



Attendance Verification Using Machine Learning

Cuneyt Aksoy

B.S., Computer Science

Senior Capstone Project

Supervised by Dr. Kafi Rahman, Ph.D.

Fall 2023

THE IDEA



- Inspired by the boom of ChatGPT + Artificial Intelligence
- Combining machine learning with a Truman State based system
- Main online systems include TruView and Brightspace, with no attendance recording
- No centralized method exists to record attendance at Truman State
- Serve as a stepping stone for experimentation
- Missed opportunities with attendance information, improving retention rates
- Different methods might be employed, which can raise issues

ARCHITECTURE



- Uses HTML, CSS, JavaScript (Front-End), Python, PHP (Back-End), Apache server (Hosting)
 - PHP is used to manage user sessions, handle form submissions, and interact with the database
 - Python server is responsible for machine learning using the **scikit-learn** machine learning library to train and test the signature verification model
 - Apache server handles HTTP requests and responses
- MySQL stores student records and attendance logs

AI - MACHINE LEARNING



Compares the signature entered by user with the signature samples the model was trained on

Signature verification mainly consists of 4 steps:

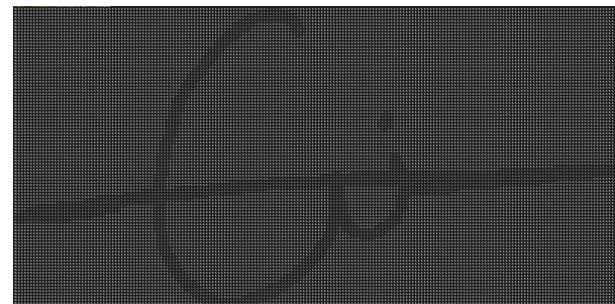
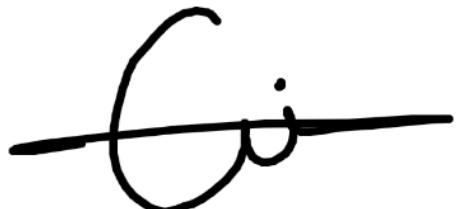
- Preprocessing
 - Images are cleaned and made ready for feature extraction
- Feature Extraction
 - Extracting distinctive quantifiable attributes or patterns from signature samples
- Model Evaluation using 10-fold cross validation
- Training

PRE-PROCESSING



Priming for feature extraction

1. Take an image as input trim from all sides
2. Convert trimmed image to a 200x100 matrix
3. Convert this image into a gray scale image
4. Processes the image pixel by pixel, if the value in a pixel is greater than 200, then it considers it as 0, else 1
5. This yields the binary version of the image as shown below





DISTANCE METHOD

- Then, the program goes through each line of the matrix and calculates the distances between every line of the signature and multiplies them with a coefficient to assign them unique weights and finally adds them up
 - For each row of the matrix, it generates an integer value
 - Later, these integers are saved in testing files as features. e.g., 35, 48, 48, 16, 85, 36, 133, ...

35×1

$+ 24 \times 2$

$+ 16 \times 3$

$+ 4 \times 4$

$+ 17 \times 5$

$+ 5 \times 6$

$+ 19 \times 7$

$+ 2 \times 8$

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

⋮ MASKING METHOD ⋮

- Like the previous approach, image is first converted into gray-scale mode
- The idea is to determine the number of occurrences of certain patterns (masks) in the image
- Occurrences of each mask is used as a feature
- Note: Different sizes of masks can be used

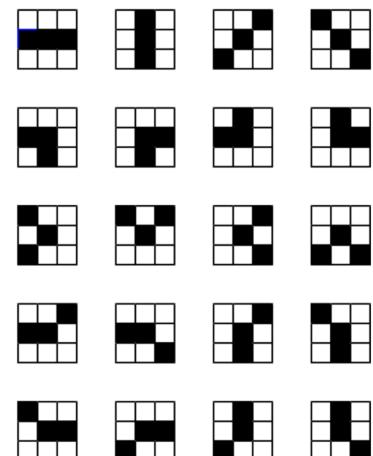


Fig 1: All possible 3x3 Masks

10-FOLD CROSS VALIDATION

- Evaluate the performance and generalizability of a model
- The dataset is divided into ten subsets (folds). The model is trained on nine of these folds and tested on the remaining fold. This process is repeated ten times, with each fold being used as the test set exactly once.
- Performance metrics obtained from each iteration (fold) are averaged to provide an overall estimation of how the model is likely to perform on unseen data. It helps to assess how well the model generalizes to new data beyond the training set.

PERFORMANCE



Algorithm	Masking Method	Distance Method
Decision Tree	60%	49%
Random Forest	73%	80.75%
SVM (Support Vector Machine)	45%	69%
Gaussian Naive Bayes	42%	75%
KNN (K-Nearest Neighbor)	71%	59%
Neural Networks	6%	22%

Higher percentages indicate better performance.

DETECTION



- Finally, ML algorithm is used to compare the provided testing data to the training data stored in database
- If machine learning algorithm can classify it as the user who provided the input, attendance is recorded
- Attendance for this user is recorded in the database, stamped with the date

Your signature is accepted!

[Close](#)

CLOSING THOUGHTS



Relevant coursework:

- CS180 - Foundations of Computer Science I
- CS181 - Foundations of Computer Science II
- CS260 - Object-Oriented Programming and Design
- CS310 - Data Structures and Algorithms
- CS315 - Internet Programming
- CS430 - Database Systems

Ideas for improvement

</THANKS>

<? concerns ?>