ECE/CS/ME 539 Introduction to Artificial Neural Networks

Project Final Report

# Multi-Language Alphabetical Classifier

Team # : 11

Team Members:

Caleb Federman, Department of Electrical and Computer Engineering Undergraduate

Elias Cassis, Department of Electrical and Computer Engineering Undergraduate

Connor Braun, Department of Computer Sciences Undergraduate

Tate Waugh, Department of Computer Sciences Undergraduate

[Github](https://github.com/calebfederman/handwritten-alphabet-language-classifier)

ABSTRACT

This study introduces a novel machine learning framework capable of recognizing and classifying characters from multiple alphabets, specifically English, Russian, and Arabic, and subsequently identifying their associated language. Utilizing Convolutional Neural Networks (CNNs), our approach innovatively combines individual and grouped character classification methods. The framework's backbone consists of multi-layered CNNs, applying multiple 2D convolutions with increasing channels and periodic ReLU activations. This structure optimizes feature selection, while pooling and dropout strategies effectively manage model complexity and training time. Our models underwent rigorous testing, comparing individual and grouped CNN classification performances, both with and without transformation (slight random affine) adjustments. Adjustments in learning rate and weight initialization parameters were crucial for enhancing model learning and accuracy. This approach sets a new standard in multi-language character recognition, extending the boundaries of traditional language classification methods.

Problem Statement

The primary challenge addressed in this project is the accurate classification and recognition of characters from diverse alphabets (English, Russian, and Arabic) using machine learning techniques. Existing solutions often focus on single-language classification, limiting their scope and applicability in multilingual contexts. Our objective was to create a robust framework that not only recognizes individual characters but also determines the corresponding language. This required overcoming challenges in model design, such as selecting the appropriate neural network architecture and tuning hyper-parameters to handle the complexities of multiple languages. The project aims to surpass the baseline accuracy metric of 86% achieved in prior single-language classification studies. It also seeks to explore the feasibility of combined CNN models for multi-language classification, an area not extensively explored in current literature. Our work is positioned at the intersection of character recognition and language identification, pushing the frontiers of multilingual character classification.

Related Work

CNN for handwritten letters classification1:

This project prioritized single language classification, and was trained and tested on a russian dataset. The CNN structure in this model greatly inspired our own.

This project initially classified a single language using a CNN model, achieving ~86% accuracy. They then add additional steps to automatically detect and remove the background, thus isolating the character and making classification easier. With these steps, they achieved ~99% accuracy.

Due to the similarities between their initial CNN and goal there, our baseline accuracy metric is around 86% accuracy.

**More Complex Models:**

Cross lingual handwritten character recognition using long short term memory network with aid of elephant herding optimization algorithm2:

Utilizes detailed preprocessing steps with multiple algorithms before applying an LSTM to classify letters from English, Kannada, and Arabic. By doing this, they achieved accuracies of 99.66%, 96.67%, and 99.93% respectively. This project shows how utilizing more complex algorithms and neural networks can result in high transcription accuracy.

Cross-Lingual Text Image Recognition via Multi-Hierarchy Cross-Modal Mimic3:

Explores how letter/word classification can be done in a single process, as opposed to the more traditional method of separating classification and translation into two separate discrete processes. This project shows more extrapolation from the goal of classification and details how classification can both be of assistance to other goals as well as how it may be able to assist them (hence the joined classification/translation) process.

Our Baseline accuracy metric: 86%

Data

The source of our 3 datasets was Kaggle and the original datasets can be found at these links.

<https://www.kaggle.com/datasets/mohneesh7/english-alphabets>4 - English Fig1

# of Samples: 6831

<https://www.kaggle.com/datasets/tatianasnwrt/russian-handwritten-letters>5-Russian Fig2

# of Samples: 14190

<https://www.kaggle.com/datasets/insafbenlamari/arabic-letters>6 - Arabic Fig3

# of Samples: 8418

The format of the samples differed slightly. The English letters were white with a black background, the Russian letters were drawn directly on paper, and the Arabic letters were black with a white background. We partitioned all data into 70/15/15/ train/test/val splits before sending them through the CNN. Our pre-processing consisted of grayscaling, normalization of values and resizing each image to 28 \* 28, which ensured consistency across the dataset. We also applied one hot encoding. Once we were able to do this for our data, our resulting datasets after splitting for all languages were:

English:

* Characters: (6831, 28, 28, 1)
* Labels: (6831, 26)

Arabic:

* Characters: (8718, 28, 28, 1)
* Labels: (8718, 28)

Russian:

* Characters: (14190, 28, 28, 1)
* Labels: (14190, 33)

Tasks Performed

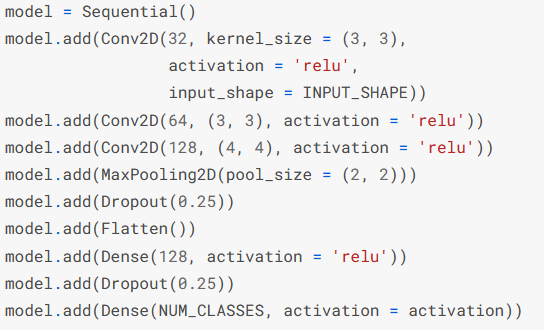
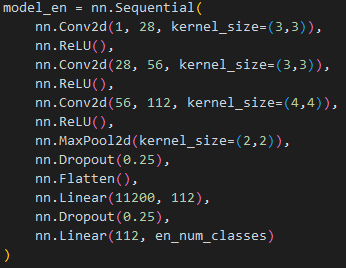
We developed four separate models for this project. All models were CNNs, one corresponding to each or our chosen languages (English, Arabic, Russian) as well as a combined CNN utilizing the combined dataset or all three (3) languages.

We adapted the baseline model from BryanB’s CNN language classifier1 (image difference shown in Fig. 1a and Fig. 1b) to fit the use case for our four different models.

Thus our models consist of multiple 2d convolutions on the input image with an increasing amount of channels with periodic ReLU activations in order to best select features. The model then utilizes a pooling layer in order to keep the model focused and reduce unnecessary complexity and training time. Dropout layers are used to keep the model from overfitting, and linear layers are used to move the model to predicting the required number of classes (English, Arabic, Russian, or combined amount of letters).

We measured our models performance based on accuracy, and compared results in a variety of different situations. All tests (with the exception of the grouped confidence based test) were run with and without a slight affine transformation performed to some of the data. This transformation would rotate the images up to 90 degrees, translate them up to 20%, and scale them up to a 20% zoom. Our tests were: Individual classification, where the individual language models classified the test set from their own languages dataset.Grouped CNN classification, where the CNN trained on all the datasets was tested on the test set containing all the datasets. Grouped confidence based classification, where each individual language CNN tried to classify data from a combined dataset of all the languages, with the result being decided by the most confident CNN (calculated using the softmax). Finally, we used the grouped CNN to determine language accuracy as well.

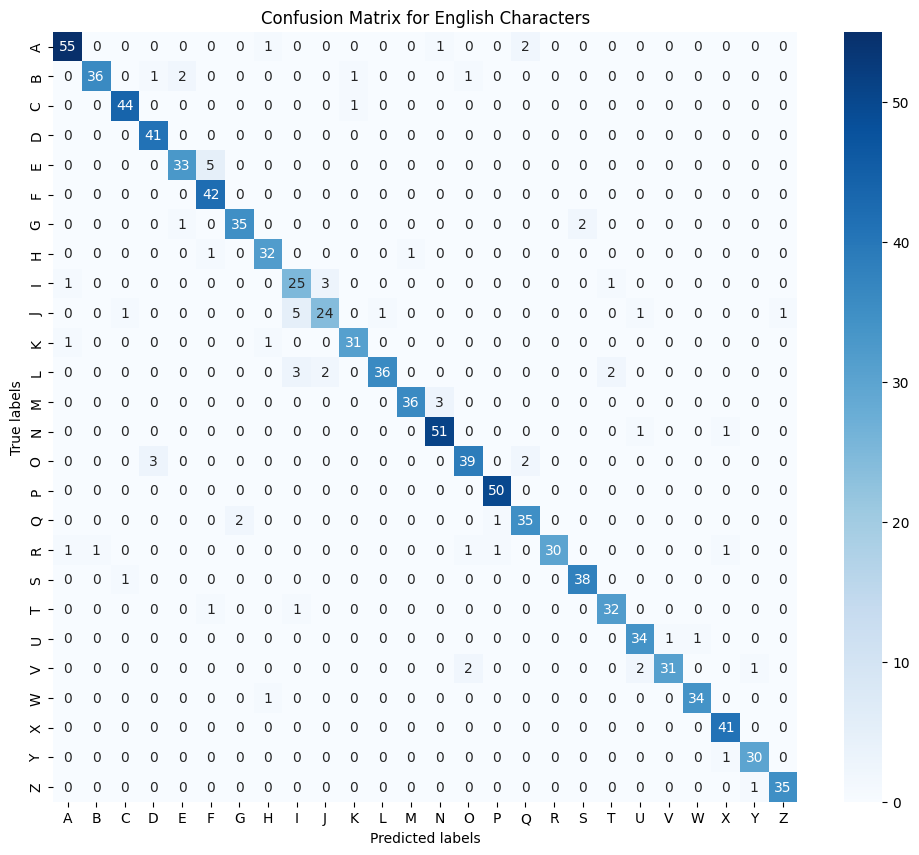
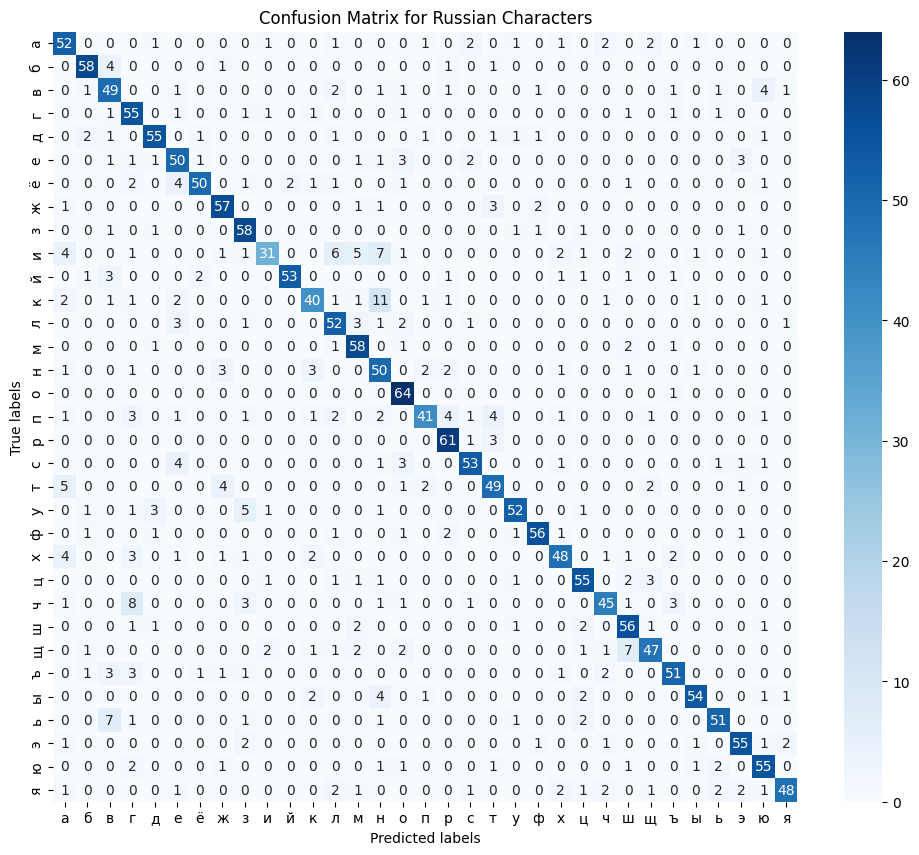
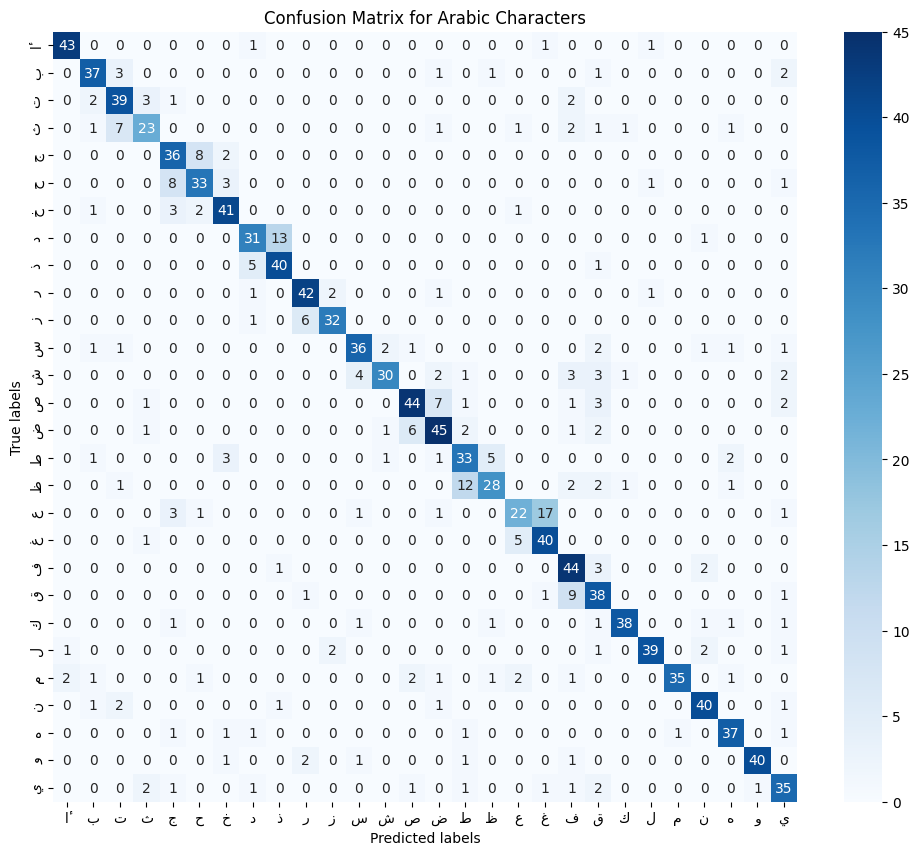
In terms of parameter selection, we adjusted our learning rate to be slightly lower in order for a cleaner, more reliable, convergence. Our initialization of weights also required some changes as initially our standard deviation was too small, and the model did not train well, and stagnated on test set classification.

Figure 4a: BryanB’s CNN Figure 4b: Our CNN

Results

Model Performance:

* Separate CNN Models: Each model was tailored to its specific language, with the English, Arabic, and Russian models achieving accuracies of 85.29%, 67.39%, and 70.93% respectively. These results were derived under the testing conditions without transformations. You can see the confusion matrices for each of these models in Figure 5a below.

Figure 5a

* Combined CNN Model: The combined mode presented an accuracy of 77.73%. This was slightly lower than the English model, yet higher than both the Arabic and Russian models. Figure 5b below demonstrates example images from our test set along with their true and predicted values.

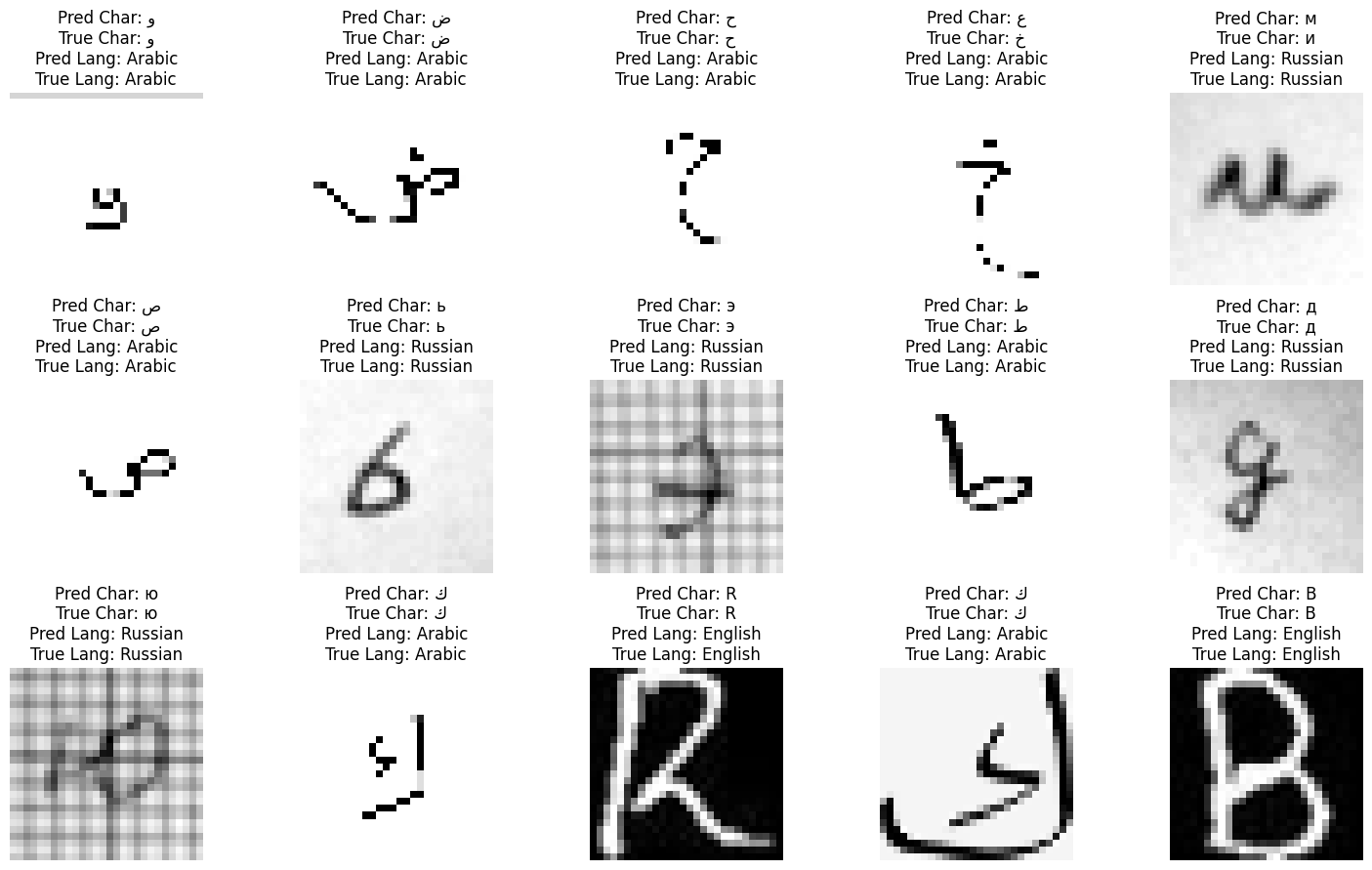


Figure 5b

* Grouped Confidence-Based Classification: The confidence-based classification took the three individual language models and selected the highest confidence between them for the final character classification. This method resulted in the worst performance of any trial with just 44.87%

Transformation Impact:

* The inclusion of transformations (slight random affine adjustments) impacted the model performances. The separate CNN models showed a marginal decrease in accuracy, whereas the combined model exhibited an improvement up to 80.62%.

Language Recognition Accuracy:

* A notable achievement was the high accuracy in language recognition. Despite the character recognition models not meeting our goal, the model accurately identified the language of the characters with a remarkable accuracy of 99.93%. This is demonstrated in the confusion matrix in Figure 5c below.

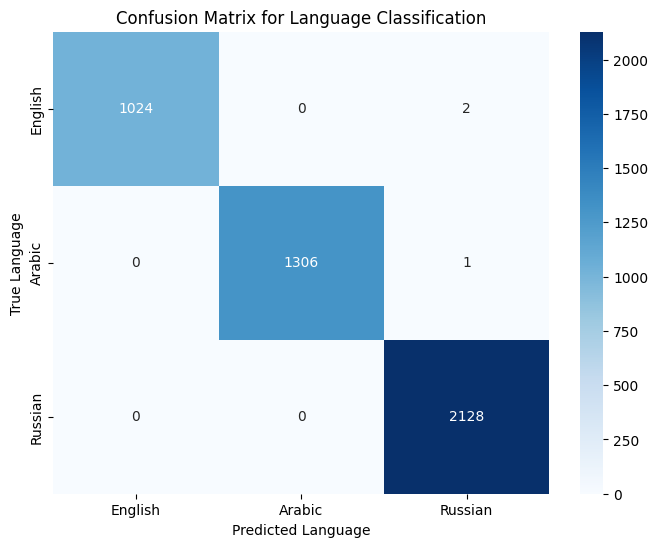


Figure 5c

Performance Comparison:

* When compared to our target baseline accuracy of 86% the English model performs to the standard, however, our Arabic and Russian models perform well below our target. The combined model performed better than the Arabic and Russian models, yet still below 86%. We discuss suspected reasoning and potential ways to improve in the next section.

Discussion

Based on the results yielded from this experiment, there are various takeaways that led to the conclusions brought about through our outputs, graphs, and confusion matrices. First we can reflect on the accuracy of English, Arabic, and Russian yielding 85.29%, 67.39%, and 70.93% respectively. First, for the English dataset, it was the only one to utilize a black background with white text font for each handwritten letter. We think that this led to an increase in training and test accuracy as even with a smaller dataset when compared relatively to the other two, it trained the best out of the three. However, not only was this an advantage that the English dataset carried, but the Russian and Arabic datasets also had their disadvantages. For Arabic, the dataset was slightly bigger, yet yielded the worst accuracy of the three. We believe this to be because oftentimes the handwritten letters had pixelated segments missing from what should’ve been a continuous writing stroke. Because of this, characters became slightly segmented as illustrated in Figure 5b. Finally with regards to the Russian dataset, despite it having almost 1.7 times as many samples as the second highest dataset, it performed with only ~71% accuracy. This could be a number of reasons, but we mainly deduce it to the perceived “graph paper” background that each Russian letter has which dramatically increases the busyness of the picture. In fact, in Kaggle user BryanB’s CNN language classifier1 he completed a project where he trained the dataset on the characters with the graph paper background, as well as one where he isolated & removed the background and found a testing accuracy increase of up to 99%! Therefore, despite that obstacle being out of the scope of our current project, that action could be a worthy investment for our future training on our classification project.

Next, as we cumulatively combined our CNNs of each individual language’s character classification we wanted to create a feature to accurately guess the language of any character of a given language. We implemented this through a confidence-based classification that was generated by taking the arbitrary letter (of any of the three languages) and declaring a confidence value from the letter to a given language. This resulted in the worst of our trials yielding just a ~45% accuracy. However, we believe this to be the case because of the intrinsic shape and style of many of the characters in the various languages. For example, in the Russian language, there are a few characters that resemble the ‘b’ shape of English, therefore when our confidence-based classification must make a decision, there is a “split-vote” between the various ‘b’ shaped letters of Russian, versus a well-defined letter ‘b’ in the English alphabet. Thus, even if the true value of the shape was a variant of this ‘b’ shaped letter in Russian, there was a much higher confidence associated in English, leading to a strong chance of misclassification based on skewed confidence guesses.

One issue we ran into while developing and training the models was a difficulty in learning and plateaued classification accuracy of the test set. This ended up being a result of an initial standard deviation for the model that was too small. By increasing this, we were able to get better performance in both training and testing classification. Further changing these initial parameters could be done to try and achieve faster and more accurate learning. We also noticed that between when transformations were and weren’t applied had little effect on model performance. In fact, the transformations slightly decreased the accuracy of predictions. We think this is due to our datasets being somewhat standardized. However, we believe this is somewhere additional testing and experimentation could be done. Using the models trained on transformed and untransformed datasets with different datasets (or manually inputted images) in order to see how their relative performance may more accurately show which model is more generalizable. Equally, testing with different types of transformation may prove helpful for better generalization with a nonstandard input image, as well as potentially increasing accuracy on the Russian dataset as the increased generalization could help it ignore the graph paper background on some images, and thus more accurately classify the letter.

While our project is more focused on individual letter and language classification, this can be used in a variety of different circumstances, such as translation and transcription, or even included as a subset of a larger goal, such as grammar checking or predicting missing segments or written passages. To this purpose more advanced models have been developed and can be seen in our related works field2,3.

References

1 Bryan, B. "CNN for Handwritten Letters Classification." Kaggle, 12/14/2023.<https://www.kaggle.com/code/bryanb/cnn-for-handwritten-letters-classification>

2 Nirmala S Guptha, V. Balamurugan, Geetha Megharaj, Khalid Nazim Abdul Sattar, J. Dhiviya Rose,

Cross lingual handwritten character recognition using long short term memory network with aid of elephant herding optimization algorithm,

Pattern Recognition Letters,

Volume 159,

2022,

Pages 16-22,

ISSN 0167-8655,

https://doi.org/10.1016/j.patrec.2022.04.038.

(<https://www.sciencedirect.com/science/article/pii/S0167865522001490>)

3 Chen, Zhuo & Yin, Fei & Yang, Qing & Liu, Cheng-Lin. (2022). Cross-Lingual Text Image Recognition via Multi-Hierarchy Cross-Modal Mimic. IEEE Transactions on Multimedia. PP. 1-13. 10.1109/TMM.2022.3183386.

4 Mohneesh\_Sreegirisetty. “English Alphabets.” *Kaggle*, 18 July 2018, [www.kaggle.com/datasets/mohneesh7/english-alphabets](http://www.kaggle.com/datasets/mohneesh7/english-alphabets)

5 Snauwaert, Tatiana. “Russian Handwritten Letters.” *Kaggle*, 7 Apr. 2020, [www.kaggle.com/datasets/tatianasnwrt/russian-handwritten-letters](http://www.kaggle.com/datasets/tatianasnwrt/russian-handwritten-letters)

6 Benlamari, Insaf. “Arabic\_letters.” *Kaggle*, 6 June 2023, [www.kaggle.com/datasets/insafbenlamari/arabic-letters](http://www.kaggle.com/datasets/insafbenlamari/arabic-letters)

7 J. Memon, M. Sami, R. A. Khan and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," in *IEEE Access*, vol. 8, pp. 142642-142668, 2020, doi: 10.1109/ACCESS.2020.3012542.

8 Islam, Noman, et al. “A Survey on Optical Character Recognition System.” Rnal of Information & Communication Technology-JICT, vol. 10, no. 2, Jan. 2016, pp. 4–7. jms.ilmauniversity.edu.pk/index.php/JICT/article/download/641/362.

9 Chaudhuri, A., Mandaviya, K., Badelia, P., Ghosh, S.K. (2017). Optical Character Recognition Systems. In: Optical Character Recognition Systems for Different Languages with Soft Computing. Studies in Fuzziness and Soft Computing, vol 352. Springer, Cham. <https://doi.org/10.1007/978-3-319-50252-6_2>

10 R. Avyodri, S. Lukas and H. Tjahyadi, "Optical Character Recognition (OCR) for Text Recognition and its Post-Processing Method: A Literature Review," *2022 1st International Conference on Technology Innovation and Its Applications (ICTIIA)*, Tangerang, Indonesia, 2022, pp. 1-6, doi: 10.1109/ICTIIA54654.2022.9935961.

11 Dhivya S and Usha Devi G. 2021. Study on Automated Approach to Recognize Characters for Handwritten and Historical Document. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 20, 3, Article 37 (May 2021), 24 pages. https://doi.org/10.1145/3396167