

Projet annuel

DIGIT RECOGNITION

Equipe:

CHELLI Celia

MEDJADI Idir

MOMIN Faruk

ABBAS Gharib



Sommaire

- 01 Présentation du contexte
 - 02 Présentation du groupe et des rôles de chacun
 - 03 Présentation du projet
 - 04 Conclusion et retours d'expériences



PROBLÉMATIQUE

On a tous une écriture différent,
parfois pas trop lisible comment L'IA
pourra nous aider, a ce comprendre
les uns des autres ?

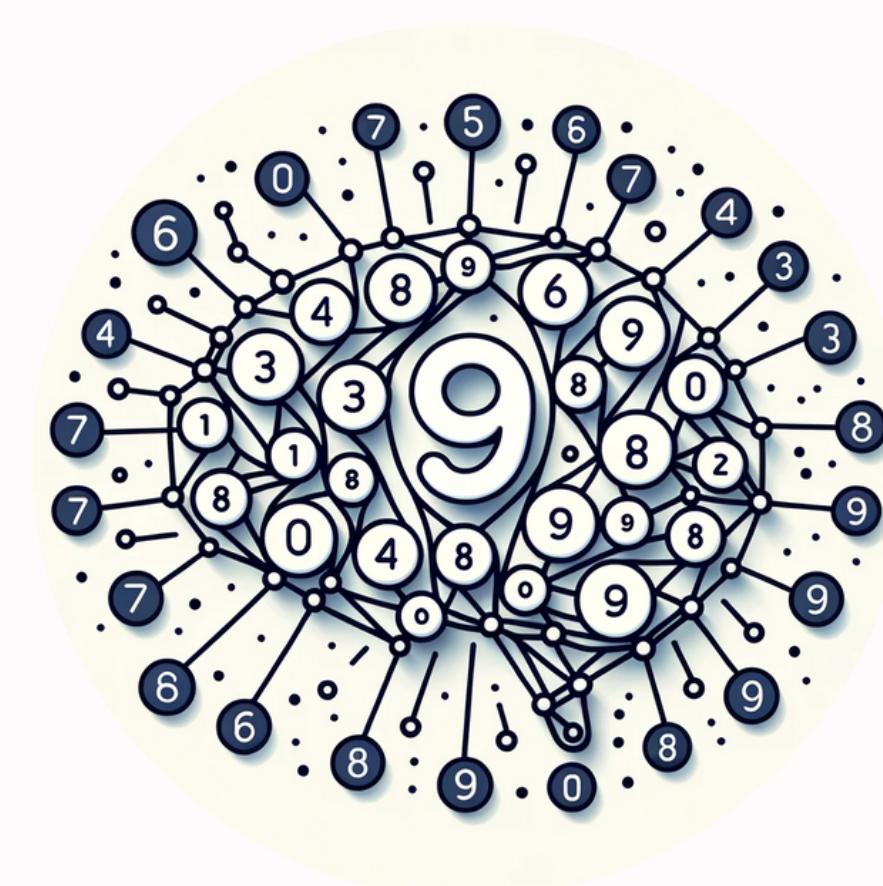
Contexte

la reconnaissance de chiffres manuscrits (Digit Recognition).

01



Le projet vise à développer un système capable de reconnaître des chiffres manuscrits.



L'application est développée en utilisant :

- Le framework Django (Python).
- Le package scikit-learn.
- MongoDB-Atlas et pymongo pour stocker
- Html, css et Javascript



02

03



Le jeu de données MNIST est un ensemble d'images en noir et blanc de chiffres manuscrits de 0 à 9. Ces images ont déjà été traitées pour obtenir le format CSV.

Présentation du groupe et des rôles de chacun

Notre équipe se compose de :

CHELLI Celia



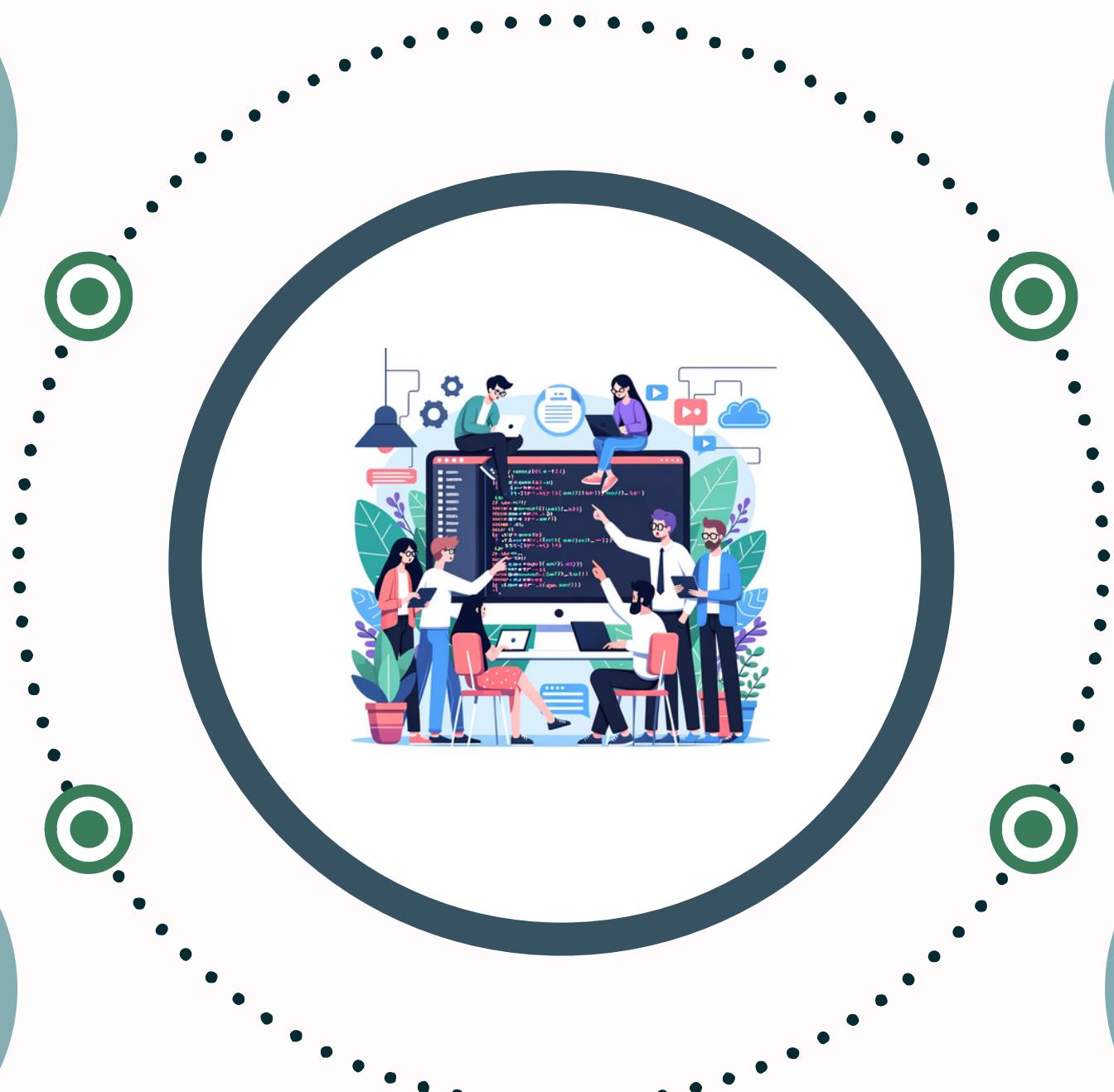
MEDJADI Idir



MOMIN Faruk



ABBAS Gharib



Trello Espaces de travail < Plus +

space de travail de Celia Chelli

Tableau < Filtre Partage

Projet annuel IA

Tâches a réaliser

- Gharib: Construction du Modèle (GA)
- Farouk: Évaluation et Optimisation du Modèle (F)
- Celia: Test et Déploiement du Modèle (CC)

Tâches faites (ok)

- Idir - Préparation des Données: (CC, MI)
- Ensemble de l'équipe: Documentation et Présentation (CC, F, GA, MI)
- Préparer une présentation (CC)

+ Ajouter une carte

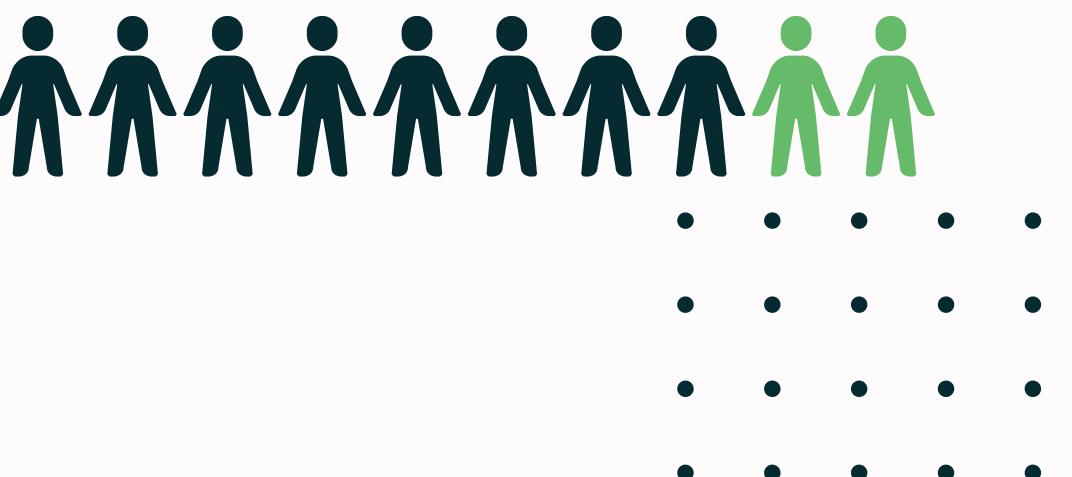
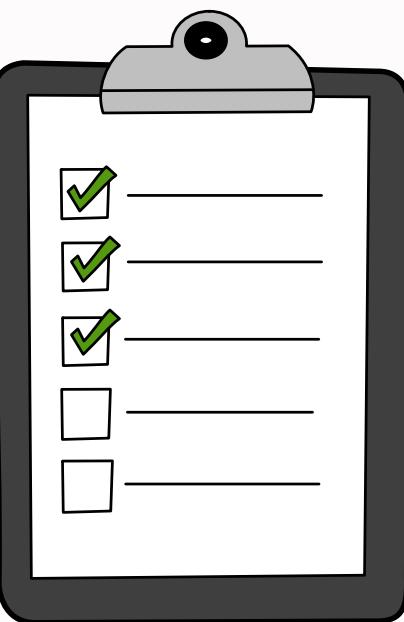
+ Ajouter une carte

Essayer Premium gratuitement

The screenshot shows a Trello board titled "Projet annuel IA". It has two main columns: "Tâches a réaliser" (Tasks to do) and "Tâches faites (ok)" (Tasks done). In the "Tâches a réaliser" column, there are three cards: "Gharib: Construction du Modèle" (status: GA), "Farouk: Évaluation et Optimisation du Modèle" (status: F), and "Celia: Test et Déploiement du Modèle" (status: CC). In the "Tâches faites (ok)" column, there are three cards: "Idir - Préparation des Données:" (status: CC, MI), "Ensemble de l'équipe: Documentation et Présentation" (status: CC, F, GA, MI), and "Préparer une présentation" (status: CC). There are also buttons to "+ Ajouter une carte" (Add a card).

Tâches

Nous avons partagé les tâches du projet tous ensemble en utilisant l'interface Trello, et dans l'ensemble nous avons travaillé ensemble pour que tout puisse nous s'apporter des connaissances.



DIGIT RECOGNITION



Notre application

Front-end

Digit Recognition

A

Pratiquer l'Alphabet

Apprenez et pratiquez les lettres de
l'alphabet

5

Pratiquer les Chiffres

Apprenez et pratiquez l'écriture des
chiffres



Ajouter des Données

Améliorer le modèle en ajoutant plus de
données

© 2023 Digit Recognition.

Nous avons utilisé html, css et javascript

Notre application

Back-end

```
2  import numpy as np
3  from bidict import bidict
4  from random import choice
5  from tensorflow import keras
6  from django.shortcuts import render, redirect
7  from django.template import loader
8  import cv2
9  import ssl
10
11 # Connexion à MongoDB
12 client = MongoClient('mongodb+srv://faruktest:wPViRRiItg2dbVjk@clustermachinelearning.hajw2e7.mongodb.net/', tls=True,
13                      tlsAllowInvalidCertificates=True)
14
15 db = client.data_csv
16
17 data_collection = db.data
18 for document in data_collection.find():
19     print(document)
20
21 ENCODER = bidict({
22     'A': 1, 'B': 2, 'C': 3, 'D': 4, 'E': 5, 'F': 6,
23     'G': 7, 'H': 8, 'I': 9, 'J': 10, 'K': 11, 'L': 12,
24     'M': 13, 'N': 14, 'O': 15, 'P': 16, 'Q': 17, 'R': 18,
25     'S': 19, 'T': 20, 'U': 21, 'V': 22, 'W': 23, 'X': 24,
26     'Y': 25, 'Z': 26
27 })
28
29 def index(request):
30     return render(request, "index.html")
```

Notre application

Base de donnée

The screenshot shows the MongoDB Compass interface. The top navigation bar has a cluster icon and the text "clustermachine...". A dropdown menu is open, showing "Documents" and "data_csv.data" (which is highlighted). Below the navigation is a sidebar with "My Queries", "Databases" (with "data_csv" selected), and a search bar. The main area is titled "data_csv.data" and shows a table of documents. The table header includes "Documents", "Aggregations", "Schema", "Indexes", and "Validation". A filter bar at the top of the table says "Type a query: { field: 'value' } or Generate query". To the right of the table, it says "60.0k DOCUMENTS 1 INDEXES". At the bottom of the table, there are buttons for "ADD DATA", "EXPORT DATA", and "OPTIONS". The table itself lists document IDs from 0 to 56, each with a value column. The values are: 0: 0, 1: 0, 2: 0, 3: 0, 5: 0, 9: 0, 11: 37, 14: 48, 16: 0, 18: 28, 23: 252, 24: 48, 25: 0, 26: 237, 27: 0, 30: 0, 35: 0, 36: 0, 39: 0, 43: 253, 45: 0, 46: 0, 49: 0, 55: 0, 56: 122.

ID	Value
0	0
1	0
2	0
3	0
5	0
9	0
11	37
14	48
16	0
18	28
23	252
24	48
25	0
26	237
27	0
30	0
35	0
36	0
39	0
43	253
45	0
46	0
49	0
55	0
56	122

MongoDB

Entraînement des données

The screenshot shows a Jupyter Notebook interface with the title bar "alphabet_machine_learning". The Explorer sidebar on the left displays a file tree for a project named "ALPHABET_MACHINE_LEA...". The main area contains a code cell with the following Python code:

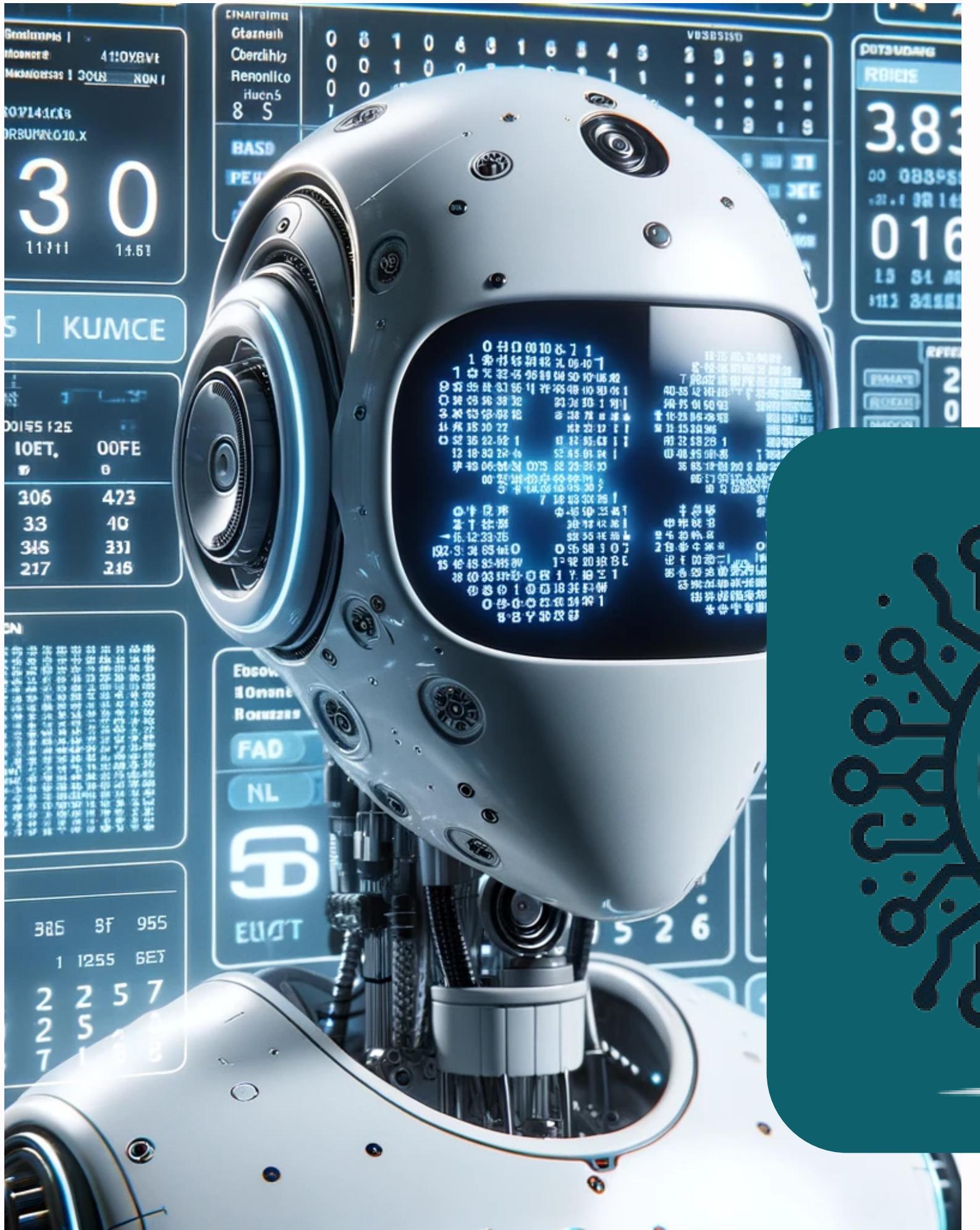
```
model.fit(X_train, y_train, validation_data=(X_val, y_val), epochs=10, batch_size=32)
```

Below the code cell, the output shows the training progress over 10 epochs:

```
[12] 70m 32.3s Python
```

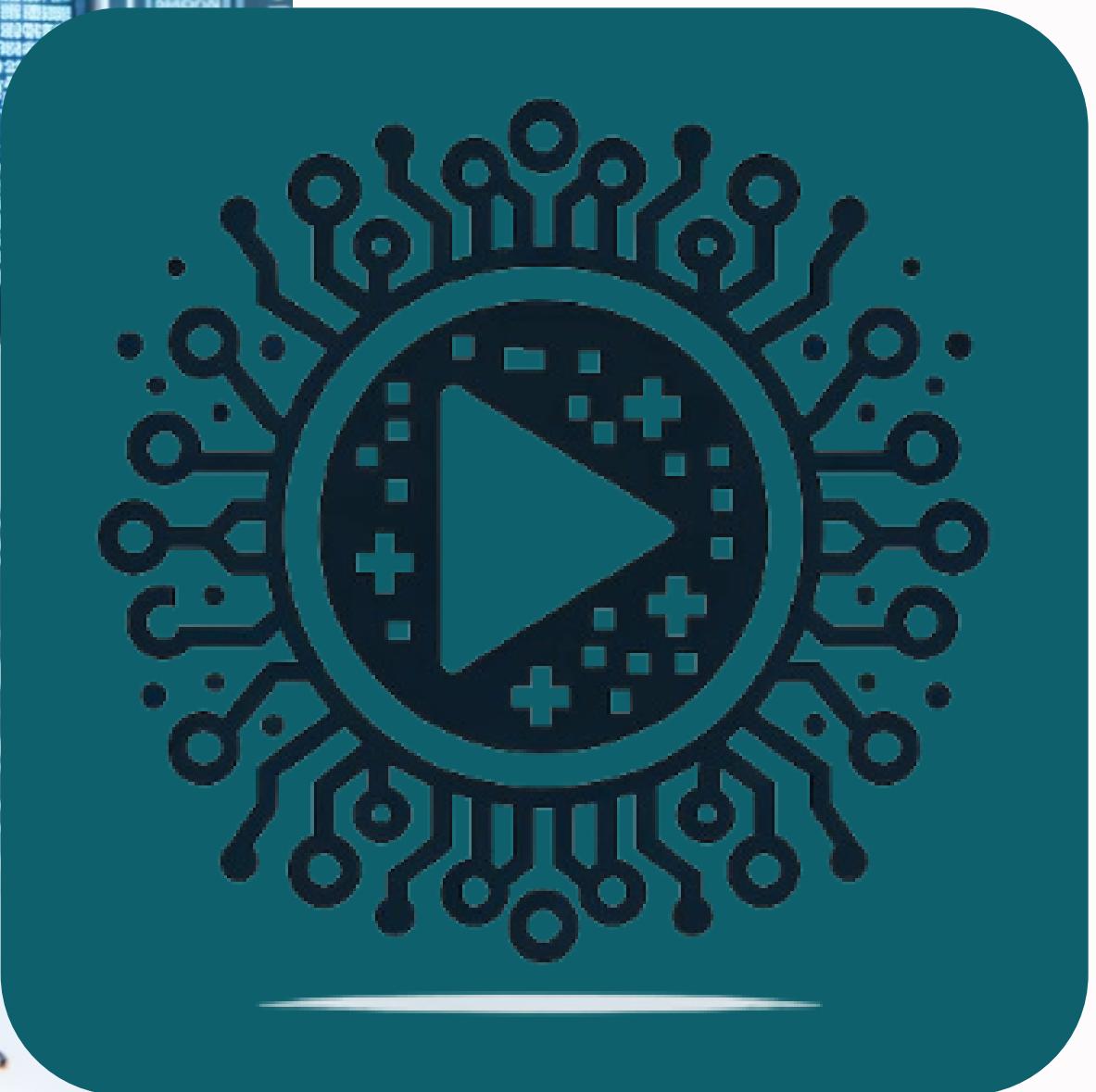
```
... Epoch 1/10  
1500/1500 [=====] - 612s 403ms/step - loss: 11.2158 - accuracy: 0.0000e+00 - val_loss: 11.5918  
Epoch 2/10  
1500/1500 [=====] - 608s 405ms/step - loss: 10.9378 - accuracy: 0.0000e+00 - val_loss: 12.4100  
Epoch 3/10  
1500/1500 [=====] - 536s 357ms/step - loss: 10.8890 - accuracy: 0.0000e+00 - val_loss: 13.1065  
Epoch 4/10  
1500/1500 [=====] - 403s 269ms/step - loss: 10.8664 - accuracy: 0.0000e+00 - val_loss: 13.7497  
Epoch 5/10  
1500/1500 [=====] - 417s 278ms/step - loss: 10.8537 - accuracy: 0.0000e+00 - val_loss: 14.3629  
Epoch 6/10  
1500/1500 [=====] - 370s 246ms/step - loss: 10.8454 - accuracy: 0.0000e+00 - val_loss: 14.9512  
Epoch 7/10  
1500/1500 [=====] - 350s 234ms/step - loss: 10.8396 - accuracy: 0.0000e+00 - val_loss: 15.5098  
Epoch 8/10  
1500/1500 [=====] - 412s 275ms/step - loss: 10.8351 - accuracy: 0.0000e+00 - val_loss: 16.0282  
Epoch 9/10  
1500/1500 [=====] - ETA: 0s - loss: 10.8314 - accuracy: 0.0000e+00
```

The bottom status bar shows "Ln 1, Col 1" and "12:39 13/10/2023".



Demo

Voici une démonstration de notre application.



4
7

9
2

