KAZAKH-BRITISH TECHNICAL UNIVERSITY

KBTU

HANDWRITING RECOGNITION

in the Kazakh Language

Alar Akilbekov (Math, Code)

Baktiyar Toksanbay (Code)

2024

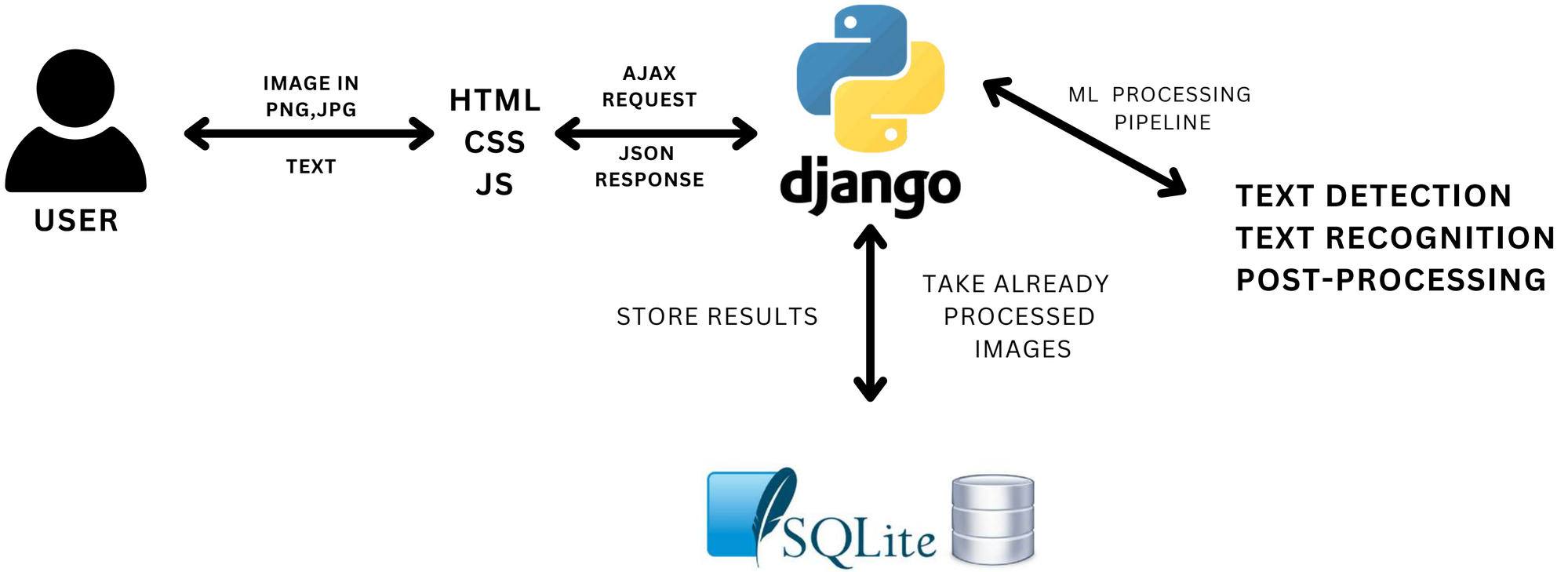
DESCRIPTION

Handwritten Text Recognition (HTR)

Ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices.

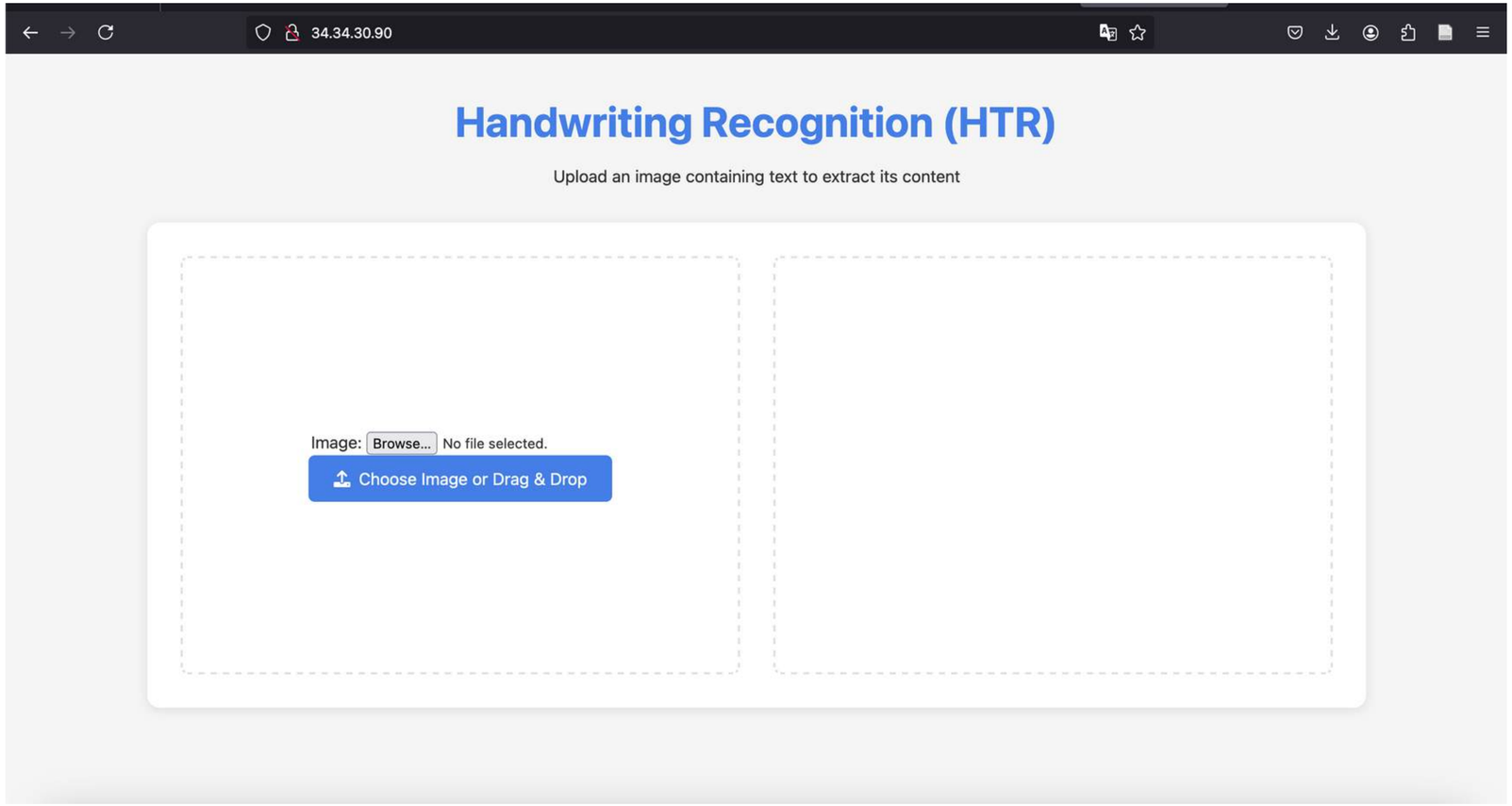
PROJECT STACK

PROJECT ARCHITECTURE



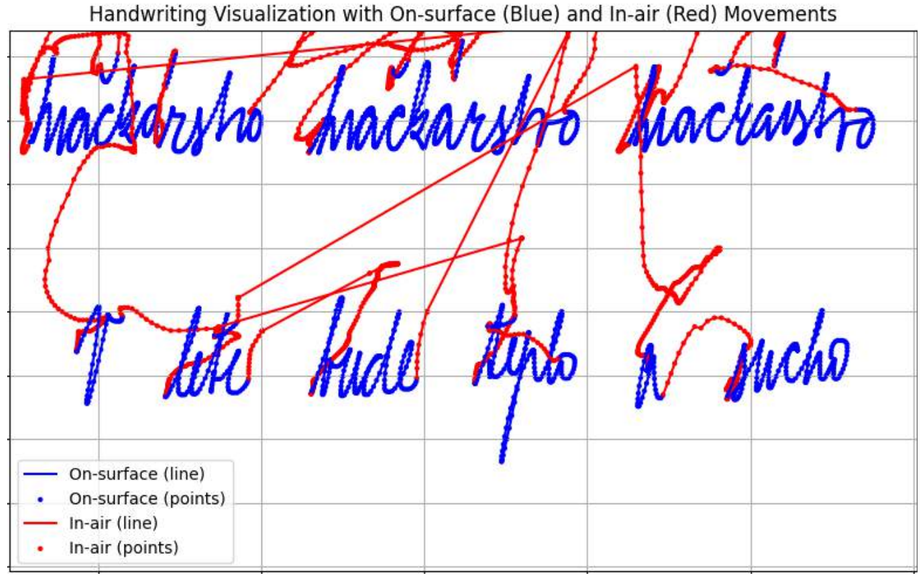
TRY IT OUT

34.34.30.90



On-line:

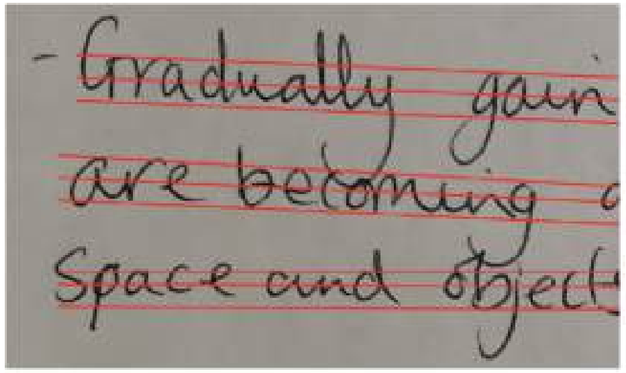
- pen-based computer screen surface
- pen-up and pen-down switching
- pen pressure
- velocity/changes of writing direction
- specifically

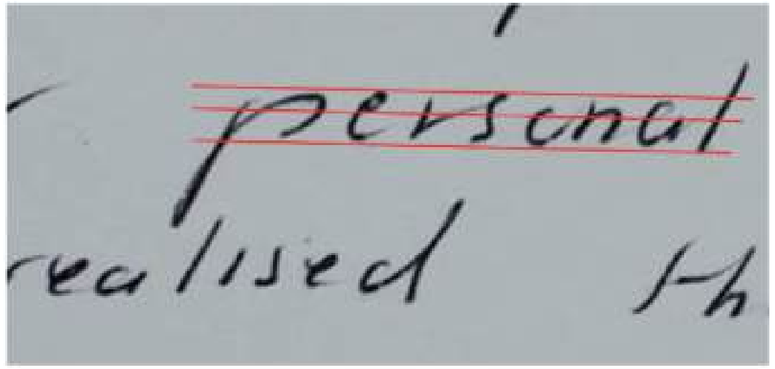


(https://github.com/alarxx/Dysgraphia-Diagnosis)

Off-line:

- piece of paper
- image





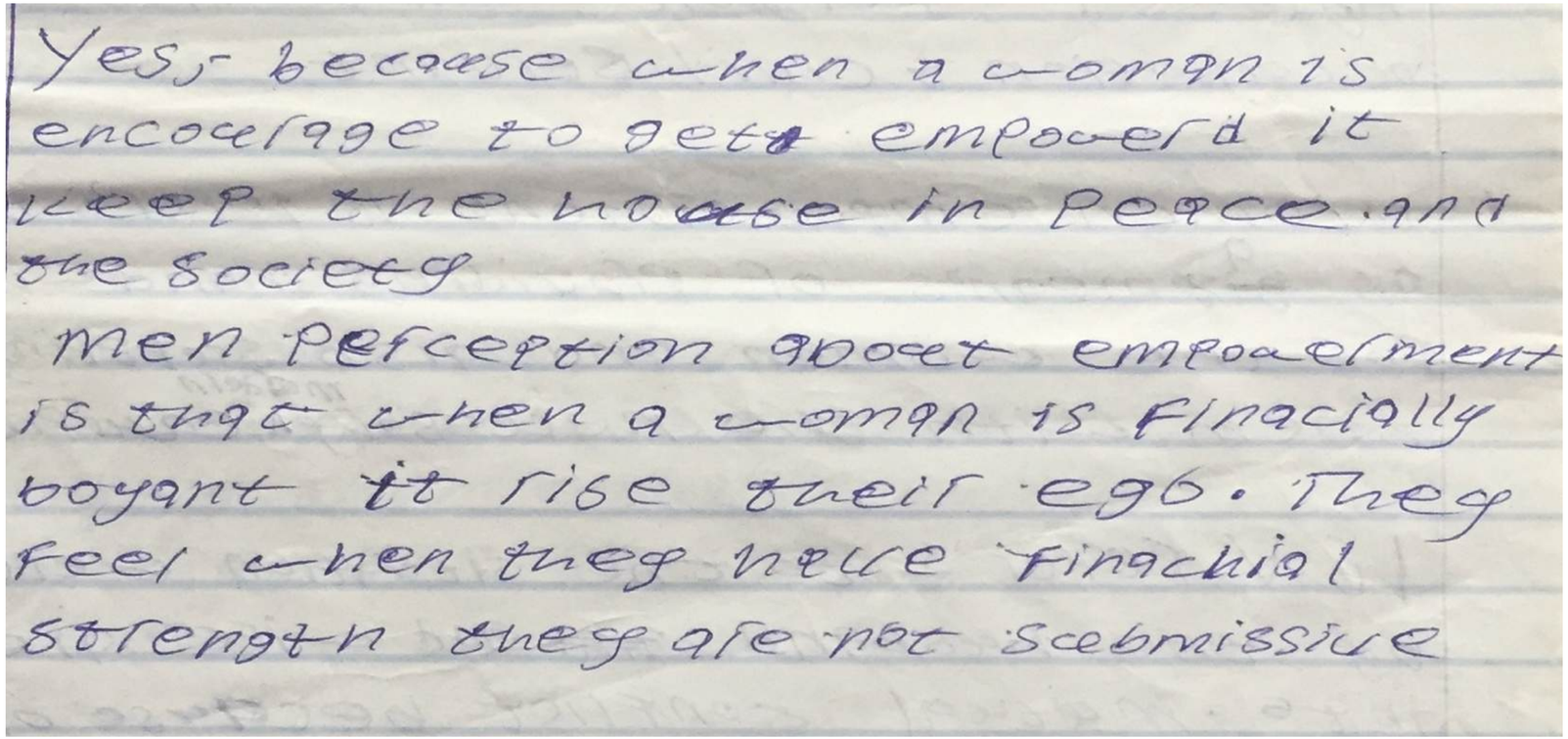
(GNHK: A Dataset for English Handwriting in the Wild)

https://dl.acm.org/doi/10.1007/978-3-030-86337-1_27

https://github.com/GoodNotes/GNHK-dataset

PROBLEM

Input data:



(GNHK: A Dataset for English Handwriting in the Wild)

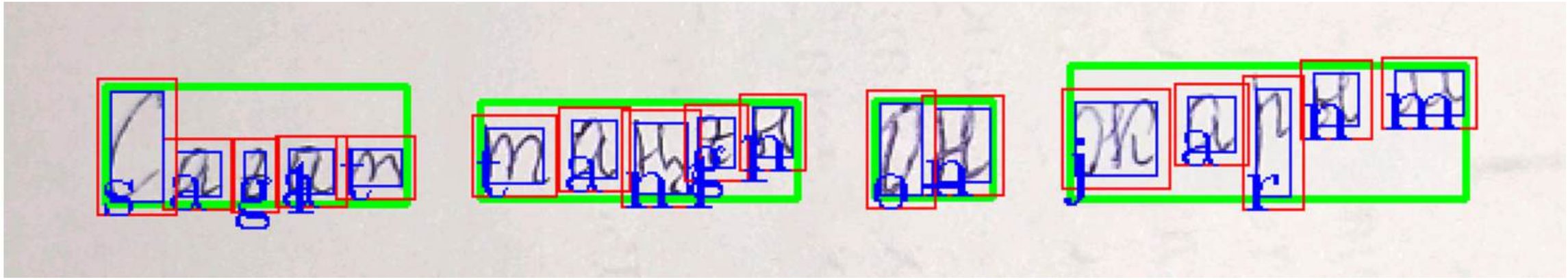
Output data: written text on the image

- **Text Detection**
- **Classification**

SOLUTION APPROACHES

- **Optical Character Recognition (OCR)**
- **Intelligent Character Recognition (ICR)**
- **Intelligent Word Recognition (IWR)**

OCR engines are primarily focused on character-by-character machine printed text recognition from a scanned document and **ICR** for different fonts or even handwritten text:



Intelligent Word Recognition (IWR) is the recognition of unconstrained handwritten words. IWR recognizes entire handwritten words or phrases instead of character-by-character, like its predecessors.

CLASSIFICATION DATASETS

Kazakh Alphabet Letters (CMNIST)

https://github.com/bolattleubayev/cmnist

Cyrillic Handwriting Dataset

https://www.kaggle.com/datasets/constantinwerner/cyrillic-handwriting-dataset/data

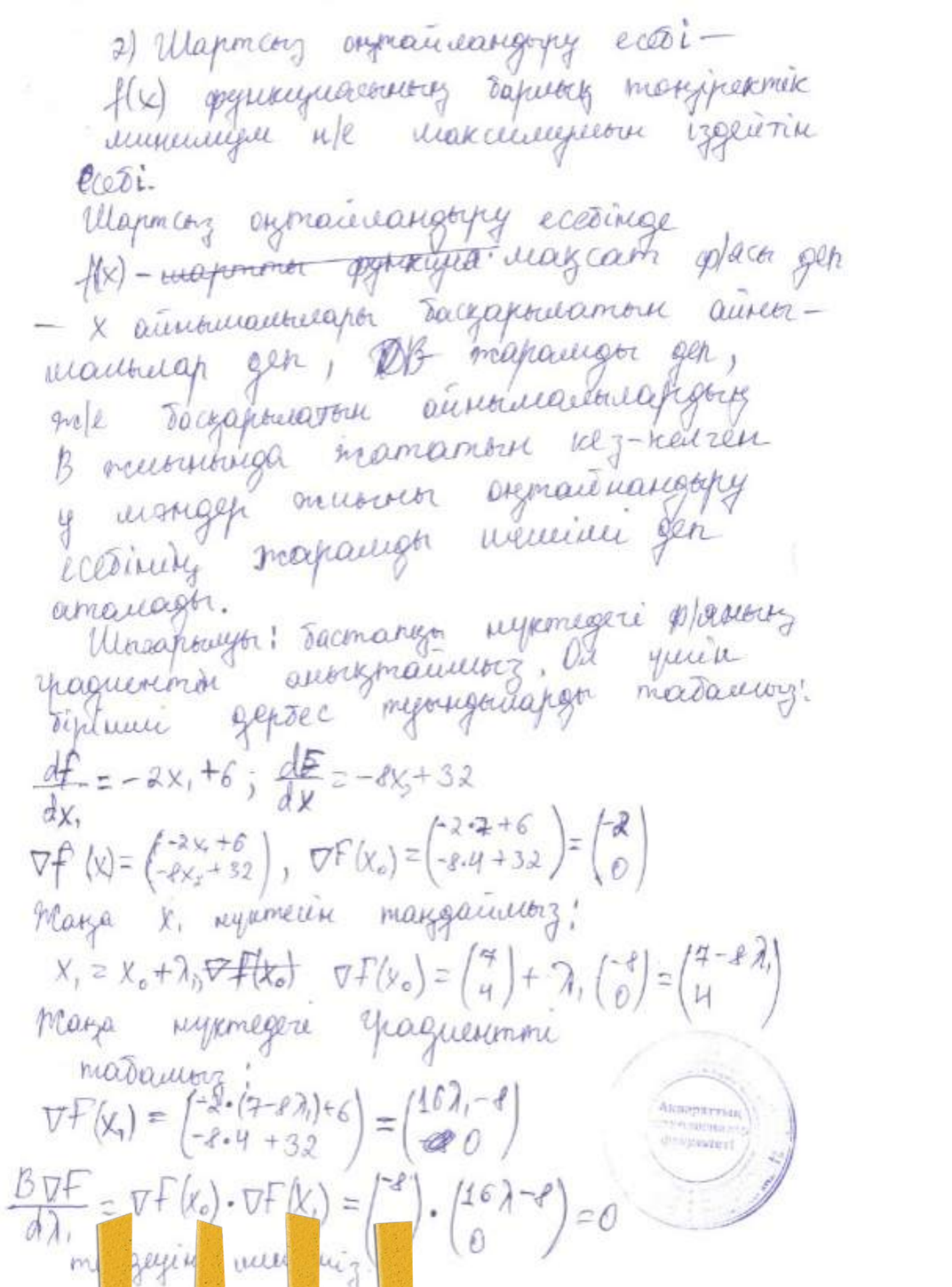
DATASET

KOHTD: Kazakh offline handwritten text dataset

Toiganbayeva, N., Kasem, M., Abdimanap, G., Bostanbekov, K., Abdallah, A., Alimova, A., & Nurseitov, D. (2022). Kohtd: Kazakh offline handwritten text dataset. Signal Processing: Image Communication, 108, 116827.

Chicago

Github: https://github.com/abdoelsayed2016/KOHTD



TEXT DETECTION DATASETS

MSRA Text Detection 500 Database (MSRA-TD500)

http://www.iapr-tc11.org/mediawiki/index.php?title=MSRA_Text_Detection_500_Database_(MSRA-TD500)

TextOCR

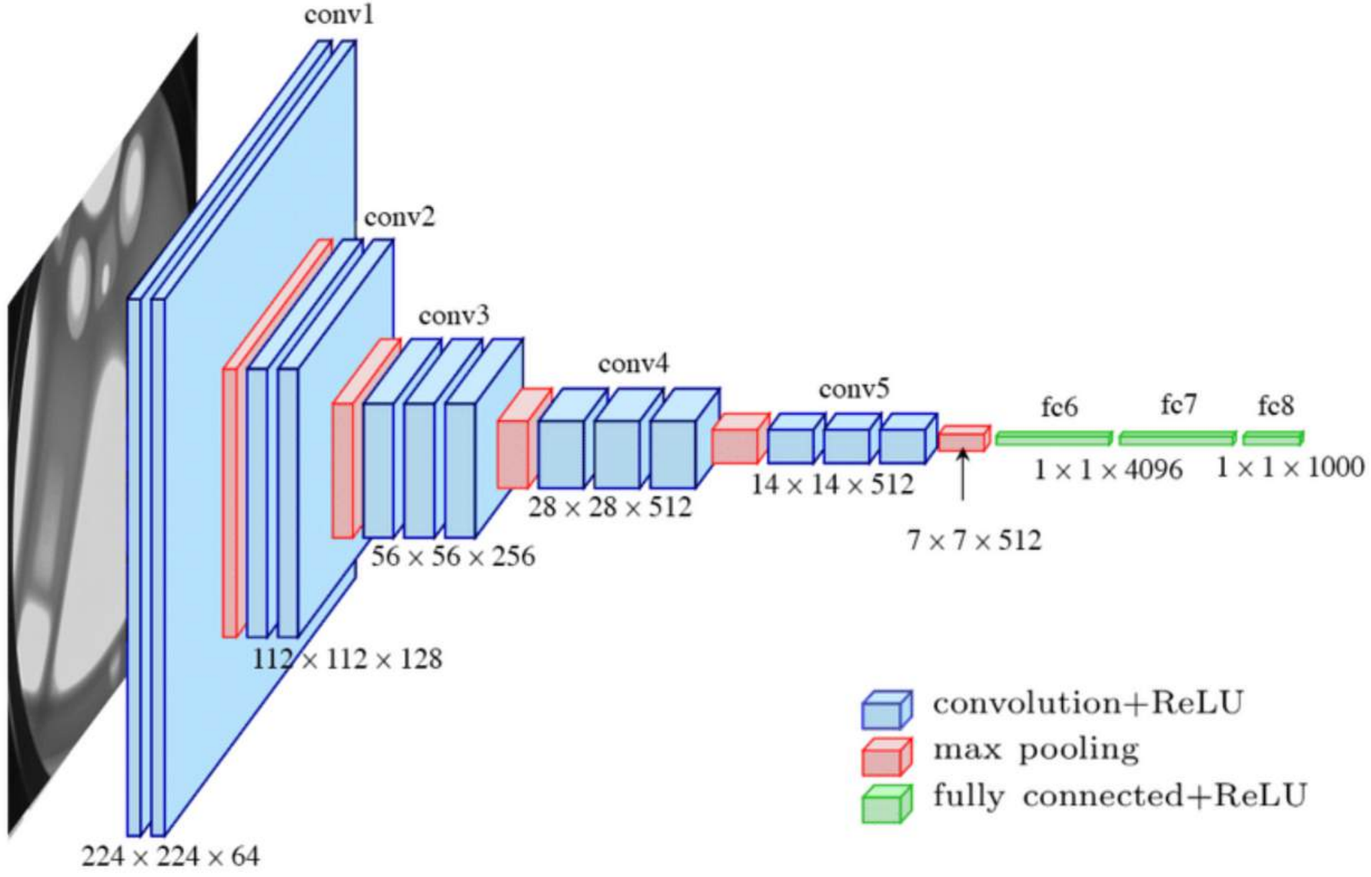
https://paperswithcode.com/dataset/textocr

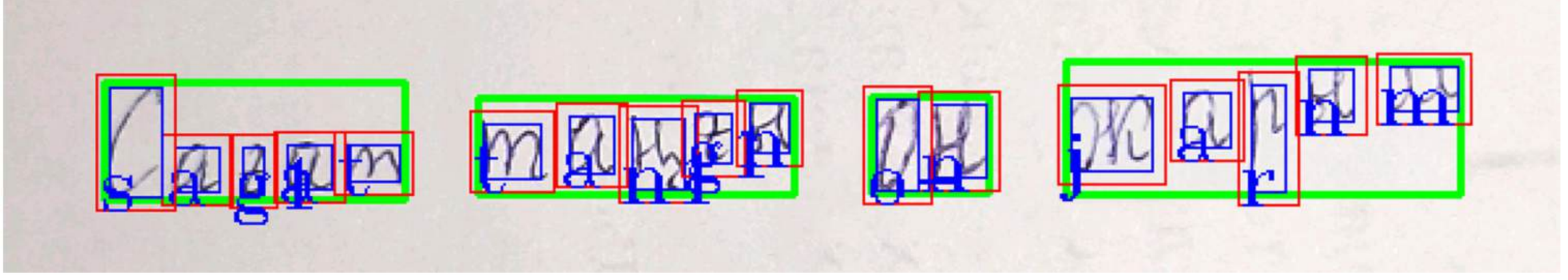
INTELLIGENT WORD RECOGNITION (IWR)

Application of deep learning: convolutional neural networks (CNNs) and recurrent neural networks (RNNs), including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), to handwriting image processing.

Recently, transformers, especially those that use attention mechanisms, have become popular for sequence processing, including handwriting recognition. They allow the model to focus on important parts of the input data when generating each character.

GANs can be used to generate realistic handwritten text, which helps to increase the available data for training recognition models.


conv1
conv2
conv3
conv4
conv5
fc6
fc7
fc8
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU



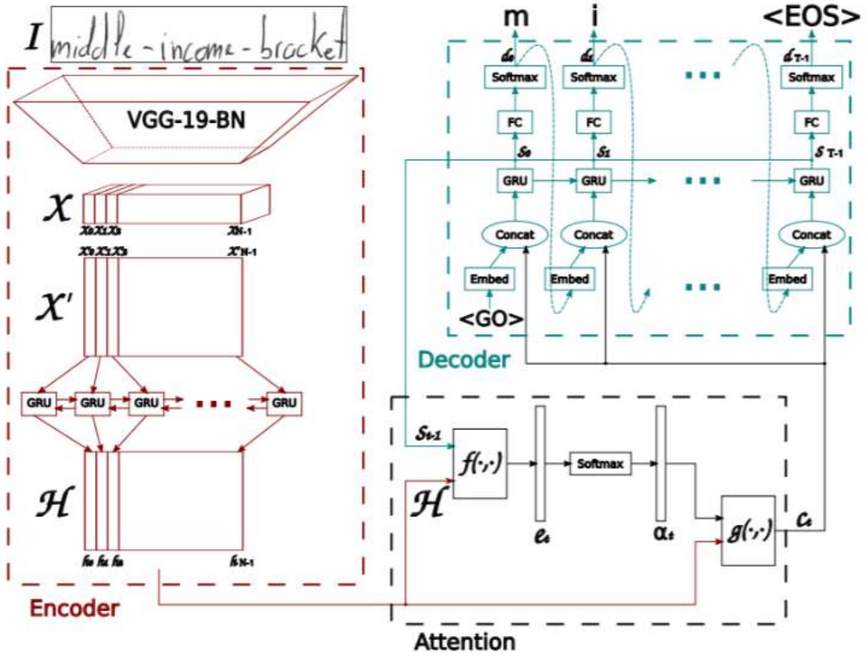


Fig. 1: Architecture of the seq2seq model with attention mechanism.

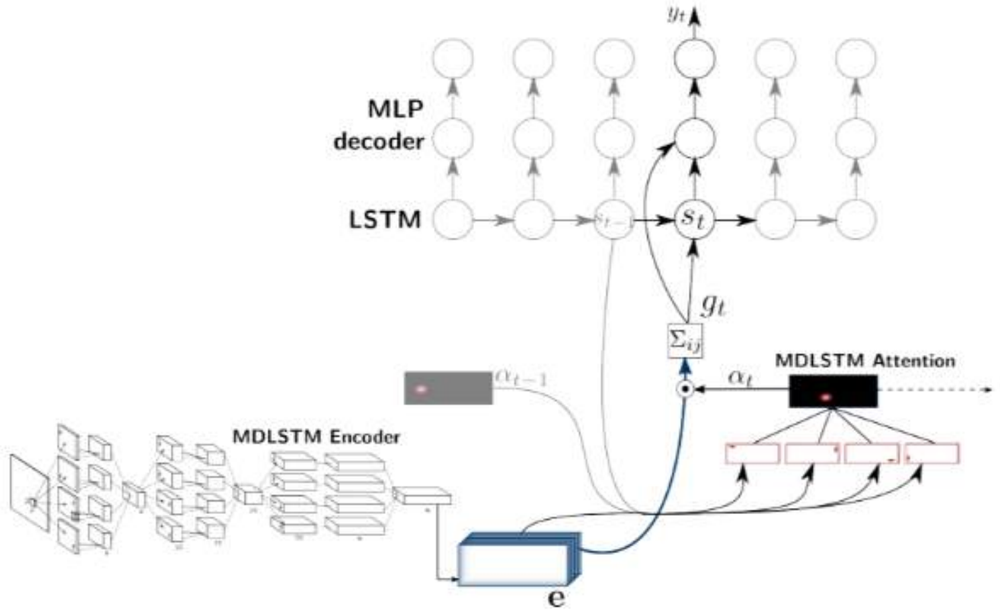


Figure 2: Proposed architecture. The encoder network has the same architecture as the standard network of Figure 1, except for the collapse and softmax layers. At each timestep, the feature maps, along with the previous attention map and state features are fed to an MDLSTM network which outputs new attention weights at each position. The weighted sum of the encoder features is computed and given to the state LSTM, and to the decoder. The decoder also considers the new state features and outputs character probabilities.

https://nanonets.com/blog/handwritten-character-recognition/

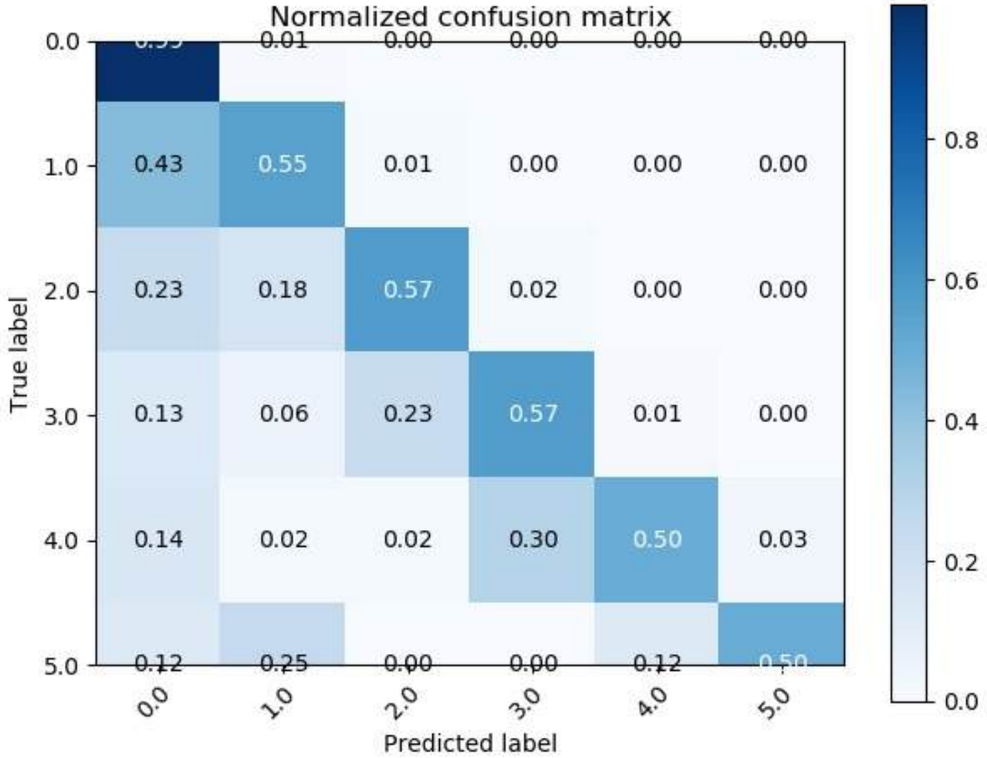
POST-EVALUATION

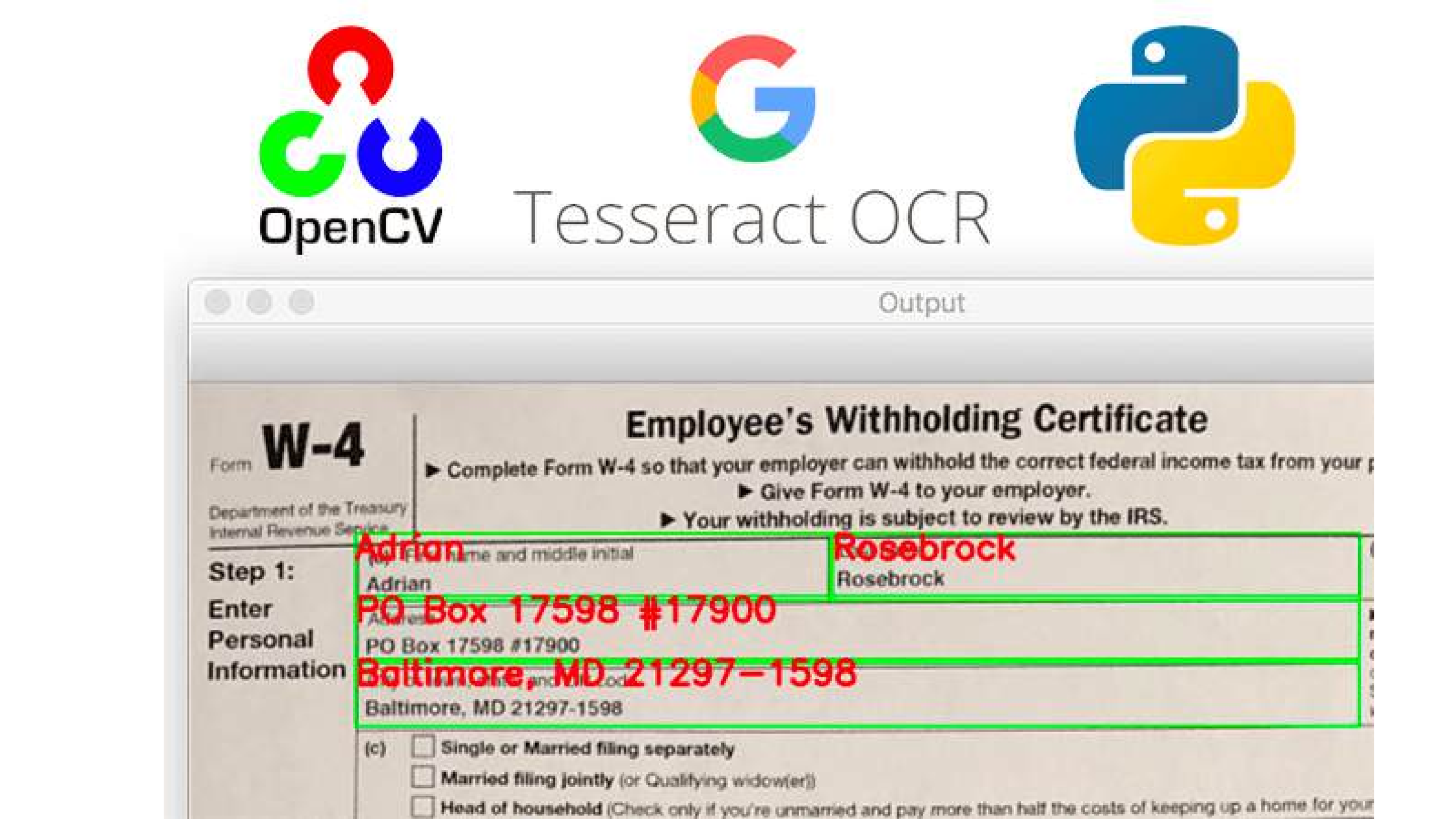
		h	e	l	l	o
	0	1	2	3	4	5
k	1	1	2	3	4	5
e	2	2				
l	3					
m	4					

Levenshtein distance

Analysing the probability of words in a sentence

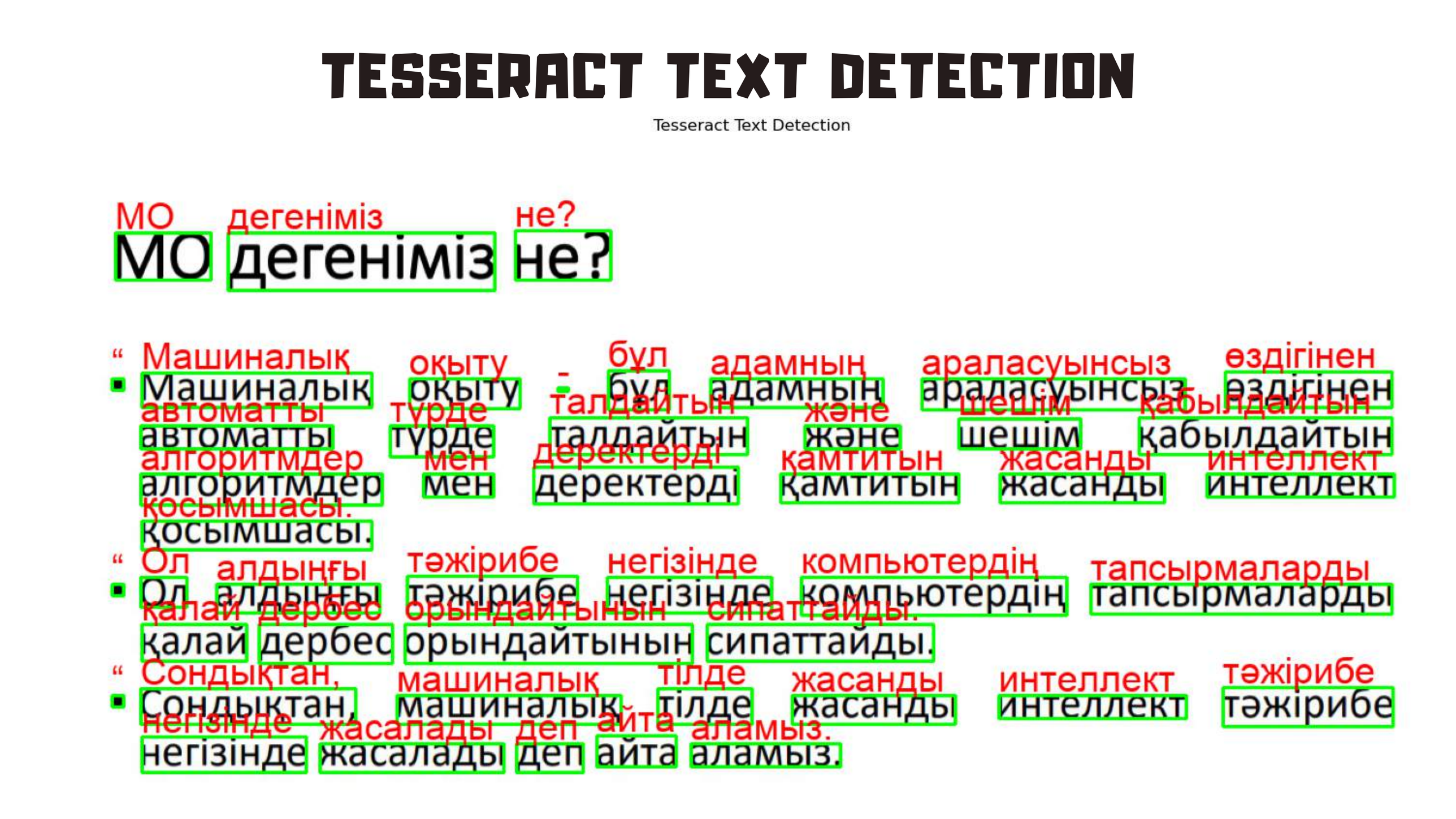
CONFUSION MATRIX FOR LETTERS



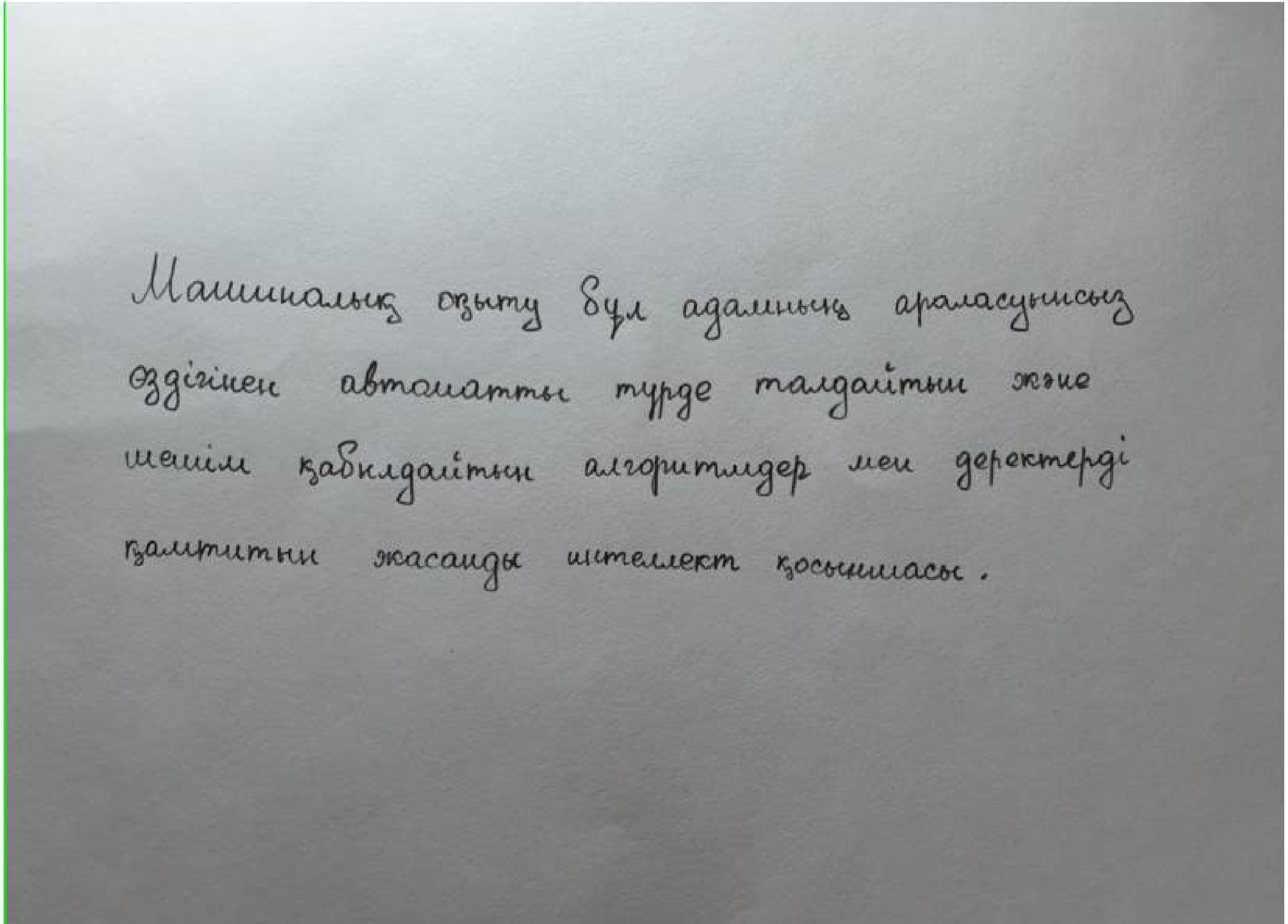

OpenCV
Tesseract OCR
Output
Form W-4
Department of the Treasury
Internal Revenue Service
Employee's Withholding Certificate
► Complete Form W-4 so that your employer can withhold the correct federal income tax from your
► Give Form W-4 to your employer.
► Your withholding is subject to review by the IRS.
Step 1:
Enter
Personal
Information
Adrian
Rosebrock
PO Box 17598 #17900
Baltimore, MD 21297−1598
Adrian
Rosebrock
PO Box 17598 #17900
Baltimore, MD 21297-1598
(c) Single or Married filing separately
Married filing jointly (or Qualifying widow(er))
Head of household (Check only if you're unmarried and pay more than half the costs of keeping up a home for your

TESSERACT TEXT DETECTION

Tesseract Text Detection

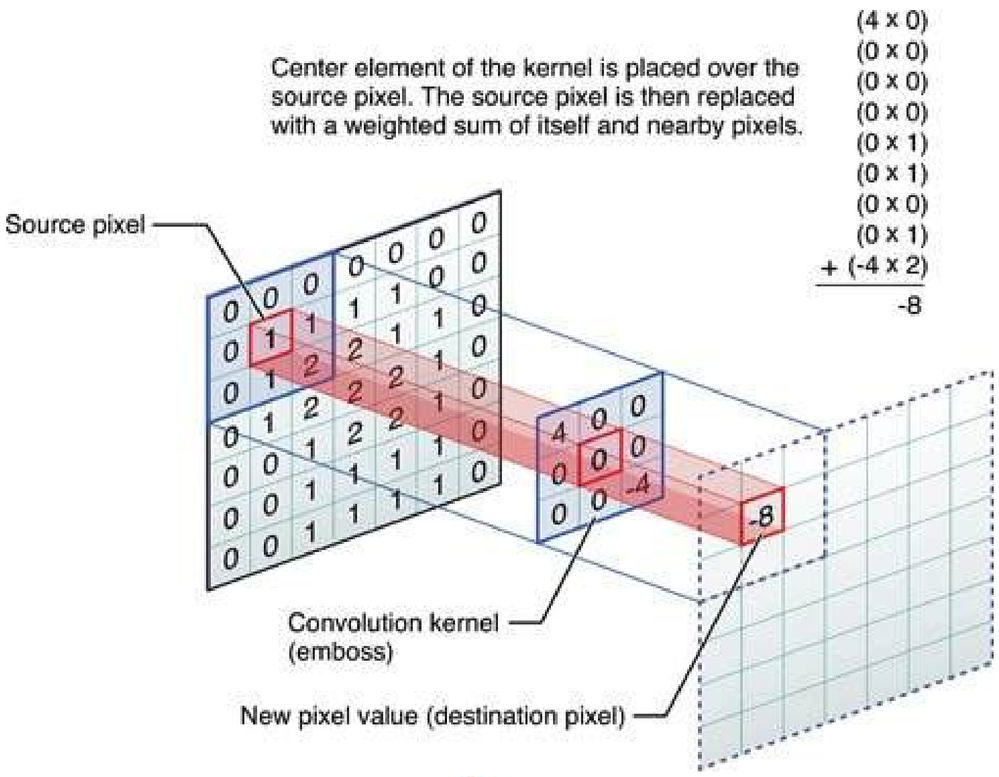


Tesseract Handwritten Text Detection



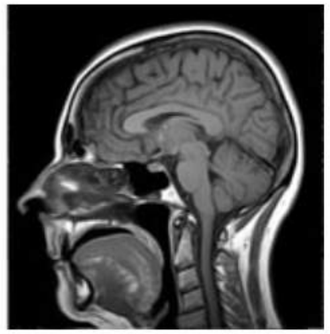
TEXT DETECTION

Edge Detector - searches for object contours by looking for a brightness difference



$$[f * g](t) = \int_0^t f(x)\, g(t-x)\, dx$$

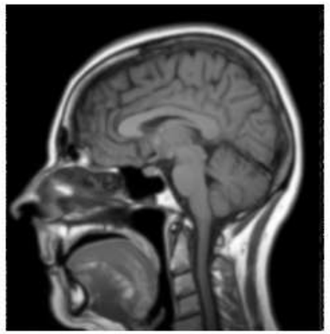
0	0	0
0	1	0
0	0	0



Identity Kernel: Just the starting image itself.

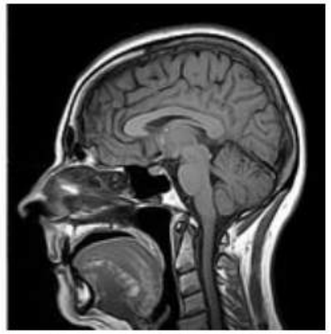
$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1



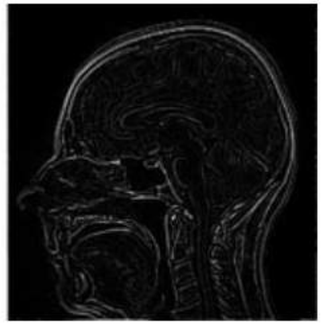
Box Blur Kernel: Each pixel is averaged equally with its 8 neighbors.

0	-1	0
-1	5	-1
0	-1	0



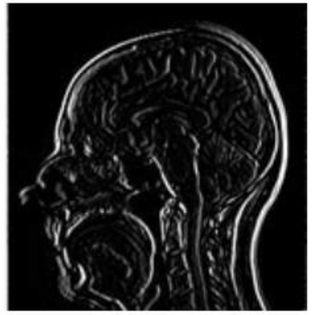
Sharpen Kernel:: Differences emphasized with its adjacent pixel values.

-1	-1	-1
-1	8	-1
-1	-1	-1



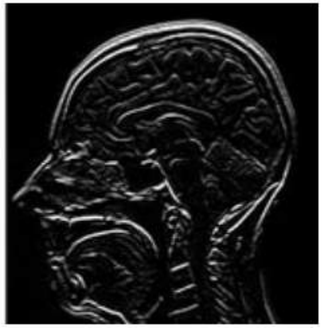
Outline ("Edge") Kernel: Highlights neighboring pixels with large differences of intensity.

1	0	-1
2	0	-2
1	0	-1



Left-to-right Sobel Kernel:. Used to accentuate differences between pixels along the horizontal axis

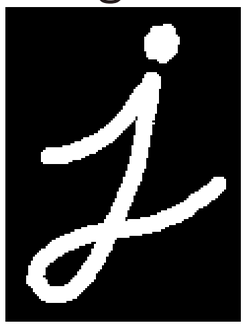
-1	-2	-1
0	0	0
1	2	1



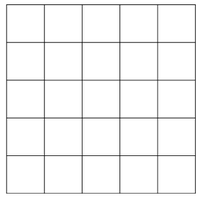
Superior-to-inferior Sobel Kernel: Used to accentuate differences between pixels along the vertical axis.

Morphological Transformations

Image



+



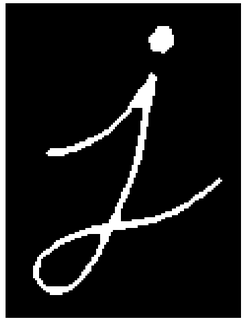
A pixel in the resulting image will be considered 1

1) only if all the pixels

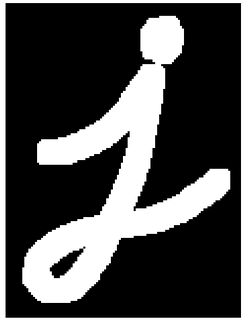
2) or at least one

under the kernel is 1

1) Erosion



2) Dilation



3) Opening

4) Closing

https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html

1.

The Dangerous and Thrilling Documentation Chronicles

Kismet Chameleon

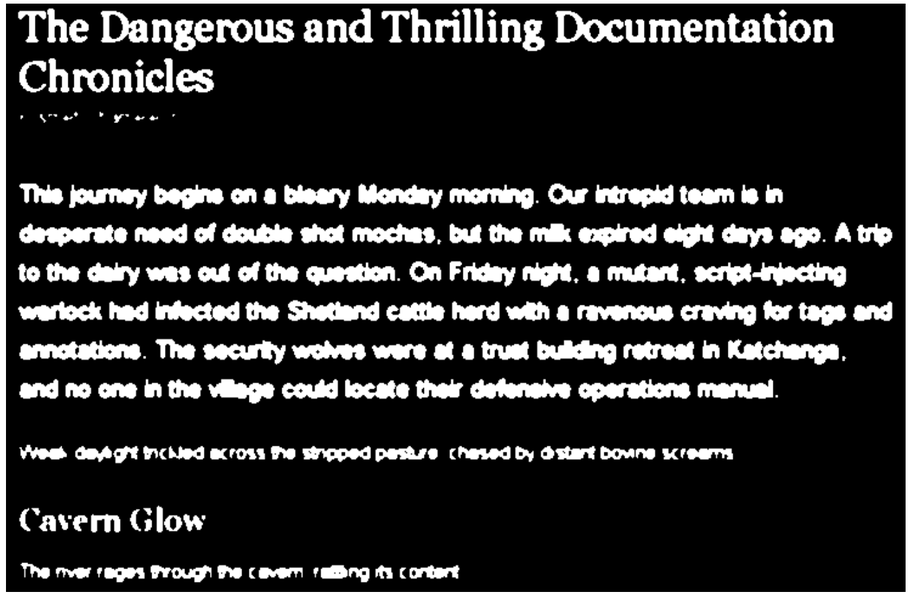
This journey begins on a bleary Monday morning. Our intrepid team is in desperate need of double shot mochas, but the milk expired eight days ago. A trip to the dairy was out of the question. On Friday night, a mutant, script-injecting warlock had infected the Shetland cattle herd with a ravenous craving for tags and annotations. The security wolves were at a trust building retreat in Katchanga, and no one in the village could locate their defensive operations manual.

Weak daylight trickled across the stripped pasture, chased by distant bovine screams…

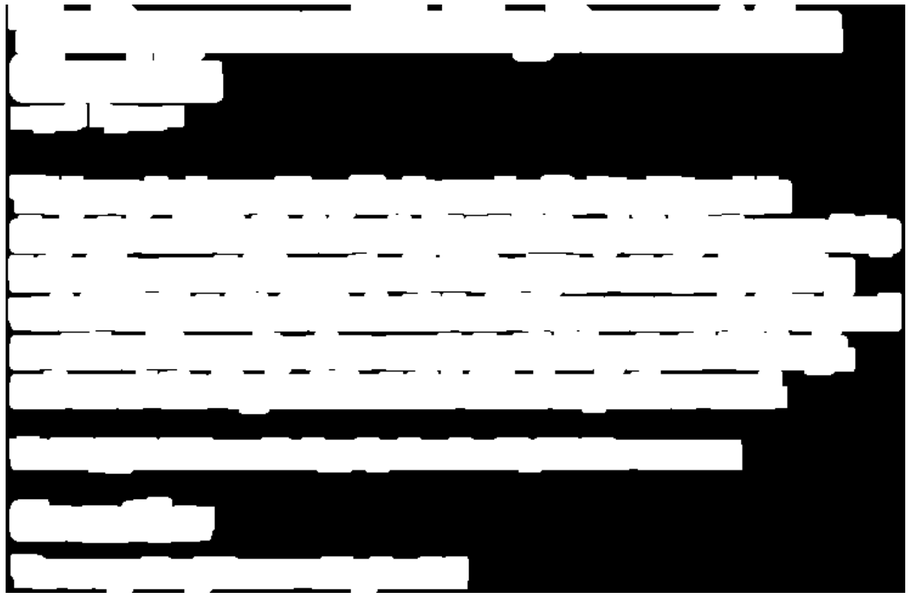
Cavern Glow

The river rages through the cavern, rattling its content…

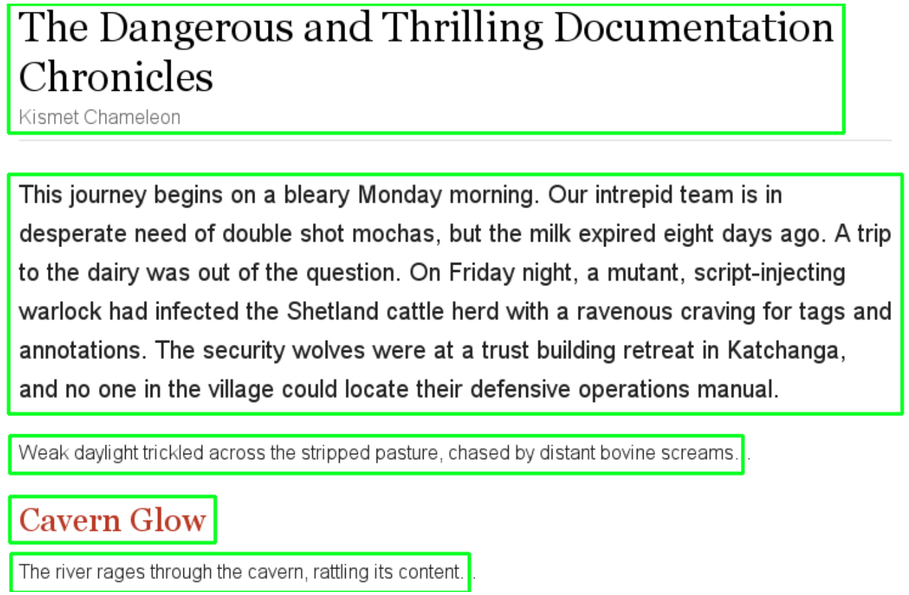
2.



3.



4.

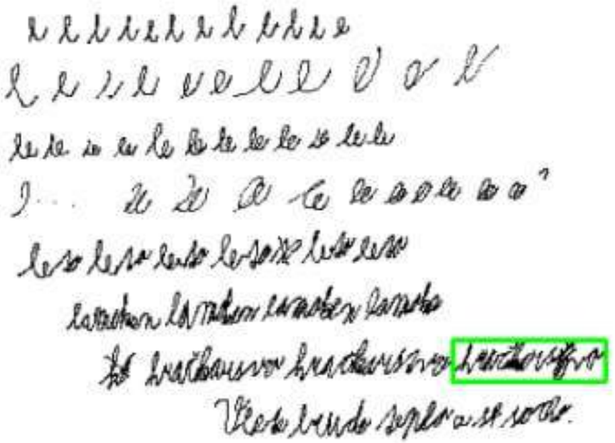


https://stackoverflow.com/questions/57249273/how-to-detect-paragraphs-in-a-text-document-image-for-a-non-consistent-text-stru

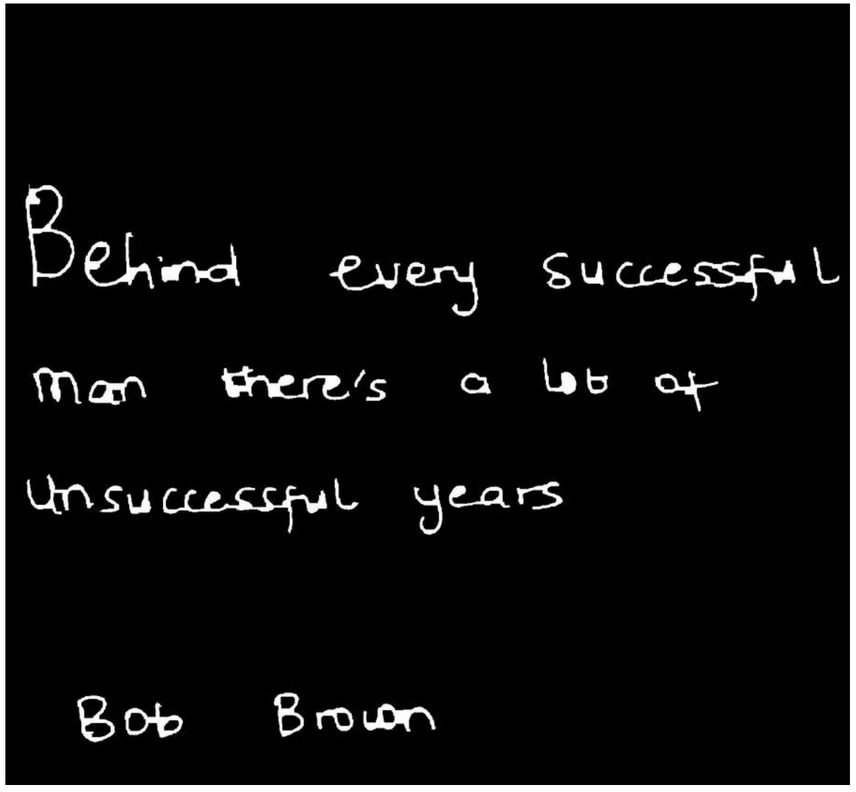
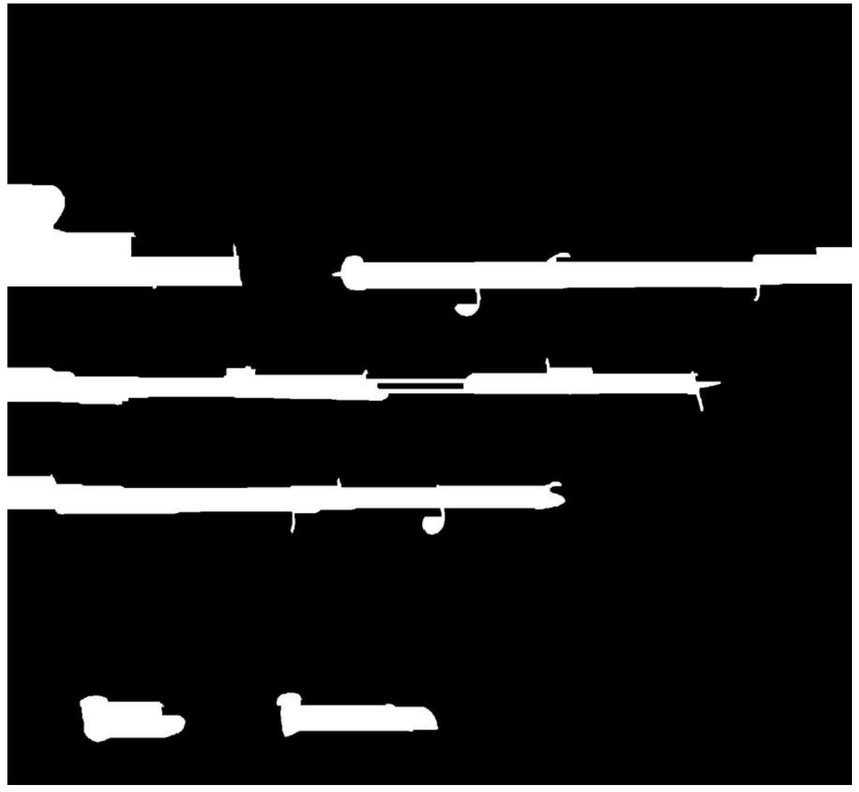
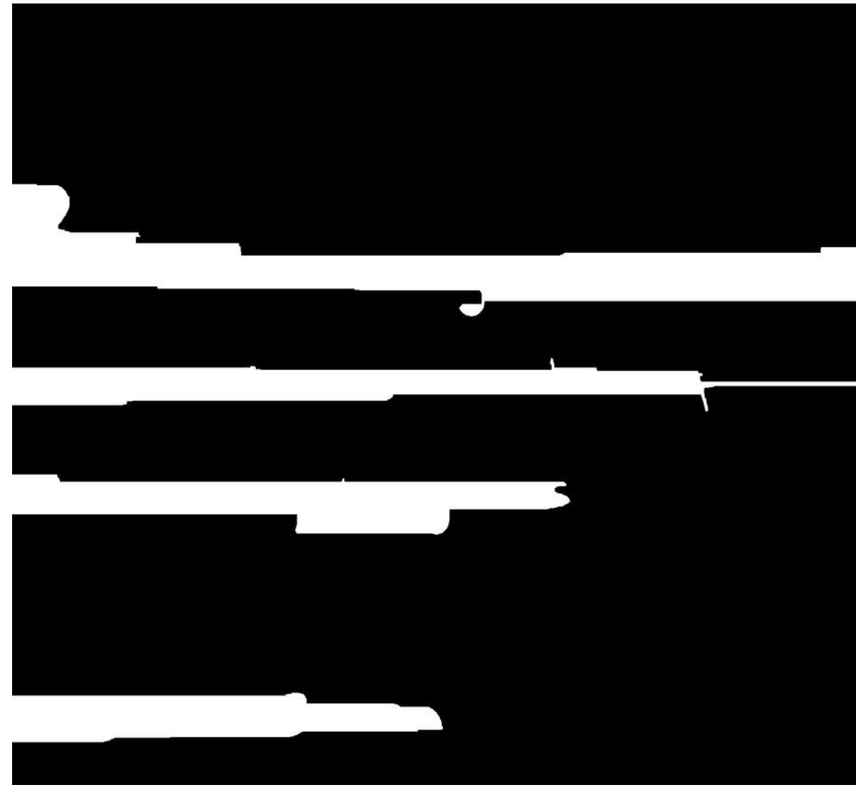
Creating a one-word picture:

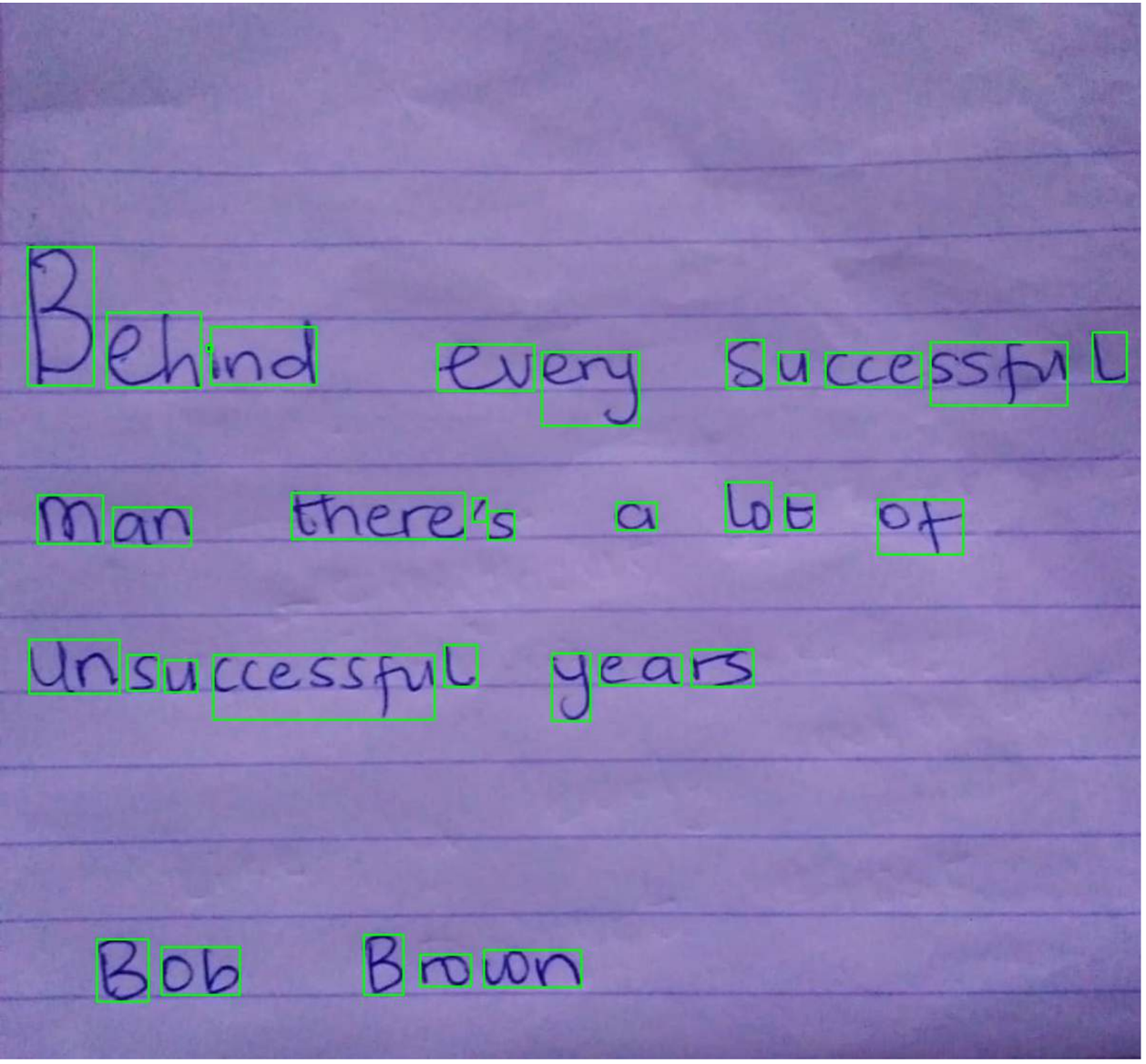
After Morphological Closing

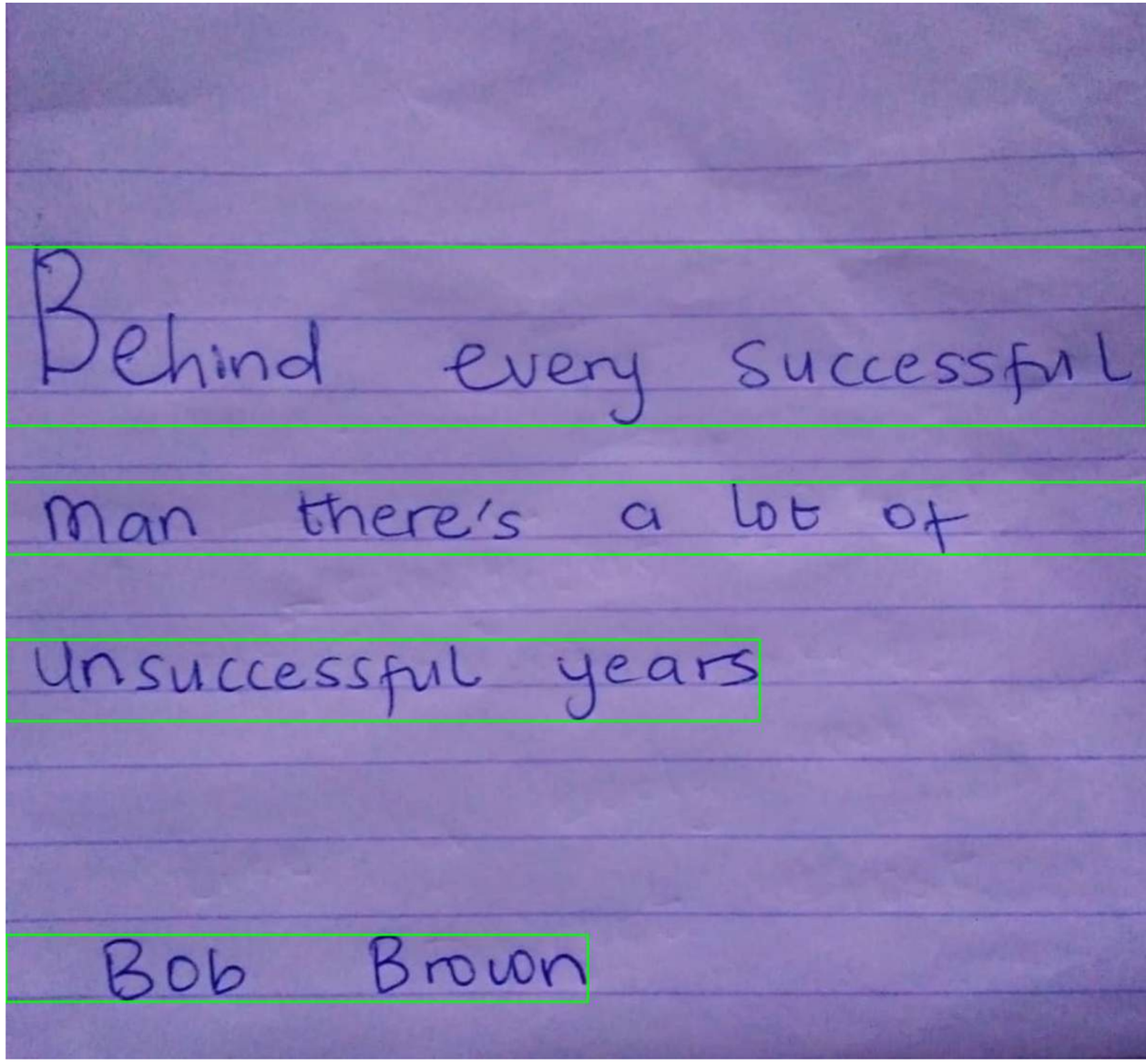
Original Image with Boundary of Word #1



224x224 Image with Word #1


Behind every successful
man there's a lot of
unsuccessful years
Bob Brown




Behind every Successful
Man there's a lot of
Unsuccessful years
Bob Brown


Behind every successful
Man there's a lot of
Unsuccessful years
Bob Brown

EAST: An Efficient and Accurate Scene Text Detector

Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang

Megvii Technology Inc., Beijing, China
{zxy, yaocong, wenhe, wangyuzhi, zsc, hwr, liangjiajun}@megvii.com

Abstract

Previous approaches for scene text detection have already achieved promising performances across various benchmarks. However, they usually fall short when dealing with challenging scenarios, even when equipped with deep neural network models, because the overall performance is determined by the interplay of multiple stages and components in the pipelines. In this work, we propose a simple yet powerful pipeline that yields fast and accurate text detection in natural scenes. The pipeline directly predicts words or text lines of arbitrary orientations and quadrilateral shapes in full images, eliminating unnecessary intermediate steps (e.g., candidate aggregation and word partitioning), with a single neural network. The simplicity of our pipeline allows concentrating efforts on designing loss functions and neural network architecture. Experiments on standard datasets including ICDAR 2015, COCO-Text and MSRA-TD500 demonstrate that the proposed algorithm significantly outperforms state-of-the-art methods in terms of both accuracy and efficiency. On the ICDAR 2015 dataset, the proposed algorithm achieves an F-score of 0.7820 at 13.2fps at 720p resolution.

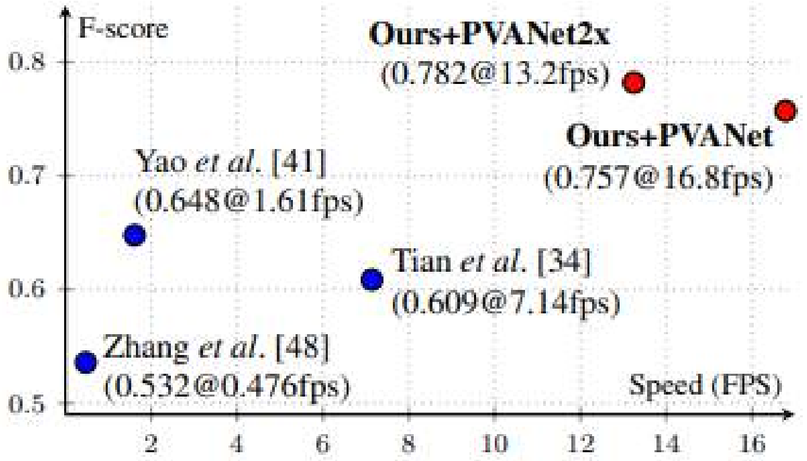


Figure 1. Performance versus speed on ICDAR 2015 [15] text localization challenge. As can be seen, our algorithm significantly surpasses competitors in accuracy, whilst running very fast. The specifications of hardware used are listed in Tab. 6.

features are manually designed [5, 25, 40, 10, 26, 45] to capture the properties of scene text, while in deep learning based methods [3, 13, 11, 12, 7, 48] effective features are directly learned from training data.

However, existing methods, either conventional or deep neural network based, mostly consist of several stages and components, which are probably sub-optimal and time-

EAST

(a) (b) (c)

Figure 5. Qualitative results of the proposed algorithm. (a) ICDAR 2015. (b) MSRA-TD500. (c) COCO-Text.

EAST

Gradient Boosting

In subject areas: Computer Science

Gradient boosting is a type of ensemble supervised machine learning algorithm that combines multiple weak learners to create a final model. It sequentially trains these models by placing more weights on instances with erroneous predictions, gradually minimizing a loss function. The predictions of the weak learners are compared with actual values, and the difference represents the error rate of the model. This error rate is used to calculate the gradient, which is used to find the direction for model parameter adjustment in the next round of training. Unlike a neural network model, where a single model is used, gradient boosting combines the predictions of multiple models to minimize the error.

AI generated definition based on: Machine Learning Guide for Oil and Gas Using Python, 2021

About this page + Add to Mendeley Set alert Discover other topics >

On this page

Chapters and Articles

You might find these chapters and articles relevant to this topic.

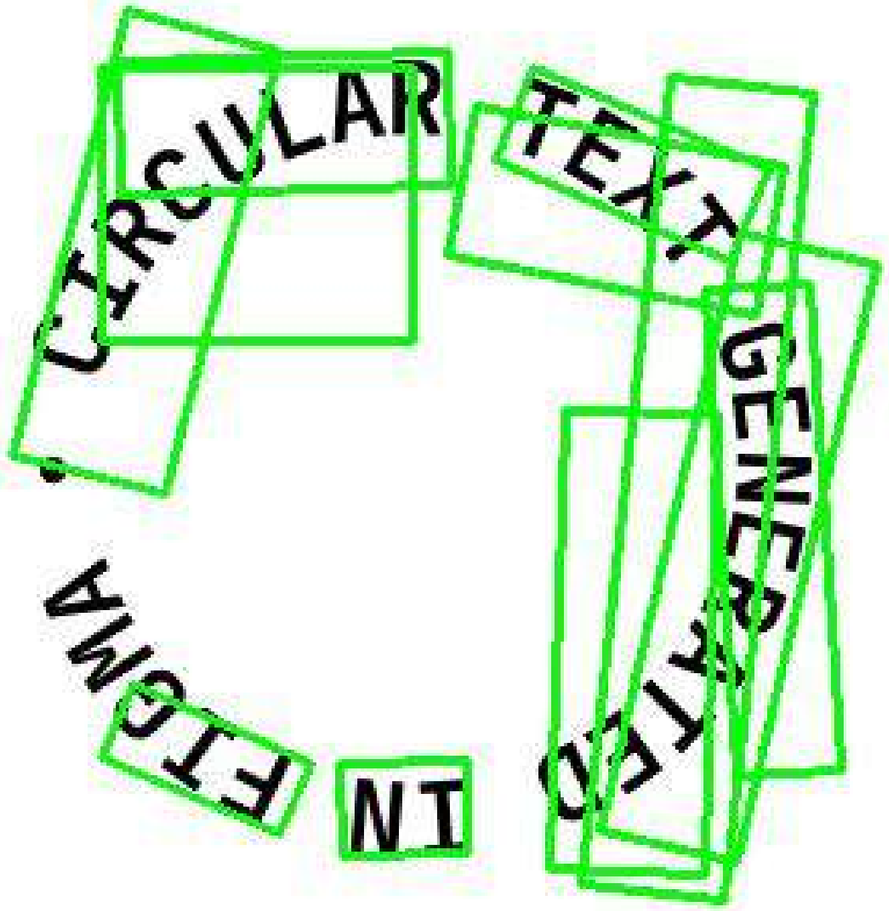
Supervised learning

Hossein Bevdadi, Alireza Haghighat, in Machine Learning Guide for Oil and Gas Using Python, 2021

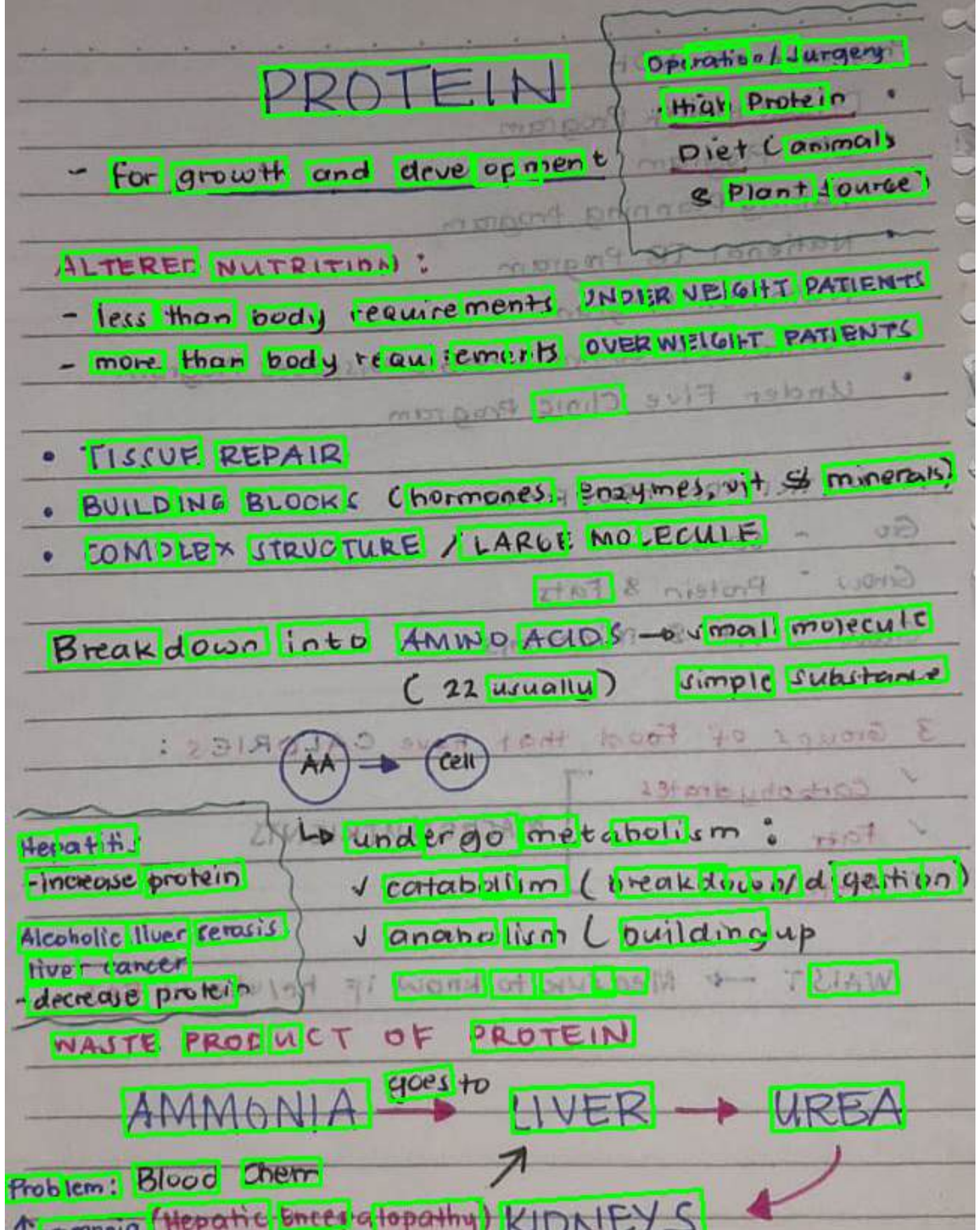
Gradient boosting

Gradient boosting is another type of ensemble supervised ML algorithm that can be used for both classification and regression problems. The main reason why algorithms such as random forest, extra trees, gradient boosting, etc., are called **ensemble** is because a final model is generated based on many individual models. As described under the bagging, boosting comparison section, gradient boosting will train many models sequentially by placing more weights on instances with erroneous predictions. Therefore, challenging cases are the focus

EAST



EAST


PROTEIN
- For growth and development
Operation / Surgery
High Protein
Diet (animals & plant source)
ALTERED NUTRITION:
- less than body requirements UNDERWEIGHT PATIENTS
- more than body requirements OVERWEIGHT PATIENTS
• TISSUE REPAIR
• BUILDING BLOCKS (hormones, enzymes, vit & minerals)
• COMPLEX STRUCTURE / LARGE MOLECULE
Breakdown into AMINO ACIDS → small molecule
(22 usually) simple substance
AA → cell
↳ undergo metabolism:
✓ catabolism (breakdown/digestion)
✓ anabolism (building up
Hepatitis
-increase protein
Alcoholic liver cerosis
liver cancer
-decrease protein
WASTE PRODUCT OF PROTEIN
AMMONIA goes to LIVER → UREA
Problem: Blood Chem
(Hepatic Encephalopathy) KIDNEYS

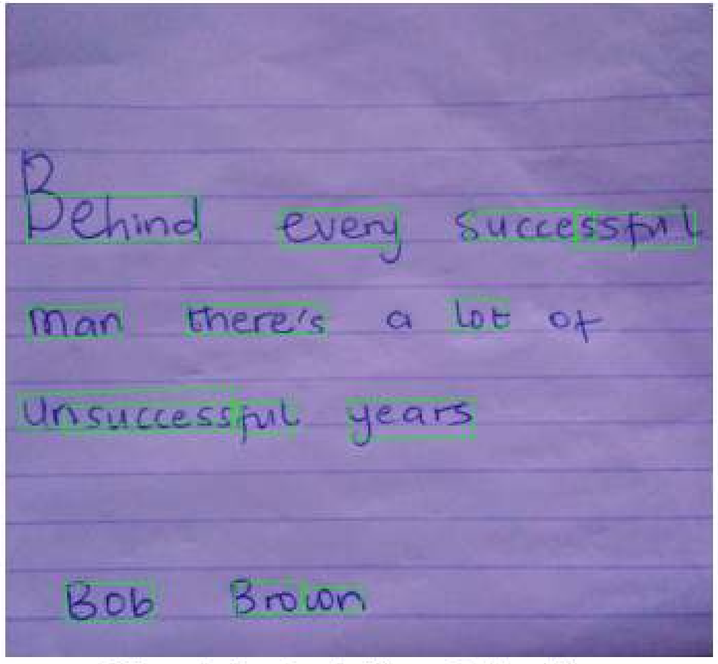
Behind every successful

man there's a lot of

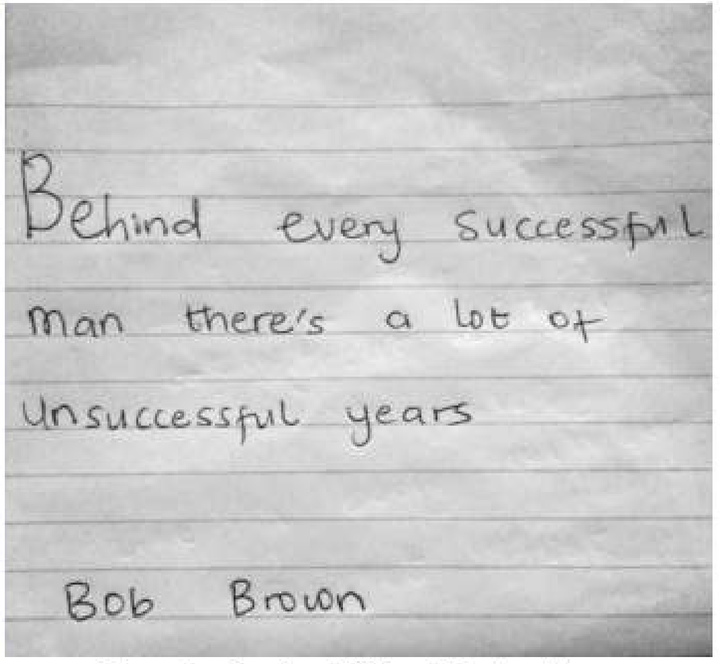
unsuccessful years

Bob Brown

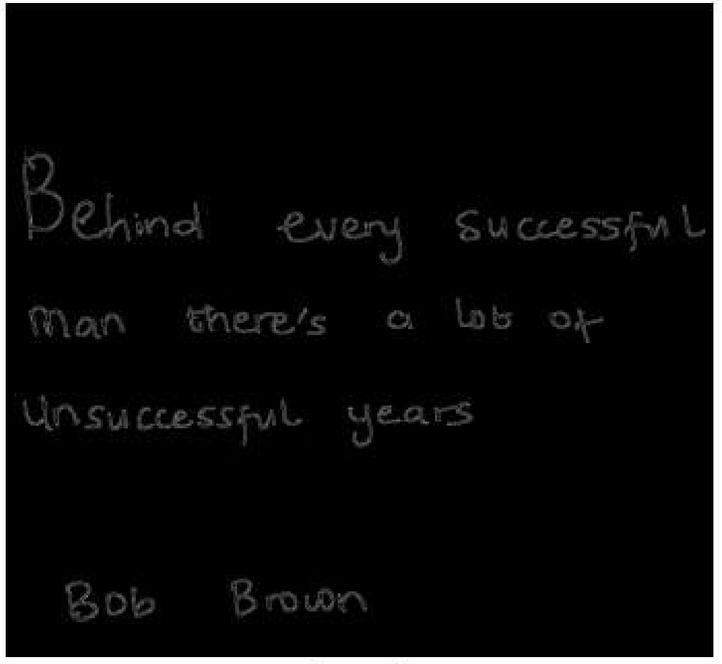
EAST: 69.0



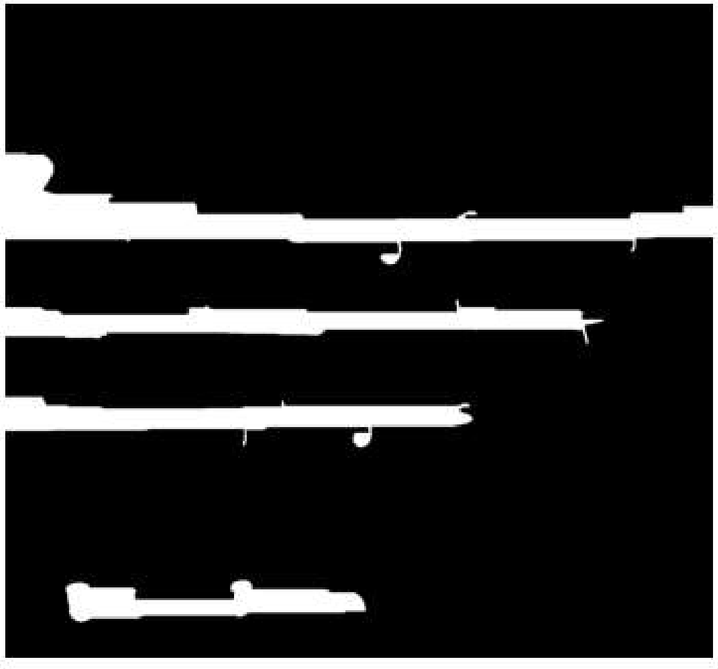
Grayscale Image



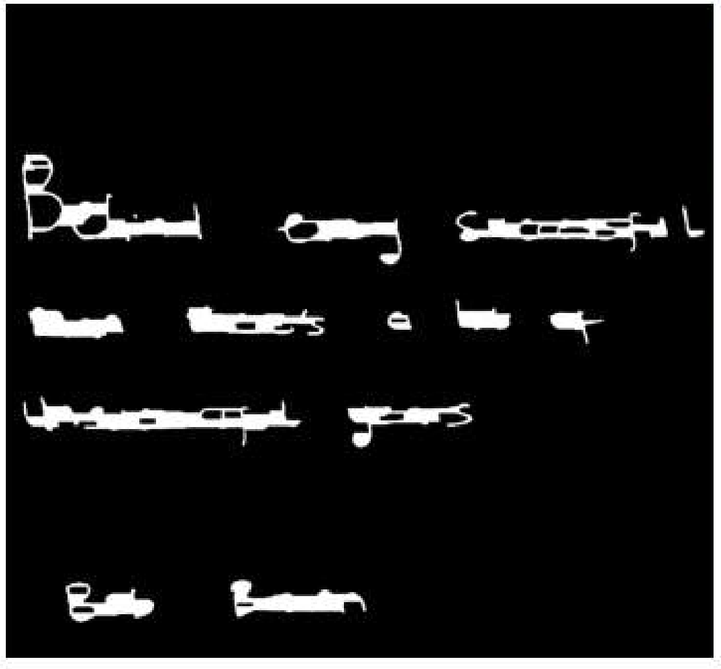
Canny Edges



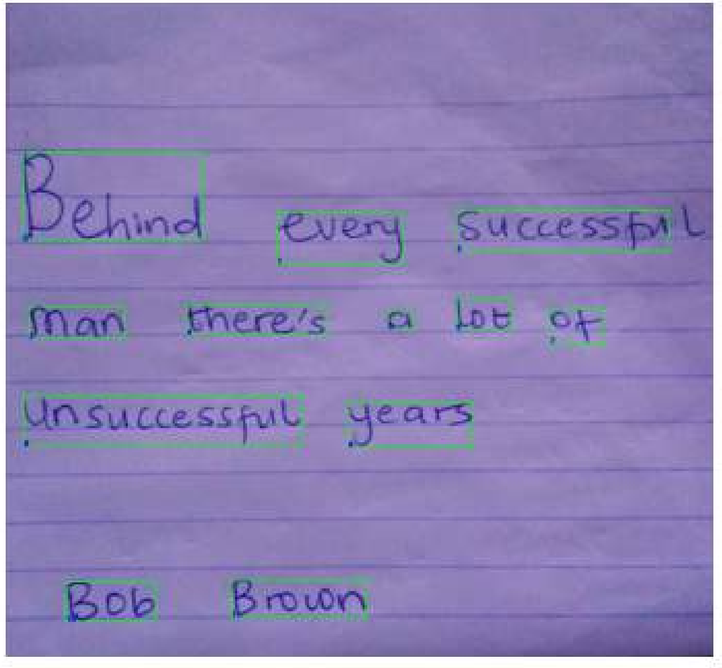
Morphological Line Detection



Morphological Word Detection



Result



General Algorithm of the applied Text Detection

1. Estimation of average text height using EAST
2. Morphological transformation finds text regions.
3. Visualization and cropping out text regions for further analysis.

The approach combines the advantages of deep model (EAST) with classical image processing methods.

Limitations

The method is designed to detect individual words on white paper, where the lines of text are written horizontally. This is a limitation, because handwritten text can be written in different structures, in a circle, in tables, and there can be different formulas and this requires a different research.

Kazakh Alphabet Letters (CMNIST)

https://github.com/bolattleubayev/cmnist

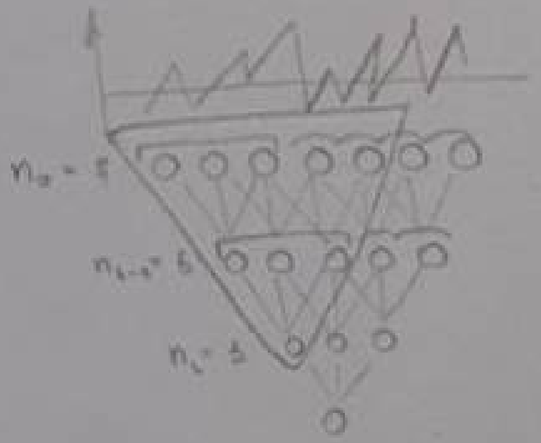
VGG-16

Receptive Field

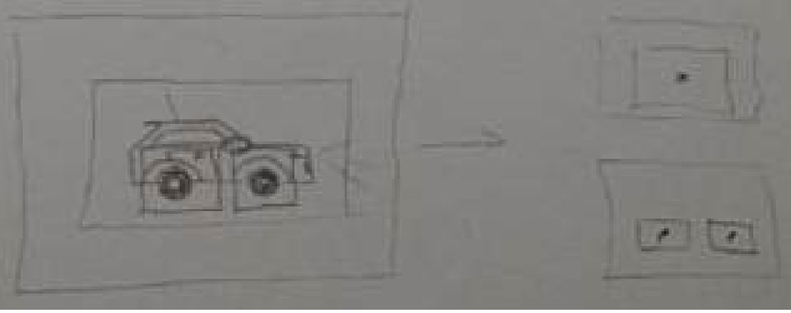
https://www.almabetter.com/bytes/articles/receptive-field-for-cnn-in-layperson-s-words

$(L-1)\cdot S$

видит нейрон

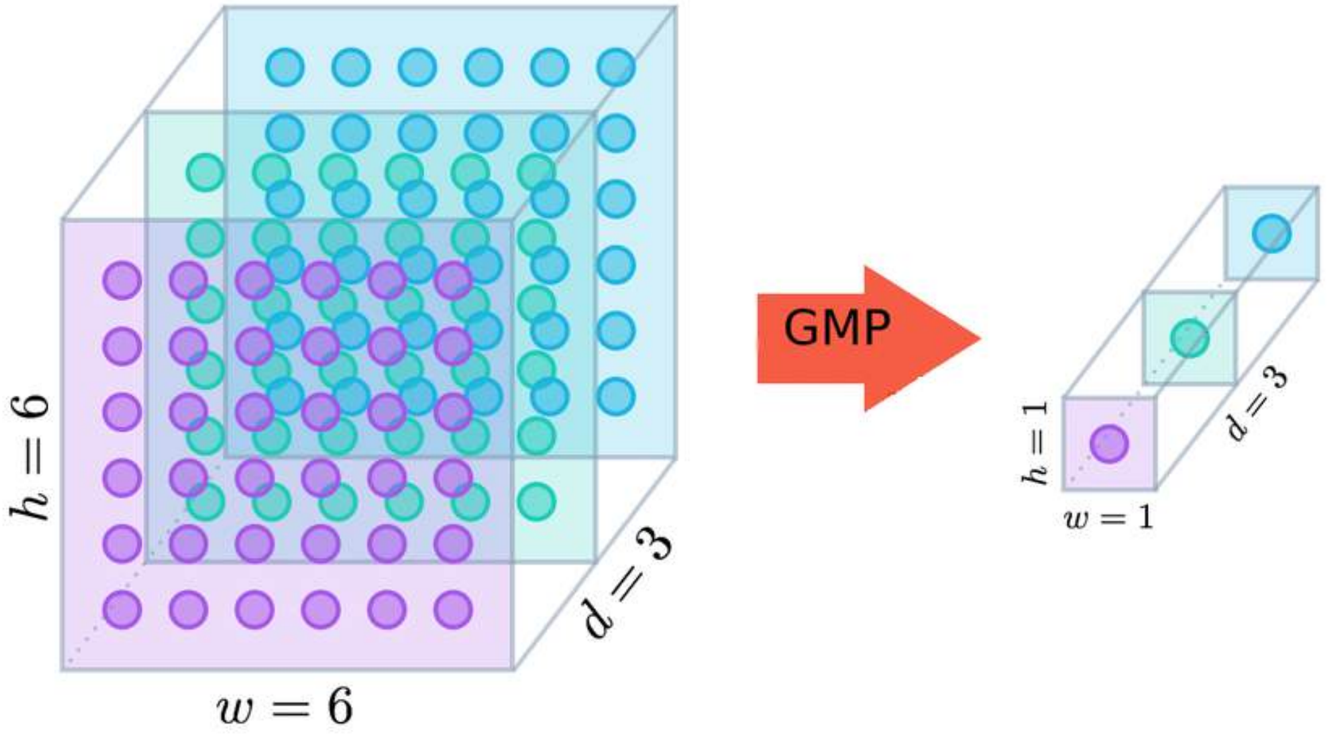


Receptive field нейрон может распознавать
размера, или относительное расположение патерна.



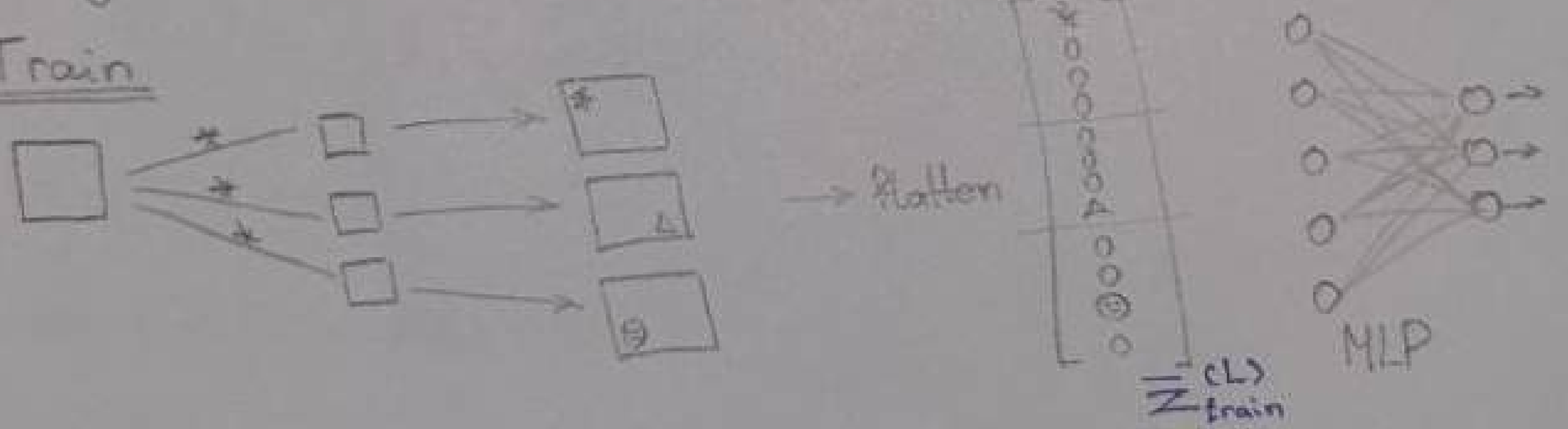
Receptive Field

3	8×64×64
5	64×64×64
10	
12	64×32×32
14	128×32×32
28	
30	128×16×16
32	256×16×16
64	
66	256×8×8
68	512×8×8


$h = 6$
$w = 6$
$d = 3$
GMP
$h = 1$
$w = 1$
$d = 3$

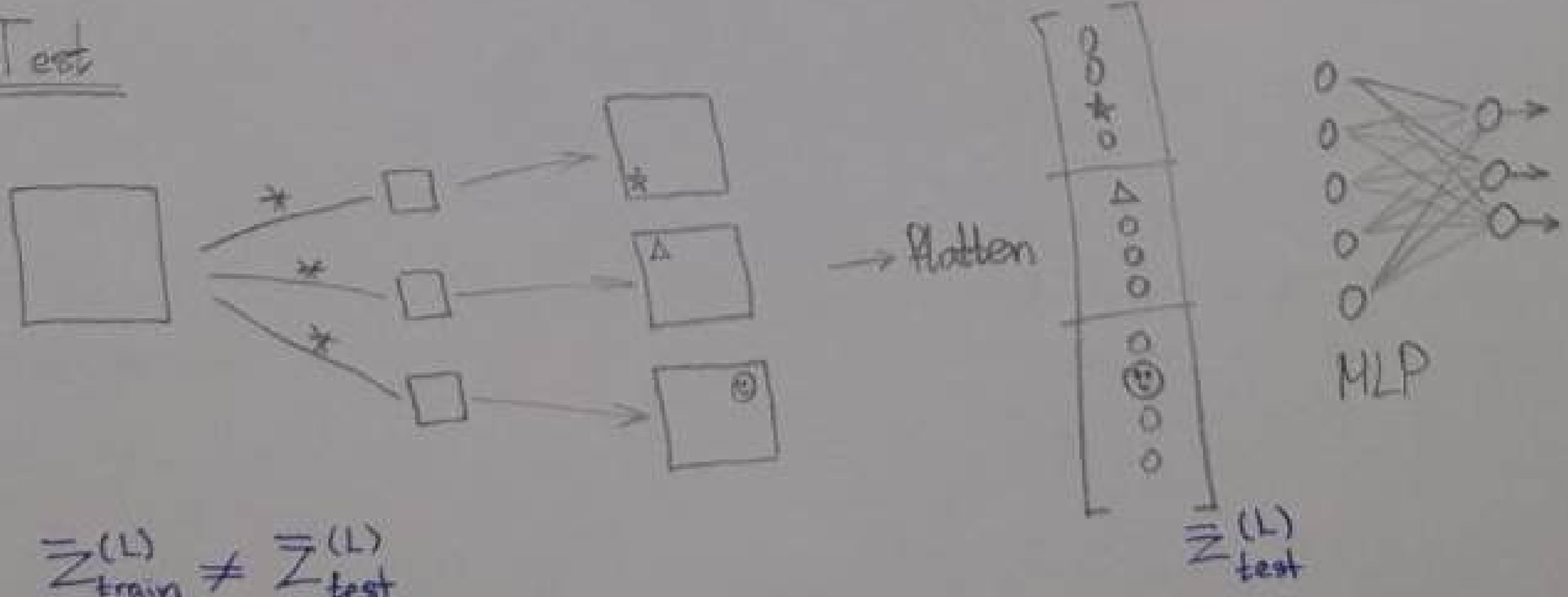
Why to use GMP

Train



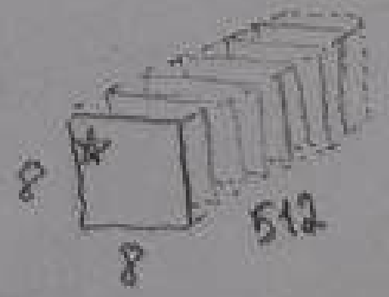
$\vec{Z}^{(L)}_{train}$

Test

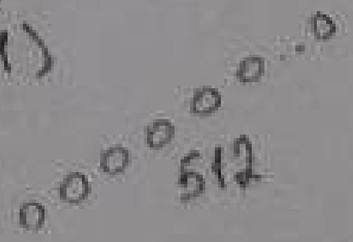


$\vec{Z}^{(L)}_{test}$

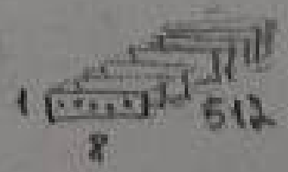
$$\vec{Z}^{(L)}_{train} \neq \vec{Z}^{(L)}_{test}$$

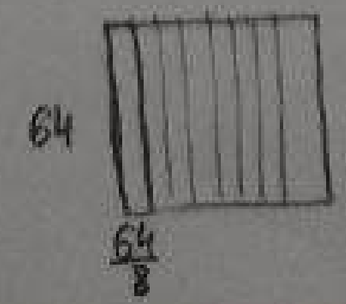


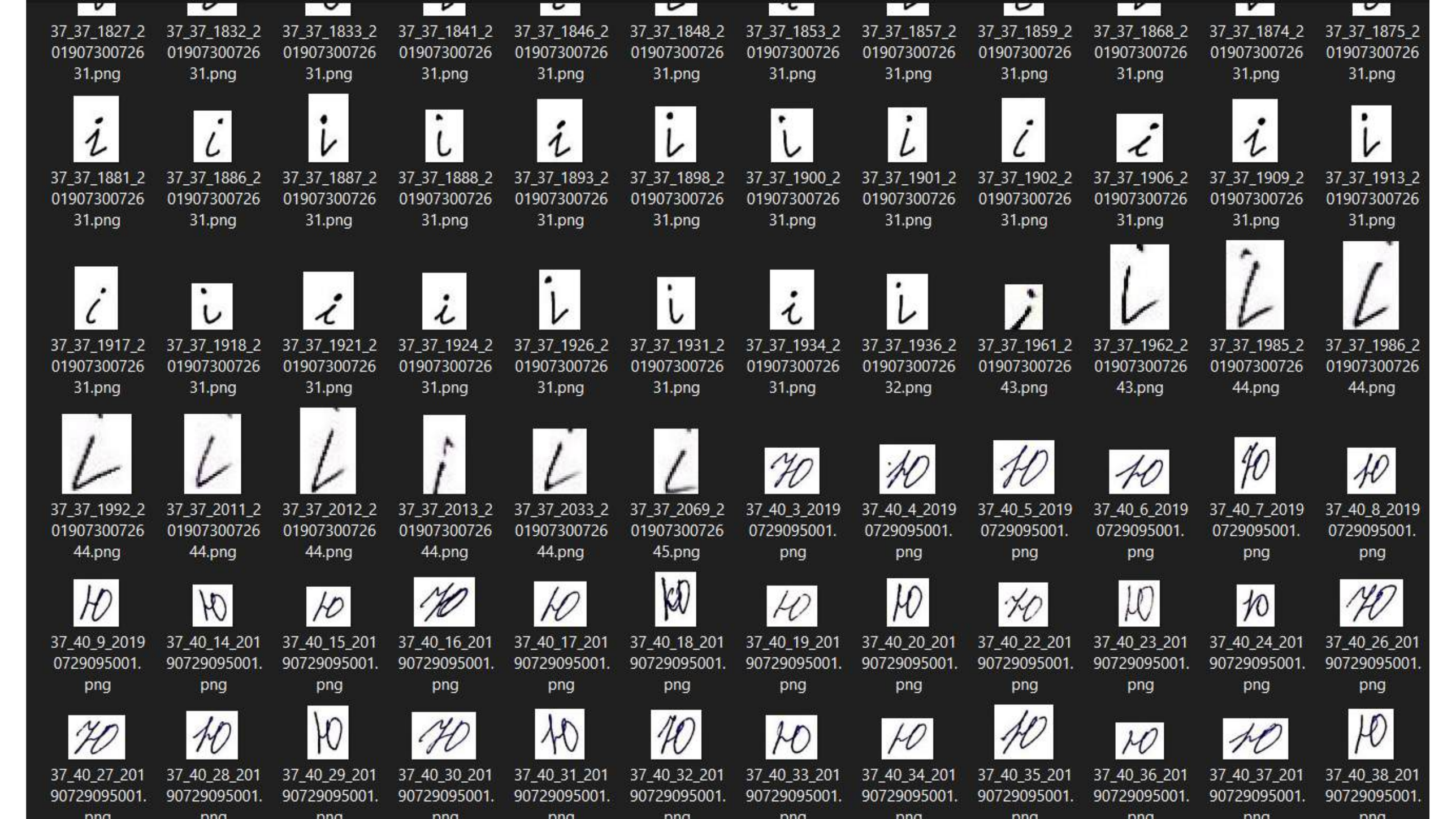
1) GMP (1, 1)

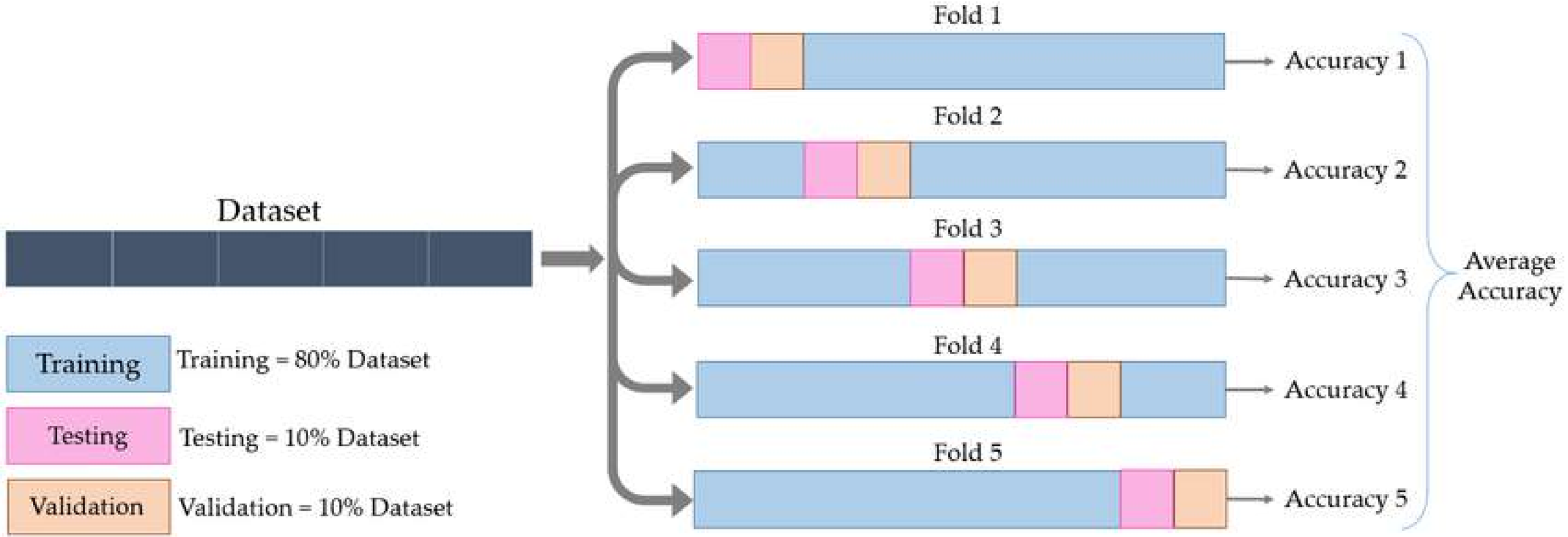


2) GMP (1, *)

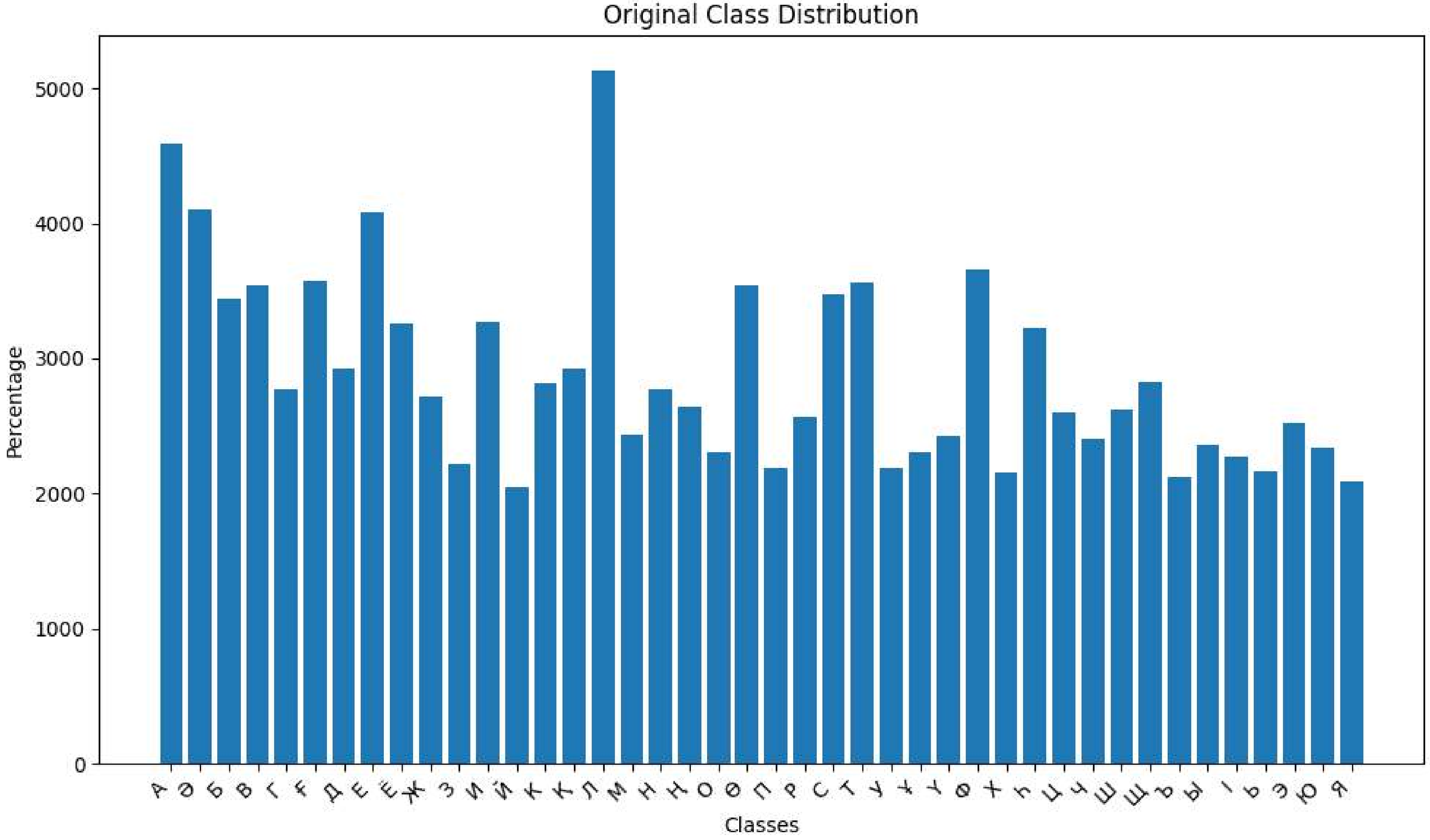


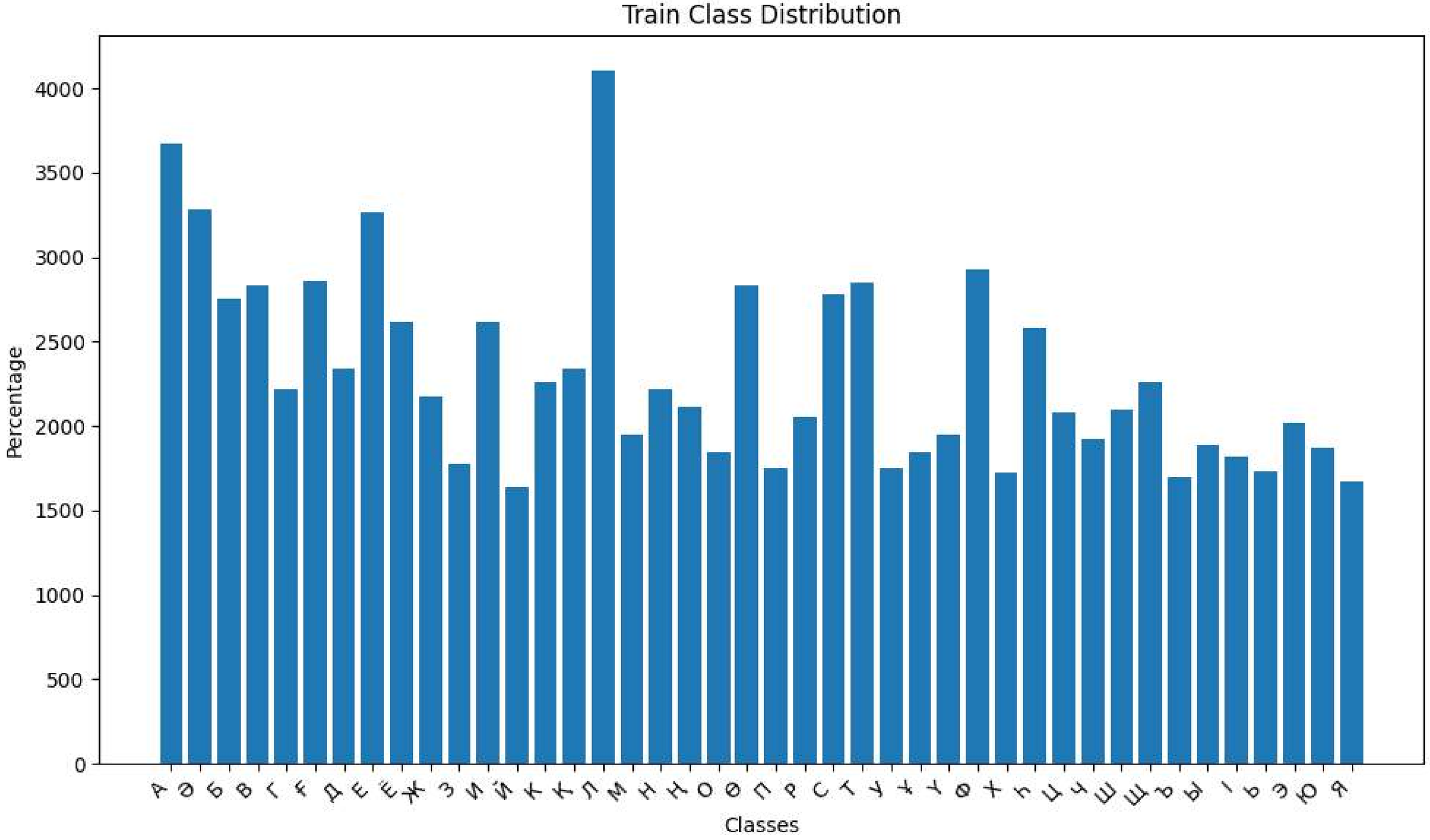


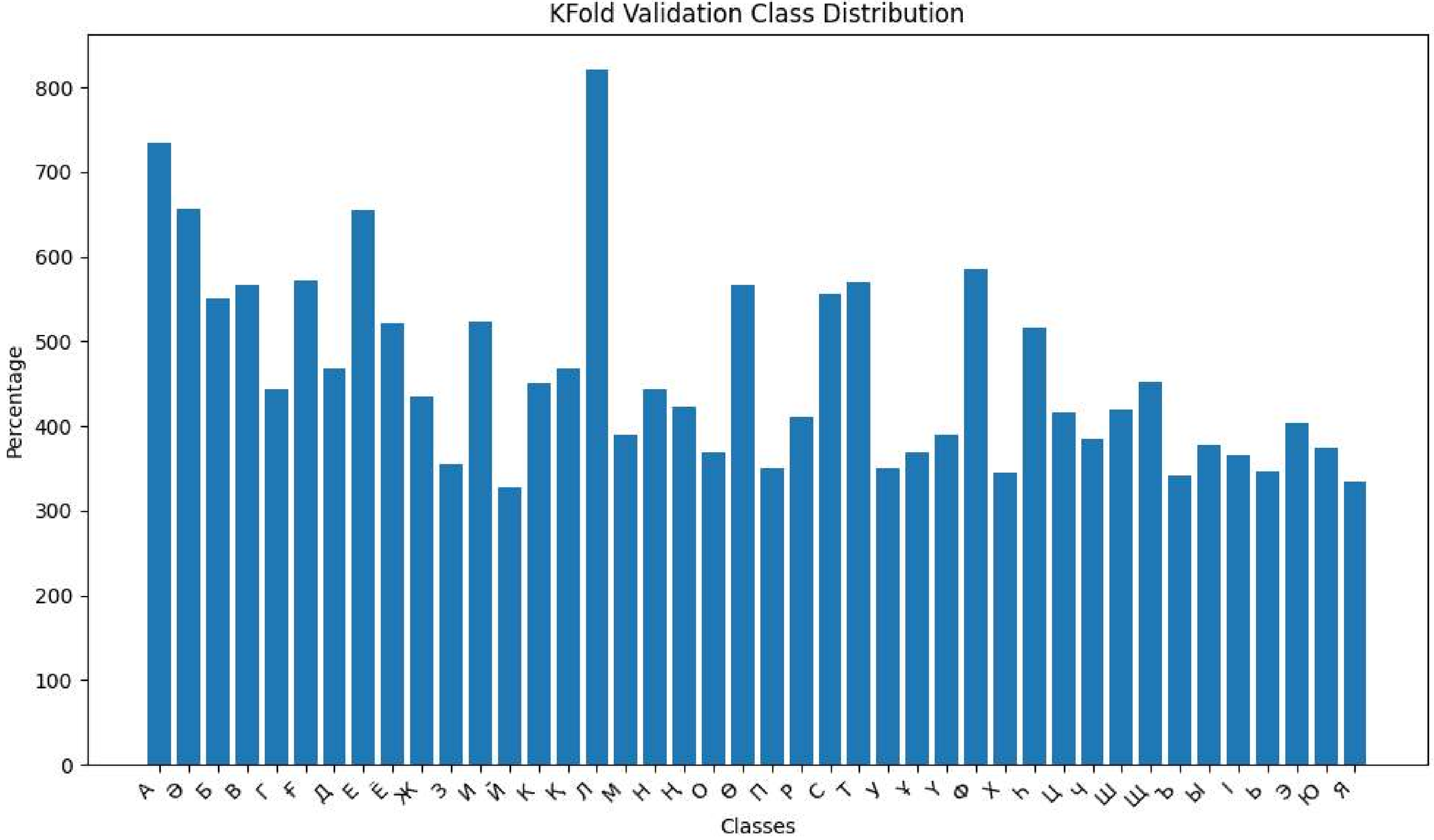

37_37_1827_2 01907300726 31.png
37_37_1832_2 01907300726 31.png
37_37_1833_2 01907300726 31.png
37_37_1841_2 01907300726 31.png
37_37_1846_2 01907300726 31.png
37_37_1848_2 01907300726 31.png
37_37_1853_2 01907300726 31.png
37_37_1857_2 01907300726 31.png
37_37_1859_2 01907300726 31.png
37_37_1868_2 01907300726 31.png
37_37_1874_2 01907300726 31.png
37_37_1875_2 01907300726 31.png
37_37_1881_2 01907300726 31.png
37_37_1886_2 01907300726 31.png
37_37_1887_2 01907300726 31.png
37_37_1888_2 01907300726 31.png
37_37_1893_2 01907300726 31.png
37_37_1898_2 01907300726 31.png
37_37_1900_2 01907300726 31.png
37_37_1901_2 01907300726 31.png
37_37_1902_2 01907300726 31.png
37_37_1906_2 01907300726 31.png
37_37_1909_2 01907300726 31.png
37_37_1913_2 01907300726 31.png
37_37_1917_2 01907300726 31.png
37_37_1918_2 01907300726 31.png
37_37_1921_2 01907300726 31.png
37_37_1924_2 01907300726 31.png
37_37_1926_2 01907300726 31.png
37_37_1931_2 01907300726 31.png
37_37_1934_2 01907300726 31.png
37_37_1936_2 01907300726 32.png
37_37_1961_2 01907300726 43.png
37_37_1962_2 01907300726 43.png
37_37_1985_2 01907300726 44.png
37_37_1986_2 01907300726 44.png
37_37_1992_2 01907300726 44.png
37_37_2011_2 01907300726 44.png
37_37_2012_2 01907300726 44.png
37_37_2013_2 01907300726 44.png
37_37_2033_2 01907300726 44.png
37_37_2069_2 01907300726 45.png
37_40_3_2019 0729095001. png
37_40_4_2019 0729095001. png
37_40_5_2019 0729095001. png
37_40_6_2019 0729095001. png
37_40_7_2019 0729095001. png
37_40_8_2019 0729095001. png
37_40_9_2019 0729095001. png
37_40_14_201 90729095001. png
37_40_15_201 90729095001. png
37_40_16_201 90729095001. png
37_40_17_201 90729095001. png
37_40_18_201 90729095001. png
37_40_19_201 90729095001. png
37_40_20_201 90729095001. png
37_40_22_201 90729095001. png
37_40_23_201 90729095001. png
37_40_24_201 90729095001. png
37_40_26_201 90729095001. png
37_40_27_201 90729095001. png
37_40_28_201 90729095001. png
37_40_29_201 90729095001. png
37_40_30_201 90729095001. png
37_40_31_201 90729095001. png
37_40_32_201 90729095001. png
37_40_33_201 90729095001. png
37_40_34_201 90729095001. png
37_40_35_201 90729095001. png
37_40_36_201 90729095001. png
37_40_37_201 90729095001. png
37_40_38_201 90729095001. png



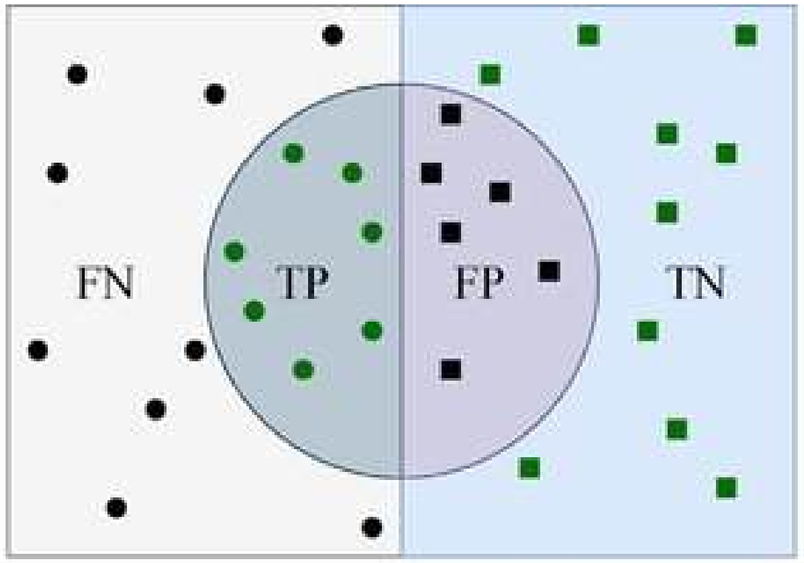
https://www.researchgate.net/figure/sual-representation-of-the-training-test-and-validation-split-using-cross-validation_fig3_338676025

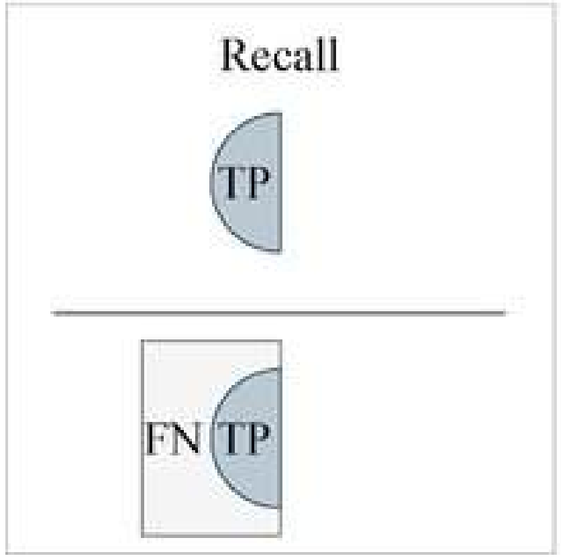

Original Class Distribution
Percentage
Classes
0
1000
2000
3000
4000
5000
А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я

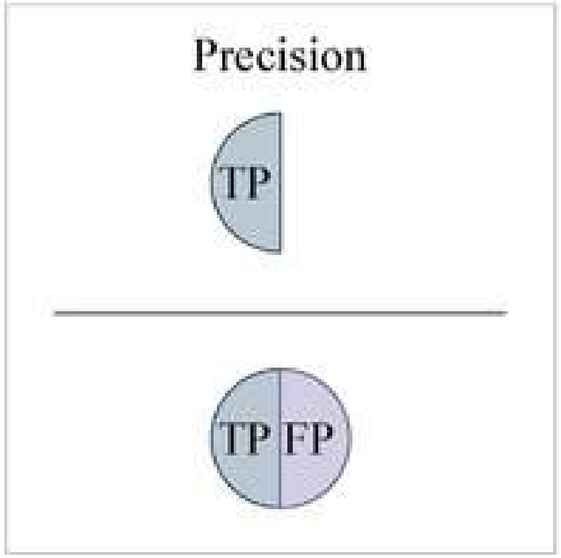

Train Class Distribution
Percentage
Classes
0
500
1000
1500
2000
2500
3000
3500
4000
А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я

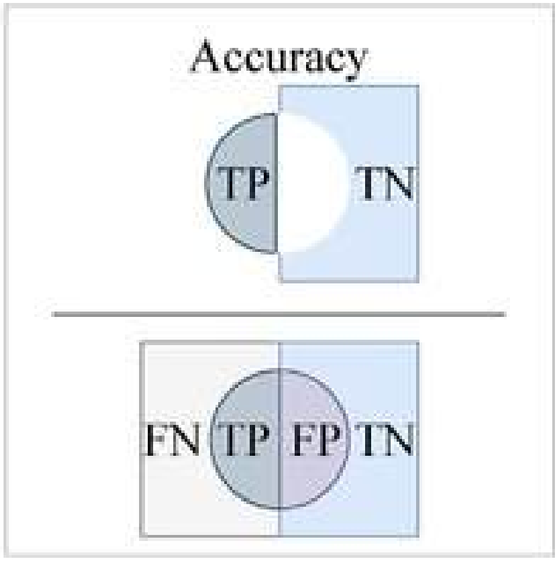

KFold Validation Class Distribution
Percentage
0
100
200
300
400
500
600
700
800
А О б в г ғ д е ё ж з и й к қ л м н ң о ө п р с т у ұ ү ә х һ ц ч ш щ ъ ы і ь э ю я
Classes

PRECISION & RECALL









F1-SCORE

Harmonic mean of the **precision** and **recall**

$$F_1 = \frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} = 2\frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

EVALUATION METRICS

Character Error Rate (CER):

$$CER = \frac{\text{Number of incorrect characters}}{\text{Total number of characters in the reference text}} \times 100\%$$

Word Error Rate (WER):

$$WER = \frac{\text{Number of incorrect words}}{\text{Total number of words in the reference text}} \times 100\%$$

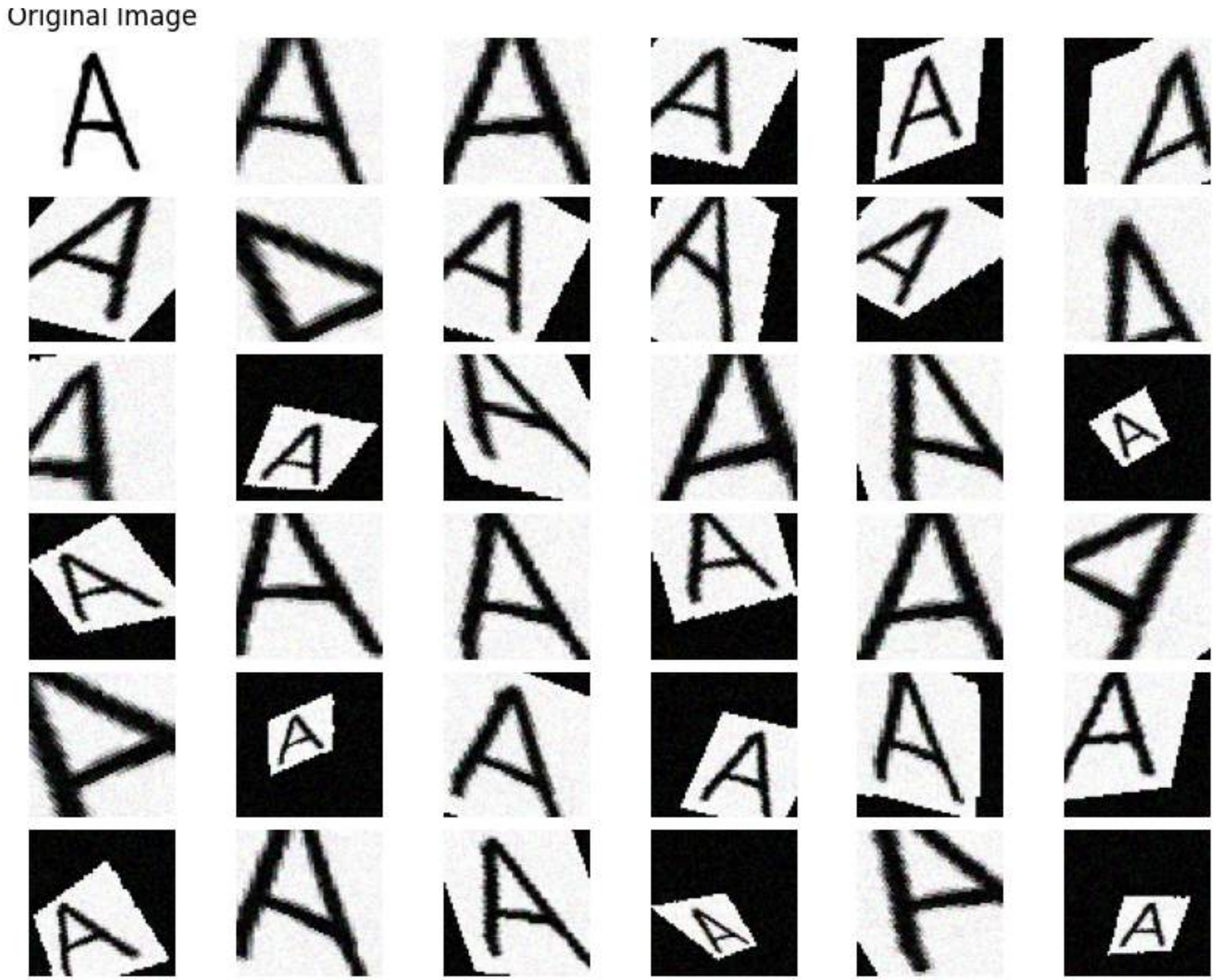
https://medium.com/@tam.tamanna18/deciphering-accuracy-evaluation-metrics-in-nlp-and-ocr-a-comparison-of-character-error-rate-cer-e97e809be0c8

2.1. Label Error Rate

In this paper, we are interested in the following error measure: given a test set $S' \subset \mathcal{D}_{\mathcal{X} \times \mathcal{Z}}$ disjoint from S, define the *label error rate* (LER) of a temporal classifier h as the normalised edit distance between its classifications and the targets on S', i.e.

$$LER(h, S') = \frac{1}{Z} \sum_{(\mathbf{x}, \mathbf{z}) \in S'} ED(h(\mathbf{x})) \qquad (1)$$

where Z is the total number of target labels in S', and $ED(\mathbf{p}, \mathbf{q})$ is the edit distance between the two sequences $\mathbf{p}$ and $\mathbf{q}$ — i.e. the minimum number of insertions, substitutions and deletions required to change $\mathbf{p}$ into $\mathbf{q}$.

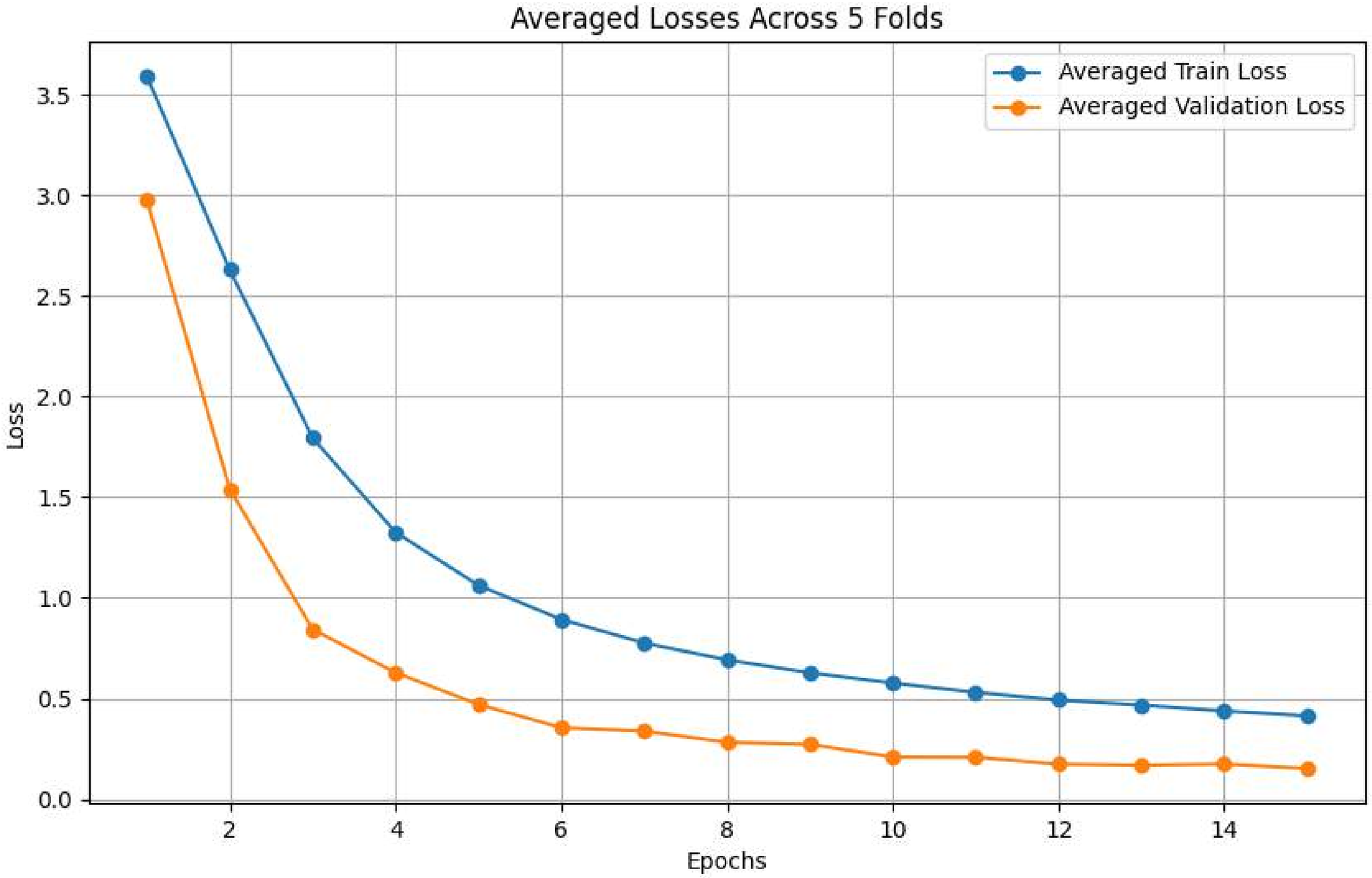


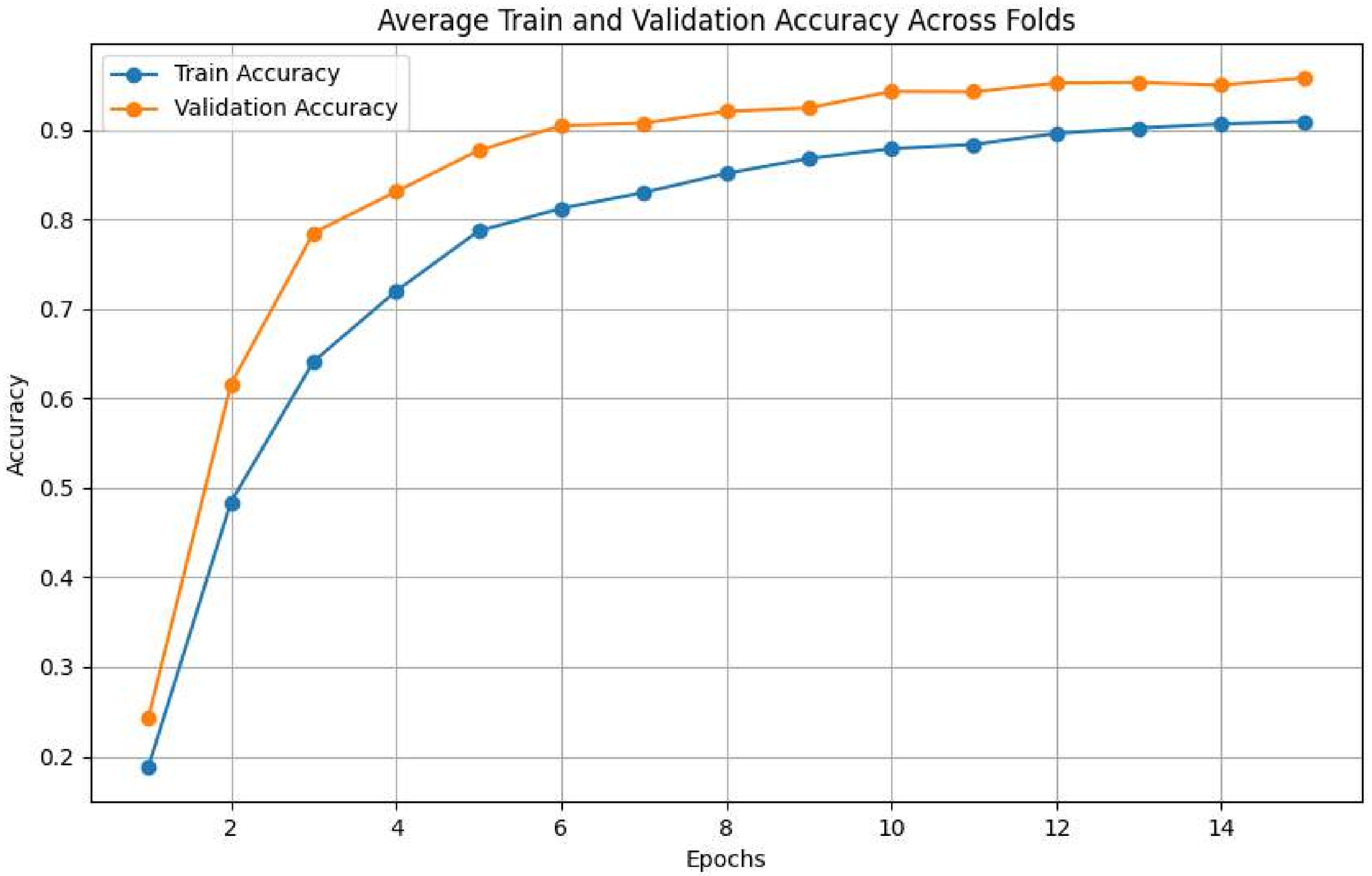
Data Augmentation

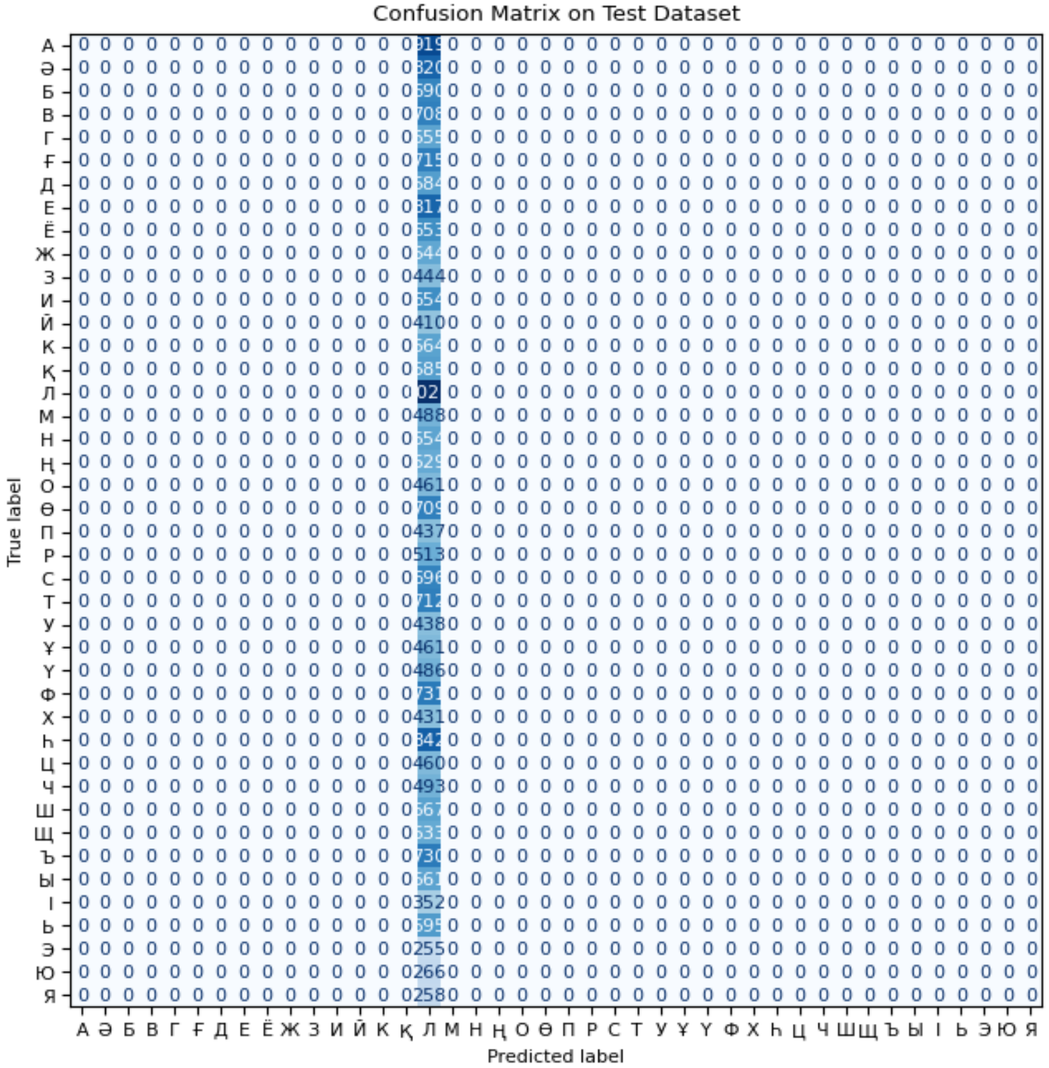
RandomAffine(rotate, translate, scale, shear)

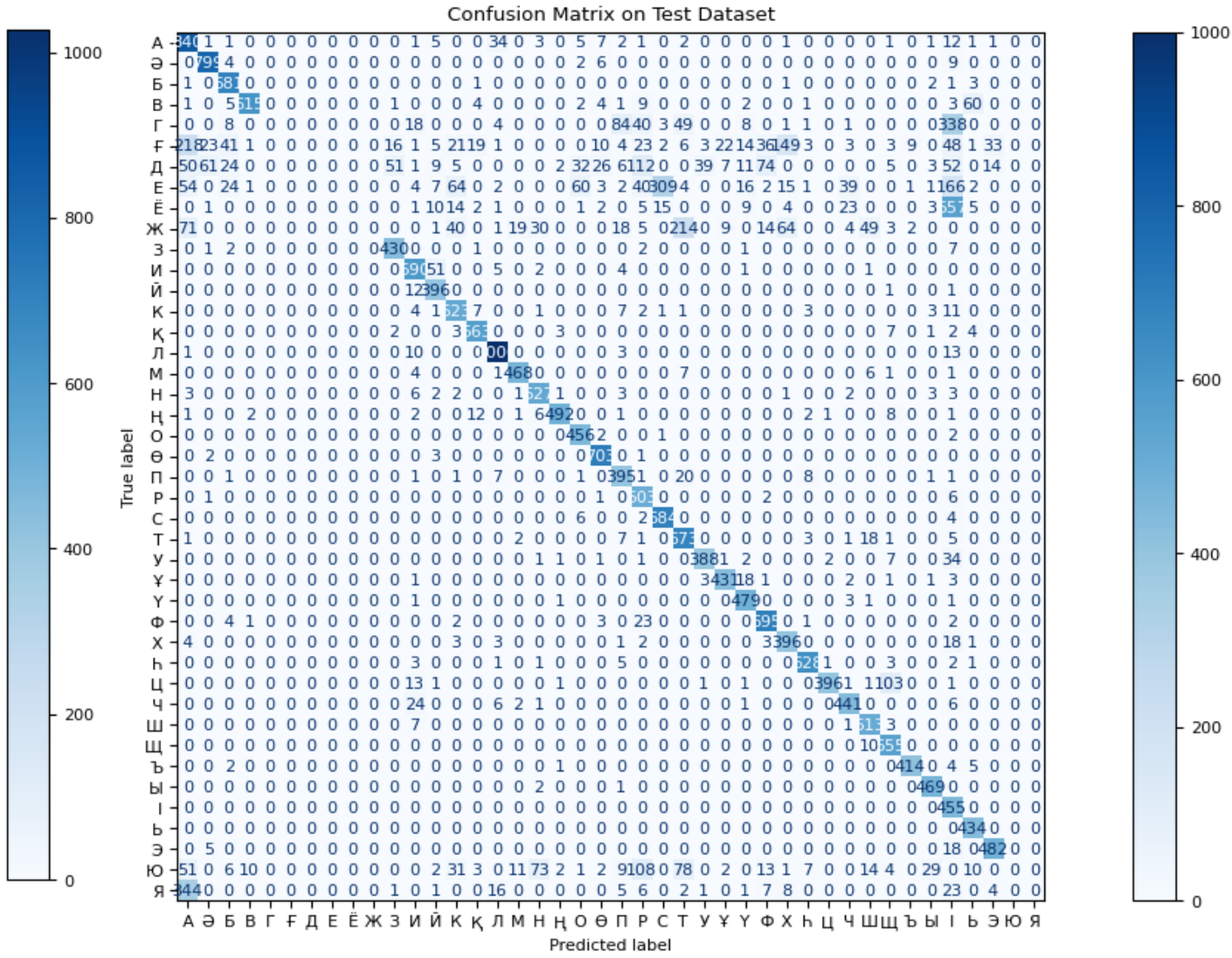
RandomPerspective

RandomNoise


Averaged Losses Across 5 Folds
Averaged Train Loss
Averaged Validation Loss
Loss
Epochs
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
2
4
6
8
10
12
14


Average Train and Validation Accuracy Across Folds
Train Accuracy
Validation Accuracy
Accuracy
Epochs
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
2
4
6
8
10
12
14





ImageFolder Nuance: 0, 1, 2, 3, 4 ... -> 0, 1, 10, 11, ... 2

Learning Duration: 4 hours 18

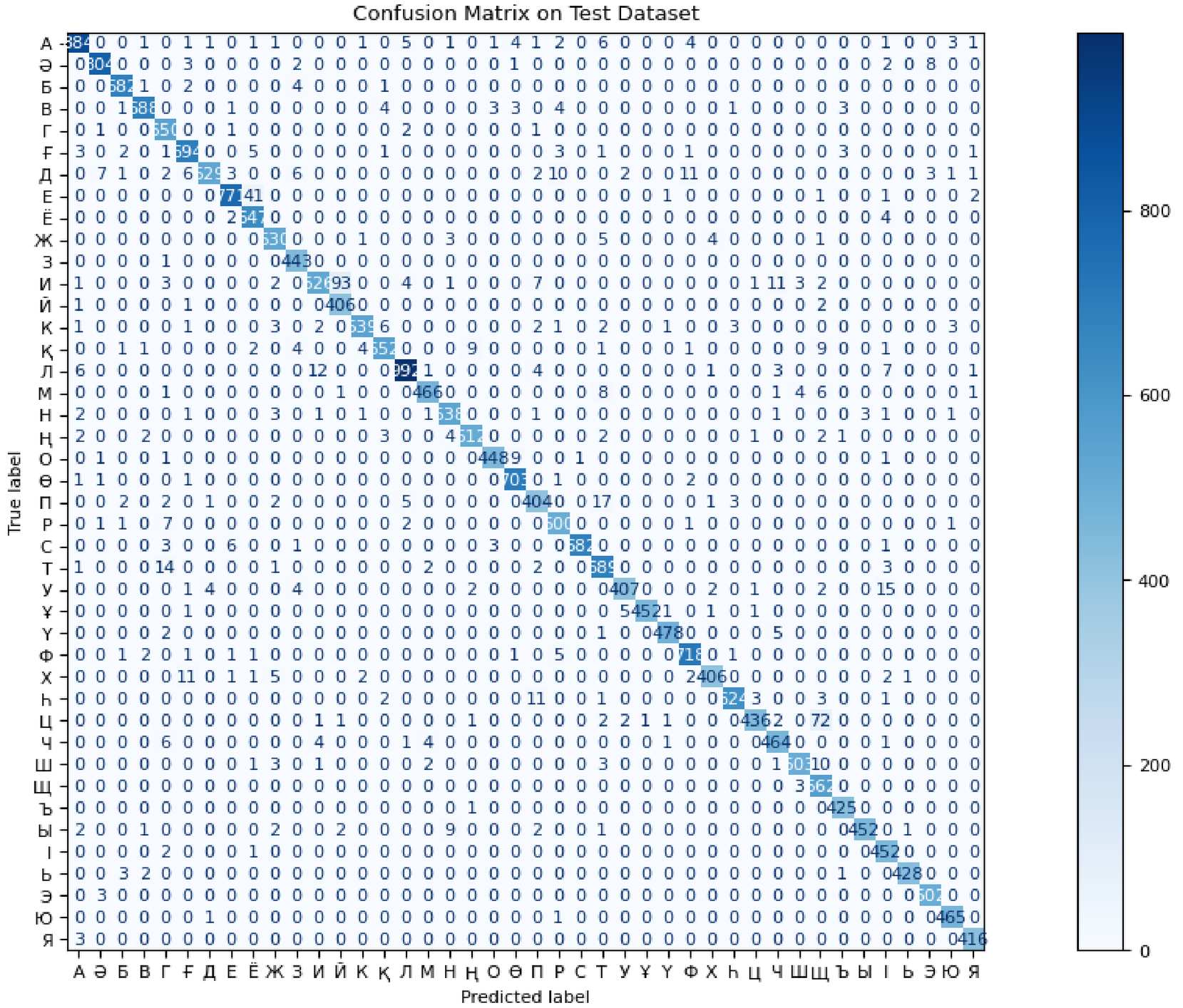
Best model is from fold 4 with

Validation F-Score: 0.9639

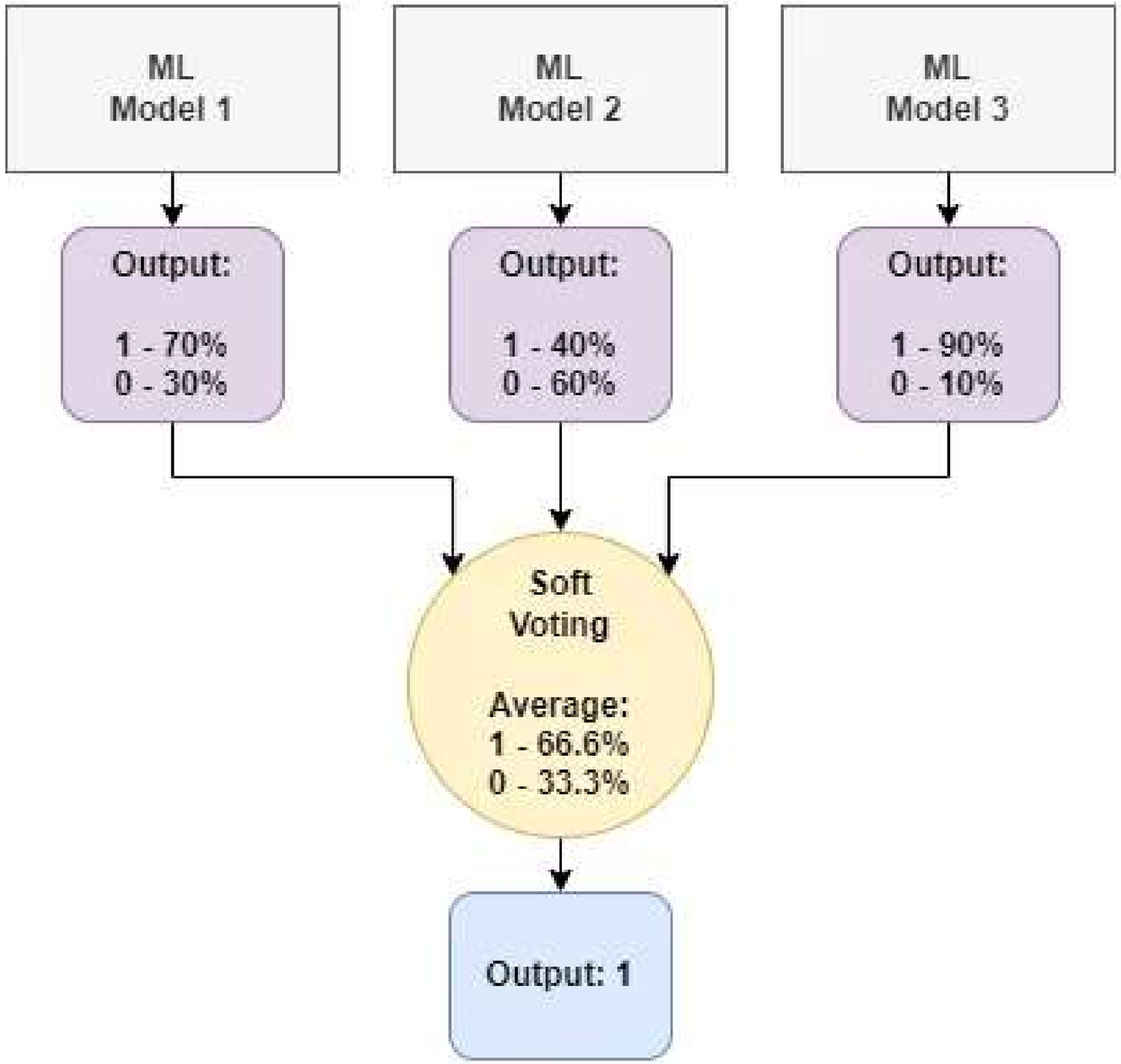
Test dataset: 24247

Test Accuracy: 96.38%

Test F1-Score: 96.33%



Voting



https://ilyasbinsalih.medium.com/what-is-hard-and-soft-voting-in-machine-learning-2652676b6a32

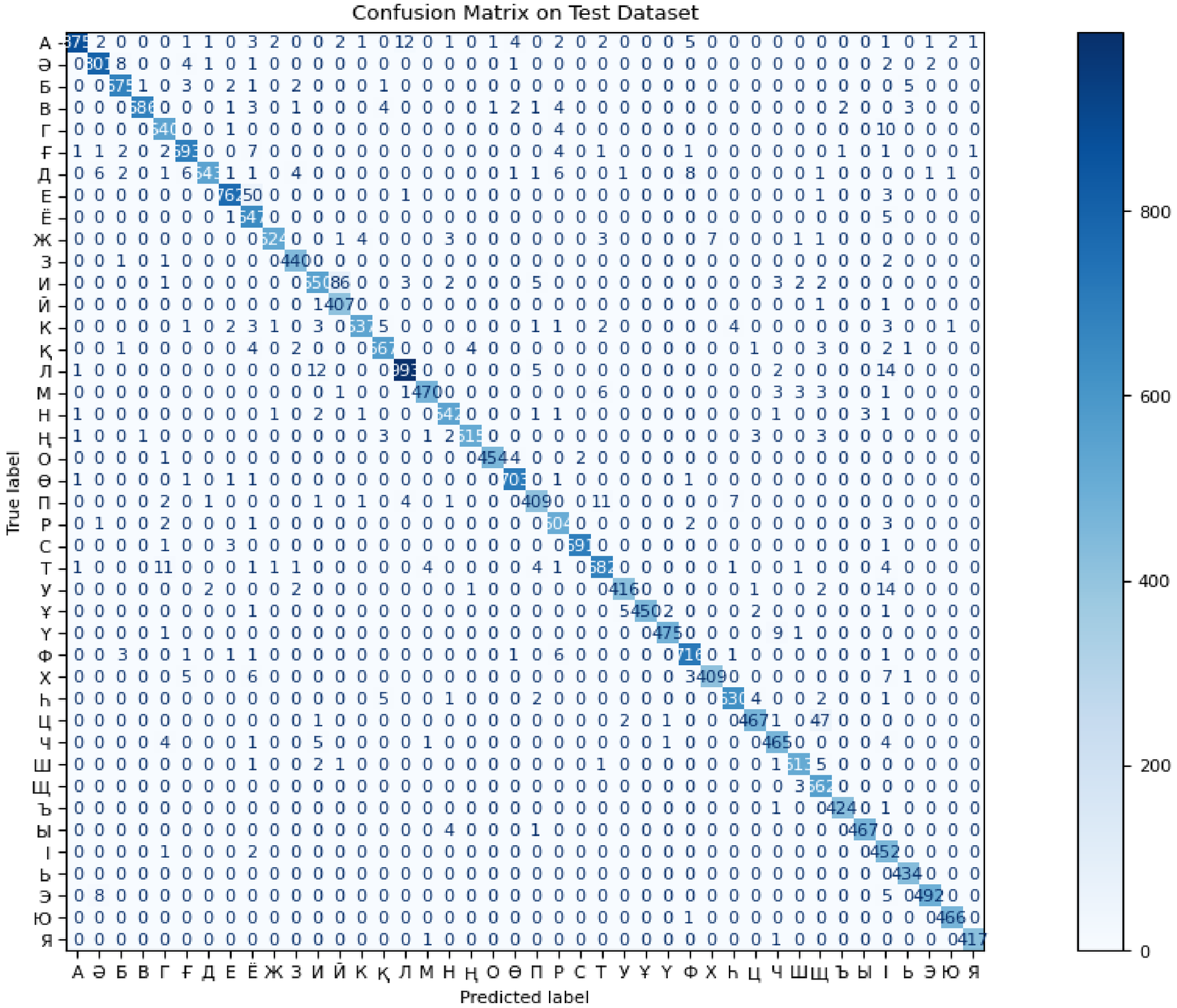
Fusion

Test dataset: 24247

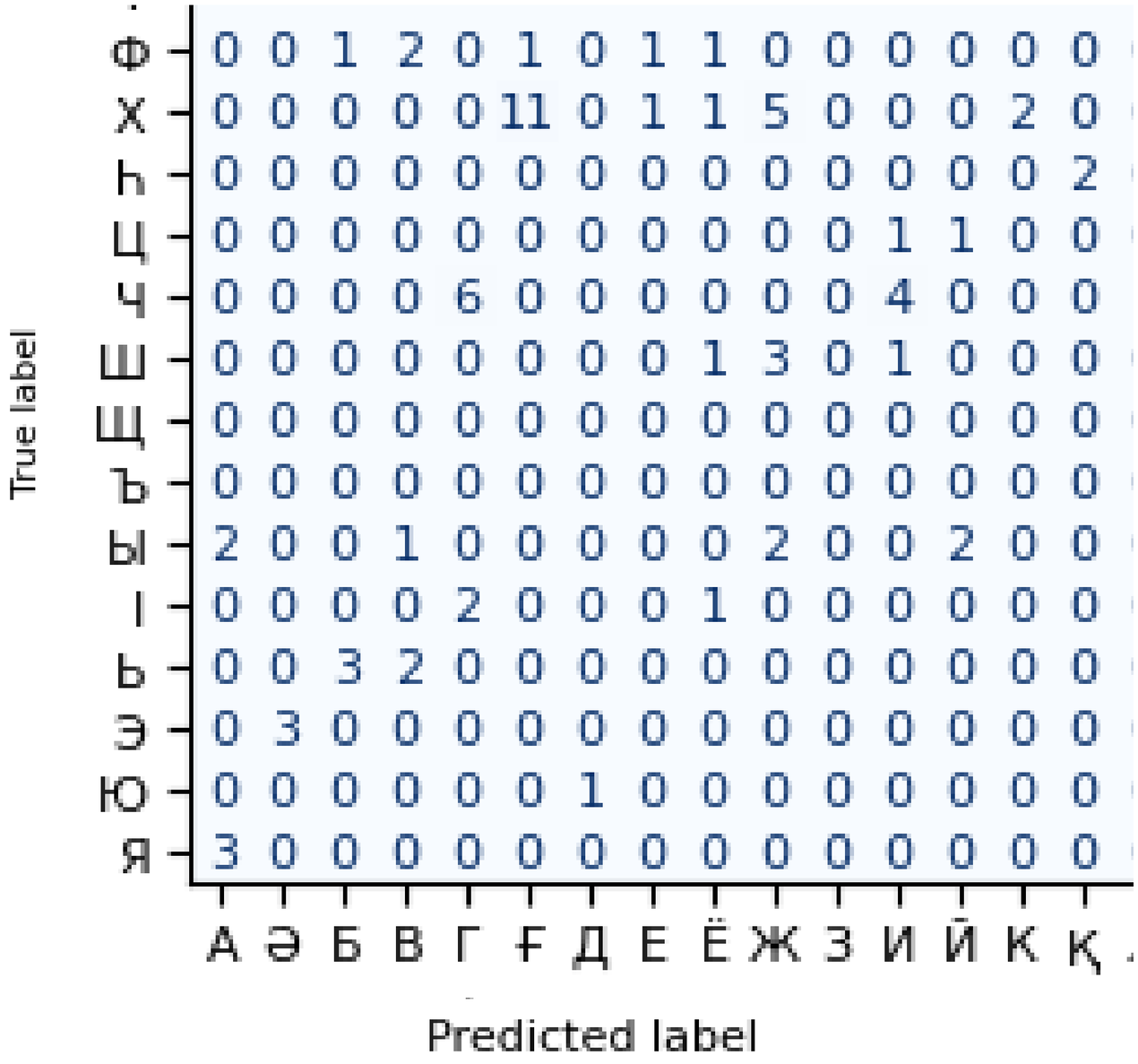
Test Accuracy: 96.38% -> 96.77%

F1-Score: 96.33% -> 96.78%

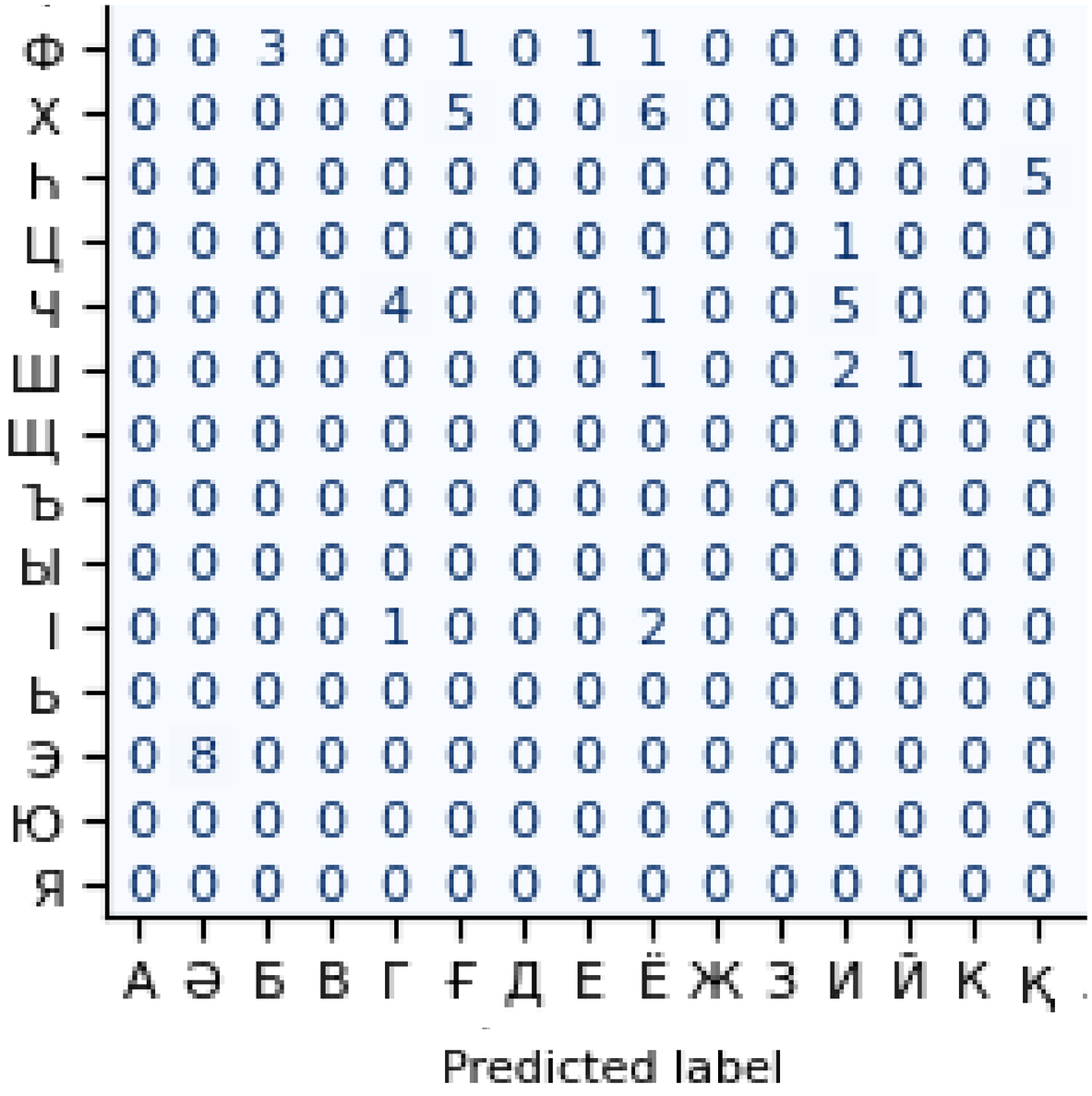
0.0039 * 24247 = 94 samples



Fold 4



Fusion

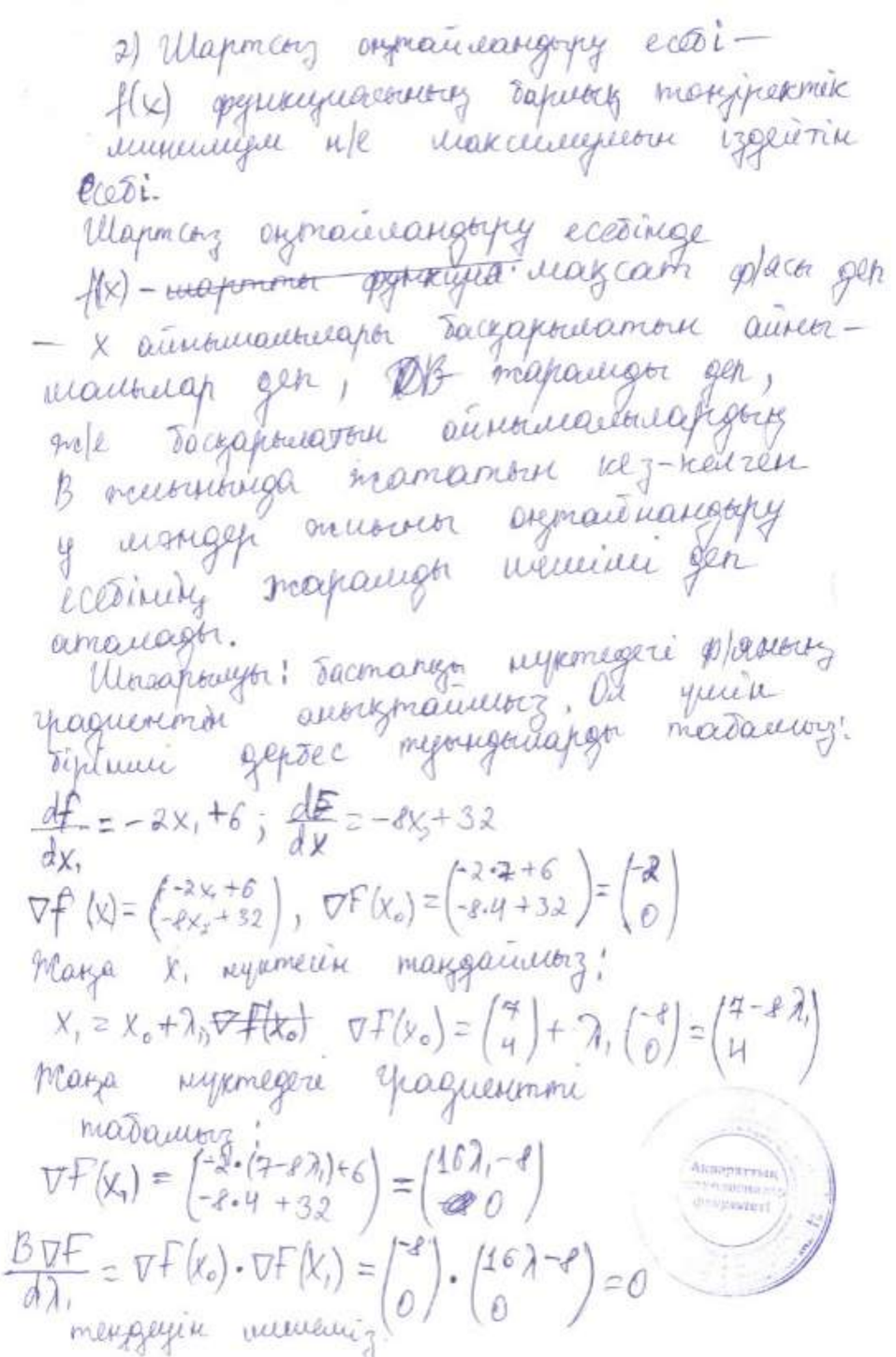


KOHTD: Kazakh offline handwritten text dataset

Toiganbayeva, N., Kasem, M., Abdimanap, G., Bostanbekov, K., Abdallah, A., Alimova, A., & Nurseitov, D. (2022). Kohtd: Kazakh offline handwritten text dataset. Signal Processing: Image Communication, 108, 116827.

Chicago

Github: https://github.com/abdoelsayed2016/KOHTD



Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks

Alex Graves[1] ALEX@IDSIA.CH
Santiago Fernández[1] SANTIAGO@IDSIA.CH
Faustino Gomez[1] TINO@IDSIA.CH
Jürgen Schmidhuber[1,2] JUERGEN@IDSIA.CH

[1] Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, 6928 Manno-Lugano, Switzerland
[2] Technische Universität München (TUM), Boltzmannstr. 3, 85748 Garching, Munich, Germany

Abstract

Many real-world sequence learning tasks require the prediction of sequences of labels from noisy, unsegmented input data. In speech recognition, for example, an acoustic signal is transcribed into words or sub-word units. Recurrent neural networks (RNNs) are powerful sequence learners that would seem well suited to such tasks. However, because they require pre-segmented training data, and post-processing to transform their outputs into label sequences, their applicability has so far been limited. This paper presents a novel method for training RNNs to label unsegmented sequences directly, thereby solving both problems. An experiment on the TIMIT speech corpus demonstrates its advantages over both a baseline HMM and a hybrid HMM-RNN.

1. Introduction

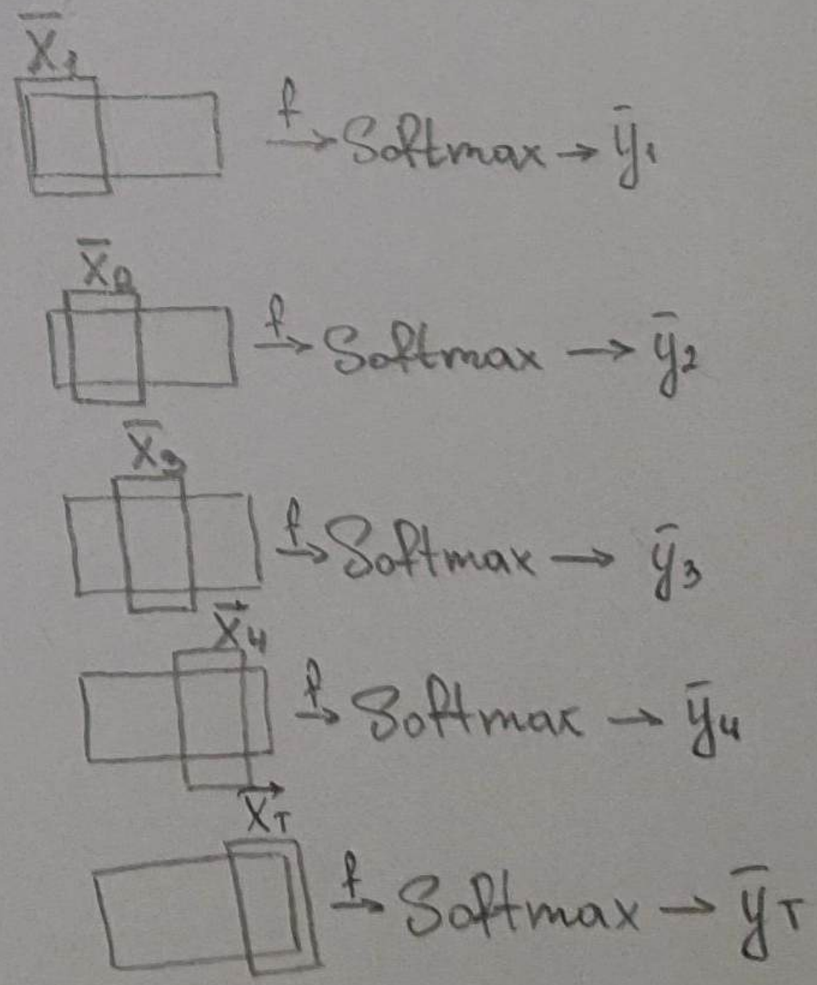
Labelling unsegmented sequence data is a ubiquitous problem in real-world sequence learning. It is particularly common in perceptual tasks (e.g. handwriting recognition, speech recognition, gesture recognition) where noisy, real-valued input streams are annotated

belling. While these approaches have proved successful for many problems, they have several drawbacks: (1) they usually require a significant amount of task specific knowledge, e.g. to design the state models for HMMs, or choose the input features for CRFs; (2) they require explicit (and often questionable) dependency assumptions to make inference tractable, e.g. the assumption that observations are independent for HMMs; (3) for standard HMMs, training is generative, even though sequence labelling is discriminative.

Recurrent neural networks (RNNs), on the other hand, require no prior knowledge of the data, beyond the choice of input and output representation. They can be trained discriminatively, and their internal state provides a powerful, general mechanism for modelling time series. In addition, they tend to be robust to temporal and spatial noise.

So far, however, it has not been possible to apply RNNs directly to sequence labelling. The problem is that the standard neural network objective functions are defined separately for each point in the training sequence; in other words, RNNs can only be trained to make a series of independent label classifications. This means that the training data must be pre-segmented, and that the network outputs must be post-processed to give the final label sequence.

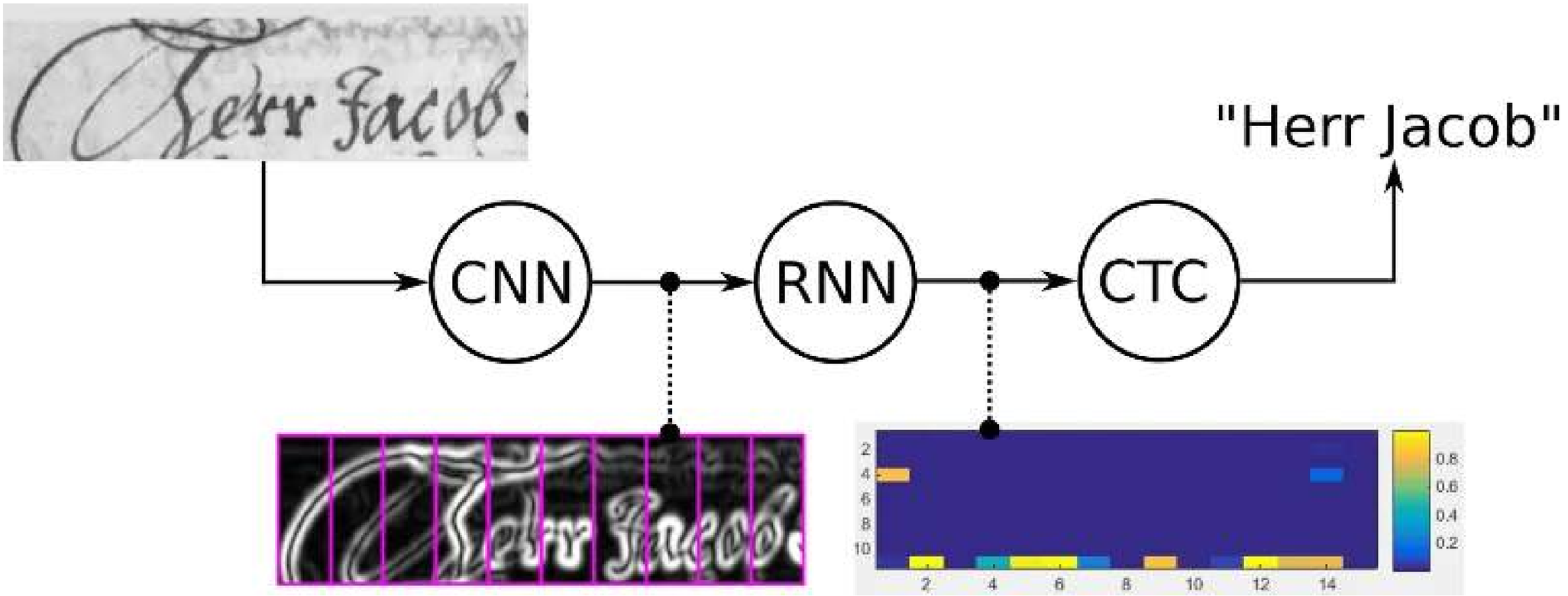
At present, the most effective use of RNNs for se-



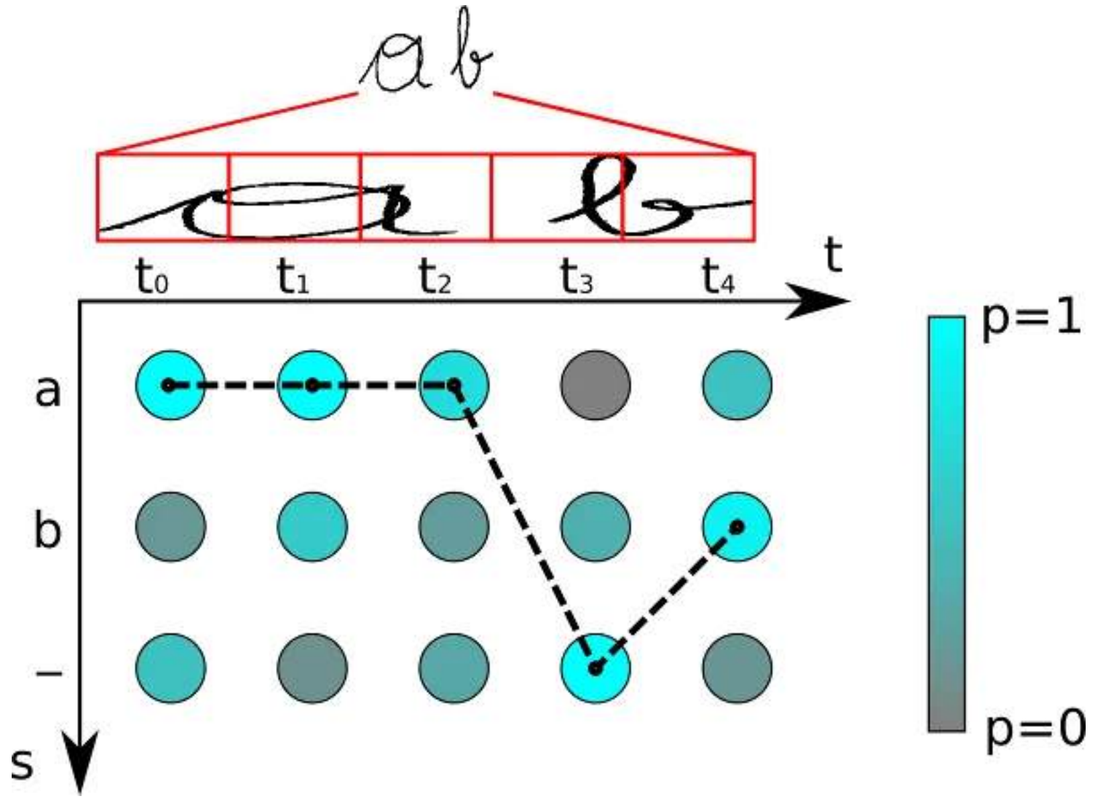
T - кол-во шагов

-WW--OOO-R-DDDD-

Z = WORD



https://towardsdatascience.com/intuitively-understanding-connectionist-temporal-classification-3797e43a86c



$$p(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} p(\pi|\mathbf{x}).$$

$$h(\mathbf{x}) = \arg\max_{\mathbf{l} \in L^{\leq T}} p(\mathbf{l}|\mathbf{x}).$$

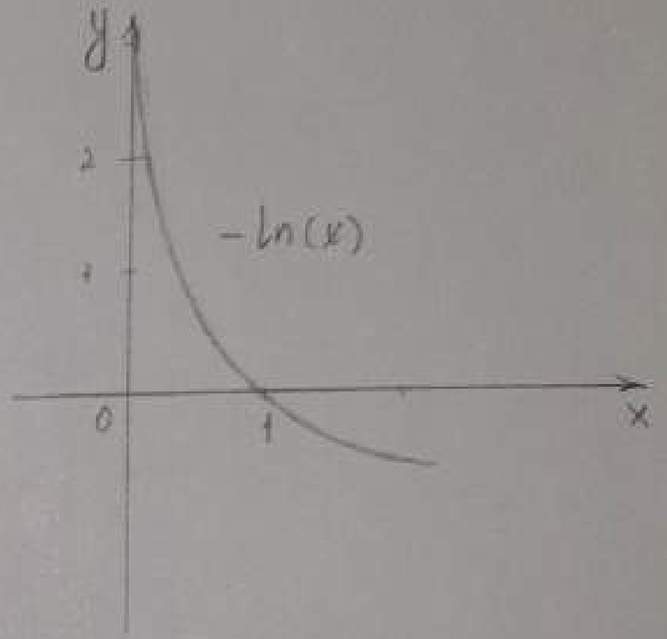
Image from: https://towardsdatascience.com/intuitively-understanding-connectionist-temporal-classification-3797e43a86c

Loss function

$$O^{ML}(S, N_w) = -\sum_{(x,z)\in S} \ln(p(z|x))$$

$$P(S, N_w) = \prod_{(x,z)\in S} p(z|x) \to \max$$

$$-\ln(P(S, N_w)) = -\sum_{(x,z)\in S} \ln(p(z|x)) \to \min$$



O - objective function

ML - maximum likelihood

S - dataset

N_w - model

x - input sequence.

Z - target sequence.

$$\frac{dO^{ML}(\{(x,z)\}, N_w)}{dy_k^t} = \frac{-d\ln(p(z|x))}{dy_k^t}$$

partial derivative by some y_k^t

Forward Backward Algorithm

$$\alpha_t(s) \stackrel{\text{def}}{=} \sum_{\substack{\pi \in N^T : \\ \mathcal{B}(\pi_{1:t}) = \mathbf{l}_{1:s}}} \prod_{t'=1}^{t} y^{t'}_{\pi_{t'}}.$$

$$\alpha_1(1) = y^1_b$$
$$\alpha_1(2) = y^1_{\mathbf{l}_1}$$
$$\alpha_1(s) = 0, \ \forall s > 2$$

$$\beta_t(s) \stackrel{\text{def}}{=} \sum_{\substack{\pi \in N^T : \\ \mathcal{B}(\pi_{t:T}) = \mathbf{l}_{s:|\mathbf{l}|}}} \prod_{t'=t}^{T} y^{t'}_{\pi_{t'}}$$

$$\beta_T(|\mathbf{l}'|) = y^T_b$$
$$\beta_T(|\mathbf{l}'| - 1) = y^T_{\mathbf{l}_{|\mathbf{l}|}}$$
$$\beta_T(s) = 0, \ \forall s < |\mathbf{l}'| - 1$$

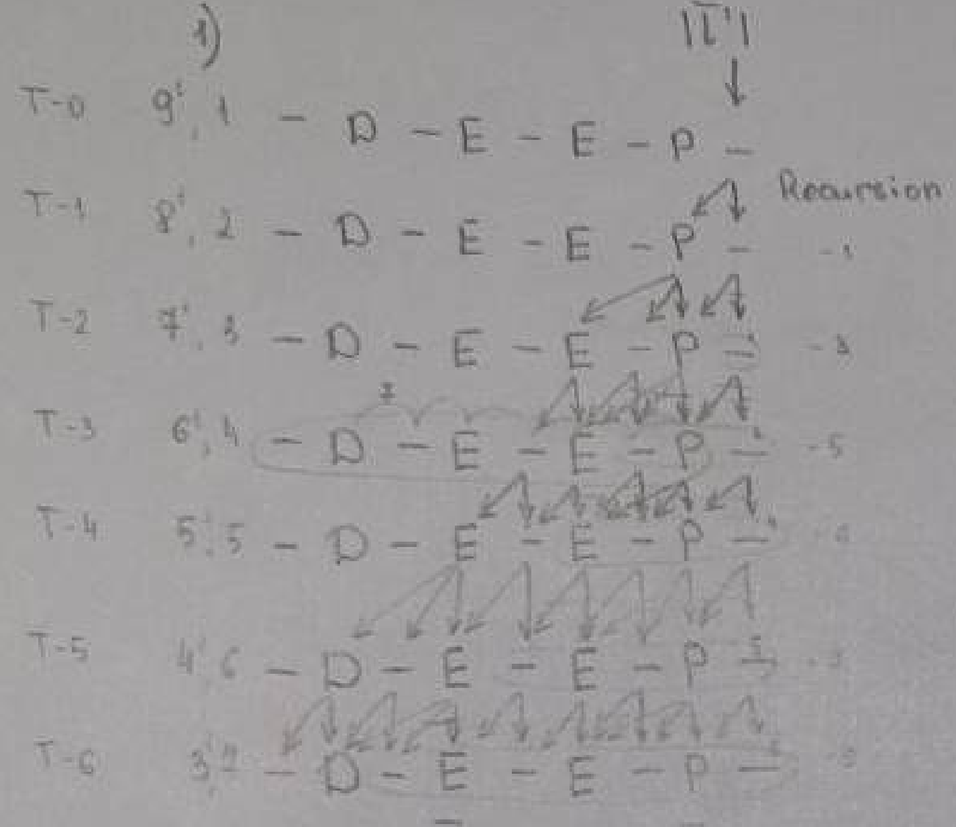
Image ▭ n×m

Forward - Backward Algorithm

$$p(\bar{l} \mid x) = d_T(1, |\bar{l}'|) + d_T(1, |\bar{l}'|-1)$$

(1) (2)

$$d_t(s) = \begin{cases} y^t_{l'_s} \cdot (d_{t-1}(s) + d_{t-1}(s-1)) & \text{if } l'_s = b \text{ or } l'_s = l'_{s-2} \\ y^t_{l'_s} \cdot (d_{t-1}(s) + d_{t-1}(s-1) + d_{t-1}(s-2)) & \text{otherwise} \end{cases}$$

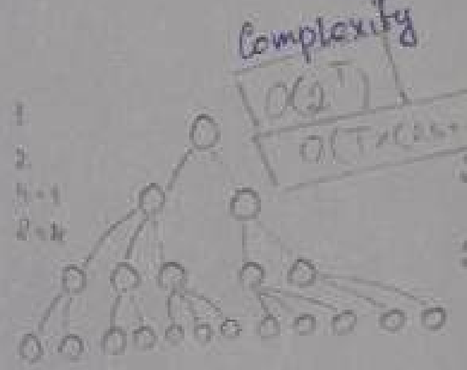


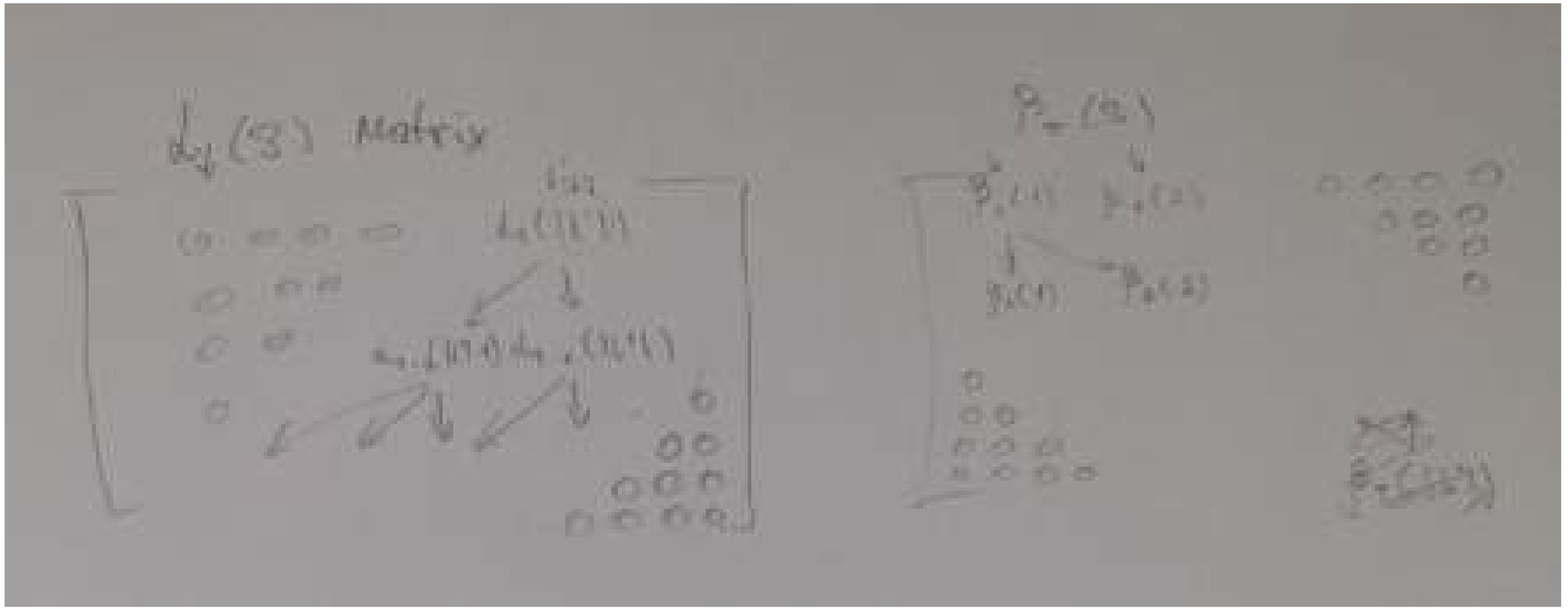
$$d_T(1, |\bar{l}'|) = y^T_b \cdot (d_{T-1}(1, |\bar{l}'|) + d_{T-1}(1, |\bar{l}'|-1))$$

(1) (2)

1) $d_{T-1}(1, |\bar{l}'|) = y^{T-1}_{l_{1:T}}(d_{T-2}(1, |l'|) + d_{T-2}(1, |\bar{l}'|-1))$ — sym. error

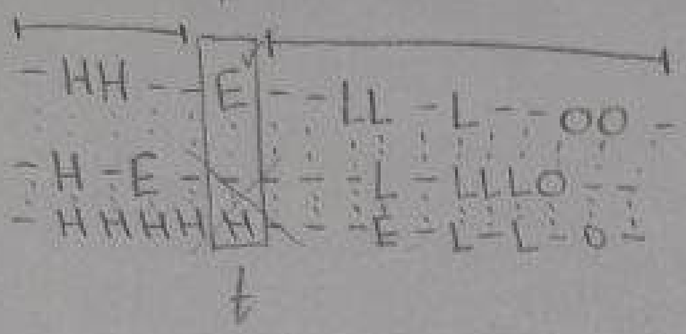
2) $d_{T-1}(1, |\bar{l}'|-1) = y^{T-1}_{l_{1:T-1}}(d_{T-2}(1, |l'|-1) + d_{T-2}(1, |l'|-2) + d_{T-2}(1, |\bar{l}'|-3))$





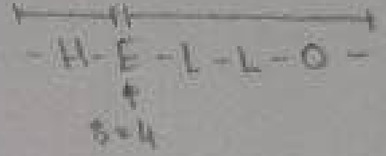
Forward - Backward Algorithm

$$\alpha_t(s) \cdot \beta_t(s) = \sum_{\pi \in B^{-1}(l): \pi_t = l'_s} y^t_{l'_s} \cdot \prod_{t'=1}^{T} y^{t'}_{\pi_{t'}}$$



$t = 6$

$l'_s = "E"$



$\alpha_t(s)$ - вероятността да t угадаме до s , по всяка $l'_{1:s}$

$\beta_t(s)$ - вероятността да t угадаме от s до края , по всяка $l'_{s:|l'|}$

$$\alpha_t(s) \cdot \beta_t(s) = \sum_{\substack{\pi \in B^{-1}(l): \\ \pi_t = l'_s}} \overbrace{y^t_{l'_s}}^{\text{const}} \cdot \underbrace{\prod_{t'=1}^{T} y^{t'}_{\pi_{t'}}}_{p(\pi|x)}$$

$t = \text{const}$, $s = \text{const}$

$\alpha_1(3) = 0$

$\alpha_2(5) = 0$

$\dots$

$$p(\pi|x) = \prod_{t=1}^{T} y^t_{\pi_t}, \quad \forall \pi \in L'^T, \quad L'^T \in L'^*$$

$$\alpha_t(s) \cdot \beta_t(s) \cdot \frac{1}{y^t_{l'_s}} = \sum_{\substack{\pi \in B^{-1}(l): \\ \pi_t = l'_s}} p(\pi|x)$$

1) $$p(l|x) = \sum_{\pi \in B^{-1}(l)} p(\pi|x) =$$, но тут t нет и нам будто по всем t

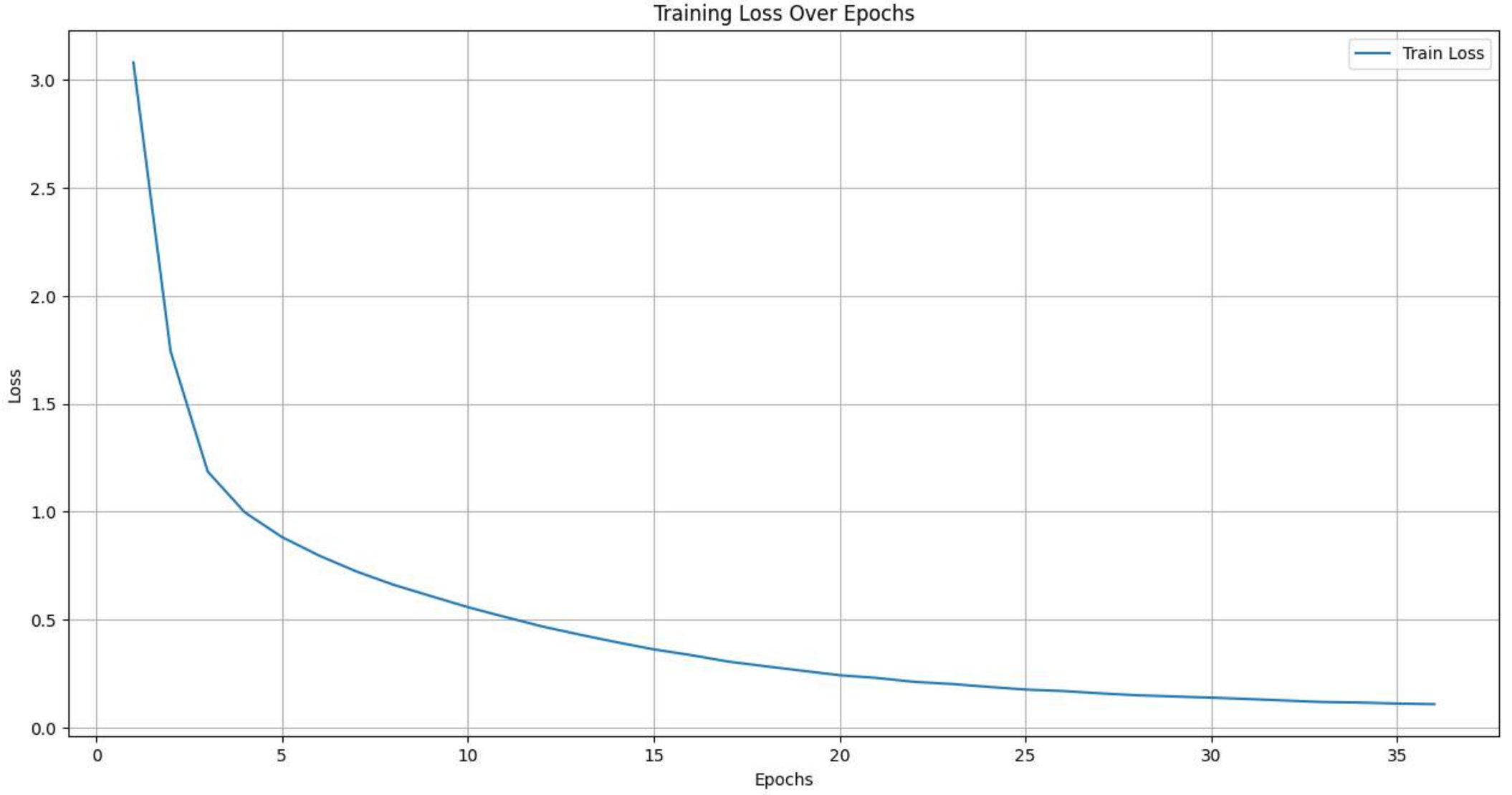
2) $$p(l|x)_t = \sum_{s=1}^{|l'|} \frac{1}{y^t_{l'_s}} \cdot \alpha_t(s) \cdot \beta_t(s)$$, но тут $t = \text{const}$, нам кажется $1) \neq 2)$

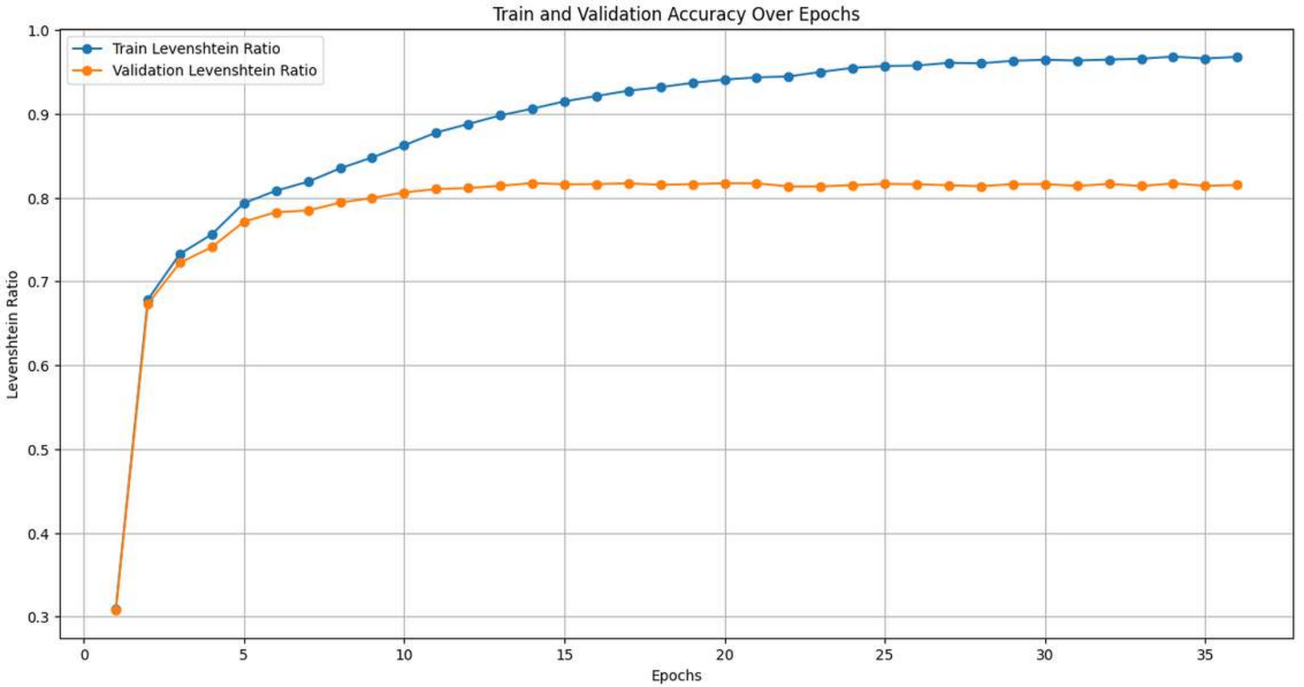
← полный случай $p(l|x)$

$$\sum_{s=1}^{|l'|} \sum_{\substack{\pi \in B^{-1}(l): \\ \pi_t = l'_s}} p(\pi|x), \quad t$$

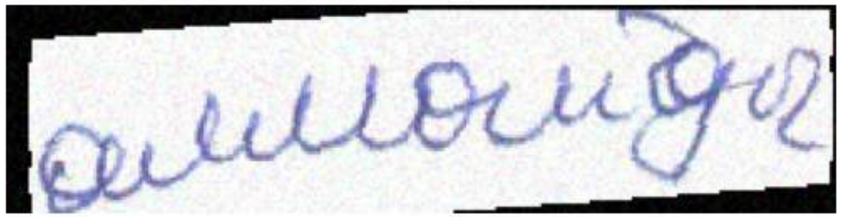
$$E = O^{ML}(S, X_\omega) = -\sum_{(x,z) \in S, \; l=z} \ln(p(l|x))$$

Итак, t в голове мы не сможем посчитать

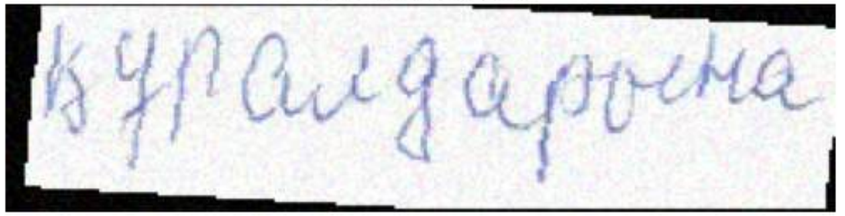

Training Loss Over Epochs
Train Loss
Loss
Epochs
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
5
10
15
20
25
30
35


Train and Validation Accuracy Over Epochs
Train Levenshtein Ratio
Validation Levenshtein Ratio
Levenshtein Ratio
Epochs
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
5
10
15
20
25
30
35

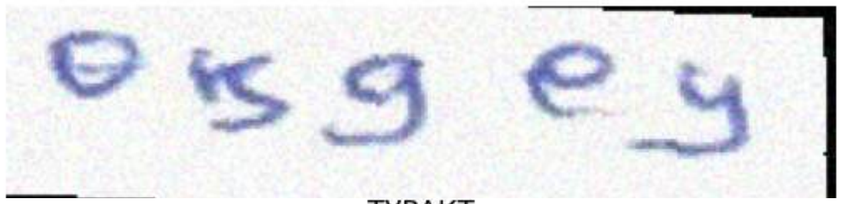
АЛМАЙДЫ



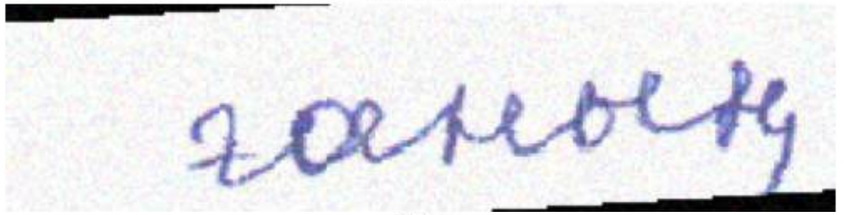
ҚҰРАЛДАРЫНА



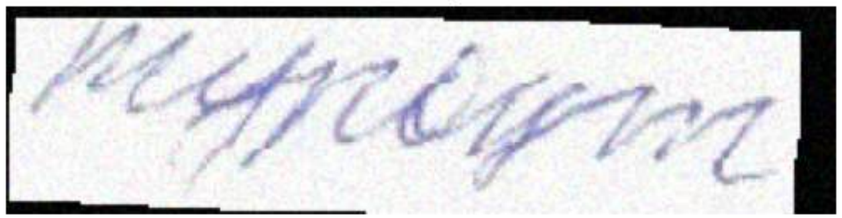
ӨҢДЕУ



ҒАНЫҢ

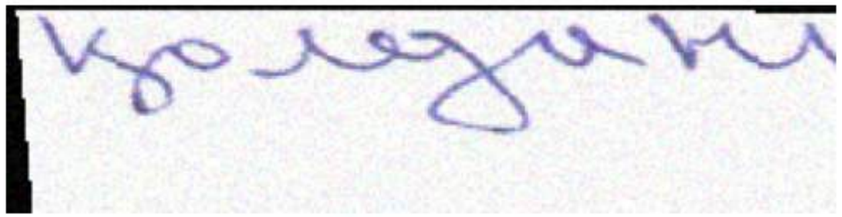


ТУРАҚТ

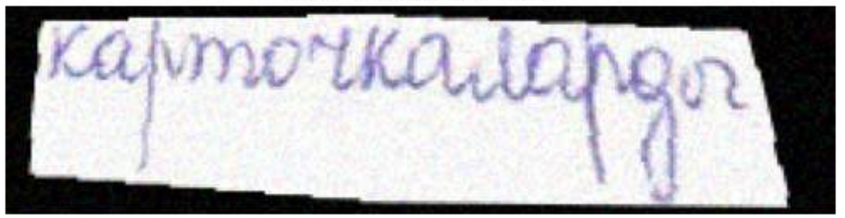


АҚЫ

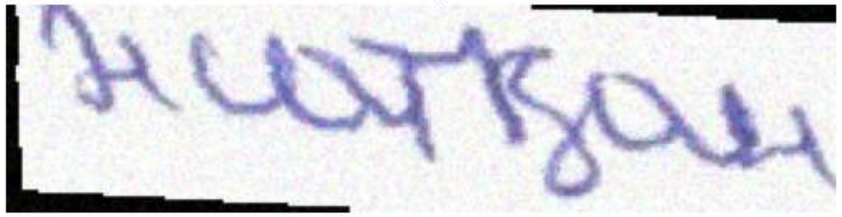
ҚОЛДАНЫС



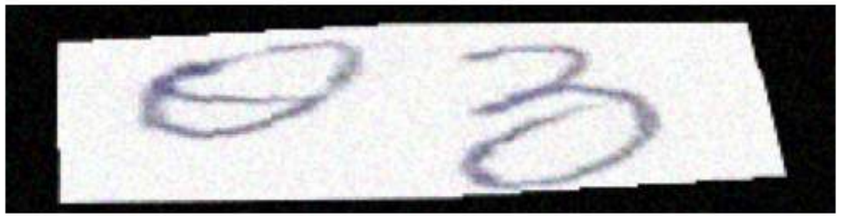
КАРТОЧКАЛАРДЫ

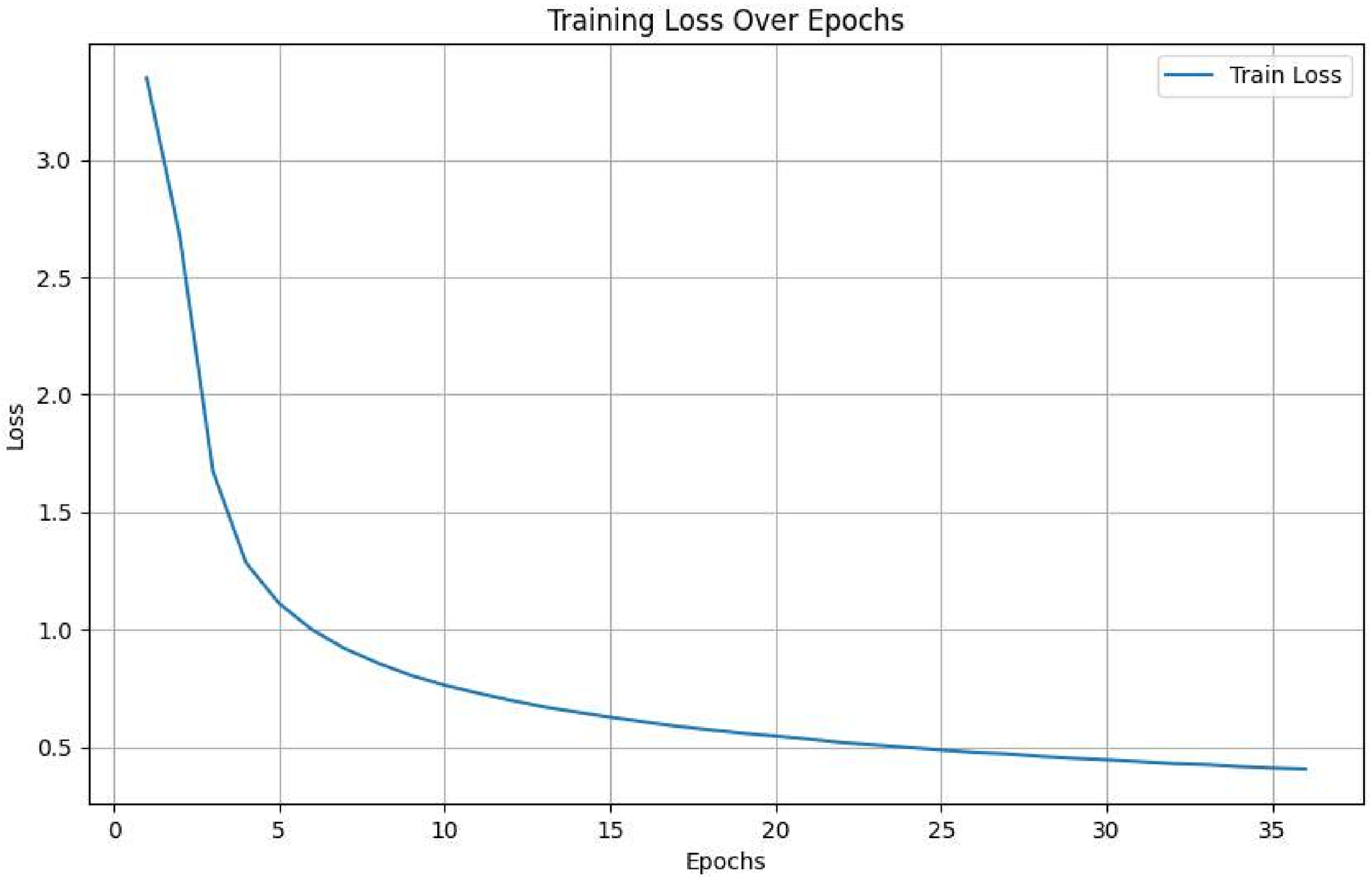


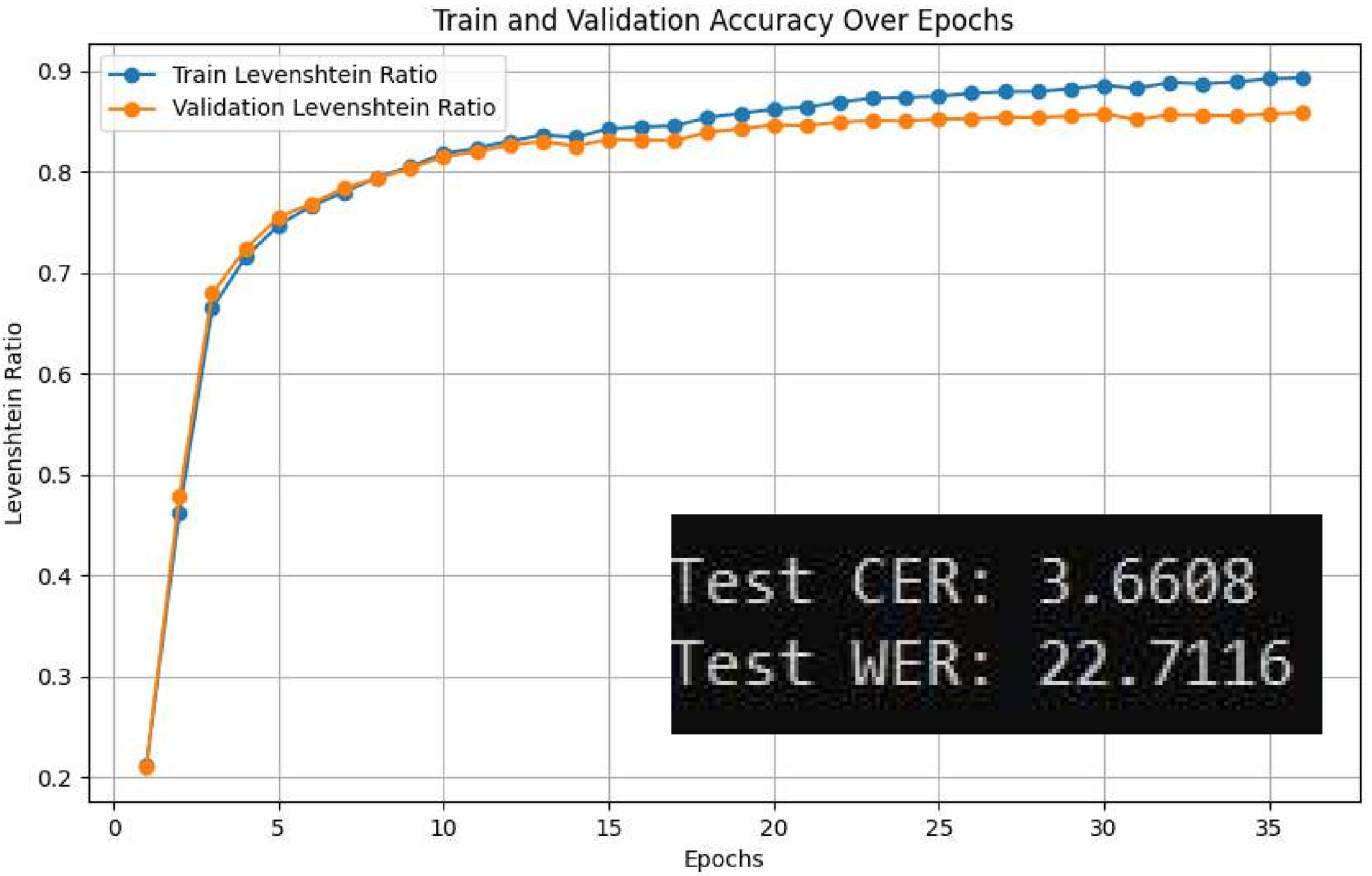
ЖАТҚАН



ӨЗ




Training Loss Over Epochs
Train Loss
Loss
Epochs
3.0
2.5
2.0
1.5
1.0
0.5
0
5
10
15
20
25
30
35


Train and Validation Accuracy Over Epochs
Train Levenshtein Ratio
Validation Levenshtein Ratio
Levenshtein Ratio
Epochs
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
5
10
15
20
25
30
35
Test CER: 3.6608
Test WER: 22.7116

GT: АЛМАТЫ
Predicted: АЛМАТЫ
Avg Conf: 0.9713

GT: ӨНДІРУДІҢ
Predicted: ӨНДІРУДІҢ
Avg Conf: 0.9969

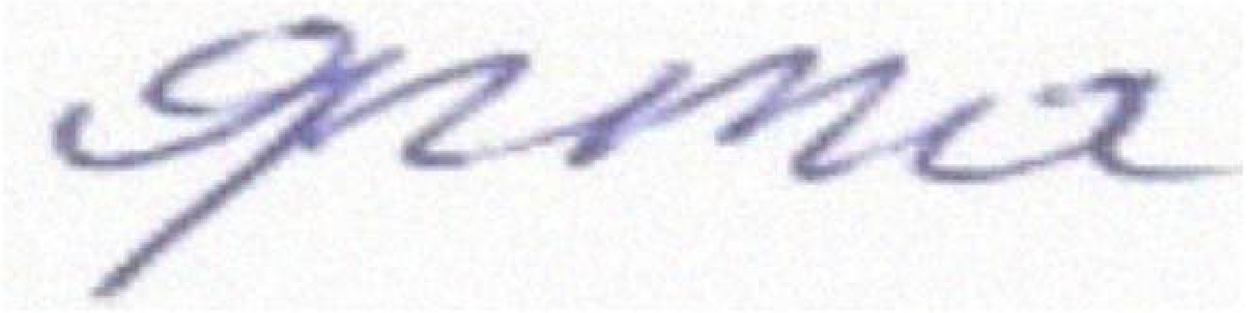
GT: НЕМЕСЕ
Predicted: НЕМСЕ
Avg Conf: 0.9251

GT: ОСЫ
Predicted: ОСЫ
Avg Conf: 0.9167

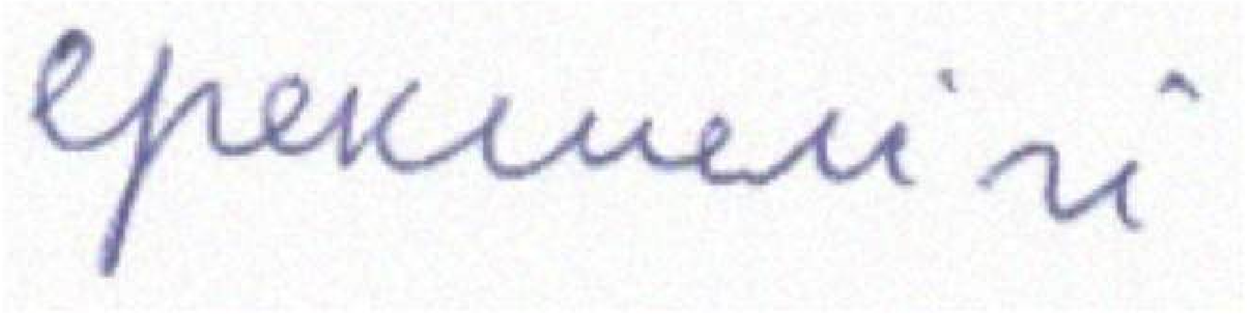
GT: АДАМДАР
Predicted: АДАМДАР
Avg Conf: 0.9403

GT: ЖҮЙ
Predicted: ЖҮБ
Avg Conf: 0.9958

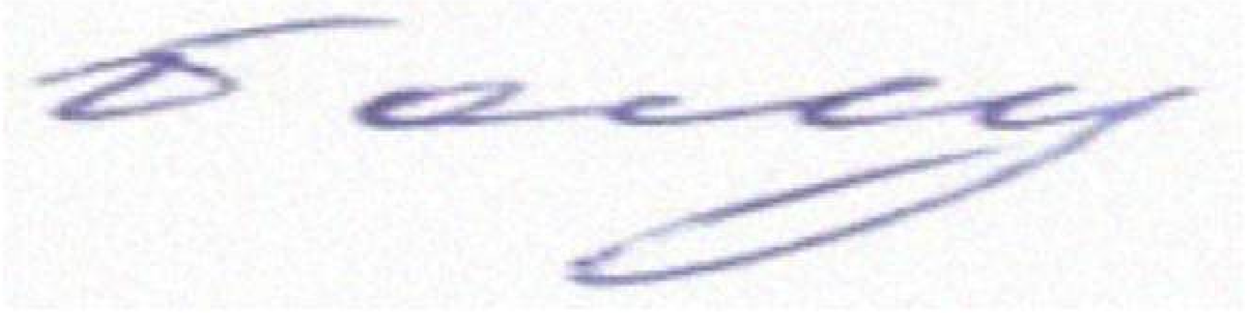
GT: ОРТА
Predicted: ДРТА
Avg Conf: 0.8590



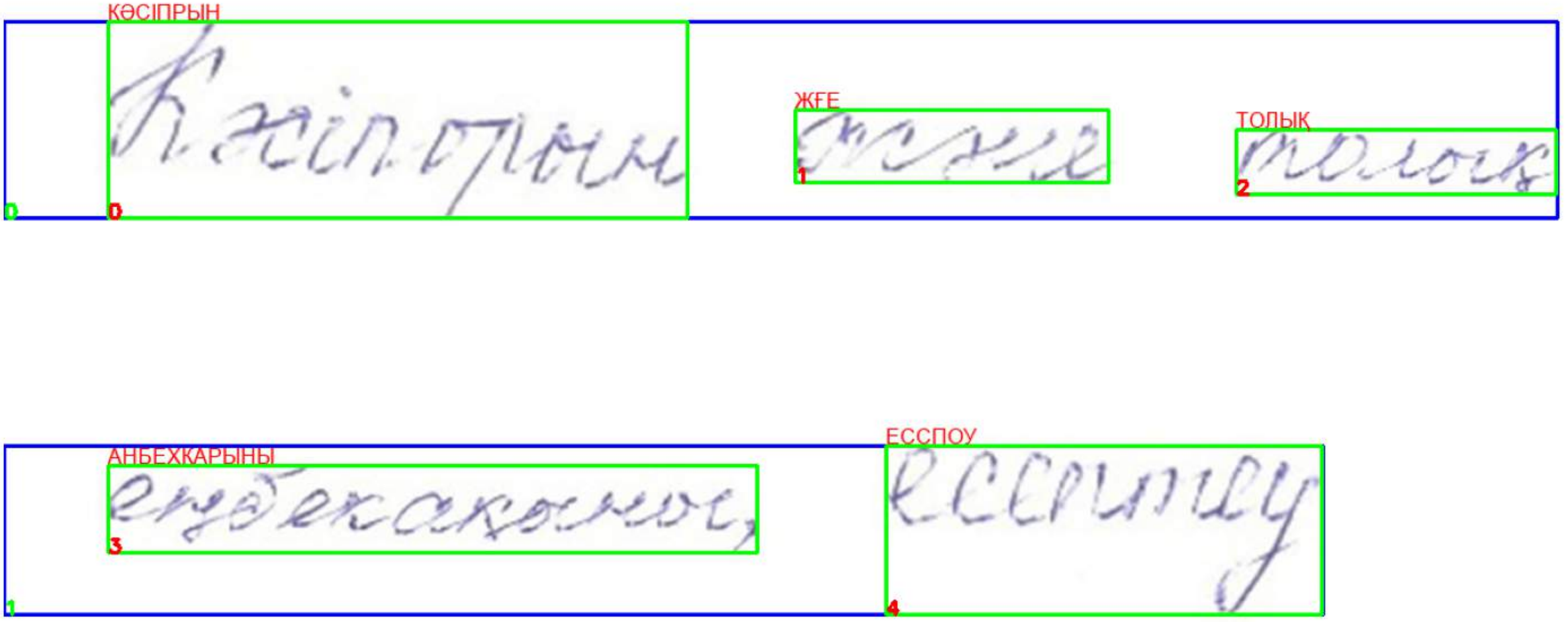
GT: ЕРЕКШЕЛІГІ
Predicted: ЕРЕКШШЕІГІ
Avg Conf: 0.9248



GT: БОЛУ
Predicted: ЫАЛУ
Avg Conf: 0.7767



Recognized Text Overlaid



FUTURE RESEARCH

- Circular text recognition
- Table detection and structure recognition
- Sequential patterns of texts for recognition or post-processing
- End-to-end model for text detection and recognition
- Speech Recognition

REFERENCES

Lee, A. W., Chung, J., & Lee, M. (2021). GNHK: a dataset for English handwriting in the wild. In Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part IV 16 (pp. 399-412). Springer International Publishing.

C. Yao, X. Bai, W. Liu, Y. Ma and Z. Tu. Detecting Texts of Arbitrary Orientations in Natural Images. CVPR 2012 (PDF).

Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., & Hassner, T. (2021). Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8802-8812).

Matcha, A. C. N. (n.d.). Handwriting Recognition with ML (An In-Depth Guide). Nanonets Blog | AI Document Processing & Workflow Automation. https://nanonets.com/blog/handwritten-character-recognition/

Toiganbayeva, N., Kasem, M., Abdimanap, G., Bostanbekov, K., Abdallah, A., Alimova, A., & Nurseitov, D. (2022). Kohtd: Kazakh offline handwritten text dataset. Signal Processing: Image Communication, 108, 116827.

Kumar, A., & Pati, P. B. (2023). Offline HWR Accuracy Enhancement with Image Enhancement and Deep Learning Techniques. Procedia Computer Science, 218, 35-44.

Nurseitov, D., Bostanbekov, K., Kurmankhojayev, D., Alimova, A., Abdallah, A., & Tolegenov, R. (2021). Handwritten Kazakh and Russian (HKR) database for text recognition. Multimedia Tools and Applications, 80(21), 33075-33097.

Toiganbayeva, N., Kasem, M., Abdimanap, G., Bostanbekov, K., Abdallah, A., Alimova, A., & Nurseitov, D. (2022). Kohtd: Kazakh offline handwritten text dataset. Signal Processing: Image Communication, 108, 116827.

Patterson, J., & Gibson, A. (2017). Deep learning: A practitioner’s approach (First edition). O’Reilly.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. The MIT press.

Raschka, S., Liu, Y. H., & Mirjalili, V. (2022). Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python. Packt Publishing Ltd.

ANY QUESTIONS?

THANK YOU!