Arabic Language Classification

By:
Ahmed Jaheen
Aedan Ounsamone

TABLE OF CONTENTS

01

Proposal Summary

02

Progress on Deploying the baseline model.

03

Progress on the proposed Solutions.

04

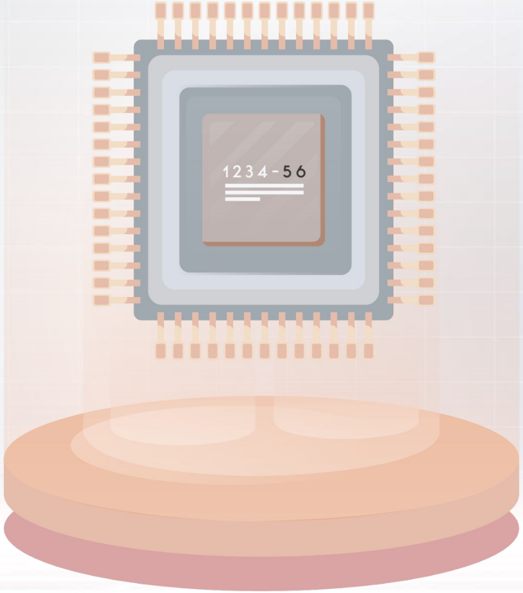
Initial Results and findings.

05

Updates to Milestone II and Final

06

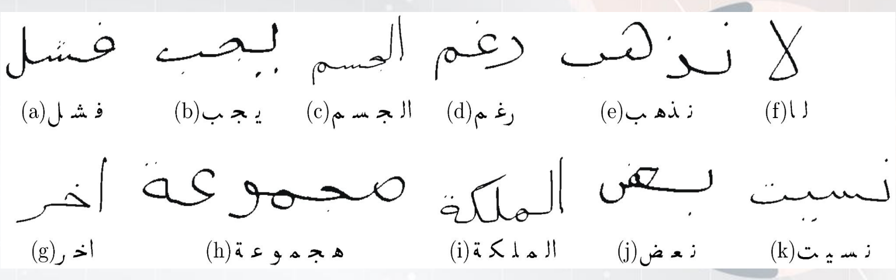
Members' contributions.



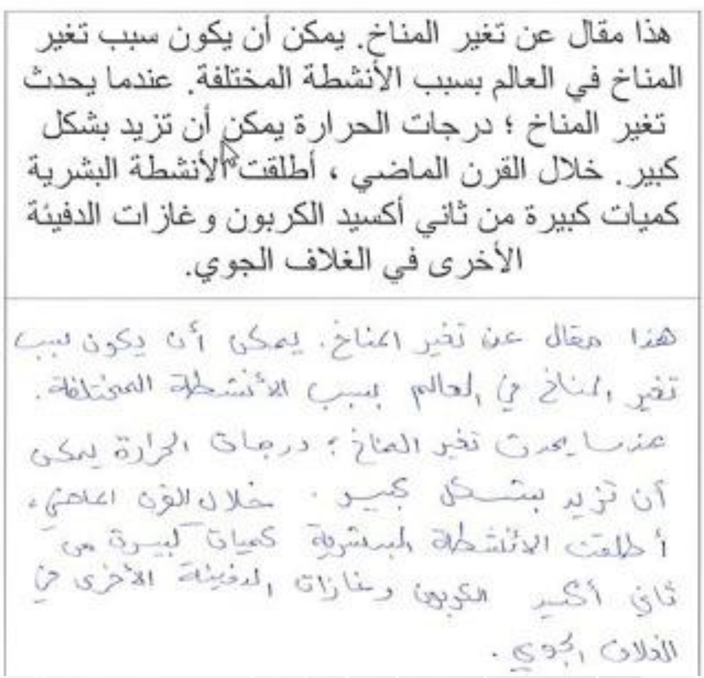
01

Proposal Summary

Problem statement



The advancement in text recognition on scanned images has opened up numerous possibilities, from searching texts in extensive documents to automating postal sorting. Arabic handwriting recognition, given its unique challenges, has garnered attention later than other scripts, leading to diverse methodologies tailored for different image types. On the other hand, recognizing Arabic characters and digits is pivotal due to the script's cursive nature and shape variations. Researchers have developed models to tackle this problem, and our project aims to explore these models, evaluate datasets, and propose performance enhancements.

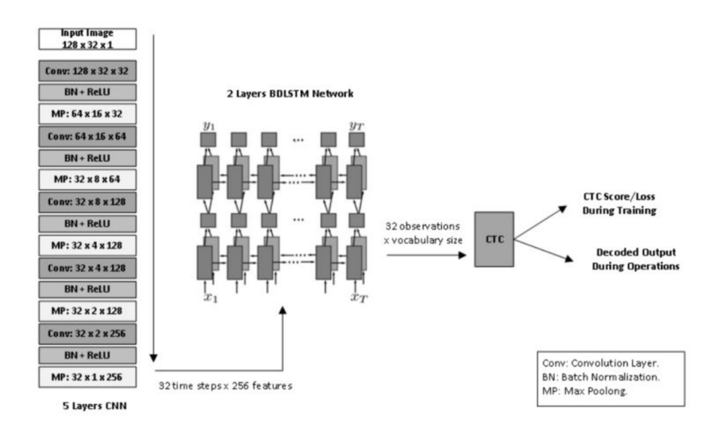


هذا مقال عن تغير المناخ. يمكن أن يكون سبب تغير
المناخ في العالم بسبب الأنشطة المختلفة. عندما يحدث
تغير المناخ ؛ درجات الحرارة يمكن أن تزيد بشكل
كبير. خلال القرن الماضي ، أطلقت الأنشطة البشرية
كميات كبيرة من ثاني أكسيد الكربون وغازات الدفيئة
الأخرى في الغلاف الجوي.

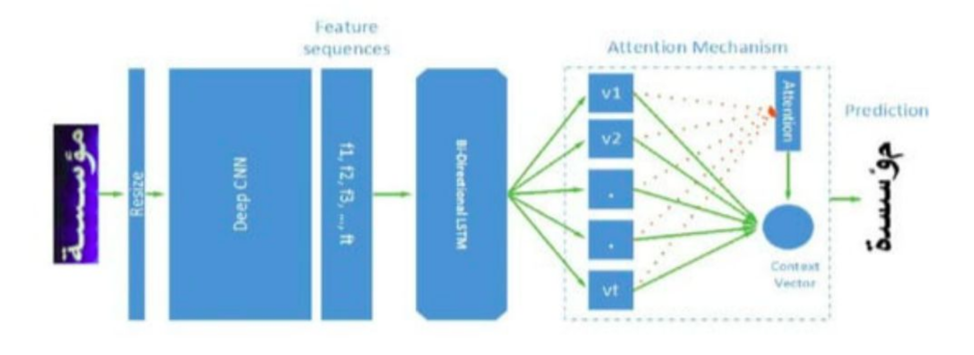
هذا مقال عن تغير المناخ. يمكن أن يكون سبب
تغير المناخ في العالم بسبب الأنشطة المختلفة.
عندما يحدث تغير المناخ ؛ درجات الحرارة يمكن
أن تزيد بشكل كبير . خلال القرن الماضي ،
أطلقت الأنشطة البشرية كميات كبيرة من
ثاني أكسيد الكربون وغازات الدفيئة الأخرى في
الغلاف الجوي .

Related Work

We surveyed 3 models: Hybrid CNN with Bi-LSTM (Bi-Directional Long Short Term Memory), CNN & RNN with Attention Mechanism, and other pre-trained models.

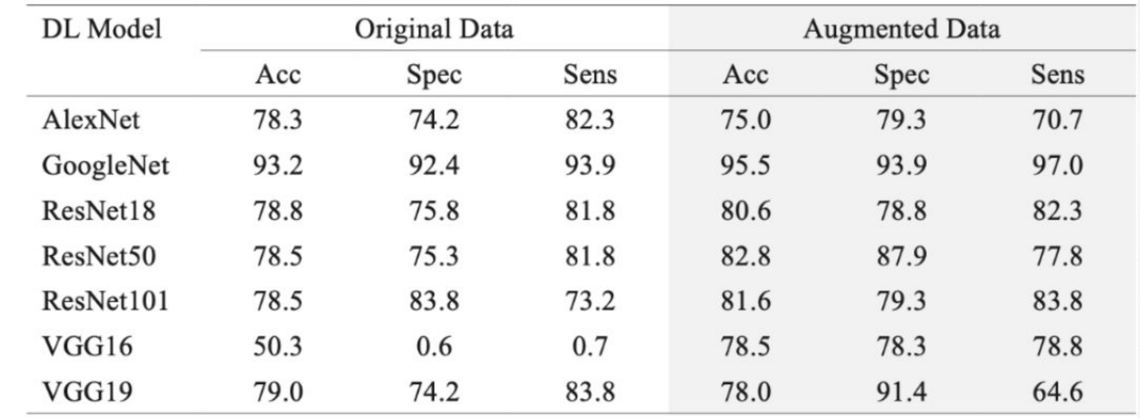


CNN with Bi-LSTM



CNN & RNN with Attention

Baseline Model

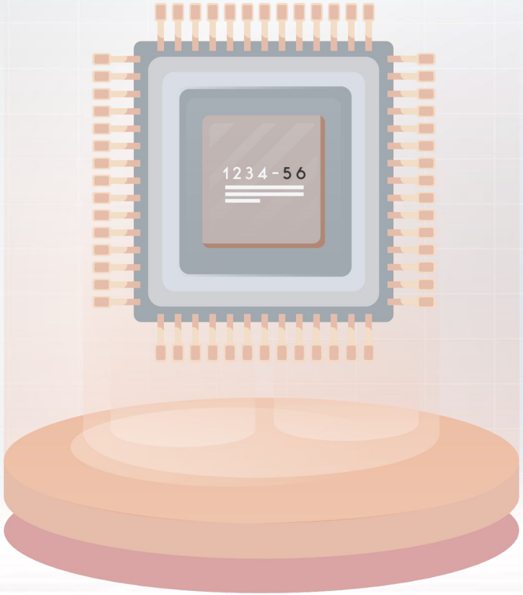


DL Model	Original Data			Augmented Data		
	Acc	Spec	Sens	Acc	Spec	Sens
AlexNet	78.3	74.2	82.3	75.0	79.3	70.7
GoogleNet	93.2	92.4	93.9	95.5	93.9	97.0
ResNet18	78.8	75.8	81.8	80.6	78.8	82.3
ResNet50	78.5	75.3	81.8	82.8	87.9	77.8
ResNet101	78.5	83.8	73.2	81.6	79.3	83.8
VGG16	50.3	0.6	0.7	78.5	78.3	78.8
VGG19	79.0	74.2	83.8	78.0	91.4	64.6

The paper applied transfer learning on seven different networks: AlexNet, GoogleNet, ResNet18, ResNet50, ResNet101, VGG16, and VGG19 on two datasets containing images of Arabic hand- writing. Among the models, GoogleNet achieved the highest accuracy, recording 93.2% with the normal dataset and 95.5% with the augmented dataset, in correctly identifying native Arabic handwriting.

Key changes to Baseline Model

- Use Regularization Techniques and a Learning Rate Scheduler.
- Test different pre-trained models & evaluate accuracies on data
- Fine-tuning best pre-trained model.
- Use vision transformers model
- Use Adaboost ensemble classifiers.
- Implement data augmentation techniques.



02

Progress on Deploying the Baseline Model.

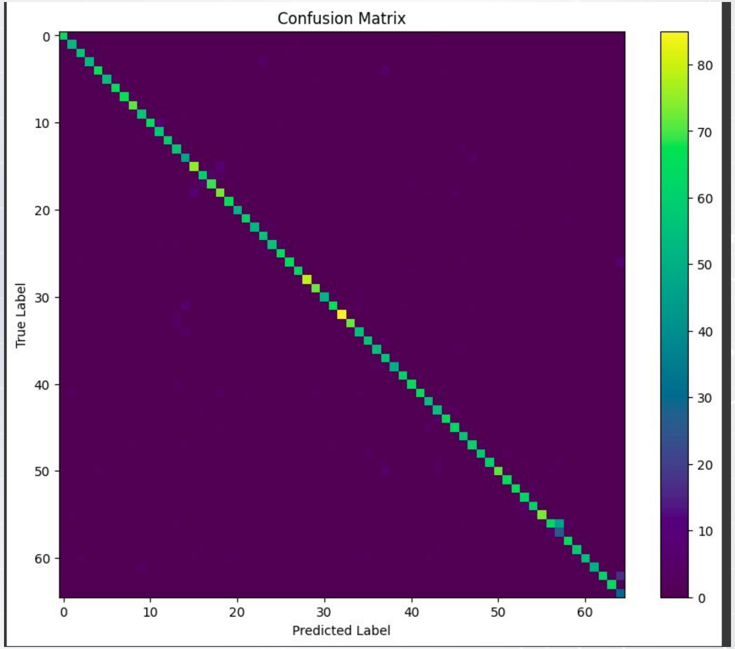
Deploying the Baseline Model.

1. We managed to deploy the baseline model successfully by training it on the AHAWP dataset.
2. We have gone through a hard time to train the model on this dataset as we were not able to find this as a whole but rather we only found the updates labels and we had to match each image with its new label.
3. We used the official TorchVision implementation of GoogleNet.

Baseline Model Results on New Dataset:

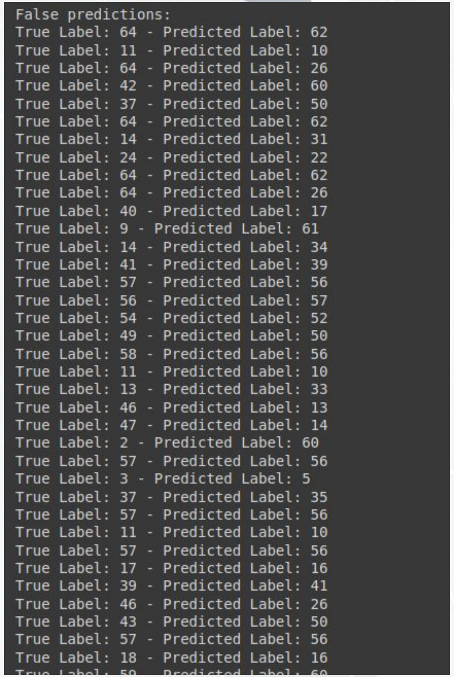
- Training Accuracy= 0.9367
- Validation Accuracy= 0.9218
- Testing Accuracy= 0.9256
- Weighted Precision= 0.9221
- Unweighted Precision = 0.9292
- Weighted Recall = 0.9218
- Unweighted Recall=0.9238
- F-1 Score=0.9200

- Confusion Matrix:



Findings after deploying the baseline Model:

1. The accuracy we got was about 3 percent lower than the reported accuracy in the paper. A possible reason is that we used another dataset.
2. The Haa', Eiin, and Waaw were the most misclassified classes. A possible reason might be that these classes are similar in writing (same shape) - have different shape in Arabic Language based on their order→ A possible solution would be data augmentation.



```
False predictions:
True Label: 64 - Predicted Label: 62
True Label: 11 - Predicted Label: 10
True Label: 64 - Predicted Label: 26
True Label: 42 - Predicted Label: 60
True Label: 37 - Predicted Label: 50
True Label: 64 - Predicted Label: 62
True Label: 14 - Predicted Label: 31
True Label: 24 - Predicted Label: 22
True Label: 64 - Predicted Label: 62
True Label: 64 - Predicted Label: 26
True Label: 40 - Predicted Label: 17
True Label: 9 - Predicted Label: 61
True Label: 14 - Predicted Label: 34
True Label: 41 - Predicted Label: 39
True Label: 57 - Predicted Label: 56
True Label: 56 - Predicted Label: 57
True Label: 54 - Predicted Label: 52
True Label: 49 - Predicted Label: 50
True Label: 58 - Predicted Label: 56
True Label: 11 - Predicted Label: 10
True Label: 13 - Predicted Label: 33
True Label: 46 - Predicted Label: 13
True Label: 47 - Predicted Label: 14
True Label: 2 - Predicted Label: 60
True Label: 57 - Predicted Label: 56
True Label: 3 - Predicted Label: 5
True Label: 37 - Predicted Label: 35
True Label: 57 - Predicted Label: 56
True Label: 11 - Predicted Label: 10
True Label: 57 - Predicted Label: 56
True Label: 17 - Predicted Label: 16
True Label: 39 - Predicted Label: 41
True Label: 46 - Predicted Label: 26
True Label: 43 - Predicted Label: 50
True Label: 57 - Predicted Label: 56
True Label: 18 - Predicted Label: 16
True Label: 59 - Predicted Label: 60
```

03

Progress on the proposed solution

&

04

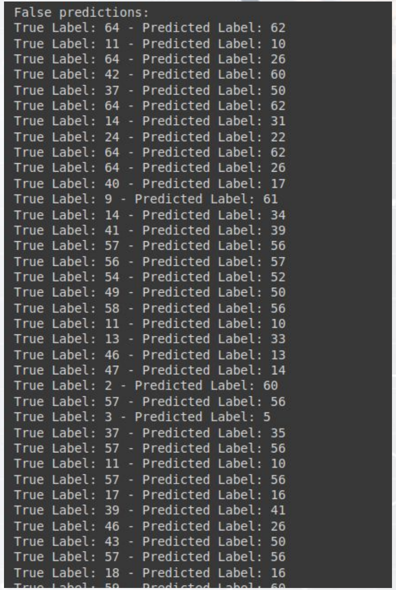
Initial results and findings

Additional Work on the Baseline Model:

1. Applying data augmentation on the AHAWP dataset and retraining the model.
2. Adding L2, Dropout regularization to prevent overfitting and retraining the model on the AHAWP dataset.
3. Adding a learning rate scheduler to dynamically adjust the learning rate during training

First, Applying data augmentation on AHAWP dataset.

1. After reviewing the baseline model results, we found that the letters 'ـعـ' ,'ع', and 'ـع' were often misclassified.
2. Upon inspecting the training data, we discovered that 'ـع' was the least common, followed by 'ـعـ' and then 'ع'.
3. To address this, we decided to boost their representation using data augmentation.
4. We added 100 images for the 'ع' and 'ـعـ' classes by increasing the line thickness and applied rotation to all 'ـع' images.



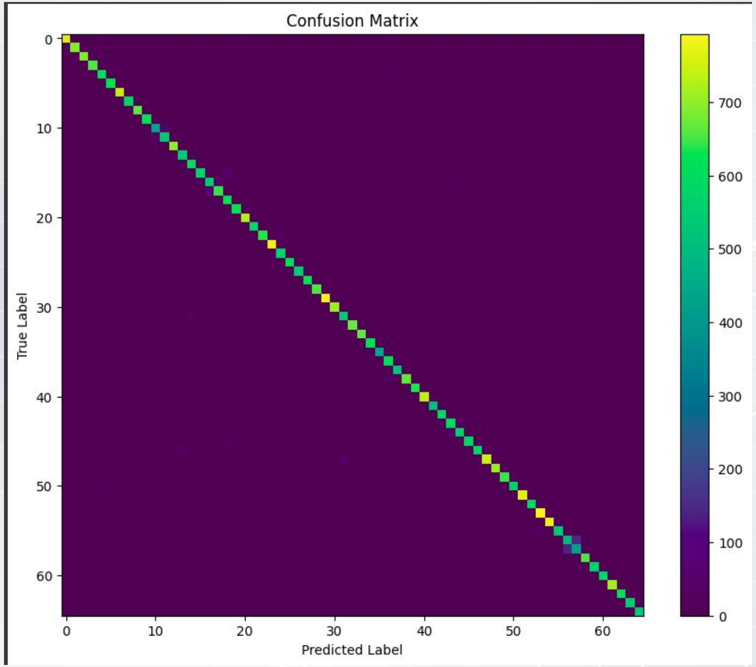
```
False predictions:
True Label: 64 - Predicted Label: 62
True Label: 11 - Predicted Label: 10
True Label: 64 - Predicted Label: 26
True Label: 42 - Predicted Label: 60
True Label: 37 - Predicted Label: 50
True Label: 64 - Predicted Label: 62
True Label: 14 - Predicted Label: 31
True Label: 24 - Predicted Label: 22
True Label: 64 - Predicted Label: 62
True Label: 64 - Predicted Label: 26
True Label: 40 - Predicted Label: 17
True Label: 9 - Predicted Label: 61
True Label: 14 - Predicted Label: 34
True Label: 41 - Predicted Label: 39
True Label: 57 - Predicted Label: 56
True Label: 56 - Predicted Label: 57
True Label: 54 - Predicted Label: 52
True Label: 49 - Predicted Label: 50
True Label: 58 - Predicted Label: 56
True Label: 11 - Predicted Label: 10
True Label: 13 - Predicted Label: 33
True Label: 46 - Predicted Label: 13
True Label: 47 - Predicted Label: 14
True Label: 2 - Predicted Label: 60
True Label: 57 - Predicted Label: 56
True Label: 3 - Predicted Label: 5
True Label: 37 - Predicted Label: 35
True Label: 57 - Predicted Label: 56
True Label: 11 - Predicted Label: 10
True Label: 57 - Predicted Label: 56
True Label: 17 - Predicted Label: 16
True Label: 39 - Predicted Label: 41
True Label: 46 - Predicted Label: 26
True Label: 43 - Predicted Label: 50
True Label: 57 - Predicted Label: 56
True Label: 18 - Predicted Label: 16
True Label: 59 - Predicted Label: 60
```

Distribution before augmentation.

Results After applying data augmentation:

- Training Accuracy= 0.9764
- Validation Accuracy= 0.9308
- Testing Accuracy= 0.9313
- Weighted Precision= 0.9301
- Unweighted Precision = 0.9335
- Weighted Recall = 0.9305
- Unweighted Recall=0.9311
- F-1 Score=0.9299

- Confusion Matrix:



Findings after data augmentation.

1. Data augmentation greatly enhanced the overall performance of the model. However, the differences between the training accuracy and the validation/testing accuracy, so we need to increase the generalization of the model.
2. The performance on the Ein and the Waaw classes were better. However, the model still was not able to classify the Haa' class → Try to use different augmentation techniques or to do interpretability study.

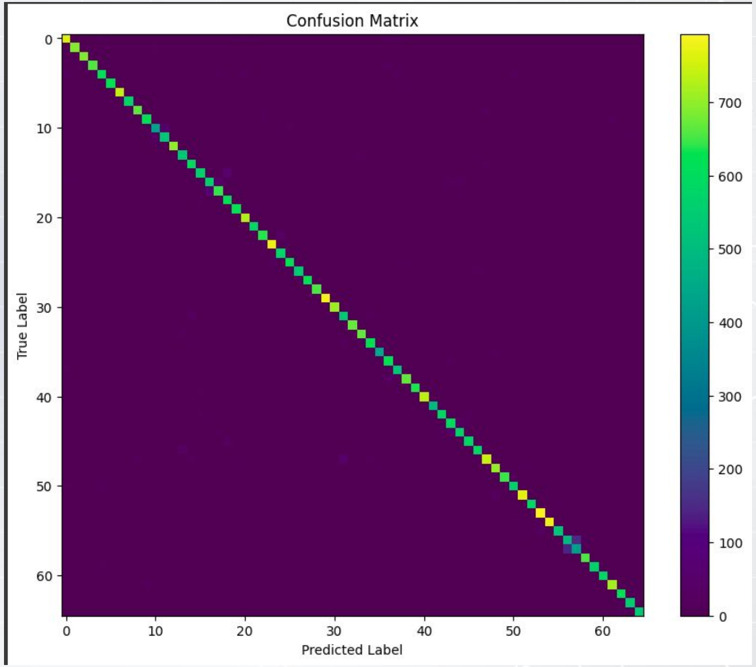
Second, Adding regularization and learning rate scheduler to the model and retrain it on the AHAWP dataset.

1. To address the issue of generalization, we opted for regularization and added learning rate scheduler.
2. We added an L2 regularizer with gamma = 0.1 and a step size 7.

Results on AHAWP dataset:

- Training Accuracy= 0.9663
- Validation Accuracy= 0.9538
- Testing Accuracy= 0.9481
- Weighted Precision= 0.9436
- Unweighted Precision = 0.9423
- Weighted Recall = 0.9431
- Unweighted Recall=0.9422
- F-1 Score=0.9431

- Confusion Matrix:



Findings after adding regularization.

1. Regularization helped in reducing the overfitting.
2. The model is still unable to classify the Haa’ different labels.

Research – handling noisy labels

1. Upon researching about different handling noisy labels techniques. We found that JNPL is one of the methods designed to improve model training on datasets with noisy labels.
2. It’s specifically designed to identify and handle noisy labels through a joint learning process that involves filtering noisy samples (negative learning) and improving the model's performance with the cleaned data (positive learning).
3. It uses a unified single-stage pipeline, and incorporating pseudo-labeling to make better use of the noisy data.

Link to the paper:
https://openaccess.thecvf.com/content/CVPR2021/papers/Kim_Joint_Negative_and_Positive_Learning_for_Noisy_Labels_CVPR_2021_paper.pdf

05

Next Steps and Project Timeline

Potential Updates for Milestone II & Final

1. We will try different data augmentation techniques on the dataset to achieve better performances.
2. Creating a proposed CNN architecture and training it on our data
3. Fine-tune model hyperparameters
4. Test different pre-trained models & evaluate accuracies on data.
5. Try to apply the proposed noisy labels handling mechanism to see its effect on the model.
6. Use Vision transformers for image classification
7. Deploying the model on a website with good User Experience

06

Members' contributions.

Members' contributions.

Jaheen

- Worked on deploying the baseline model.
- Ran the model on AHAWP dataset.
- Applied data augmentation on the dataset.
- Added regularization and dynamic learning rate scheduler.

Aedan

- Researched the noisy labels detection and handling mechanism.
- Worked on deploying the baseline model.

Thanks!