Final Delivery

Arabic Language Classification

By:
Ahmed Jaheen
Aedan Ounsamone

TABLE OF CONTENTS

01
Problem
Statement
& Original Model

02
Proposed
Model
Architecture

03
Final
Model
Architecture

04
Work Done &
Results
Last Milestone

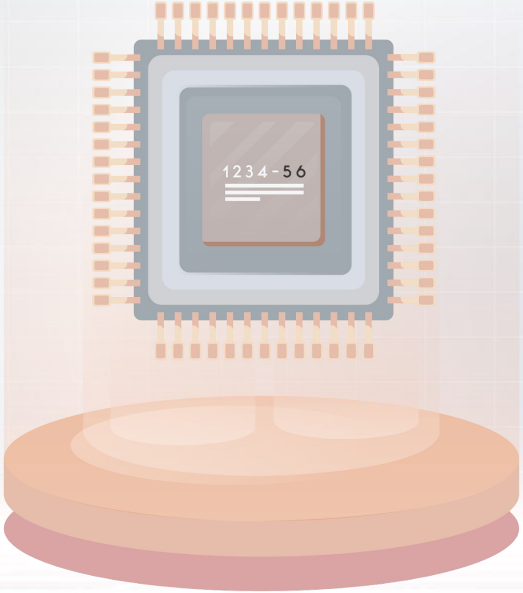
05
Progress
done &
results.

06
Final
Model &
Results

07
Demo
Website
Poster

08
Conclusion &
Future Work

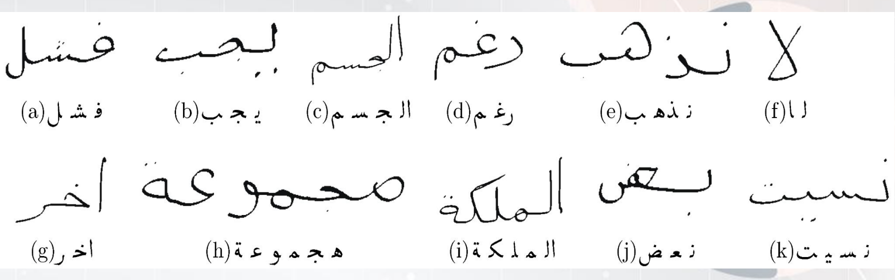
09
Members'
contributions.



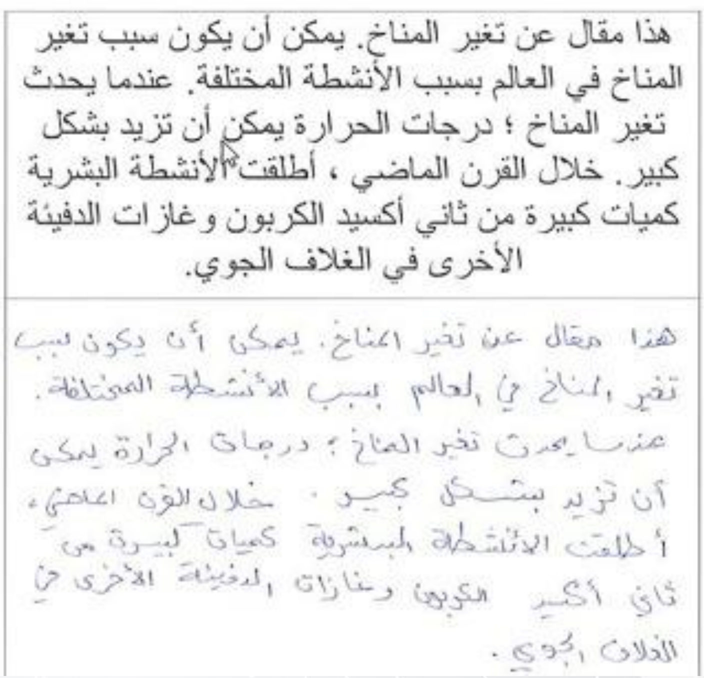
01

Problem Statement & Original Model

Problem statement



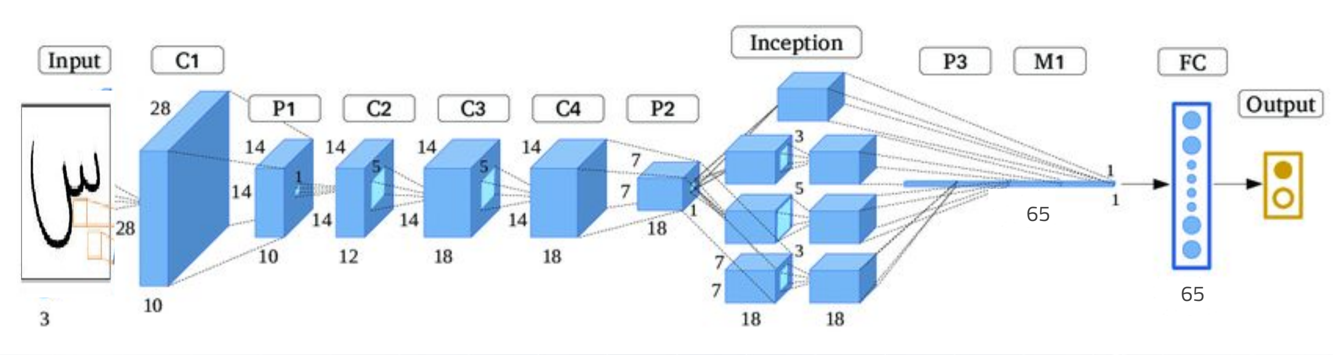
Arabic handwriting recognition presents unique challenges due to its cursive nature and character variability, which have hindered the development of robust recognition models. Our project aims to address these challenges by evaluating existing models on datasets and proposing enhancements to improve accuracy and efficiency in Arabic text recognition systems.

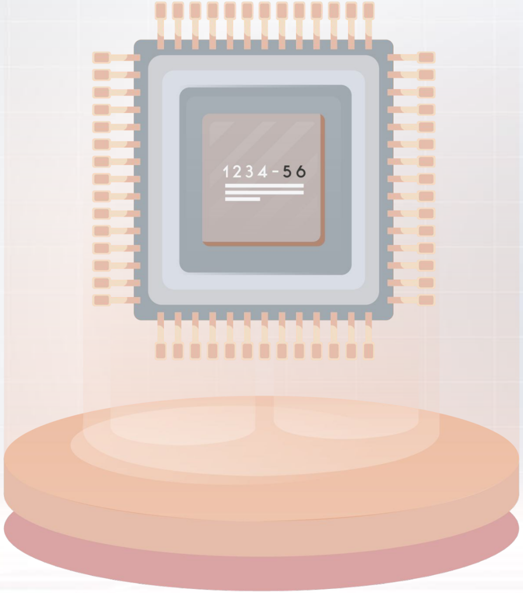


هذا مقال عن تغير المناخ. يمكن أن يكون سبب تغير
المناخ في العالم بسبب الأنشطة المختلفة. عندما يحدث
تغير المناخ ؛ درجات الحرارة يمكن أن تزيد بشكل
كبير. خلال القرن الماضي ، أطلقت الأنشطة البشرية
كميات كبيرة من ثاني أكسيد الكربون وغازات الدفيئة
الأخرى في الغلاف الجوي.

هذا مقال عن تغير المناخ. يمكن أن يكون سبب
تغير المناخ في لعالم بسبب الأنشطة المختلفة.
عندما يحدث تغير المناخ ؛ درجات الحرارة يمكن
أن تزيد بشكل كبير . خلال القرن الماضي ،
أطلقت الأنشطة البشرية كميات كبيرة من
ثاني أكسيد الكربون وغازات الدفيئة الأخرى في
الغلاف الجوي .

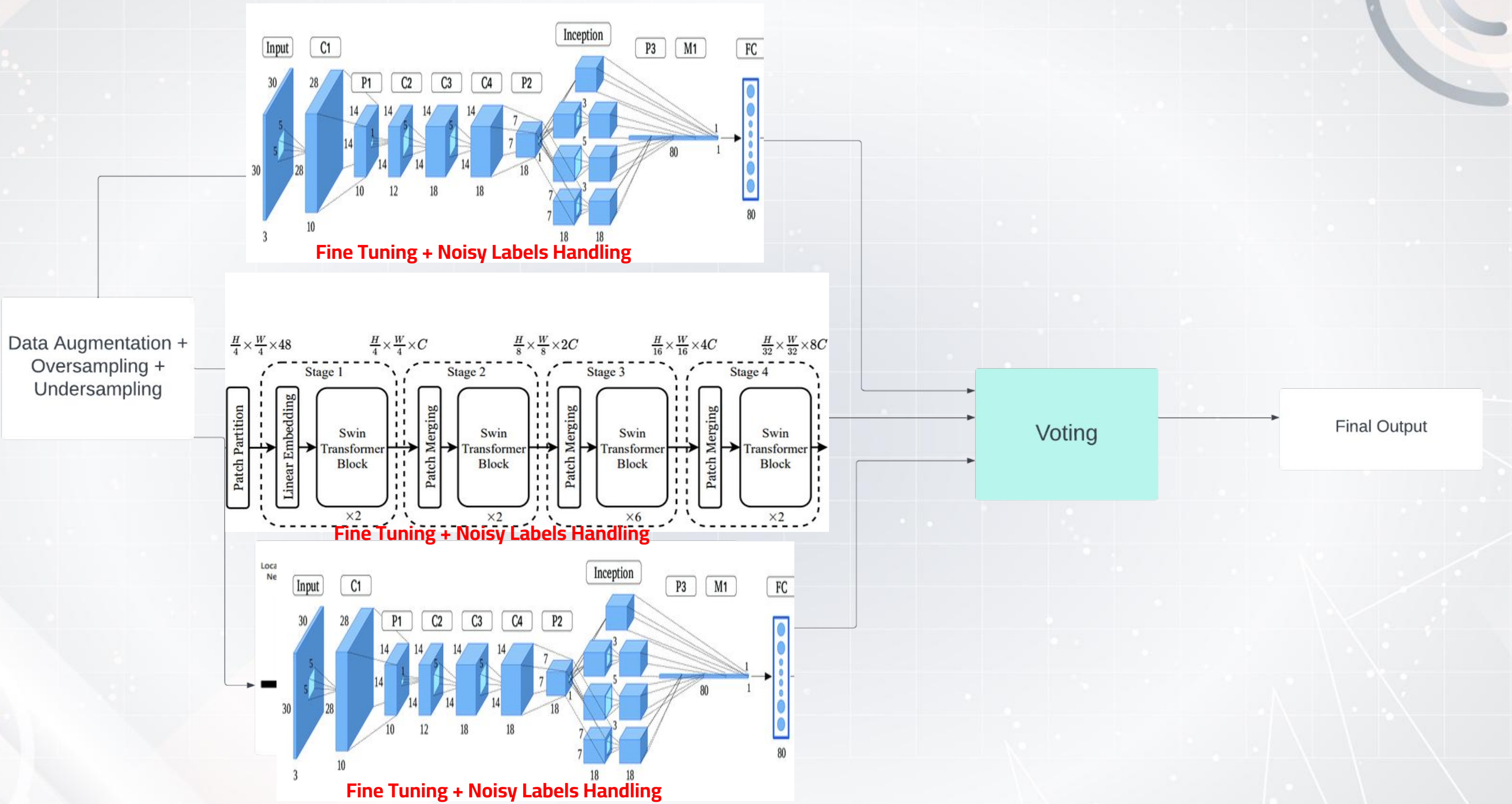
Original Model





02
Proposed Model Architecture

Proposed Model



Data Augmentation + Oversampling + Undersampling

Fine Tuning + Noisy Labels Handling

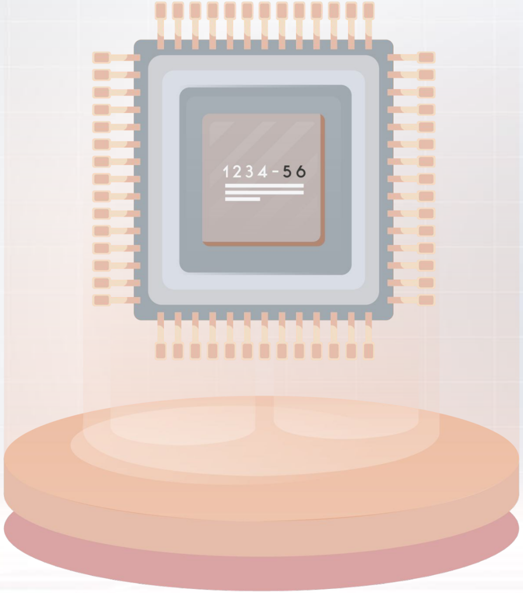
Fine Tuning + Noisy Labels Handling

Fine Tuning + Noisy Labels Handling

Voting

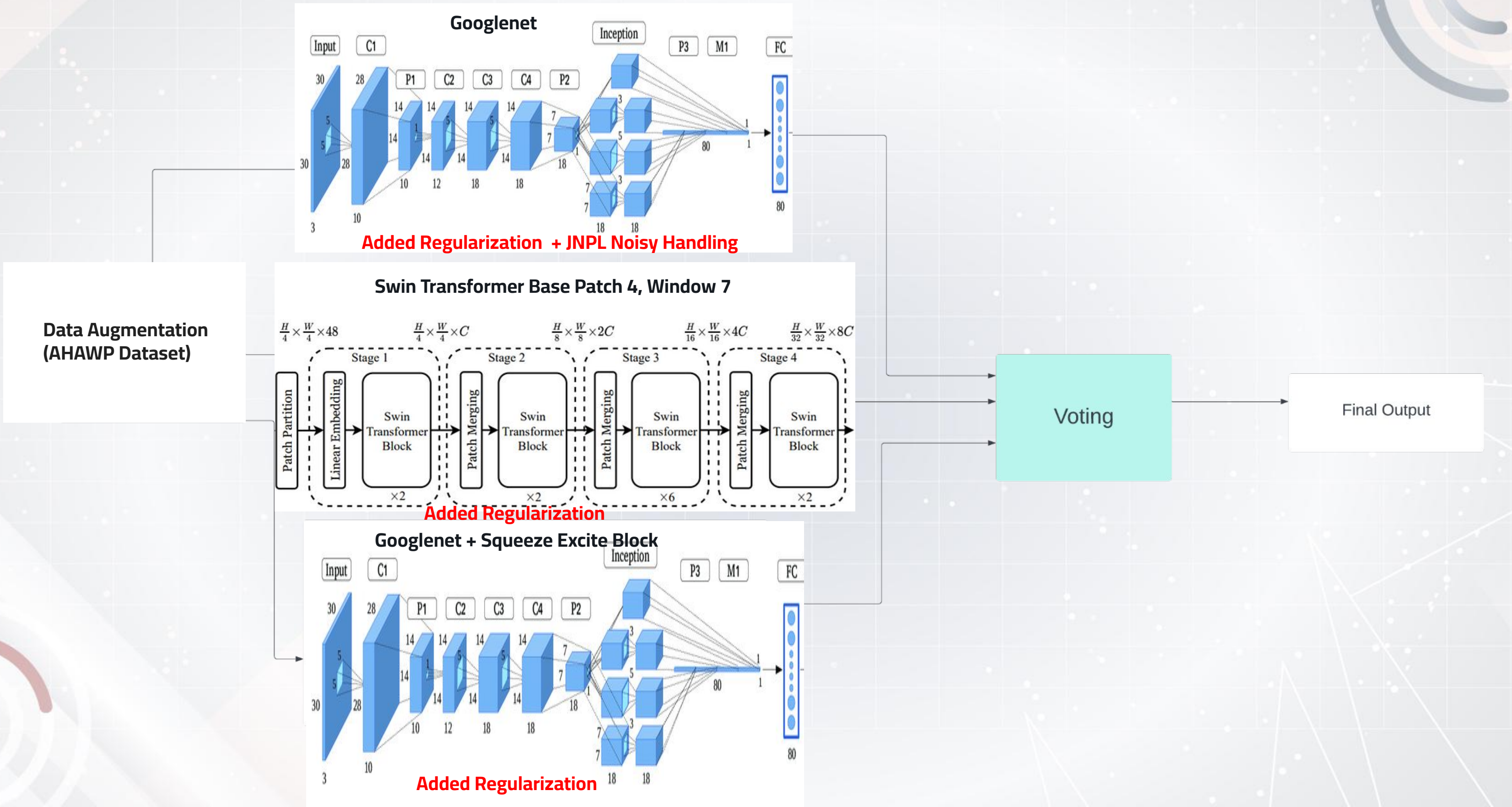
Final Output

Input
C1
P1
C2
C3
C4
P2
Inception
P3
M1
FC
30
28
14
10
12
18
7
3
5
1
80
$\frac{H}{4}\times\frac{W}{4}\times 48$
$\frac{H}{4}\times\frac{W}{4}\times C$
$\frac{H}{8}\times\frac{W}{8}\times 2C$
$\frac{H}{16}\times\frac{W}{16}\times 4C$
$\frac{H}{32}\times\frac{W}{32}\times 8C$
Stage 1
Stage 2
Stage 3
Stage 4
Patch Partition
Linear Embedding
Swin Transformer Block
Patch Merging
×2
×2
×6
×2



03
Final Model Architecture

Final Model



Data Augmentation
(AHAWP Dataset)

Googlenet

Added Regularization + JNPL Noisy Handling

Swin Transformer Base Patch 4, Window 7

Added Regularization

Googlenet + Squeeze Excite Block

Added Regularization

Voting

Final Output

Input
C1
P1
C2
C3
C4
P2
Inception
P3
M1
FC
Patch Partition
Linear Embedding
Swin Transformer Block
Patch Merging
Stage 1
Stage 2
Stage 3
Stage 4
$\frac{H}{4} \times \frac{W}{4} \times 48$
$\frac{H}{4} \times \frac{W}{4} \times C$
$\frac{H}{8} \times \frac{W}{8} \times 2C$
$\frac{H}{16} \times \frac{W}{16} \times 4C$
$\frac{H}{32} \times \frac{W}{32} \times 8C$
×2
×2
×6
×2

Updates that were not feasible:

The only update that was not feasible was running the model with OCR Techniques to validate complete words as it will require editing and resize the dataset, and it will take time that we can spend on trying more updates.

New updates implemented:

- Experimenting with more attention mechanisms.
- Experimenting with more Vision Transformers.
- Ensemble Classifiers.

04

Work Done & Results In Last Milestone

Work:

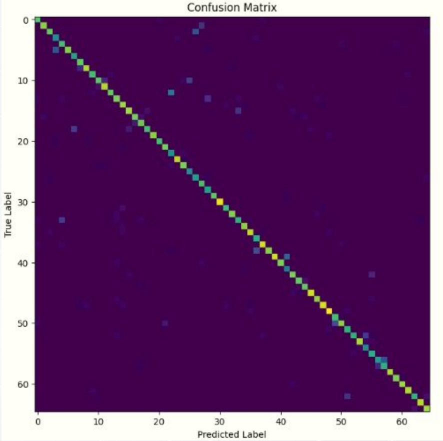
1. Successful deployment of the baseline model on AHAWP dataset.
2. Applyed Data augmentation on the AHAWP dataset.
3. Running the model on the AHAWP dataset with L2 and scheduler.
4. Experimenting with label smoothing, focal loss, and JNPL as methods of noisy labels handling.
5. After researching JNPL noisy label, we deployed it in an optimistic approach.
6. Experimenting with self attention and region based attention.
7. Researching OCR techniques to detect words.

Results:

The best results as of Milestone 2 was when running the model with regularization and with the usage of augmentation, and JNPL noisy label handling.

Accuracy : 0.9801
Weighted Precision: 0.9804
Unweighted Precision: 0.9780
Weighted recall: 0.9638
Unweighted recall: 0.9634
F-1 Score: 0.9633

Confusion Matrix:



05

Progress done
&
results.

The results of Final Phase.

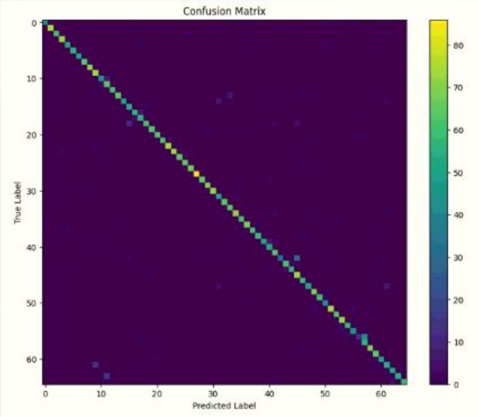
1) Running with Grid Wise attention

We tried this update as Grid-Wise attention mechanism was beneficial in one of the papers as it mainly works by transforming the image to a grid and assign different attention weights to different parts of the grid.

Results:

- Training Accuracy= 0.8409
- Validation Accuracy= 0.8650
- Testing Accuracy= 0.8578
- Weighted Precision= 0.9050
- Unweighted Precision = 0.9012
- Weighted Recall = 0.8650
- Unweighted Recall=0.8617
- F-1 Score=0.8487

- Confusion Matrix:



Conclusion:

Model Slightly Overfitting

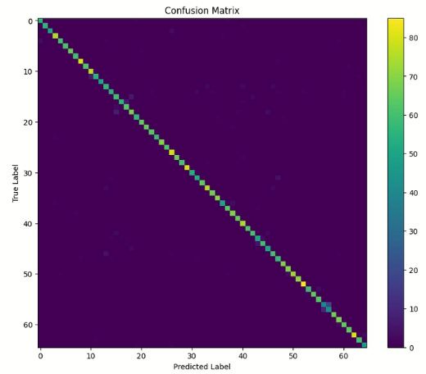
2) Running with squeeze and excitation blocks

Squeeze and excitation is designed to enhance the representation power of the neural networks especially in the context of CNNs. It is composed of two parts which are the squeeze block and the excite block. It will improve the model with less computational power as stated here: https://medium.com/@tahasamavati/squeeze-and-excitation-explained-387b5981f249

Results:

- Training Accuracy= 0.9889
- Validation Accuracy= 0.9811
- Testing Accuracy= 0.9848
- Weighted Precision= 0.9846
- Unweighted Precision = 0.9843
- Weighted Recall = 0.9840
- Unweighted Recall=0.9837
- F-1 Score=0.9845

- Confusion Matrix:



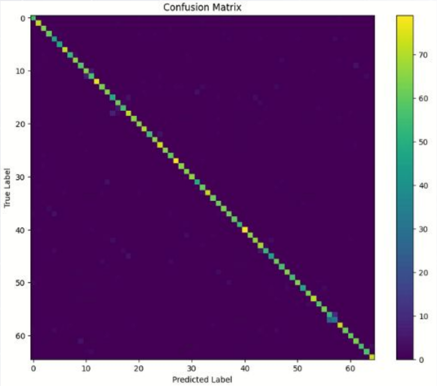
Conclusion:

The model is performing very well with less computational power.

3) Running with Original Vision Transformer Base Patch 16

Not like CNNs that apply filters to capture local features, Vision Transformers treat images as sequences of patches and use self-attention mechanisms to weigh the importance of different patches, which can capture more global context without being biased.

- Confusion Matrix:



Results:

- Training Accuracy= 0.9948
- Validation Accuracy= 0.9512
- Testing Accuracy= 0.9612
- Weighted Precision= 0.9515
- Unweighted Precision = 0.9506
- Weighted Recall = 0.9503
- Unweighted Recall=0.9494
- F-1 Score=0.9492

Conclusion:

The model is performing well but with less accuracy.

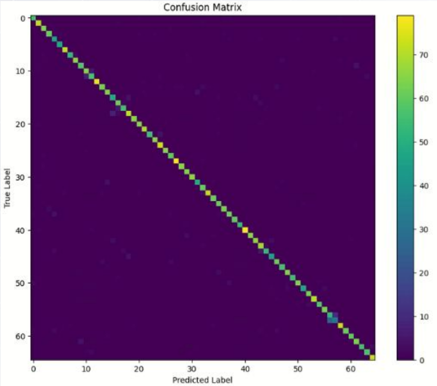
4) Running with Swin Transformer Base Patch 4, Window 7

Swin Transformer proposes hierarchical transformers where the representation is computed with shifted windows, leading to better efficiency. It merges the benefits of both CNNs and Transformers by maintaining a local window-based self-attention mechanism.

Results:

- Training Accuracy= 0.9959
- Validation Accuracy= 0.9754
- Testing Accuracy= 0.9789
- Weighted Precision= 0.9770
- Unweighted Precision = 0.9767
- Weighted Recall = 0.9756
- Unweighted Recall=0.9767
- F-1 Score=0.9760

- Confusion Matrix:



Conclusion:
The model is performing very well and predicted all the labels; changing the arch helped in that.

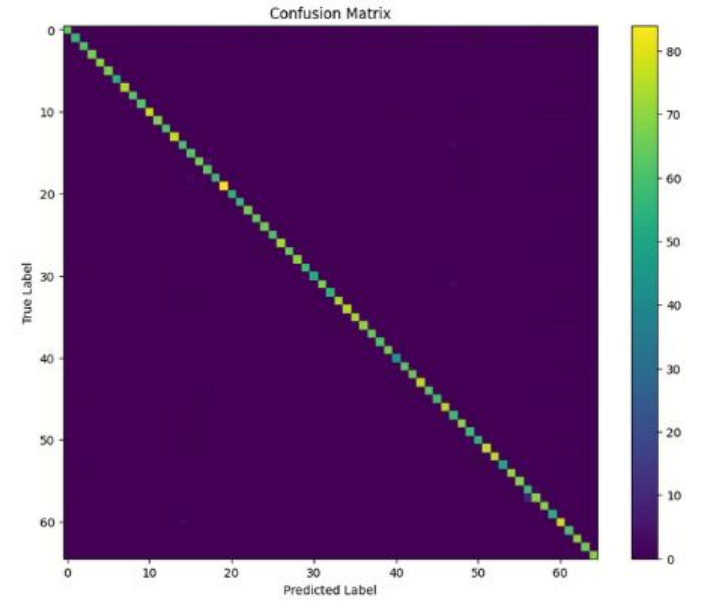
5) Running with ensemble classifiers

In this update, we tried running with an ensemble classifier that uses 3 classifiers: (The original model + regularization + JNPL), (Swin Transformer + regularization), (original model + regularization + excitation blocks). Each classifier is trained on a different distribution of the training set.

Results:

- Testing Accuracy= 0.9924
- Weighted Precision= 0.9928
- Unweighted Precision = 0.9924
- Weighted Recall = 0.9224
- Unweighted Recall=0.9929
- F-1 Score=0.9925

- Confusion Matrix:

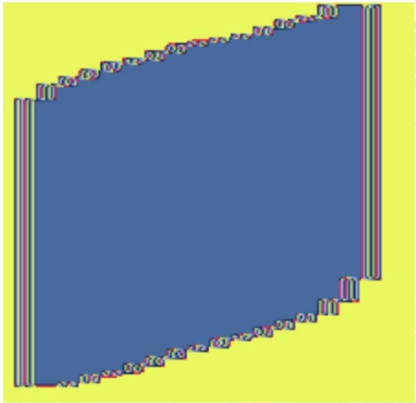


6) Trying to make the model predicts words using YOLO

In this trial, we attempted to use open-source Python library character_segmentation to break paragraphs and words into characters for the character predictor model.

Challenges faced:

1. Spaces between characters varied depending on handwriting, which made segmentation difficult
2. Any ink on the image caused inaccuracies for letters that have points on them (very common in arabic)
3. Resolution of the image made some letters seem different (more wide for instance)



Segmentation of the letter 'alef' in a bad resolution image

Results:

Simply using OpenCV with YOLO pretrained is not enough, and due to the intricacies of Arabic, another model would be required to segment images for character-level prediction

06

Final Optimal Model & Results

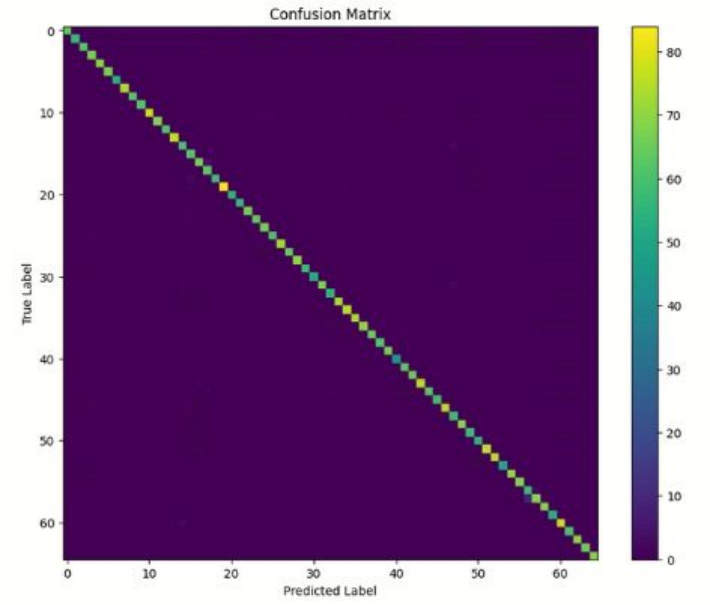
Final Optimal Model & Results:

After finishing all of our updates, we found that the ensemble classifier of the original model + regularization + JNPL, Swin Transformer and excitation blocks has the highest accuracy with a good balance between precision, recall, and accuracy. So, we selected it as our final model.

Results:

- Testing Accuracy= 0.9931
- Weighted Precision= 0.9933
- Unweighted Precision = 0.9932
- Weighted Recall = 0.9931
- Unweighted Recall=0.9932
- F-1 Score=0.9931

- Confusion Matrix:



07

Demo + Poster + Website

08

Conclusion & Future Work

Conclusions, Future work, and lessons learned

Conclusion:

1. Deep Learning is mainly about trial and error.
2. Detecting a whole “Arabic” word or paragraph is very challenging as Arabic is RTL.

Future Work:

1. Trying the same problem approach but on Arabic Paragraphs.
2. Try to use RNNs.

Lessons Learned:

1. Attention mechanisms.
2. Noisy Label Handling mechanisms.
3. Vision Transformers.
4. Transfer Learning
5. YOLO.
6. Always make checkpoints.

09

Members' contributions.

Members' contributions.

Jaheen

- Running with squeeze and excitation blocks.
- Running with Swin Transformer Base Patch 4, Window 7.
- Running with ensemble classifiers.
- Trying to make the model predicts words using YOLO.

Aedan

- Running with grid wise attention.
- Running with Original Vision Transformer Base Patch 16.

Thanks!