A Text-Independent Writer Identification System form Handwritten Document Images

Tathagata Bandyopadhyay

1305382

Overview

- Introduction
- Different Approaches
- Problem Definition
- Use cases and the functionalities
- Overall Framework
- Three Layered Design
- Function Oriented Design
- Implementation
  - SURF key points & Descriptors
  - Page level vs word level & need of word segmentation
  - Core functions
  - GUI
  - H/W and S/W Configuration Used
- Testing
  - Used Dataset
  - Results
- Conclusion and Future Scope
- References

Introduction

- Like fingerprint, Handwriting is unique for each individual[1],[2].
- Hand writing sample can help in recognizing the author.
- Wide area of applications includes but not limited to:
  - Forensic Analysis
  - Examination Candidate Verification
  - Verifying authenticity of the writer of Historical Documents
  - User Identity Verification and authentication for secure access
- Computerised systems trained with sample images of a known writer can help in predicting author identity of a new sample image.
- This makes Computerised Handwriting Recognition one of the most active and interesting area of research in the domain of Pattern Recognition.

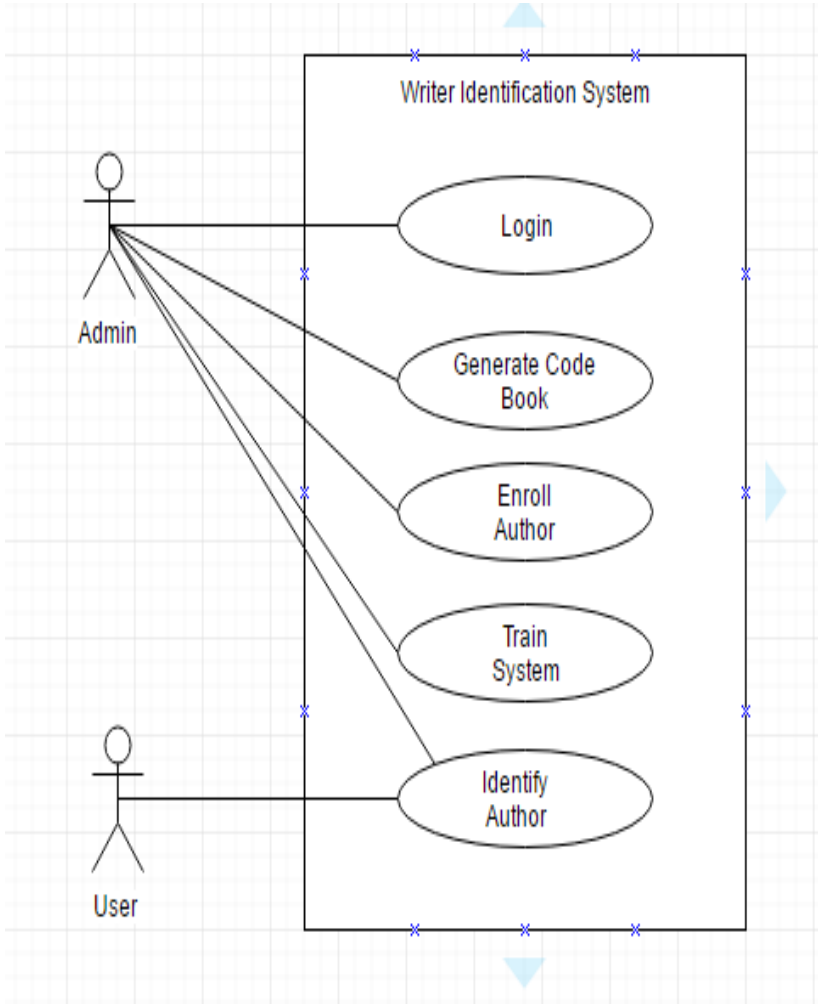
Different Approaches

- Writer Identification vs. Verification
  - **Identification:** Recognizing the writer of a new image sample as one of those registered in the database.
  - **Verification:** Similarity between two writing samples
- Offline vs Online Writer Identification
  - **Offline:** Scanned Image document. Processing is done later on.
  - **Online:** Writing in a pen-enabled electronic touch pad using a stylus. Processing is done while writing
- Text-Independent vs Text-Dependent
  - **Text-Independent:** Any text and Script
  - **Text-Dependent:** Some fixed text
- Structure based vs. Texture based
  - **Structure based:** Considers Shape Geometry
  - **Texture based:** Consider whole image as a texture pattern
- Approach adopted in the project:
  - Identification
  - Offline
  - Text-Independent
  - Structure based (Also Scale and Rotation Invariant)

Problem Definition

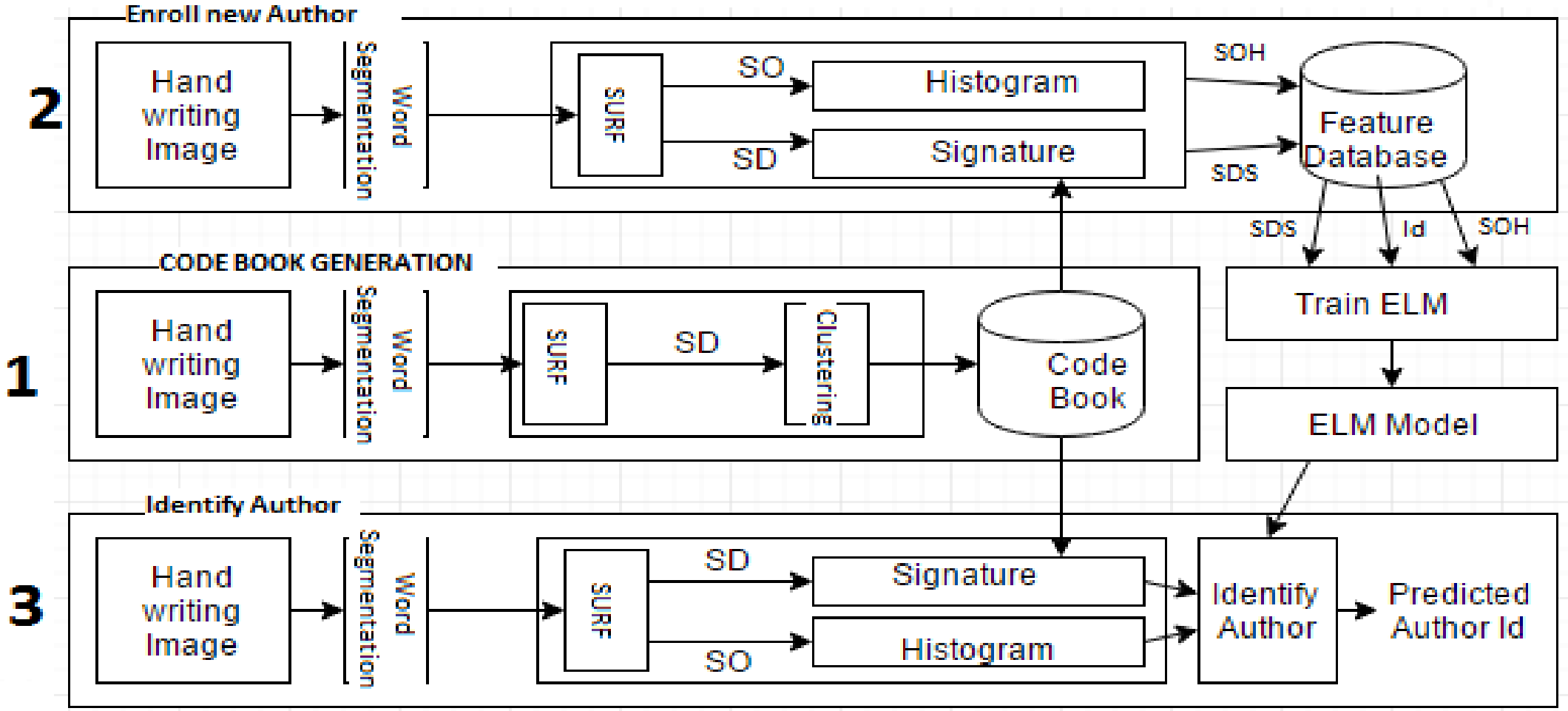
- To build a GUI application which takes a handwriting image sample as an input and predicts its author id from a pre-registered set of authors.
- System should be text independent and Scale and Rotation Invariant.
- System should also provide the functionality of registering new writer in the database.

Use cases and the Functionalities

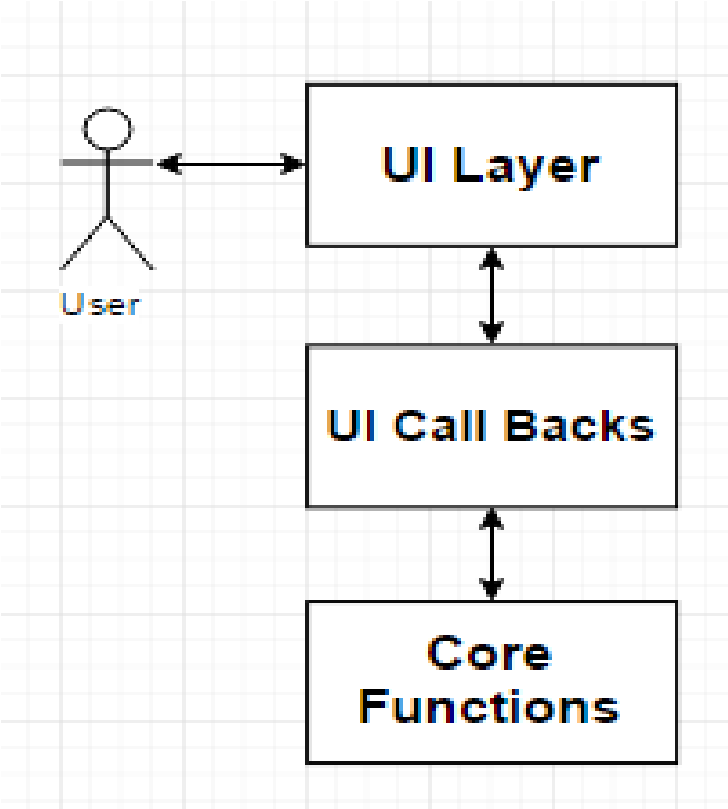


- Actors:
  - Admin and User. Admin has access to all the functionalities. User has access to only one functionality.
- Login:
  - To allow only admin to access special functionalities and restrict others.
- Generate Codebook:
  - To produce a reference for feature extractions.
- Enrol Author:
  - To store the features of a handwriting of a known author with the author id in the author database.
- Train System:
  - To train a classifier with the author database to build a author id prediction model for new handwriting samples.
- Identify Author:
  - To predict author id of a new handwriting sample with the help of the prediction model.

Overall Framework

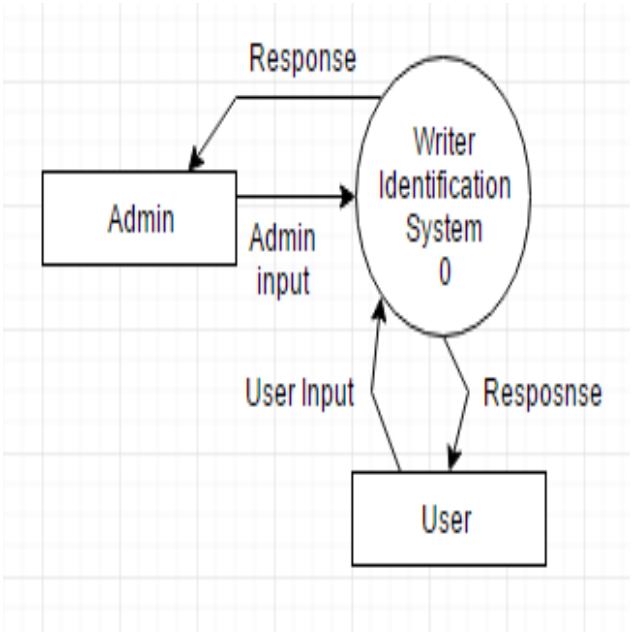


Three Layered Design

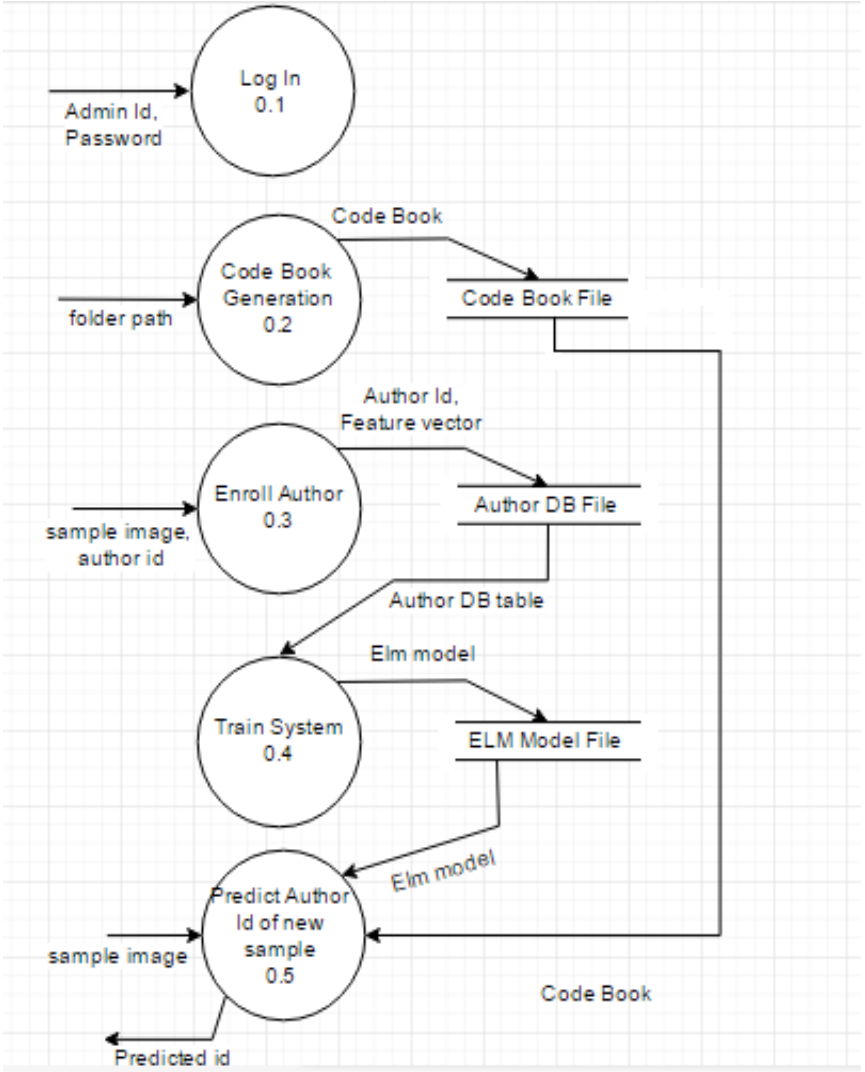


- Actor has access to only the User Interface Layer and interacts with the system through it.
- Algorithms for solving the problems are implemented in the Core Functions Layer.
- UI Call Backs acts as a bridge between UI and Core Functions Layer and controls the actor interaction with the

Function Oriented Design



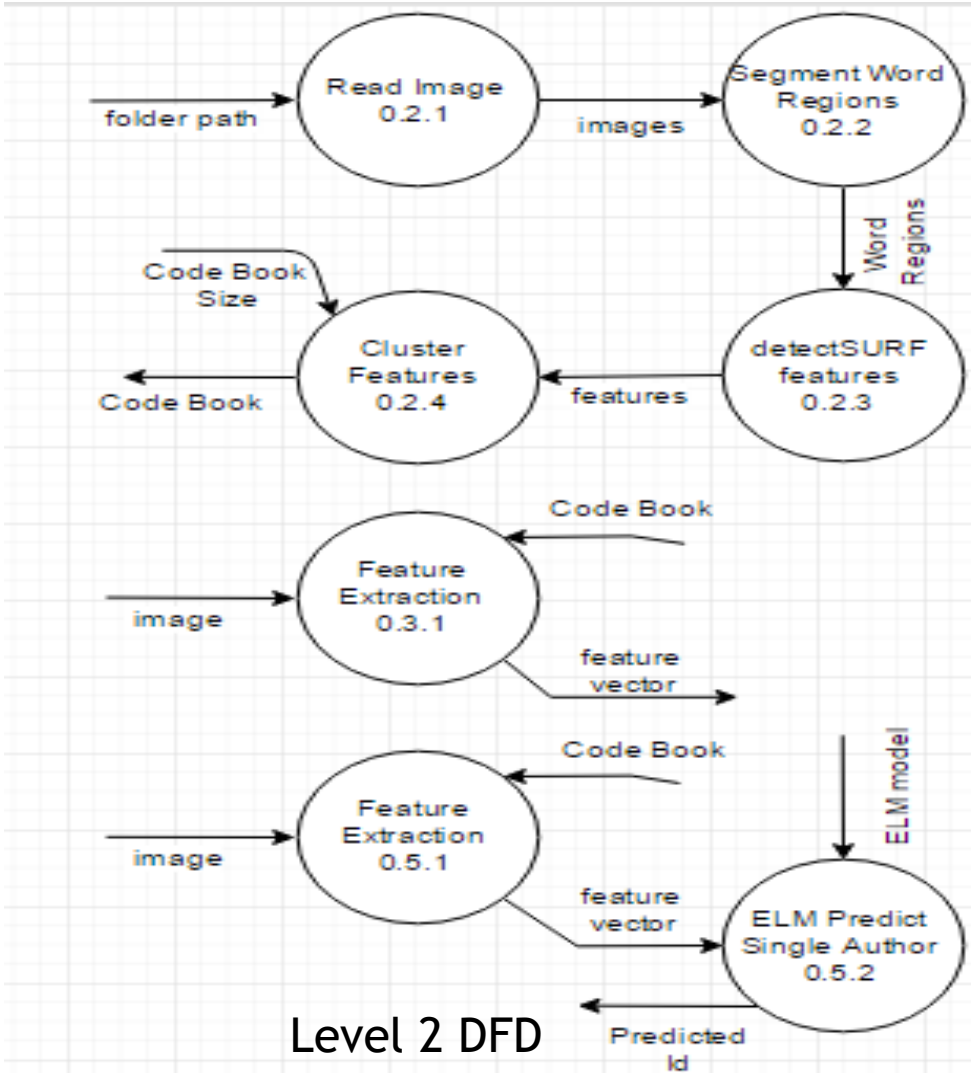
Level 0 DFD



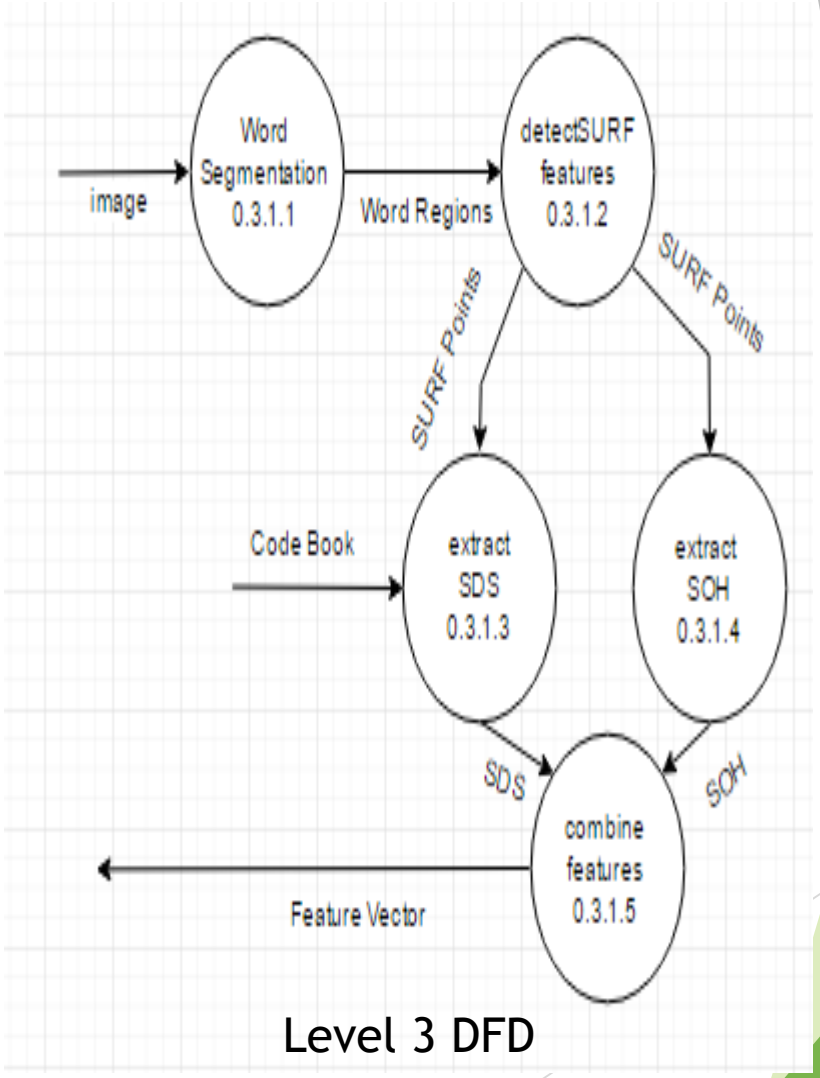
Level 1 DFD

- This project is mostly function oriented.
- Functions are decomposed up to level 3.

Function Oriented Design contd.



Level 2 DFD



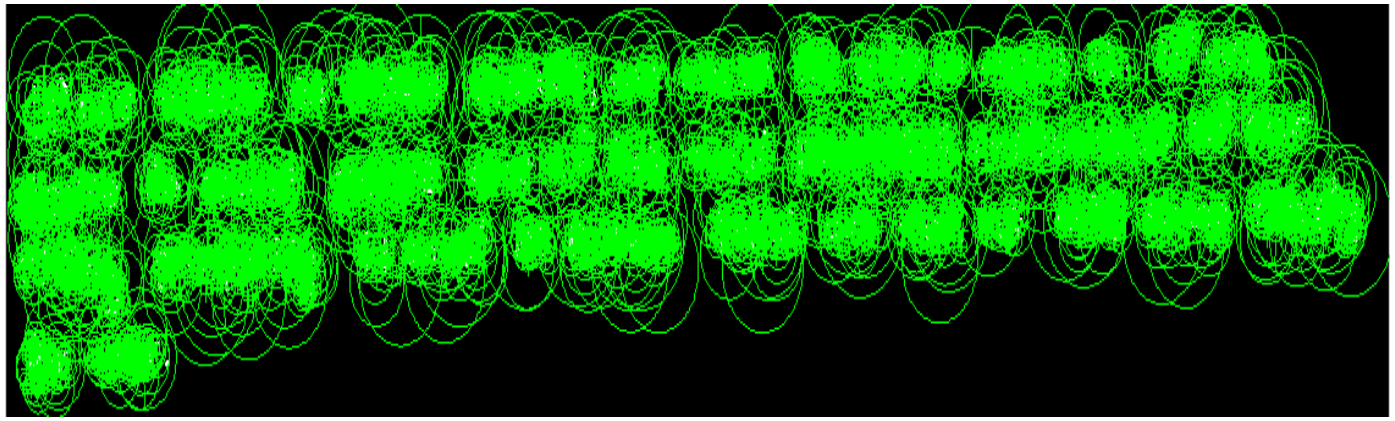
Level 3 DFD

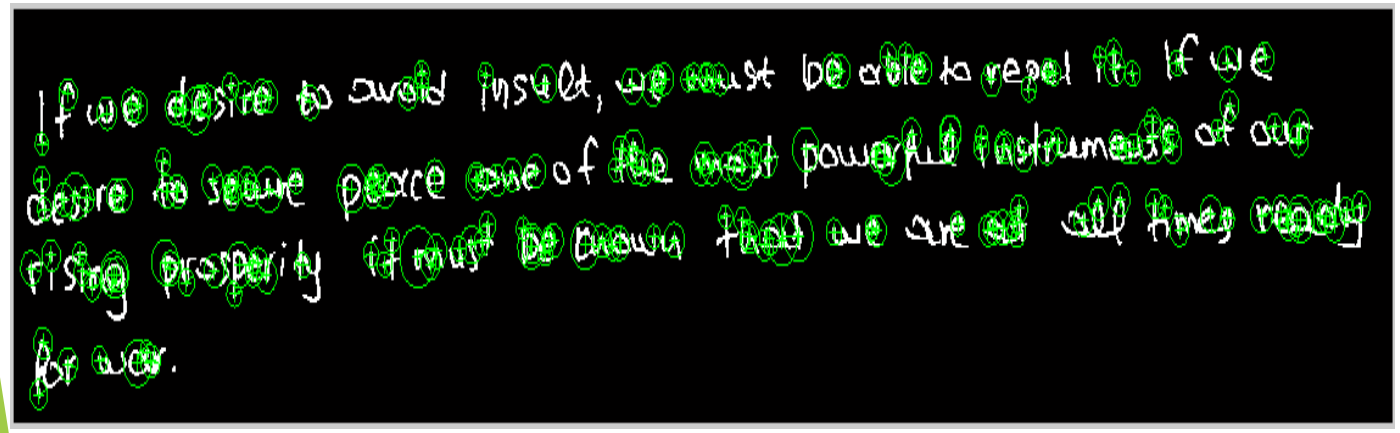
Implementation: SURF key points

- Speed Up Robust Feature(SURF)[4] is the faster version of Scale Invarient Feature Transform(SIFT).
- It detects some points (known as SURF key points) in an image which are invariant to scale and rotation.
- These points are mostly blobs or corners which are unique characteristics of a particular type of images.
- Gaussian smoothing is used to blur the noise and identify the key points.
- Each of such point is descripted by gradient histogram, known as SURF descriptor.

Implementation: Page level vs word level feature

Need for Word Segmentation





- Application of SURF on page level generates many redundant feature key points.
- Redundant features requires more resources and also reduces the efficiency.
- So word segmentation is done before applying SURF.

Implementation: Core Functions

This whole project stands on the following pillars:

- Word segmentation
  - Automatically crop the word regions.
  - Used as a pre-processing step to reduce number of redundant feature generation in feature extraction step.
  - Uses blurring and connected component analysis.
- Code Book Generation
  - Different images may generate different number of SURF key points.
  - To make the feature length fixed, SURF descriptors of the word regions of the training images are clustered into 300 categories.
  - Each cluster is represented by its centre coordinates.
  - This collection of all the cluster centres is called code book and referenced during feature extraction to build SURF Descriptor Signature (SDS) feature.
  - Uses k-means clustering with random initial points.

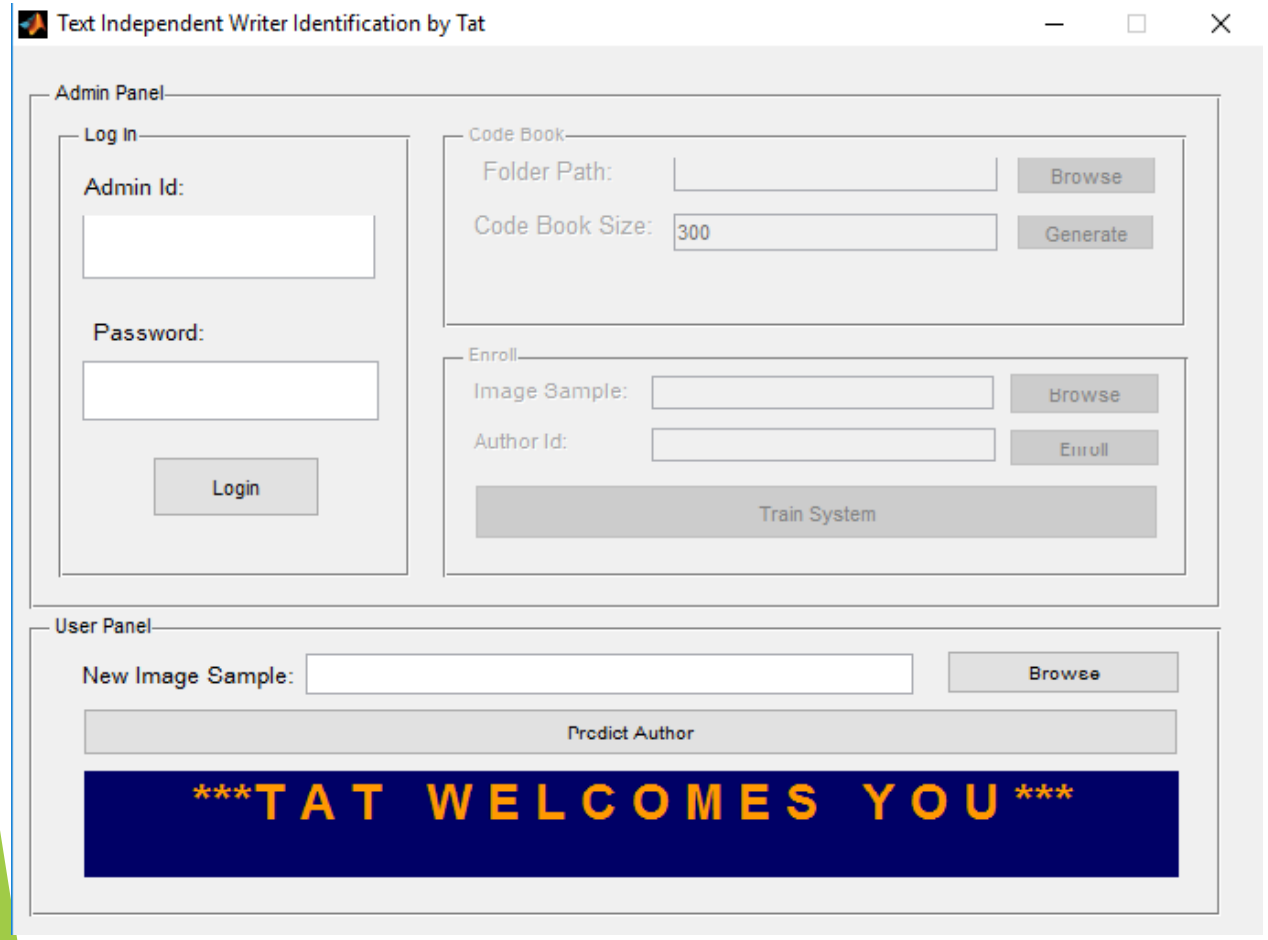
Implementation: Core Functions contd.

- Feature extraction
  - Converts an input image to a feature vector.
  - SDS and SOH two features are used
  - SDS is the histogram of SURF descriptors. Each bin of this histogram corresponds to one code (Cluster Centre) in the code book.
  - SOH is the concatenation of the histograms of SURF key points orientations for each Scale. Orientations are binned in 24 bins of interval 15 degree each.
  - This function is used by both Enrol Author and Predict Author functions.
- Enrol Author
  - This function registers the feature values of an handwriting sample of a known author into a row, identified by the author id, in the author database.
- Train System
  - This function trains an ELM [5] classifier with the author database.
  - Each individual author is treated as a class and author id gives the corresponding class label.
  - ELM with 100 Hidden neurons is used.
  - This function returns a trained ELM model.

Implementation: Core Functions contd.

- Predict Author
  - This function takes a new handwriting sample and invokes feature extraction to convert the image into feature values.
  - With the trained ELM model it calculates the score (value of output layer nodes) of the feature vector for each of the class labels.
  - The class label with the heights score is reported as predicted class.

Implementation: Graphical User Interface



- Figure Shows the Initial Screen of the GUI.
- Developed using GUIDE tool of Matlab
- More GUI functionalities in the Demo Session

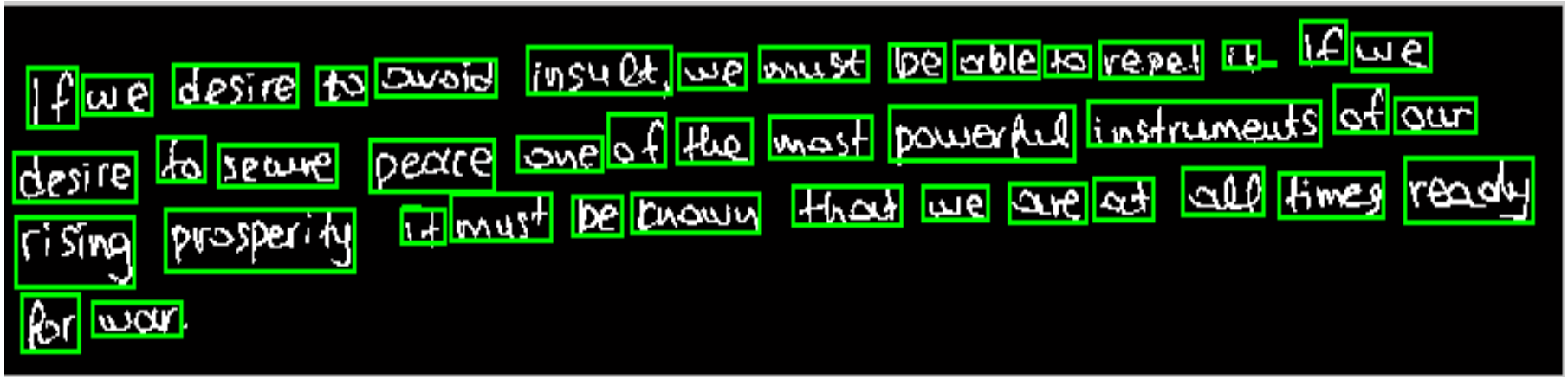
Implementation: H/W and S/W configuration

- Hardware:
  - Intel Core i5, 2.20 GHz processor
  - 8 GB DDR3 RAM
  - 1 TB HDD
- Software:
  - Matlab 2014Ra
  - GUIDE tool of Matlab

Testing: Dataset used

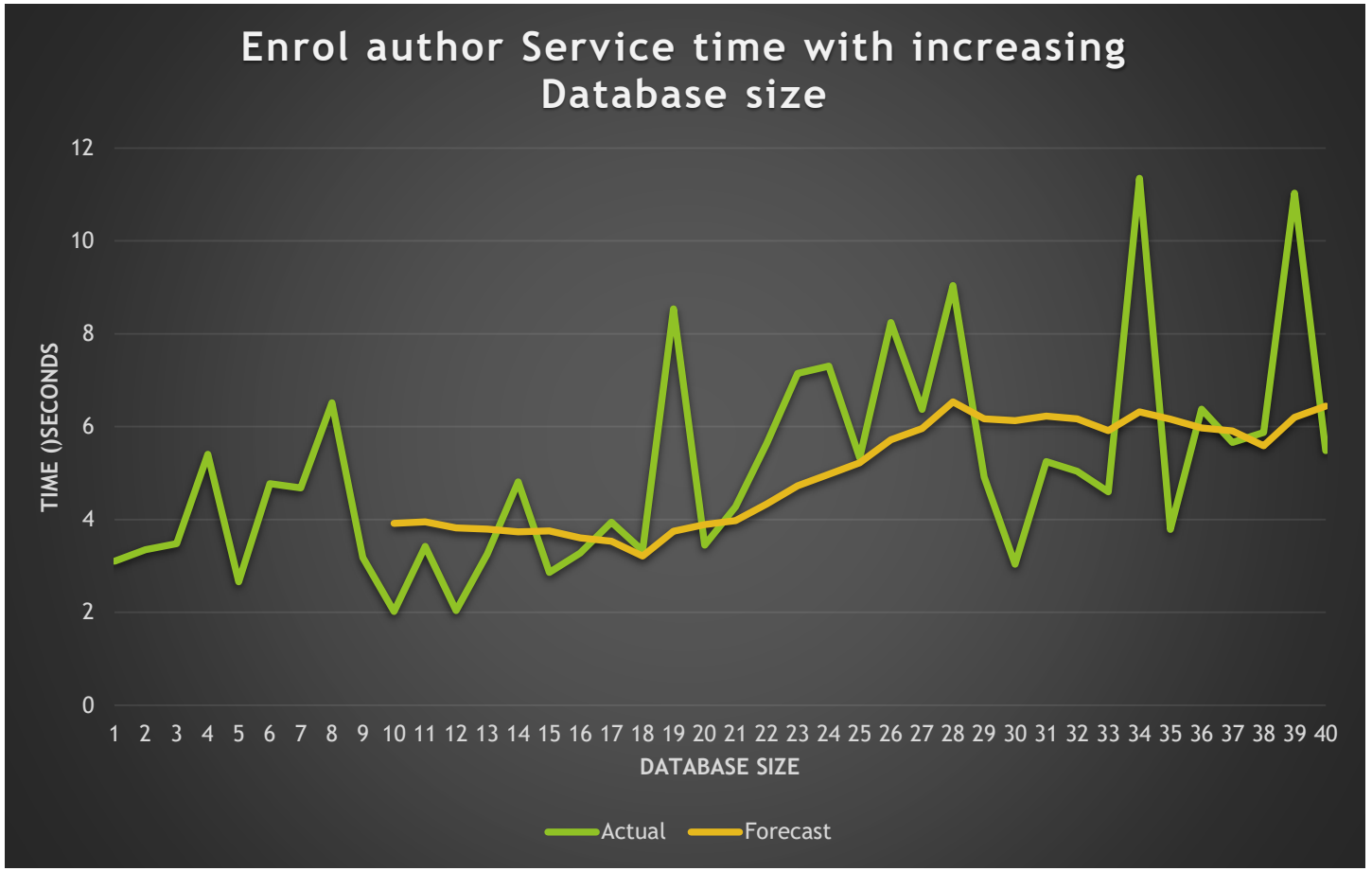
- To test the system ICDAR 2013 dataset is used.
- It contains 1000 handwriting sample images of 250 authors.
- Each author has 4 samples, 2 of which are written in English and other two in Greek.
- One English and One Greek Sample per author Kept in the train set and other two in the test set.

Testing: Results (Word Segmentation)



- Above figure shows the result of word segmentation.
- Each Green box correspond to one word region.
- This segmentation is done automatically without human intervention.

Testing: Results Contd. (Enrol Author Service Time)

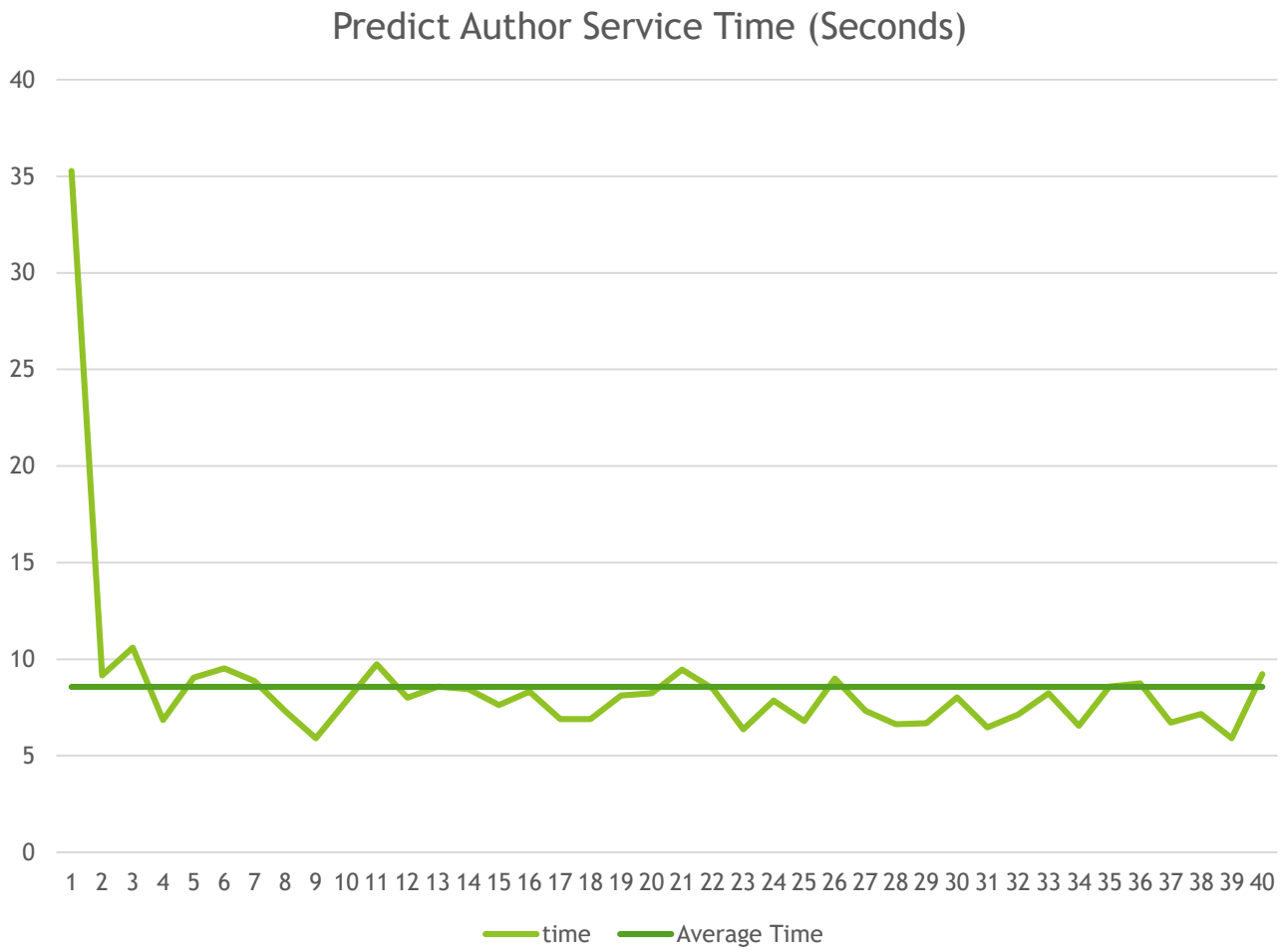


- Enrol Author functionality is tested with 40 input.
- Graph shows the actual and moving average of Enrol Author Service time.
- From the graph we can see it is almost linear.
- Average of Enrol Author Service time was found to be 5.0955 seconds.

Testing: Results Contd. (Training Time)

- The classifier model is trained with 40 size of the data base.
- The training time is found to be **0.078125 Seconds.**

Testing: Results Contd. (Prediction Service Time)



- Predicted author functionality is tested with 40 input.
- Graph shows the prediction Service time for each input.
- Also the dark horizontal line shows the average service time.
- Average service time is found to be 8.56325 seconds.

Conclusion and Future Scope

- This project is a generic structure of a writer identification system.
- All the functional requirements are met.
- Word Segmentation helps in reducing the feature key points.
- Some of the Non functional requirements yet to be achieved.
- The system can be customised for specific future uses like forensic analysis.
- As SURF has a Patent, commercialization of the system needs paid permission from the patent owner.

References

1. Haizhou Li, Liyuan Li, and Kar-Ann Toh. Advanced topics in biometrics. World Scientic, 2012.
2. James Wayman, Anil Jain, Davide Maltoni, and Dario Maio. An introduction to biometric authentication systems. Biometric Systems, pages 1-20, 2005.
3. Xiangqian Wu, Youbao Tang, and Wei Bu. Online text-independent writer identification based on scale invariant feature transform. IEEE Transactions on Information Forensics and Security, 9(3):526-536, 2014.
4. Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). Computer vision and image understanding, 110(3):346-359, 2008.
5. Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. Neurocomputing, 70(1):489-501, 2006.

Thank You