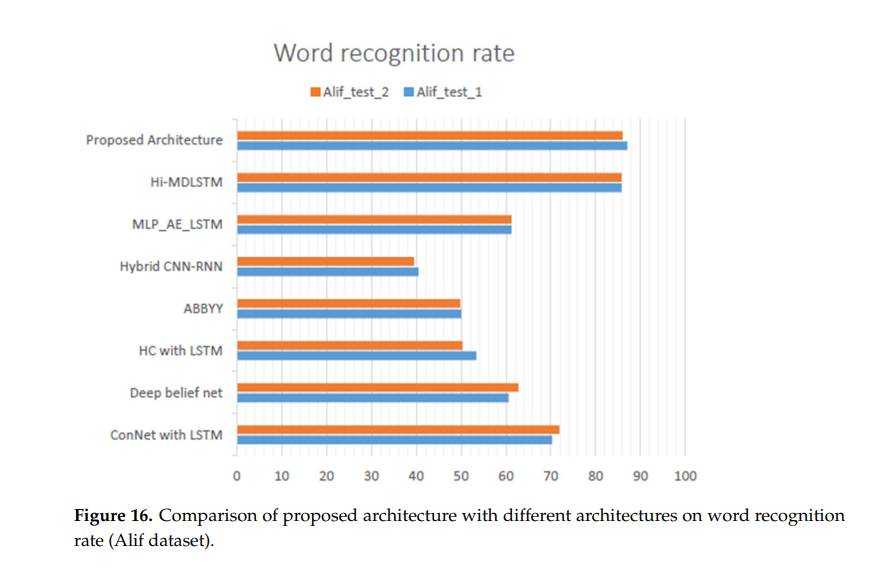


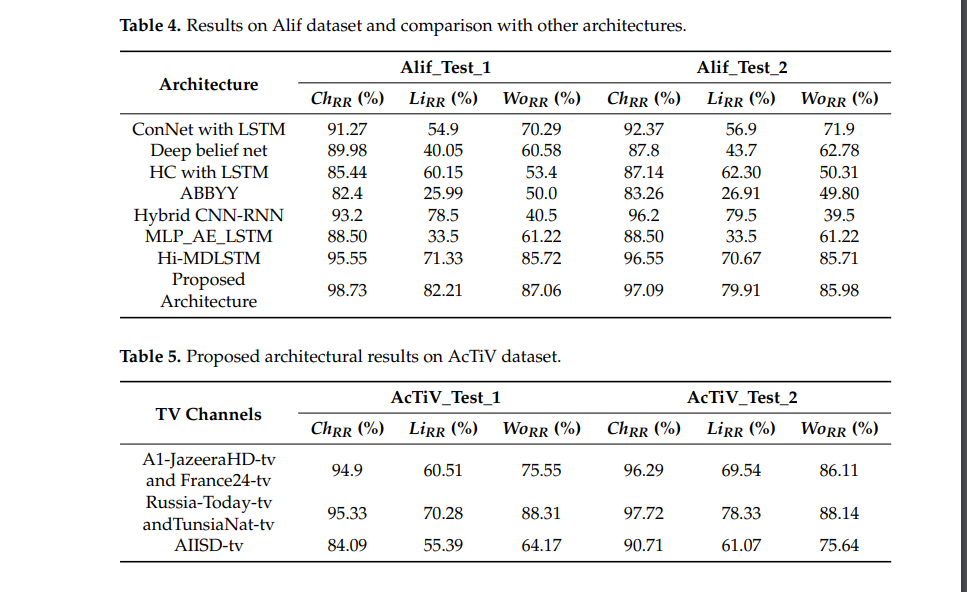


Results and Performance of word,line and character recognation rate on ALIF dataset :









HMMS :

resistant to noise, tolerate variations in writing, and the HMM tools are freely available.

HMM models have obtained a correctness rate of 73.78 % and an accuracy rate of 71.74 % for window features using horizontal and vertical edge derivatives of the image.

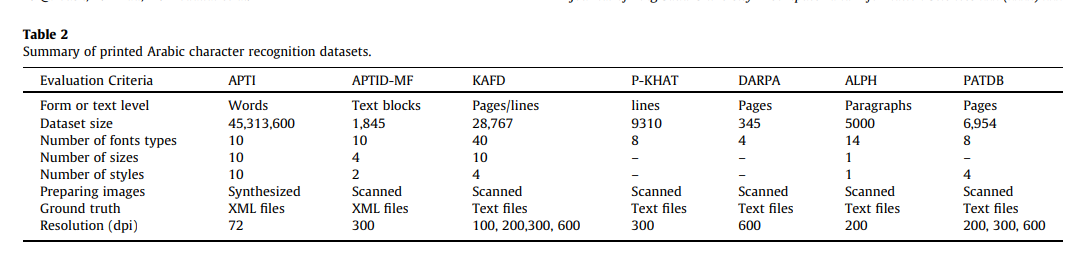
Using the hierarchal window features, the best results obtained are a correctness rate of 81.05 % and an accuracy rate of 78.04 %.

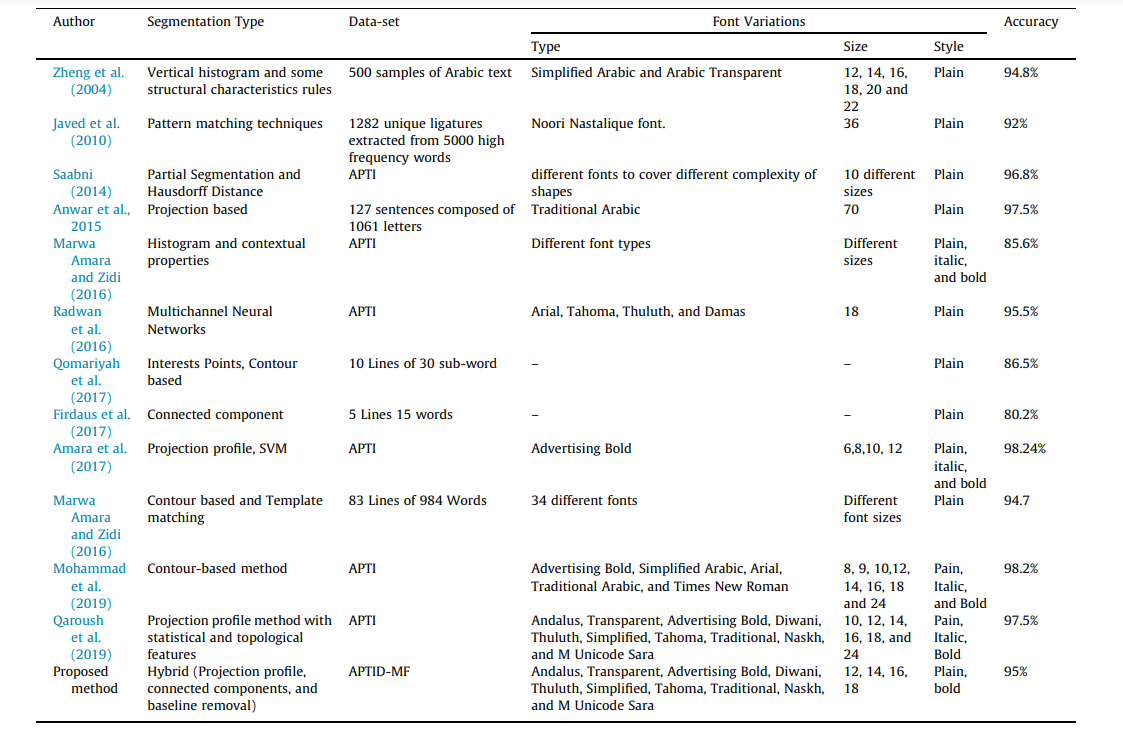
Alphanumeric VGG :

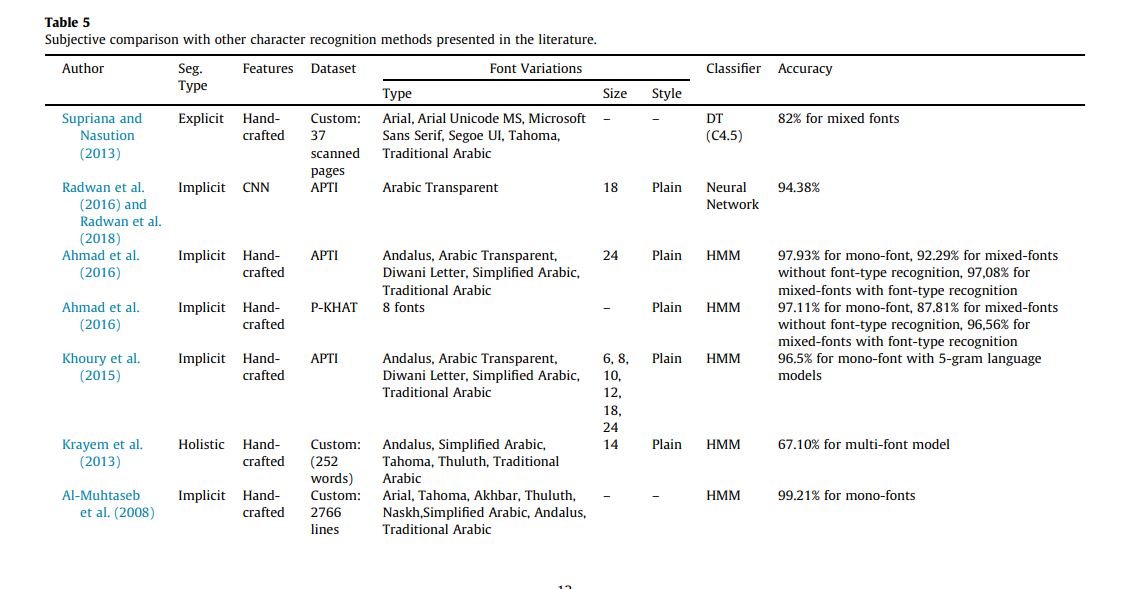
results show that: the accuracy of 97.32% for the HACDB database and 99.66% for the ADBase database have been achieved.

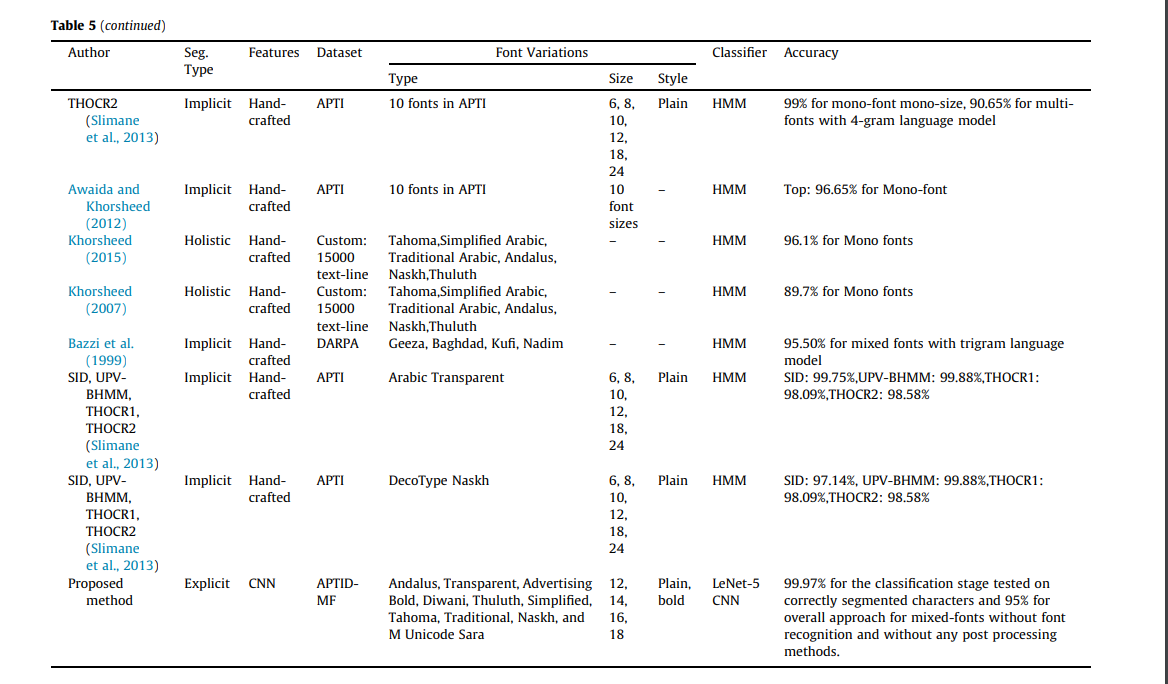
TABLES COMPARATIFS :

 DATASETS :









FOR MOBILE using android studio  :

1)ML Kit language identification(Firebase):

ML Kit is a library from Google include computer vision-related tasks, such as text recognition API.

 In build gradle(app)

implementation 'com.google.android.gms:play services-mlkit-text-recognition:16.1.1’

2)

On android studio :

using TextRecognizer API from the google vision. libraries needed are implemented in the gradle (App) . Once implemented , it makes the required APIs available in the Java code

implementation **'com.google.android.gms:play-services-vision:19.0.0'**

3)Openalpr library :

Pour detecter le texte dans les plaques des voitures

OpenALPR is an open source  library written in C++ with bindings in C#, Java, Node.js, Go, and Python. The library analyzes images and extract texts

https://github.com/SandroMachado/openalpr-android