

# **“An AI-based Arabic handwritten character recognition module as a foundational component for automated exam grading systems”**

## **Domain:**

Artificial Intelligence / Machine Learning – Computer Vision  
(Handwritten Text Recognition)

## **Problem Statement:**

Manual grading of Arabic handwritten exam papers is time-consuming, error-prone, and inconsistent due to handwriting variations. Automated grading systems require accurate recognition of handwritten Arabic text as a prerequisite step. However, Arabic handwriting recognition is challenging because of complex character shapes and visual similarity between letters. This project addresses this challenge by developing a Convolutional Neural Network (CNN) model capable of recognizing individual Arabic handwritten letters, which serves as a critical building block toward an automated Arabic exam grading system.

## **Project Statement**

This project implements a CNN-based system to classify Arabic handwritten letters from scanned images. The trained model accurately recognizes 28 Arabic characters from grayscale images and is deployed using a Streamlit interface for real-time prediction. The system demonstrates the feasibility of Arabic handwritten character recognition as a first step toward automatic grading of handwritten exams.

## **Data collection and Processing using Python**

1-anaconda installed, I used Anaconda prompt

2-Jupyter Notebook installed

3- Python libraries installed

## Data source Link :

The dataset used for training and testing was sourced from a publicly available GitHub repository consisting of Arabic handwritten letter images labeled into 28 classes.

[Arabic-Handwritten-Characters-Dataset](https://github.com/mloey/Arabic-Handwritten-Characters-Dataset)

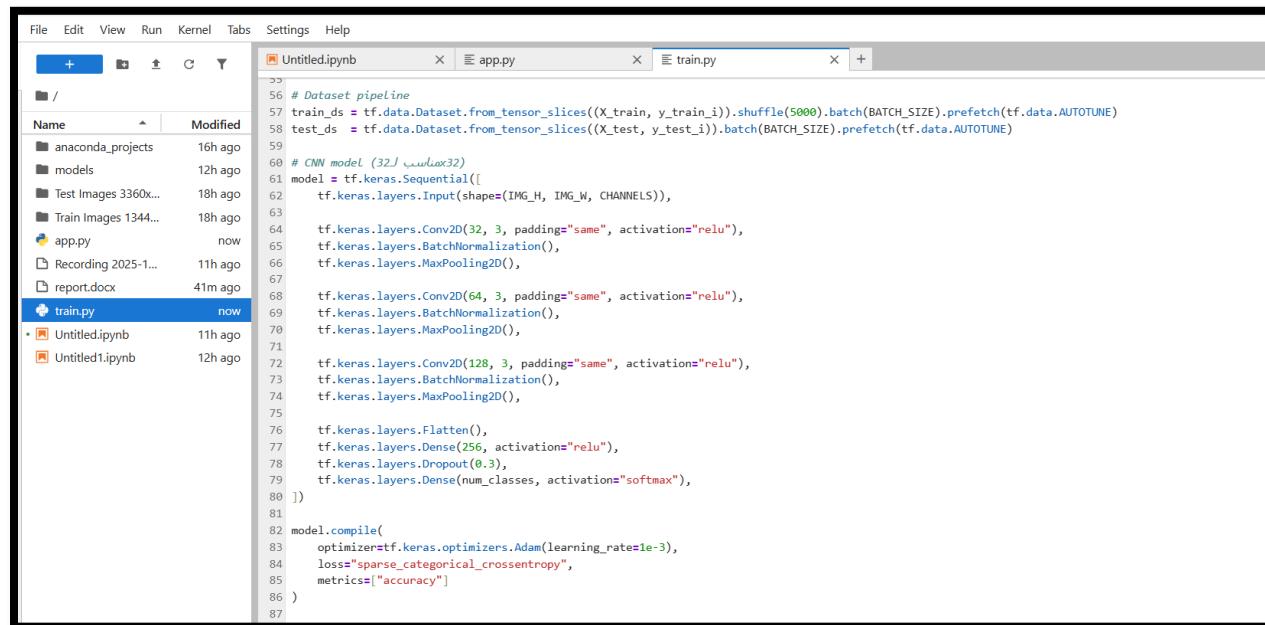
## Coding part :

Link to my files: train.py and app.py

<https://github.com/UniverseTalker/AR-HW-CNN.git>

- Input: 32x32 grayscale images
- Classes: 28 Arabic letters
- Model: **Convolutional Neural Network (CNN)**
- Framework: **TensorFlow / Keras**
- Deployment: Streamlit

In this picture part of the code related to the CNN Model training

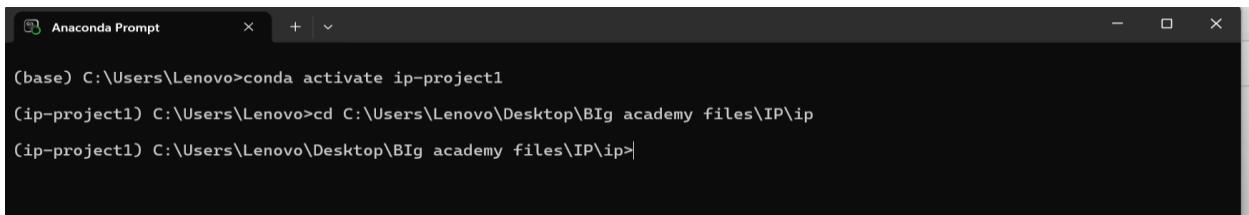


```
File Edit View Run Kernel Tabs Settings Help
Untitled.ipynb app.py train.py +
Name Modified
anaconda_projects 16h ago
models 12h ago
Test Images 3360x... 18h ago
Train Images 1344... 18h ago
app.py now
Recording 2025-1... 11h ago
report.docx 41m ago
train.py now
Untitled.ipynb 11h ago
Untitled1.ipynb 12h ago

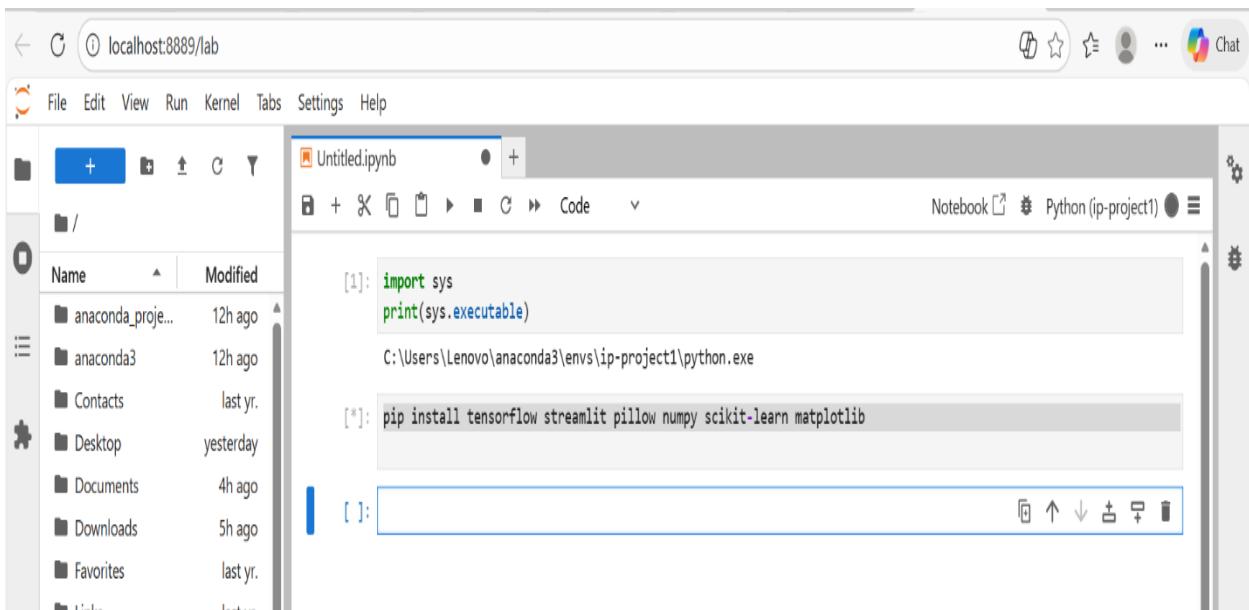
56 # Dataset pipeline
57 train_ds = tf.data.Dataset.from_tensor_slices((X_train, y_train_i)).shuffle(5000).batch(BATCH_SIZE).prefetch(tf.data.AUTOTUNE)
58 test_ds = tf.data.Dataset.from_tensor_slices((X_test, y_test_i)).batch(BATCH_SIZE).prefetch(tf.data.AUTOTUNE)
59
60 # CNN model (32x32x3)
61 model = tf.keras.Sequential([
62     tf.keras.layers.Input(shape=(IMG_H, IMG_W, CHANNELS)),
63
64     tf.keras.layers.Conv2D(32, 3, padding="same", activation="relu"),
65     tf.keras.layers.BatchNormalization(),
66     tf.keras.layers.MaxPooling2D(),
67
68     tf.keras.layers.Conv2D(64, 3, padding="same", activation="relu"),
69     tf.keras.layers.BatchNormalization(),
70     tf.keras.layers.MaxPooling2D(),
71
72     tf.keras.layers.Conv2D(128, 3, padding="same", activation="relu"),
73     tf.keras.layers.BatchNormalization(),
74     tf.keras.layers.MaxPooling2D(),
75
76     tf.keras.layers.Flatten(),
77     tf.keras.layers.Dense(256, activation="relu"),
78     tf.keras.layers.Dropout(0.3),
79     tf.keras.layers.Dense(num_classes, activation="softmax"),
80 ])
81
82 model.compile(
83     optimizer=tf.keras.optimizers.Adam(learning_rate=1e-3),
84     loss="sparse_categorical_crossentropy",
85     metrics=["accuracy"]
86 )
```

## How to run the application:

1. Activate environment (mine is: ip-project1)
2. Run training: python train.py (then model file and json file created )
3. Run streamlit app : python -m streamlit run app.py



```
Anaconda Prompt
(base) C:\Users\Lenovo>conda activate ip-project1
(ip-project1) C:\Users\Lenovo>cd C:\Users\Lenovo\Desktop\B1g academy files\IP\ip
(ip-project1) C:\Users\Lenovo\Desktop\B1g academy files\IP\ip>
```



localhost:8889/lab

File Edit View Run Kernel Tabs Settings Help

Untitled.ipynb

Notebook Python (ip-project1)

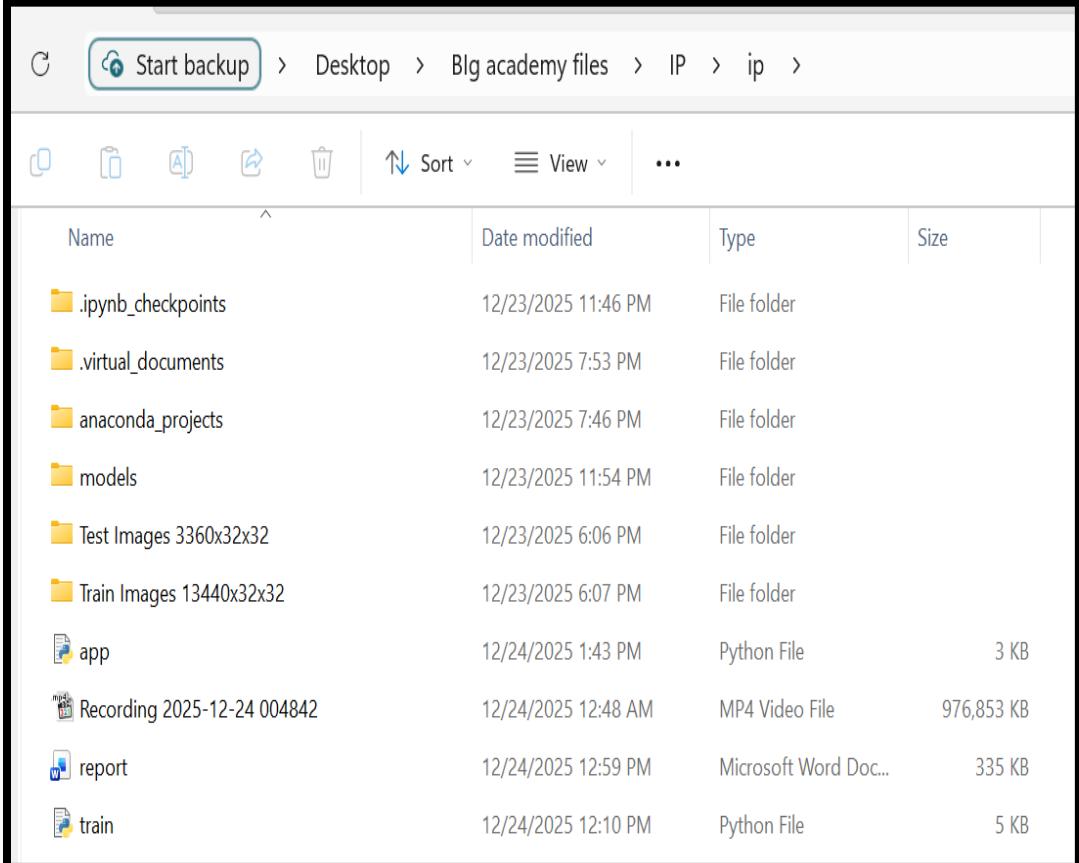
[1]: import sys  
print(sys.executable)  
C:\Users\Lenovo\anaconda3\envs\ip-project1\python.exe

[\*]: pip install tensorflow streamlit pillow numpy scikit-learn matplotlib

Name Modified

- anaconda\_project 12h ago
- anaconda3 12h ago
- Contacts last yr.
- Desktop yesterday
- Documents 4h ago
- Downloads 5h ago
- Favorites last yr.
- Index last yr.

## My project folder structure:



The screenshot shows a Windows File Explorer window with a black border. The path bar at the top shows: C: > Start backup > Desktop > Blg academy files > IP > ip >. Below the path bar is a toolbar with icons for copy, move, search, and delete, followed by 'Sort' and 'View' dropdowns and an ellipsis button. The main area is a table listing files and folders:

Name	Date modified	Type	Size
.ipynb_checkpoints	12/23/2025 11:46 PM	File folder	
.virtual_documents	12/23/2025 7:53 PM	File folder	
anaconda_projects	12/23/2025 7:46 PM	File folder	
models	12/23/2025 11:54 PM	File folder	
Test Images 3360x32x32	12/23/2025 6:06 PM	File folder	
Train Images 13440x32x32	12/23/2025 6:07 PM	File folder	
app	12/24/2025 1:43 PM	Python File	3 KB
Recording 2025-12-24 004842	12/24/2025 12:48 AM	MP4 Video File	976,853 KB
report	12/24/2025 12:59 PM	Microsoft Word Doc...	335 KB
train	12/24/2025 12:10 PM	Python File	5 KB

## Examples of Prediction :

## Arabic Letter Recognizer (CNN)

Upload a letter image (32x32 recommended). The model will predict the letter.

Upload an image

Drag and drop file here  
Limit 200MB per file • PNG, JPG, JPEG

Browse files

 id\_7\_label\_4.png 171.08

X



Uploaded image

Predicted letter



Label: 4

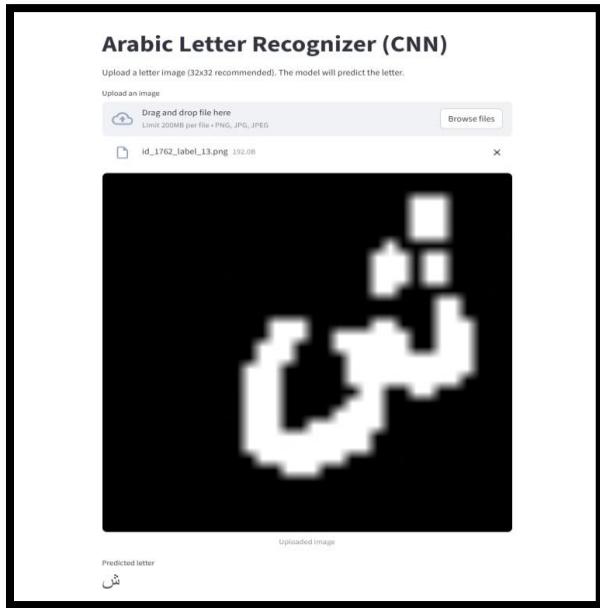
### Prediction

Confidence

99.12%

### Top 3 predictions

1. ت (label 4) - 99.12%
2. ج (label 13) - 0.66%
3. ع (label 25) - 0.11%



## Model Evaluation : Accuracy, Confusion Matrix, and Precision/Recall:

Model accuracy : 0.954

In the code we can see it inside train.py file :

```
acc = accuracy_score(y_test_i, y_pred)
print("\nTest Accuracy:", acc)
```

## Results while running file :train.py

```
Test Accuracy: 0.9541666666666667

Classification report:
      precision    recall  f1-score   support

          0       1.00     1.00     1.00      120
          1       0.99     0.98     0.99      120
          2       0.92     0.95     0.93      120
          3       0.93     0.93     0.93      120
          4       0.99     0.97     0.98      120
          5       0.93     0.98     0.96      120
          6       0.97     0.97     0.97      120
          7       0.94     0.95     0.95      120
          8       0.90     0.91     0.90      120
          9       0.91     0.97     0.94      120
```

```

accuracy                                0.95      3360
macro avg      0.95      0.95      0.95      3360
weighted avg   0.95      0.95      0.95      3360

Confusion matrix:
[[120   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   [ 0 118   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0
     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   [ 0   0 114   5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
     0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   [ 0   0   5 112   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0
     0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   [ 0   0   0   0 117   2   1   0   0   0   0   0   0   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   [ 0   0   0   0   1 118   0   0   0   0   0   0   0   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
   [ 0   0   0   0   0   3 116   0   0   0   0   0   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
   [ 0   1   0   0   0   1   0 114   2   1   0   0   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
   [ 0   0   0   0   0   0   4 109   0   2   0   0   0   0   1   1   1   0   0
     0   0   1   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
   [ 0   0   0   0   0   0   0   1   0 116   3   0   0   0   0   0   0   0   0
     0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   [ 0   0   0   0   0   0   0   1   7   9 101   0   0   0   0   0   0   0   0
     0   0   0   0   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
     0   0   0   0   1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
]

```

Jupyter Notebook

Dalal Adnan zamzam