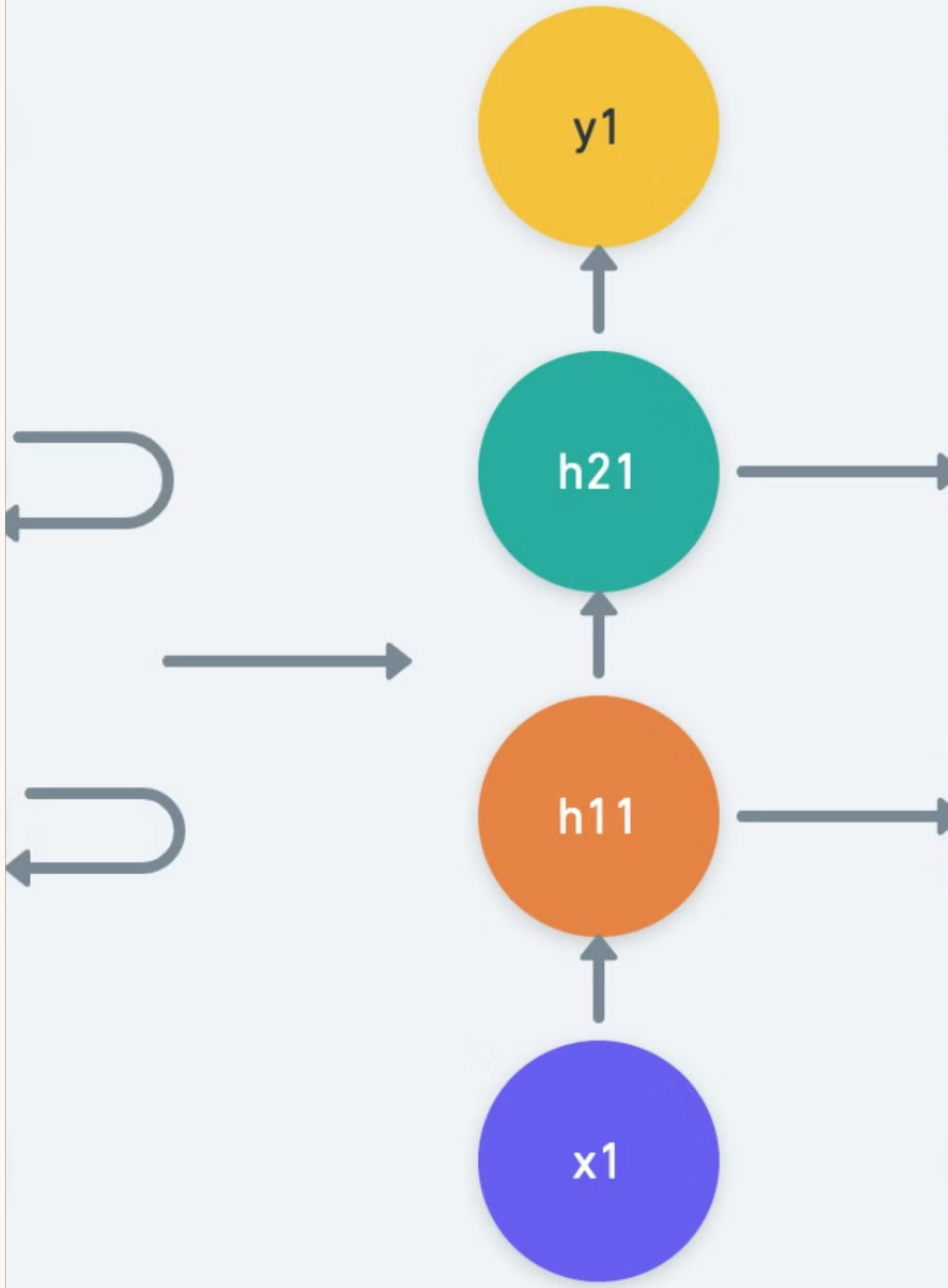
A Recurrent Neural Network Based Deep Learning Model for Offline Signature Verification and Recognition System

Journal: Expert Systems With Applications

Presented by: Amirmohammad Taghizadegan

Teacher: Dr. Yazdian



Back ground review

Introduction

Where Does the Idea Come From?

Online or Offline?

Writer Dependent or Writer Independent?

why this method?

Literature Review

Study	Year	Method/Model	Key Findings/Techniques
Justino et al.	2000	HMM with graphometric features	Used HMM based method for offline signature verification.
Kalera et al.	2004	Bayes classifier with gradient, structural, and concavity features	Presented a method for offline signature verification.
Ferrer et al.	2005	Geometric signature features with fuzzy modeling	Proposed geometric features for signature identification.
Hanmandlu et al.	2005	Fuzzy method using Takagi–Sugeno model	Utilized fuzzy modeling for offline signature verification.
Wen et al.	2009	HMM with rotation invariant structure features	Employed HMM for offline signature verification.
Pal et al.	2015	Probabilistic Neural Network (PNN)	Characterized signatures using connected components, regions, and curvelet features.
Ooi et al.	2016	Discrete Radon Transform (DRT) with PNN	Proposed a hybrid method using DRT for offline signature verification.
Yilmaz et al.	2016	Score level fusion of classifiers using HMM	Investigated WD and WI classifiers for signature verification.
Zois et al.	2016	Template matching with lattice-shaped structures	Used for WD offline signature verification.
Hadjadji et al.	2017	Curvelet Transform (CT) and PCA	Reported a method for signature identification using CT and PCA.
Hezil et al.	2018	KNN classifier	Used for recognition of offline signatures with statistical and pattern features.
Okawa	2018	Bag-of-visual words with forensic cognitive knowledge	Proposed a method considering forensic document examiners' cognitive knowledge.
Maergner et al.	2019	Graph edit distance with deep triplet networks	Combined graph edit distance with deep triplet networks for verification.
Bhunia et al.	2019	SVM with hybrid texture features	Proposed a method combining texture features for signature verification.
Soleimany and Foula	2020	Deep Multitask Metric Learning (DMML)	Introduced DMML for offline signature verification.
Jagtap et al.	2020	SNN with CNN subnetwork	Used for verification of forged offline signatures.
Jain et al.	2020	Shallow CNN	Script-independent method tested on multiple scripts for offline signature verification.

https://doi.org/10.1016/j.eswa.2020.114249

Datasets

Details of the Training and Testing Signature Datasets.

Script	Dataset	No. of Writers	Training samples per individual signature	Total No. of training samples	Total No. of test samples
Latin	GPDS synthetic	4000	12	48 000 genuine	48 000 genuine, 120 000 forged
	GPDS-300	300	12	3600 genuine	3600 genuine, 9000 forged
	MCYT-75	75	10	750 genuine	375 genuine, 1125 forged
	CEDAR	55	12	660 genuine	660 genuine, 1320 forged
Devanagari	BHSig260 Hindi	160	12	1920 genuine	1920 genuine, 4800 forged
Bengali	BHSig260 Bengali	100	12	1200 genuine	1200 genuine, 3000 forged

Proposed Method

solve

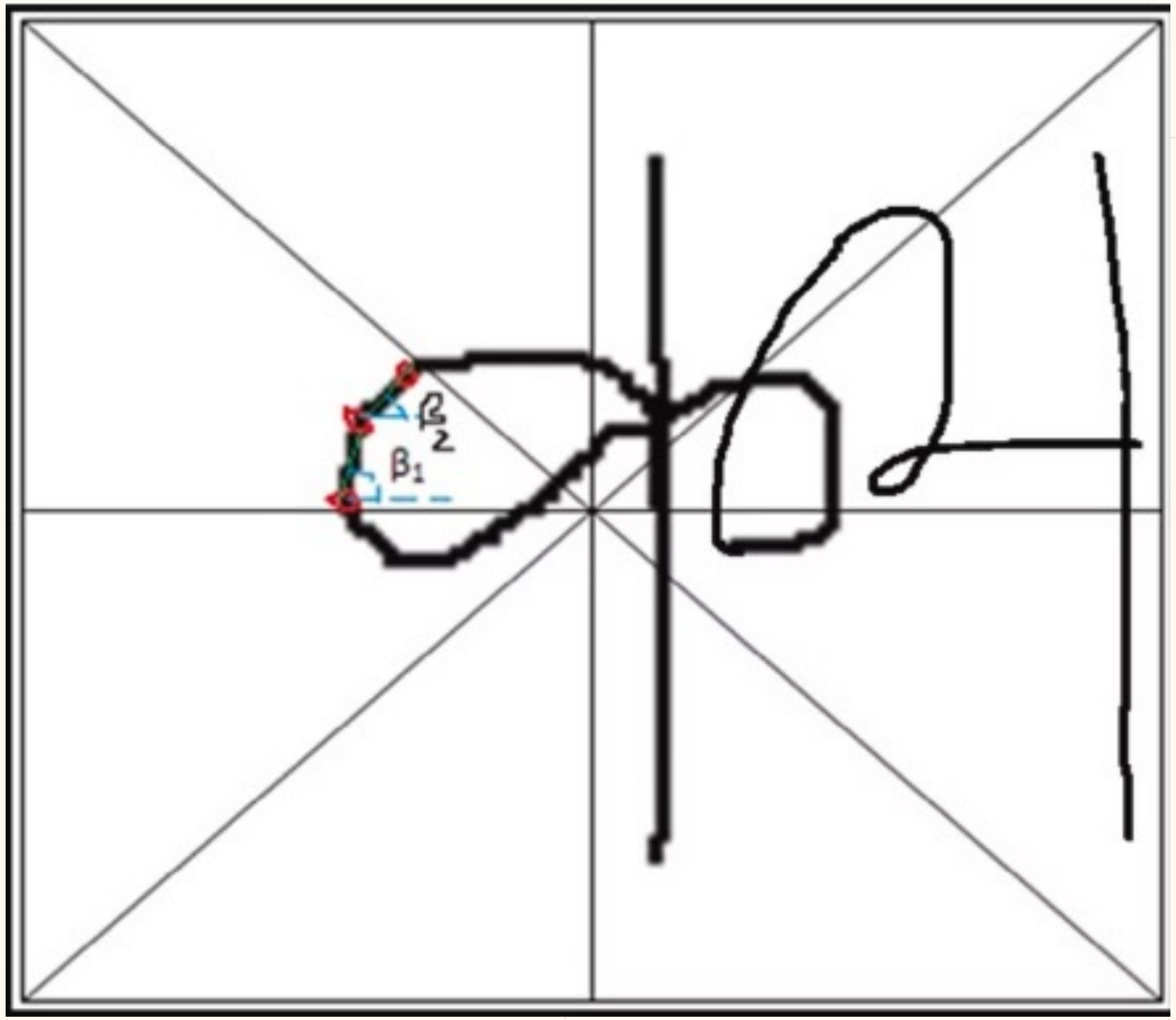
- Preprocessing
- 128 * 128 images
- Horizontal orientation

Feature Extraction

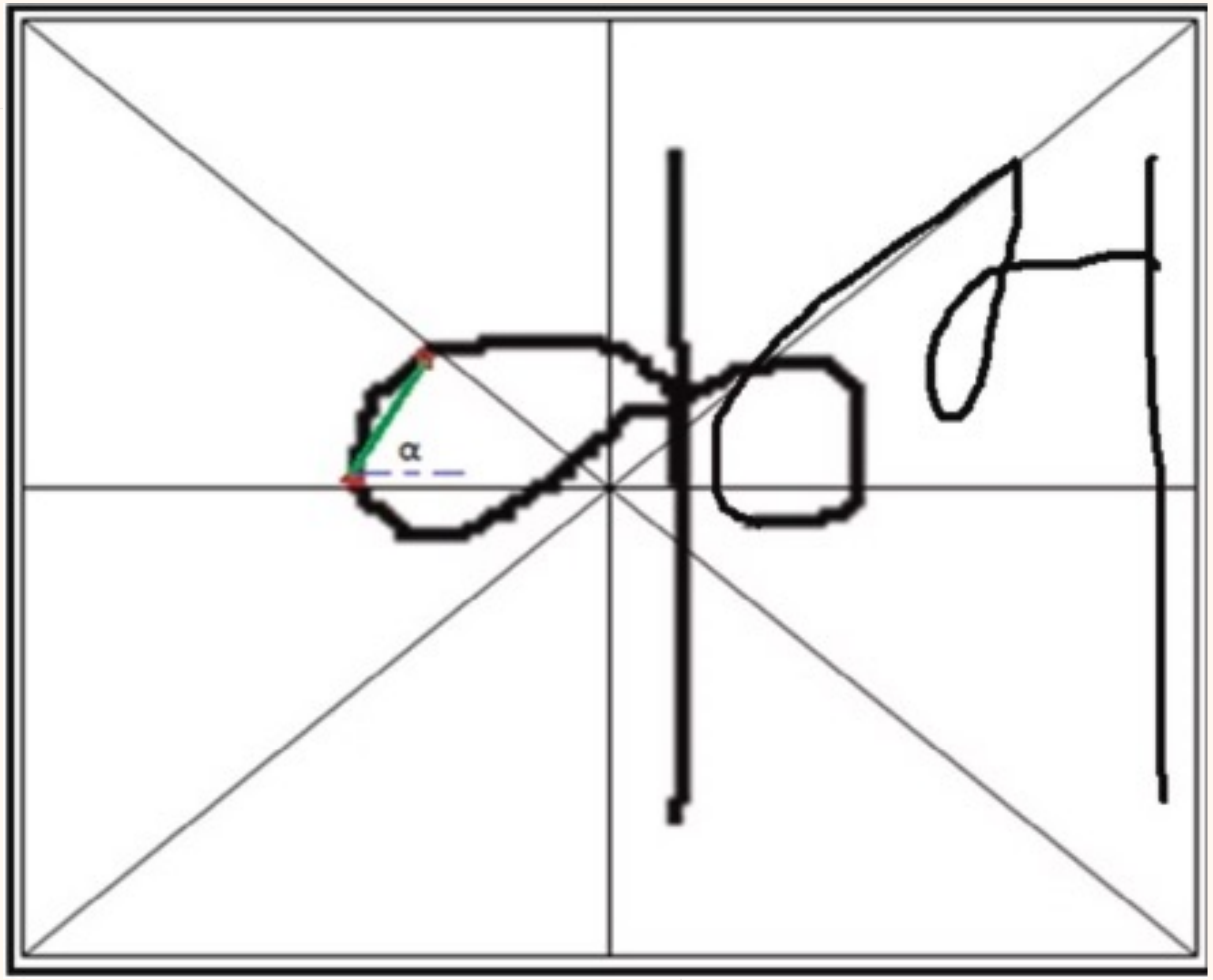
Signature verification and recognition

Feature Extraction

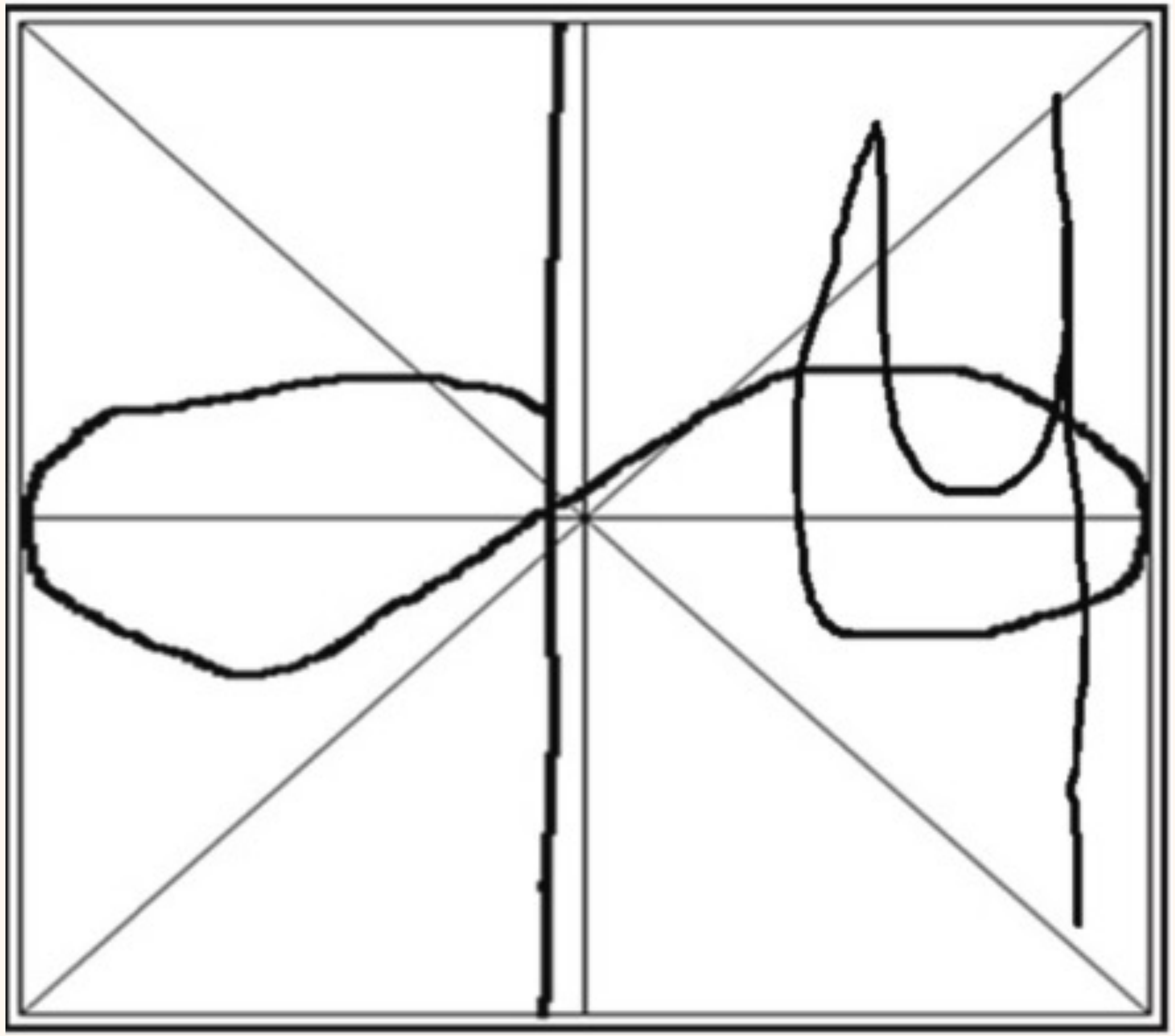
- **Change of trajectory direction (CTD):** 4 * 8 = 32 features: -1:1.
- Sine and cosine functions used.



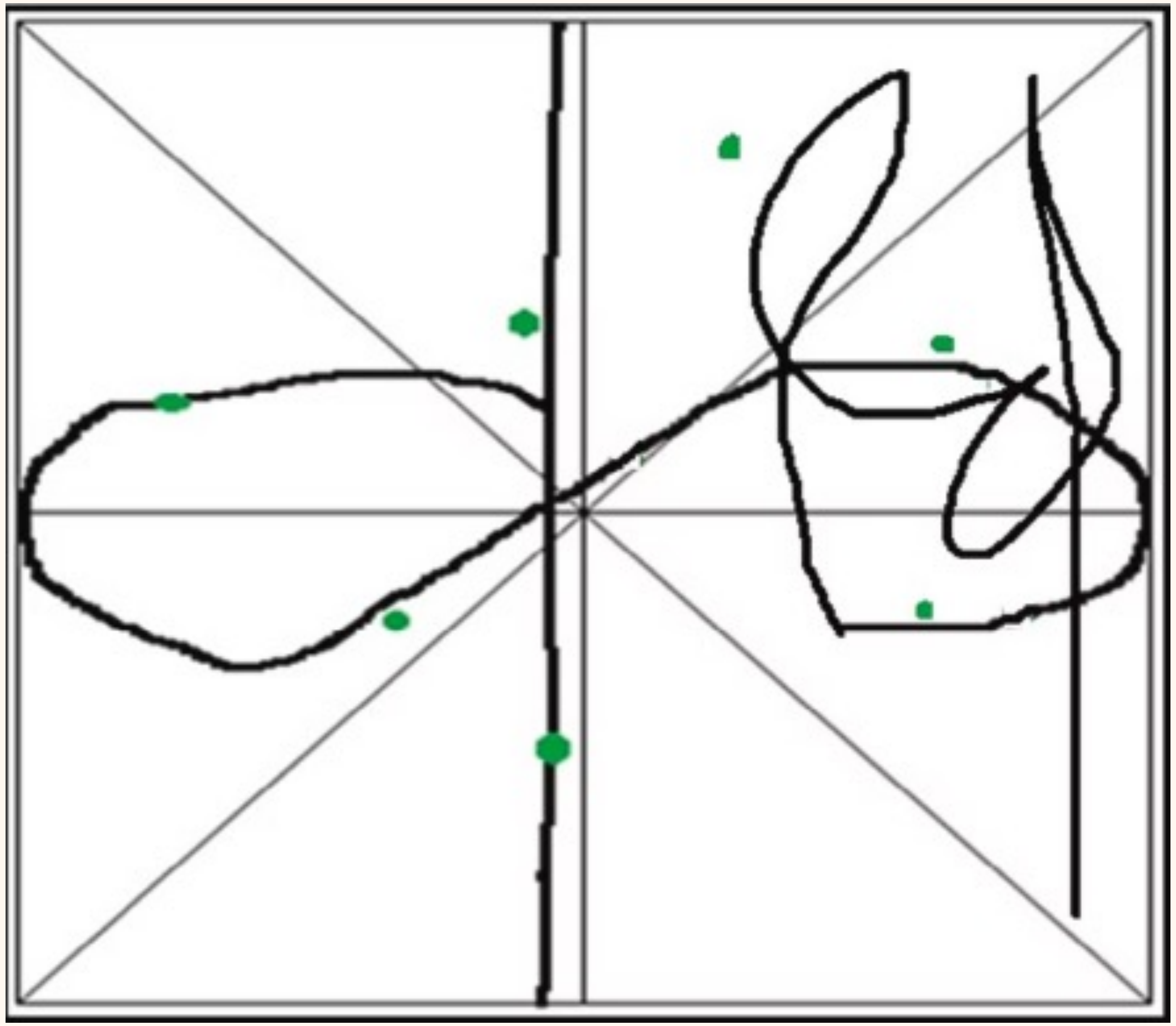
- **Trajectory slope (TS):** 1 * 8 = 8 features: -1:1.
- the cosine function used.



- **Trajectory waviness (TW):** 1 * 8 = 8 features: 0:2.
- the distance/side calculation is used.



- **Centre-of-mass (COM):** 2 * 8 = 16 features: 0:1.
- the x and y coordinates is used.



Signature verification and recognition

1

Final classification using RNN models.

-LSTM

-BLSTM

-CNN

2

Performance metrics evaluated on test datasets.

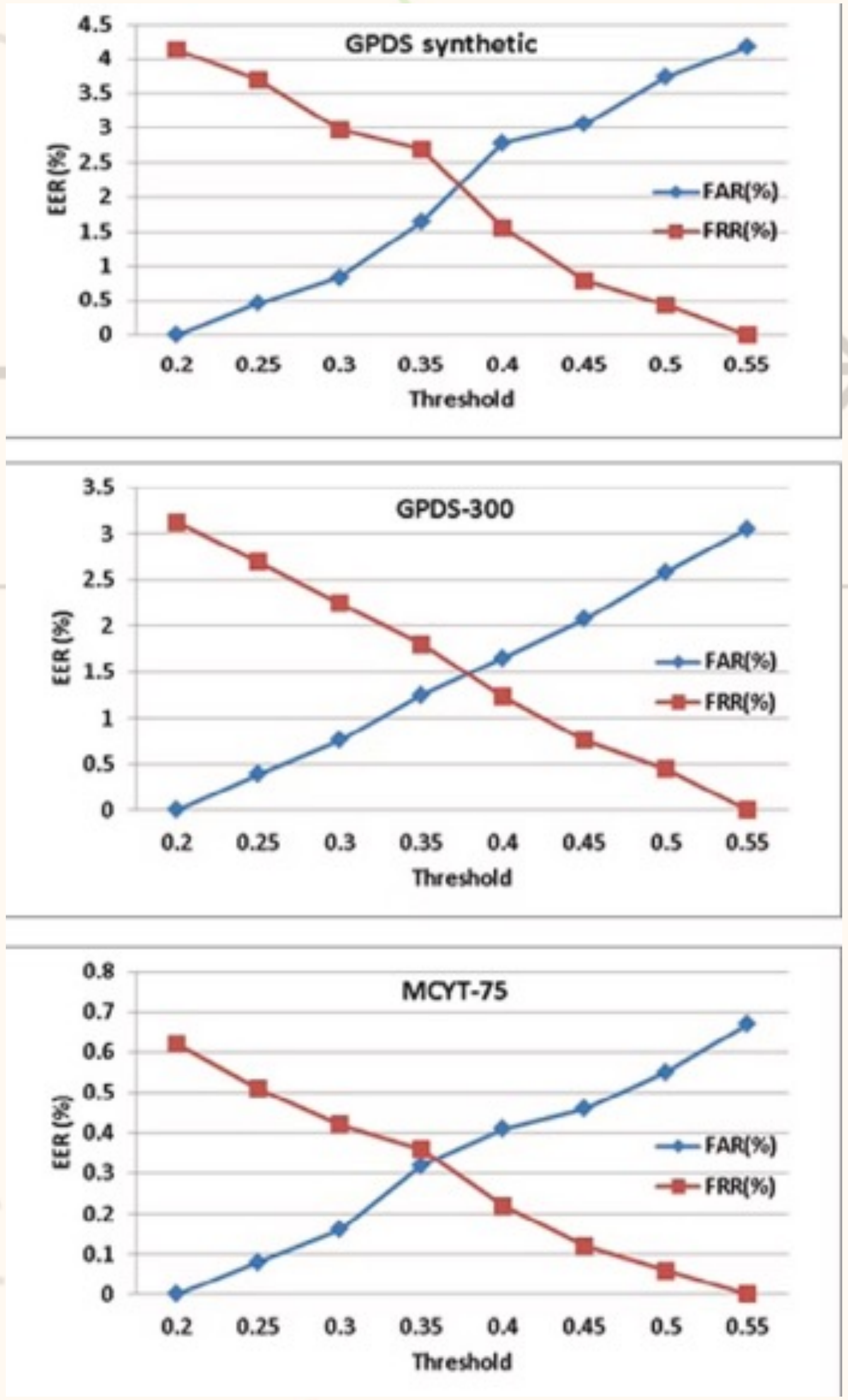
FAR(false acceptance rate)

FRR(false rejection rate)

EER(equal error rate)

3

Threshold



LSTM config

Input layer: 64 neurons

Hidden Layer:

- **Neurons**: 50 recurrently connected memory blocks.
- **Activation Functions**:
  - Block input/output: Tanh
  - Gate: Sigmoid.

Output Layer:

- depends on the number of writers

** **BLSTM Configuration**: 2 * LSTM

CNN config

Input layer: 16384 neurons

Hidden Layer:

3 convolutional layers

- **Filter Size**: 3x3 for each convolutional layer.
- **Pooling**:
  - After conv1: Max-pooling on 3x3 subareas.
  - After conv2 and conv3: Max-pooling on 2x2 subareas.

Output Layer: depends on the number of writers

** **BLSTM Configuration**: 2 * LSTM

Results Summary

Result

Dataset	Features	BLSTM Accuracy	LSTM Accuracy
GPDS synthetic	CTD, TS, TW, COM	96.08%	83.02%
	CTD, TS, TW	89.14%	78.42%
	CTD, TS, COM	85.32%	74.24%
	CTD, TW, COM	81.96%	69.32%
GPDS-300	CTD, TS, TW, COM	98.02%	85.76%
	CTD, TS, TW	90.92%	81.83%
	CTD, TS, COM	86.97%	77.79%
	CTD, TW, COM	83.64%	72.69%
MCYT-75	CTD, TS, TW, COM	99.39%	91.82%
	CTD, TS, TW	93.28%	83.78%
	CTD, TS, COM	89.64%	79.68%
	CTD, TW, COM	86.42%	74.34%
CEDAR	CTD, TS, TW, COM	99.94%	92.13%
	CTD, TS, TW	94.88%	89.34%
	CTD, TS, COM	91.24%	85.24%
	CTD, TW, COM	87.96%	80.26%
BHSig260 Hindi	CTD, TS, TW, COM	99.28%	91.61%
	CTD, TS, TW	94.32%	87.36%
	CTD, TS, COM	90.65%	82.43%
	CTD, TW, COM	87.34%	78.28%
BHSig260 Bengali	CTD, TS, TW, COM	99.37%	91.79%
	CTD, TS, TW	94.48%	86.48%
	CTD, TS, COM	90.80%	82.64%
	CTD, TW, COM	87.48%	77.34%

Signature recognition accuracy using CNN.

Dataset	Accuracy
GPDS synthetic	93.24%
GPDS-300	94.18%
MCYT-75	94.42%
CEDAR	95.31%
BHSig260 Hindi	95.12%
BHSig260 Bengali	95.19%

Suggestion

1- Using Transformer models
2- combine CNN with RNN
3- too many Paradox

Q&A

End