Assessment Management System Kiosk : Scaling Handwritten Student Assessments

Hariny G

SSN College of Engineering, Chennai - 603 110

Professor C.V. Jawahar

International Institute of Information Technology, Hyderabad, Telangana- 500 032

Assessment Management System Kiosk : Scaling Handwritten Student Assessments

Hariny G

SSN College of Engineering, Chennai - 603 110

Professor C.V. Jawahar

International Institute of Information Technology, Hyderabad, Telangana- 500 032

Abstract

With the increase in the number of students enrolled in the university system, regular assessment of student performance has become challenging. The time taken to evaluate handwritten answers is very long because the physical movement of the paper from the student to the teacher and then back to the student is a time-consuming process as it involves a lot of other activities like manual entry of marks in a database. Our idea is to deploy an android app on tablets that lets the teachers use a stylus for grading. Answers are graded on the scanned image of the answer sheet and results are automatically stored from the grading of the answer sheet in a central repository. Our solution - Assessment Management System (AMS) Kiosk helps in scaling the handwritten student assessments in a typical university setting. We aim to simplify the evaluating process by reducing the movement of the answer sheet, making the process more transparent between teachers and students and also by providing feedbacks for every answer script. Parallel and distributed assessment by multiple instructors is straightforward in an electronic workflow system. At the heart of our solution, we have - a customizable android application for high through-put, image processing algorithms that improve the quality and readability, image annotation module that process the evaluations/feedbacks as a separate layer. A number of metrics are presented to the teachers during evaluation – writer identification (handwriting recognition), plagiarism detection (similarity between various answer scripts), neatness quotient which makes it easier for them to grade the answers. The system also deploys a recommendation system of the comments that can be given to a particular answer sheet based on these metrics. With the handwriting assessment workflow system, all these recent advances in computer vision can become practical and applicable in evaluating student assessments.

Keywords : Stylus, Central Repository, Distributed Assessment, Android Application, Image Annotation, Writer Identification, Plagiarism Detection, Neatness Quotient , Recommendation System, Computer Vision.

1 Introduction

1.1 Background

With the increase in the number of e-learning platforms and Online Judges, schools and universities are less inclined towards grading the handwritten answers. Handwritten assessments have been a very effictive way to evaluate students. It shows the organization of thoughts, original expressions in comparison to the electronically formatted solutions that does not show the fingerprints of a student. It is observed that for handwritten assessments, students do not receive any detailed feedback quickly for it to be helpful enough in their next assessment, because of the time delay involved in distribution, evaluation, entry of grades etc.

The time taken to evaluate handwritten answers is very long because the physical movement of the paper from the student to the teacher and then back to the student, is a time-consuming process as it involves a lot of other activities like manual entry of marks in a database. Faculty member provide questions and students bring their solutions to classroom or submit them at a fixed location. A teaching assistant assigned by the faculty member or the faculty member herself (instructors) grade the assessments and provide quantitative and/or qualitative feedback. Finally the grades are available to students, after a brief discussion between students and instructors about evaluation corrections.

1.2 Objective

Managing student assessments consume a significant portion of the effort of a teacher. Most of the assessments are now being automated, examples include multiple choice questions, matching two sets and output based computer program evaluation.

Our idea is to deploy an android app on tablets that would let the teachers use a stylus for grading. This android app uses Samsung SPen SDK[1]. Answers are graded on the scanned image of the answer sheet and results are automatically stored from the grading of the answer sheet in a central repository. Our solution - Assessment Management System (AMS) Kiosk helps in scaling the handwritten student assessments in a typical university setting. We aim to simplify the evaluating process by reducing the movement of the answer sheet, making the process more transparent between teachers and students and also by providing feedbacks for every answer script.

Parallel and distributed assessment[2] by multiple instructors is straightforward in an electronic workflow system. At the heart of our solution, we have - a customizable android application for high through-put, image processing algorithms that improve the quality and readability, image annotation module that process the evaluations/feedbacks as a separate layer. A number of metrics are presented to the teachers during evaluation:

- writer identification[3][4] (handwriting recognition),
- plagiarism detection[4] (similarity between various answer scripts),
- neatness quotient

which makes it easier for them to grade the answers. The system also deploys a recommendation system of the comments that can be given to a particular answer sheet based on these metrics. Students digitize the hand-written document with a mobile phone based interface. Instructors can grade/assess by annotating the images online. This simple yet effective connect between the physical paper world and electronic workflow makes our solution effective and efficient.

The focus of this paper is to demonstrate a scalable paperless grading system for handwritten assessments which allows electronic submission and on-screen grading of the assessments with high transparency between instructors and students.

2 Literature Review

Even in today's world, a large number of documents are generated as handwritten documents. A majority are handwritten because it is necessary. This is specially true when the knowledge or expertise cannot be captured conveniently with electronic gadgets. Information extraction from handwritten medical records [5] , reading postal address [6], grading of handwritten answer sheets[2] are examples where document image work flow helped in scaling the system with minimal human effort and intervention. In such work flow systems, images flow across subjects who can be in different places. A prescription for medicines written in Chennai by a doctor can be used and brought in Mumbai. Similarly a postal automation module in India can take help of a person in London to recognize the address block and still continue to be efficient.

Our work is motivated with the success of these document image workflow systems that were put into practice when the handwriting recognition[3] accuracy was unacceptably low.

Writer Identification - This is to identify documents containig more than one signature style. A student typically spends several years in college. Hence a single document from student can used as unique fingerprint/signature to identify his handwriting[4] across semesters. Several works have been done in the field of handwriting recognition[3][5-6][9]. They have also achieved good results regarding the recognition. They have been restricted to just the recognition of the handwriting i.e. the person who wrote it. [15]Handwriting is a behavioral biometric which can be used for writer identification and verification. This biometric can be applied in different domains such as security [13,14], forensic and historical document analysis. In addition, it can be used for the enhancement of handwriting recognition systems [6]. Writer recognition includes writer identification and writer verification. In writer identification, the writer of query handwriting should be determined among a number of writers for each of whom sample handwriting is available. In writer verification, the goal is to determine whether the writer of two handwritten texts is the same person or not.

Plagiaism Detection - By comparing two handwritten documents a human would be able to tell if it was written by the same person or different people. Several works have been done in this field too[3][10-11][17-18]. First is the language independent way which compares word to word with the selected set of target documents which are the sources of copied materials. Second is quite similar to document check but here the target document is the set of all documents that is reachable on Internet and candidate document is searched for characteristic text or sentence. The third type is stylometry in which a language analysis algorithm is used to compare the style of different paragraphs and report if a style change has occurred. This requires a prior analysis of candidate's previous documents[12].

Document Image Processing - It is a known fact that camera-captured images are prone to various degradations such as inadequate lighting, shadows, blur and camera flash at times. Such degradations often lead to difficulties in analysis at subsequent stages of image processing. For example, degradations may result in a significant drop in the performance of Optical Handwriting Recognition (OHR)[5][9][16], word spotting and other handwritten document analysis tasks, resulting in unrecoverable information loss. [2]The degradations introduced can be classified into (i)Character level - with broken characters, touching, skewed or curved handwriting, (ii)Page level - margin noise, salt-and-pepper, ruled line, warping, curling, skew, blur or translation.

Assessment Management - An online paper correcting system has been designed[2]. It employs the above mentioned Writer Identification, Plagiarism Detection and Document Image processing. Though the students in traditional learning management systems have the comfort of submitting the handwritten assessments from any location, the assessments still have to be compressed (zipped) and uploaded to a server. Instructors will have to download the file and then evaluate the handwritten or other file based assessments. For handwritten assessments, an android application which is used by students to take pictures of the assessments and upload them to server immediately. This can be very helpful in scenarios such as a surprise or spot assessment in class room. The android application tries to qualify the images based on the visual aesthetics of the uploaded handwritten document image.

The solution allows on-screen evaluation of uploaded handwritten assessments as an app is available where the teachers can login and correct the uploaded answer sheets. The instructor can highlight, annotate and comment on document images. These annotations are saved separately along with its image coordinate details. Since these annotations are immediately available to the students, they can immediately start a discussion with the instructors. The keywords from questions, assessment image and discussions together form a rich set of evaluation annotations for an assessment platform, which can be mined for patterns and reused while evaluating a similar assessment of other students[2].

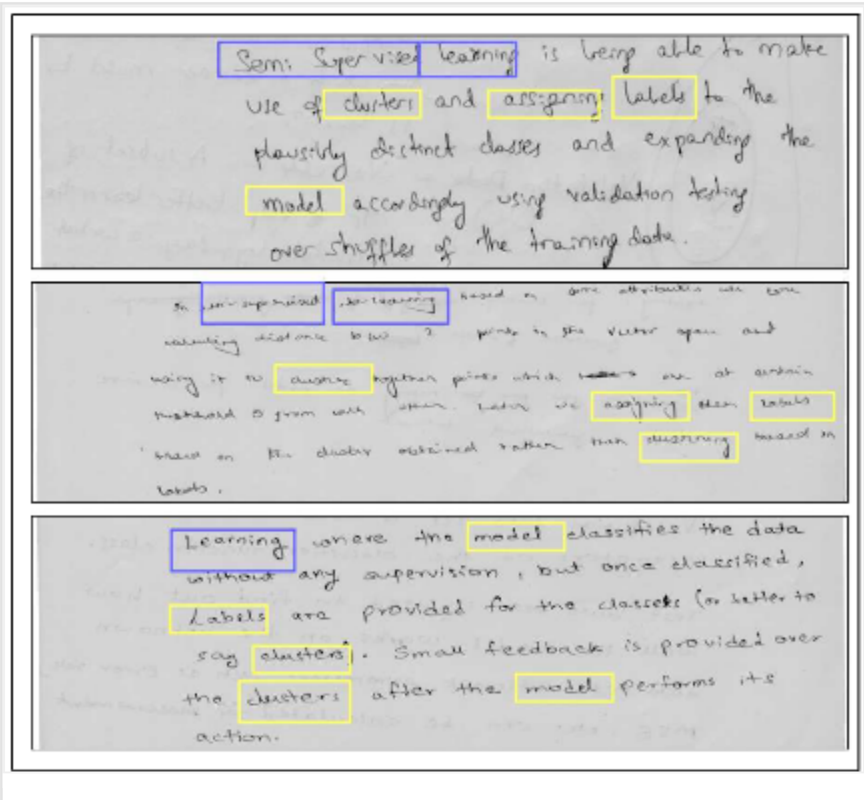


Fig 1 Annotation of answer sheets and highlighting of words.

The typical engineering homework assessment may involve sketches, formulas with special symbols, as well as calculation steps. The most time efficient way for students to do this work is by hand, on paper. The handwritten assessment of student will be available for further evaluation by instructors, using on screen evaluation tools or semi/auto evaluation methods which are still research problems as explained below.

The limitations of the present Assessment Management System are:

- Though Plagiarism detection, writer identification have been worked upon separately, it is not implemented in the current system.
- Only simple word spotting techniques are not sufficient we also need semantic techniques on handwritten text.
- In a university setting, tutors are required to evaluate several students and thousands of answers at a time. This can be cumbersome and any assistance provided to the instructors which can increase the throughput of evaluations will be a value-add.
- Though the students in traditional learning management systems have the comfort of submitting the handwritten assessments from any location, the assessments still have to be compressed (zipped) and uploaded to a server. Instructors will have to download the file and then evaluate the handwritten or other file based assessments.
- This only increases the time taken to evaluate the answer sheets and return it back to the student.

3 Methodology

We now start by looking at what can happen in a typical classroom scenario. Faculty member provide questions and students bring their solutions to classroom or submit them at a fixed location. A teaching assistant assigned by the faculty member or the faculty member herself (instructors) grade the assessments and provide quantitative and/or qualitative feedback. Finally the grades are available to students, after a brief discussion between students and instructors about evaluation corrections. In the following sections, we explain how our solution was designed to troubleshoot the pain points faced by instructors and students during the workflow process.

The focus is to demonstrate a scalable paperless grading system for handwritten assessments which allows electronic submission and on-screen grading of the assessments with high transparency between instructors and students.

3.1 Android Application - Grading App

For this we have developed an android application using Samsung SPen SDK [1]. The written answer sheets are uploaded using the present android application present[2]. Once the answer sheets are uploaded they scanned copies are stored in a central repository. The Grading App is used by the teachers to correct the uploaded answer sheets.

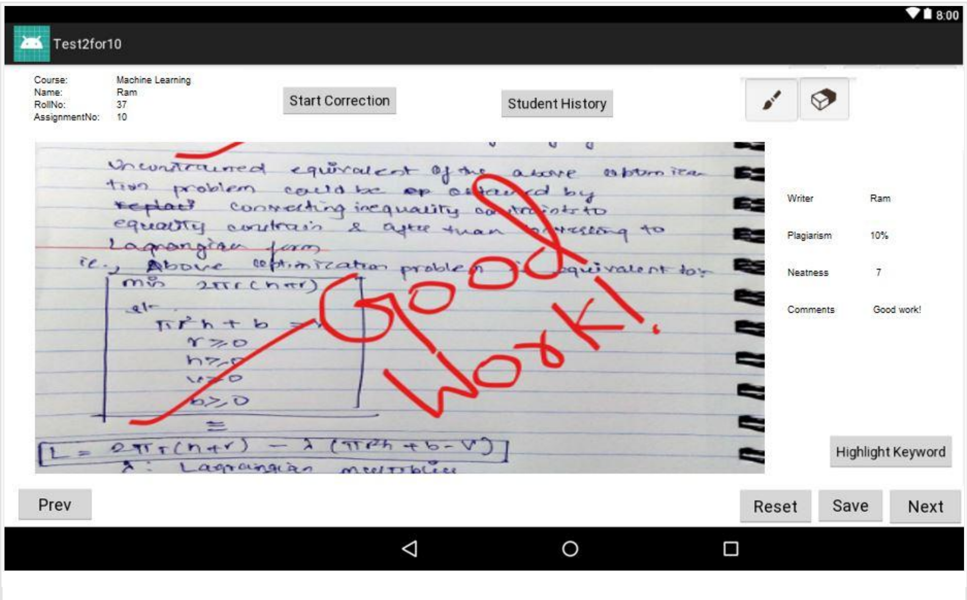
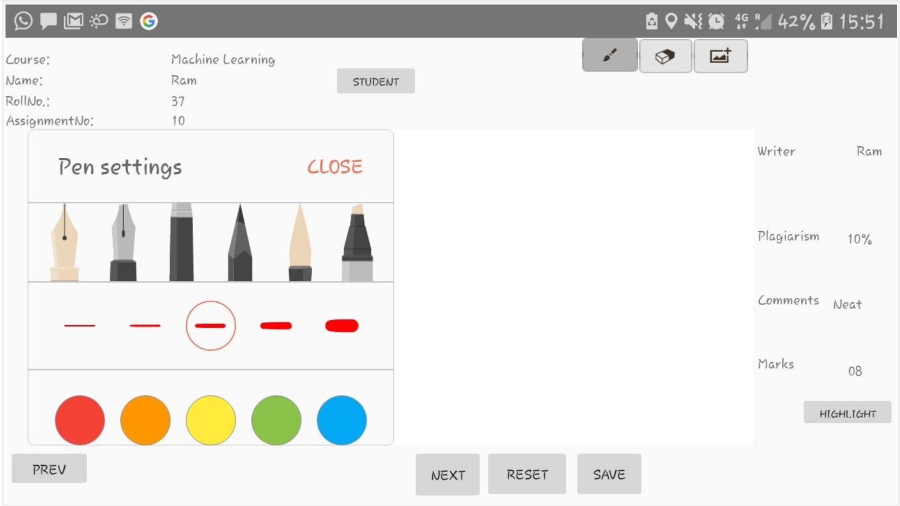


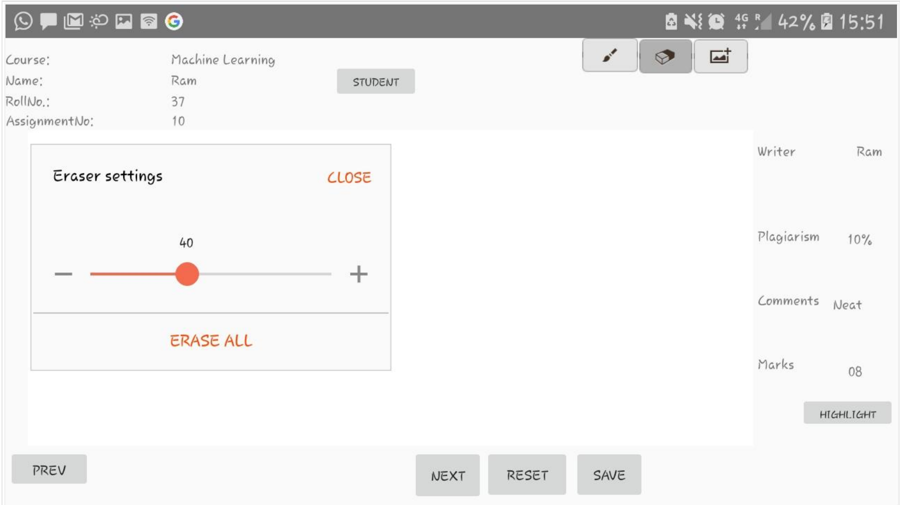
Fig 2 The Grading App- used by the teachers to correct the answer sheets. The teachers can use a stylus to correct the answer sheets. The app was emulated on an emulator.

The app has a lot of other features apart from letting the teacher correct the answer sheets. The app make use of a stylus for correcting the answer sheets. This gives the teachers an experience that is not different from the usage of a physical paper and pen to correct.The app is deployed on Samsung tablets on the devices that support the usage of a stylus. If the device does not support stylus, the teacher can correct using fingers becuse even finger strokes can be used to annotate or mark the answer sheets.

The app lets the user/teacher change the size of the pen stroke or the type of pen or the colour of the pen and also the size of the eraser to erase the strokes drawn in case the teacher wants to change it.



a) Pen Setting of the Grading App - lets you change the pen, pen colour and the size of the stroke.



b) Eraser Setting of the Grading App - lets you change the size of the eraser.

Fig 3 Different settings available in the Grading App.

There are Next and Prev buttons to lead the next and previous answer sheets. The images(answer sheets) are stored and loaded JSON files. The Save button saves the strokes i.e. the correction done as a separate layer and stores it as a bitmap so as to store as a JSON file so that it can be accessed from the central repository. Faculty member provide questions and students bring their solutions to classroom or submit them at a fixed location. A teaching assistant assigned by the faculty member or the faculty member herself (instructors) grade the assessments and provide quantitative and/or qualitative feedback. Finally the grades are available to students, after a brief discussion between students and instructors about evaluation corrections.

The image - answer sheet is loaded on SPenCanvasLayer[1] which allows the teachers to make corrections on the answer sheet as a separate layer and the underlying answer sheet picture is not modified.On top of the canvas (SPenCanvasLayer) a SPenPageDoc[1] and SPenNoteDoc[1] are present. These help in separating out the layers of answer sheets and the comments. The teacher corrects the answer sheet and then loads the next one after saving the current sheet. This way the corrections are stored in the server and there is no need for the teacher to spend separate time to manually enter the marks in a database or distribute the answer sheets which takes up a lot of time. Students digitize the handwritten document with a mobile phone based interface. Instructors can grade/assess by annotating the images online. This simple yet effective connect between the physical paper world and electronic workflow makes our solution effective and efficient. The android application tries to qualify the images based on the visual aesthetics of the uploaded handwritten document image.

The app also provides metrics like - Writer Identification, Plagiarism Detection and Neatness Score to help the teacher give the correct marks to the student. How these metrics are calculated is discussed in the later sections but they are loaded as JSON objects into the app. In traditional correction of answer sheets the teacher will have to know if the particular answer was written by the said student or not, by looking up other answer sheets of the student or by trying to identify and verify his/her handwriting manually, but this app provides the metrics thus reducing the time a teacher has to spend on any particular answer sheet.

An android application was designed to work with REST API (web services) , which also supports assisted image capture and image corrections. This application supports submission of hand-written answers, by allowing the capture of the hand-written document using the camera of the mobile device.

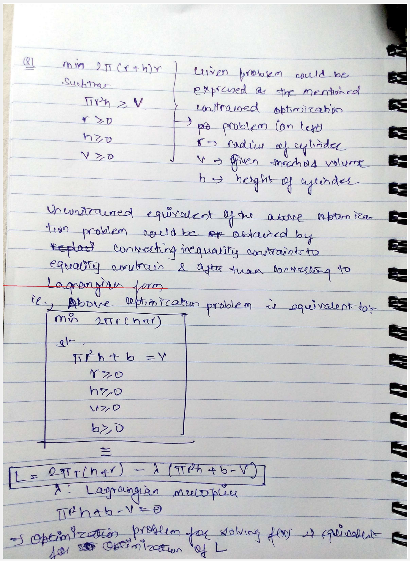
3.2 Dataset

The data is obtained by giving students the random testsand asking them to upload their scanned copies of the answer sheets and then use those as the required data to perform and calculate all the necessary features. This application supports submission of hand-written answers, by allowing the capture of the hand-written document using the camera of the mobile device.

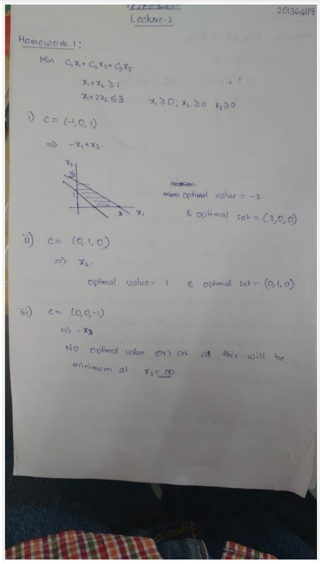
The degradations introduced can be classified into :

(i)Character level - with broken characters, touching, skewed or curved handwriting, (ii)Page level - margin

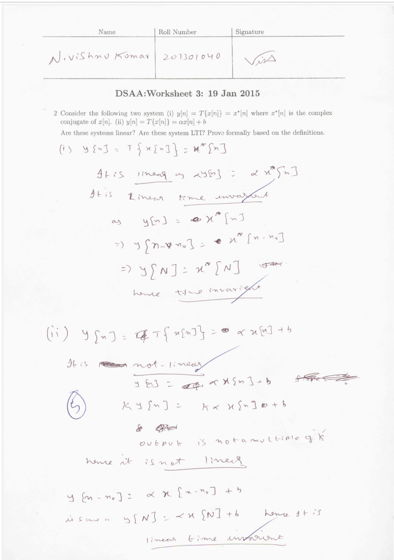
noise, salt-and-pepper, ruled line, warping, curling, skew, blur or translation. We focused on rectifying page level degradations.



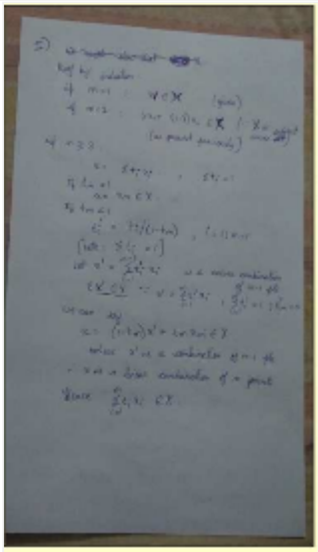
a) Answer Sheet sample 1



b) Answer Sheet sample 2



c) Answer Sheet sample 3



d) Answer Sheet Sample 4

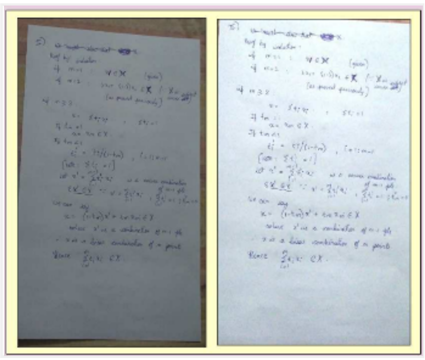
Fig 4 Dataset for the System - collected through the students when they write exams they are required to uplaod it.

3.2.1 Document Image Processing

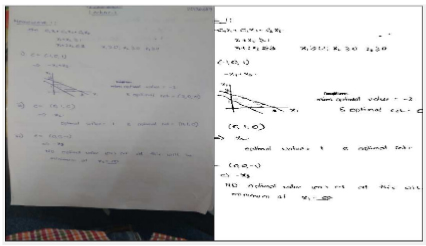
It is a known fact that camera-captured images are prone to various degradations such as inadequate lighting, shadows, blur and camera flash at times. Such degradations often lead to difficulties in analysis at subsequent stages of image processing. For example, degradations may result in a significant drop in the performance of Optical Handwriting Recognition (OHR), word spotting[24] and other handwritten document analysis tasks, resulting in unrecoverable information loss. If the data recieved is not upto the required standards then the calculation of all other metrics like Writer Identification, Plagiarism detection will not be very accurate.

The degradations introduced can be classified into

(i)Character level - with broken characters, touching, skewed or curved handwriting, (ii)Page level - margin noise, salt-and-pepper, ruled line, warping, curling, skew, blur or translation. Here our solution is concentrated on Page level degradations.

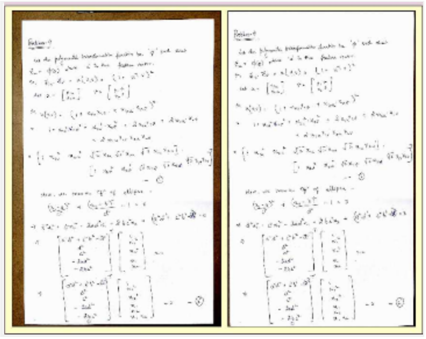


a) Processed Image 1

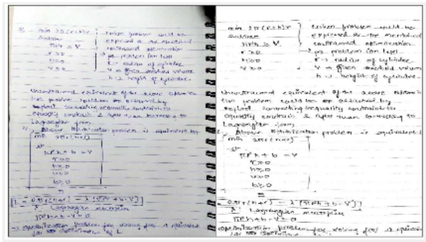


b) Processed Image 2

Fig 5 Original image and dewarped document image free from distortions (shadows and bends)



a) Processed Image 3



b) Processed Image 4

Fig 6 Original image with rule lines / bad illumination and dewarped document image free from distortions.

Compared to scanners, mobile cameras offer convenient, flexible, portable, and non-contact image capture, which enable better throughput in a document workflow management system. However, camera-captured documents may also suffer from distortions caused by non-planar document shape and perspective projection, which can lead to failure of current OCR/OHR technologies. The images were rectified based on the method explained in[25,26]. These methods share a similar hierarchical problem decomposition:

(i) Split the text into lines.

(ii) Find a warp or coordinate transformation that makes the lines parallel and horizontal. Though the cited methods were modeled for printed text, we observed that same methods worked well for camera-captured handwritten document images.

Some of the students submit there assessments in rule lined pages. Rule lines - both horizontal and vertical, should be removed to ensure better analysis at subsequent stages of image processing. We adapted methods described in [27] which uses rule line detection using Horizontal Projection Profile (HPP) and Hough Lines (HL).

The steps involved are:

(i) De-skew the image using method described in earlier section

(ii) Extract the location of horizontal lines using combination of HPP and HL

(iii) Remove the lines from the deskewed version of original document image and

(iv) Reconstitute the missing pixels.

3.3 Methods

First of all the handwritten answer sheets are captured and then they are processed as discussed in Sec. 3.2.1 . Only after the images are preprocessed, are they loaded into the Grading App for correction and also only then all the metrics are computed.

• Make the overall student assessment process efficient by removing paper movement, paper arrangements (eg. sorting pile of papers by student IDs) and additional data entry (manual entry of scores into a database explicitly).

• Bring correction/evaluation electronically as an extra annotation layer. This should enable parallel, distributed and multiple grading of the same student assessment.

• Incorporate a set of computer vision methods required to meet the immediate goal and keep the design open to introduce advanced image recognition modules at a later stage.

System Architecture:

The Assessment Management System architecture was designed with modularity, scalability and extensibility in mind. System transparency: This is implemented by processes such as double blind assessments, discussion forums, dashboards by profile hierarchies. The double blind procedure makes sure of unbiased evaluations and discussion between students and instructors. The queries and discussions on evaluations can be monitored down the work flow hierarchy.

Writer Identification -

Handwriting has continued to persist as a means of communication and recording information in day-to-day life even with the introduction of new technologies. Given its ubiquity in human transactions, machine recognition of handwriting has practical significance, as in reading handwritten notes or anwers. A student typically spends several years in college. Hence a single document from student can used as unique fingerprint/signature to identify his handwriting[4] across semesters. Several works have been done in the field of handwriting recognition[3][5-6][9].

Matching printed documents for retrieving the original documents and detecting cut-and-paste for finding plagiarism were attempted in the past by computing interest points in word images and their corresponding matches [29]. However, handwritten documents have large intra class[3] variability to reliably detect interest points. In addition, the problem of word overflow in which words from the right end of the document overflow and appear on the left end of the next line make the matching based on rigid geometry infeasible.

A similarity score is computed by detecting patterns of text re-usages between document images irrespective of the minor variations in word morphology, word ordering, layout and paraphrasing of the content. Our method does not depend on an accurate segmentation of words and lines. We formulate the document matching problem as a structured comparison of the word distributions across two document images. To match two word images, we propose a convolutional neural network (CNN)[29] based feature descriptor. Performance of this representation surpasses the state-of-the-art on handwritten word spotting. Finally, we demonstrate the applicability of our method on a practical problem of matching handwritten assignments .

Our objective fits well with the Siamese CNN[35] neural network architecture, which was first developed in 1993 to tackle the signature verification problem. [36] This type of architecture takes in two inputs and outputs a distance metric for the inputs. Bromley et al. was able to detect 95% of genuine signatures using this architecture.

To construct the pairwise dataset used to train our model, we separate writers used in our training, validation, and test sets. We do so in order to verify that the network is trained to be agnostic of specific author handwriting styles. For a positive match, we randomly sample a writer and then randomly sample two different lines written by that writer. For a negative mismatch, we sample two different writers and then randomly sample a line from each. We construct our datasets to have equal numbers of positive and negative examples. In a Siamese network, two inputs are taken and evaluated for a score. The two inputs are fed into Siamese convolutional networks that translate each image into latent encoding space. In this study, we vary these convolutional networks to determine which convolutional architecture produces the best encoding of handwriting. These latent encodings are then concatenated and used to produce class probabilities, or in this case the probabilities of whether or not two inputs are duplicates.

The writer for a particular answer sheet is identified and the data is sent to the front-end i.e. the Grading App as a JSON object and it is displayed next to the Writer label. This makes it easie for the teacher to correct the answer sheets by checking the writer and the original name provided in the information of the student.

Plagiarism Detection -

There are several cases where students do not write the answers by themselves but either copy the answers from their friends or the internet. When teachers are correcting the answers manually without any aids then it is difficult for them to identify if any answers were plagiarised. First they have to identify it and then cross-check it so that there are no mistakes. This is a very time-consuming process. To help teachers with this we have included a plagiarism detector in our solution.

While MOSS [31] automatically detects program similarity, it has no way of knowing why codes are similar. Systems like MOSS also use web-services for code comparison which makes them even more slow. It is still up to a human to go and look at the parts of the code that MOSS highlights and make a decision about whether there is plagiarism or not. Though we have integrated a custom code analyzer which uses sequence based models, it is limited to C language and better models are required to scale to large number of students.

First is the language independent way which compares word to word with the selected set of target documents which are the sources of copied materials.[3][17, 18] Second is quite similar to document check but here the target document is the set of all documents that is reachable on Internet and candidate document is searched for characteristic text or sentence. The third type is stylometry in which a language analysis algorithm is used to compare the style of different paragraphs and report if a style change has occurred. This requires a prior analysis of candidate's previous documents[12].

Neatness Score -

A subjective estimate of the visual aesthetic[37] features such as neatness of handwriting has many practical applications. For many decades, this has been used in education systems (in many countries) for promoting good handwriting by giving bonus points for neatly written solutions. In addition to this, there have been many studies and axioms that relate the quality and consistency of the handwriting to the personality of the individuals.

Our first observation has been that the evaluations by human beings is similar by and large. We analyze the human agreements on the aesthetic property of each document and observe that these variations are small. We also observe that there are more deviations in the scoring of average quality documents as compared to the excellent or poor ones. Using this human assessment for the quality of documents, [37] we are interested in training a SVM model that can be used to predict the human judgments on the aesthetic quality of handwritten documents. We also model the task of replicating human judgments on neatness as a comparative study between pairs of words and documents.

Fig 7 Metrics that are provided to the instructors to make the grading easier and a less time consuming process - Writer identification, Plagiarism detection, Neatness score and the comments.

Comments/Feedbacks Layer - Text Extraction and Recognition

Our solution allows on-screen evaluation of uploaded handwritten assessments. The instructor can highlight, annotate and comment on document images. These annotations are saved separately along with its image coordinate details. Since these annotations are immediately available to the students, they can immediately start a discussion with the instructors. Once the teacher finishes grading the answer sheets he/she has to save it. When the save button is clicked on the app then the comments and the annotations are stored as a separate layer and the original image of the answer sheet is not affected which allows multiple and parallel corrections.

So when the corrections are stored the comments are stored on a white background as shown in Fig 8.

Fig 8 Comments layer stored on a white background which is separated from the original image(answer sheet that is uploaded).

This image is now to be processed to get the text that is written so that a recoomendation can be given when the teacher is correcting the answer sheet.

Suppose the texts that are extracted are "Neat, Good, Write neatly, Correct, Wrong" so when the teacher corrects the next answer sheet these comments should be shown as a recommendation to the teacher to use it. This helps save time and also helps the teacher to not write the same comment repeatedly. The texts are to be extracted from the image and then tagged for further reference.

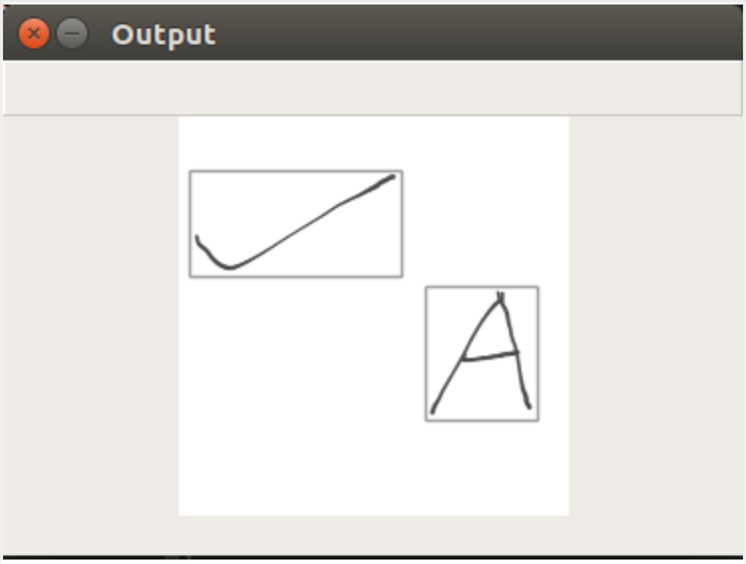
The combination of Python and OpenCV have been used to extract the text from the image. The image is saved once the Save button is clicked in the Grading App. The comments layer is then converted to GrayScale using OpenCV functions and libraries. The image is converted to Grayscale because it is easier to extract the text when the image is in grayscale than in the BGR colour scale.

An object recognition[33] algorithm identifies which objects are present in an image. It takes the entire image as an input and outputs class labels and class probabilities of objects present in that image. For example, a class label could be “cat” and the associated class probability could be 97%. On the other hand, an object detection algorithm not only tells you which objects are present in the image, it also outputs bounding boxes (x, y, width, height) to indicate the location of the objects inside the image.

At the heart of all object detection algorithms is an object recognition algorithm. Suppose we trained an object recognition model which identifies dogs in image patches. This model will tell whether an image has a dog in it or not. It does not tell where the object is located. To localize the object, we have to select sub-regions (patches) of the image and then apply the object recognition algorithm to these image patches. The location of the objects is given by the location of the image patches where the class probability returned by the object recognition algorithm is high.

Sliding window[32] play an integral role in object classification and detection, as they allow us to localize exactly where in an image an object resides. Utilizing both a sliding window we are able to detect objects in images at various scales and locations. In the context of computer vision a sliding window is rectangular region of fixed width and height that slides across an image. It is an exhaustive search for objects over the entire image. Not only do we need to search all possible locations in the image, we have to search at different scales. This is because object the text we are looking for may be of any size.

However in our solution we process the image to gray scale and then run a sliding window algorithm through it so as to extract the text (comments in our case). Then we check if the boundary of the window has or is touching any gray pixels or not. If it is then the image is not complete and so we continue searching for the image and increase the size of the window after one complete iteration through the image. Once we find a place where there are no gray pixels on the boundary then we calculate the ratio of number of grey pixels to white pixels. We then store it in a dictionary along with the (x, y) coordinates and the height and width of the window. After the algorithm has run for the required number of times we stop the sliding window and now sort the dictionary in descending order of the ratio of number of gray pixels to white pixels. We then visualise[34] the image from the data of pixels that we have stored.



a) Image output showing the regions of text in the image.

```
Total Number of Region Proposals: 15
[123  85  56  67]
[  0   0 195 200]
[  5  27 106  53]
[142  94  24  27]
[  8  29 100  48]
[126 116  50  33]
[145  94  22  26]
[123  86  56  66]
[  6  27 105  53]
[145  94  21  26]
[ 8 30 99 47]
[141  88  28  34]
[  6  27 105  52]
[  6  27 104  52]
[142  88  27  33]
0.13456304807983066
0.04792019021006447
0.042120004308951846
0.045182200046909315
0.040987253138358576
0.04329834856863409
0.04542486708329447
0.050506471581316825
0.0460042608403146
0.047818338054167596
```

b) Result showing the co-ordinates of the image and their ratio values.

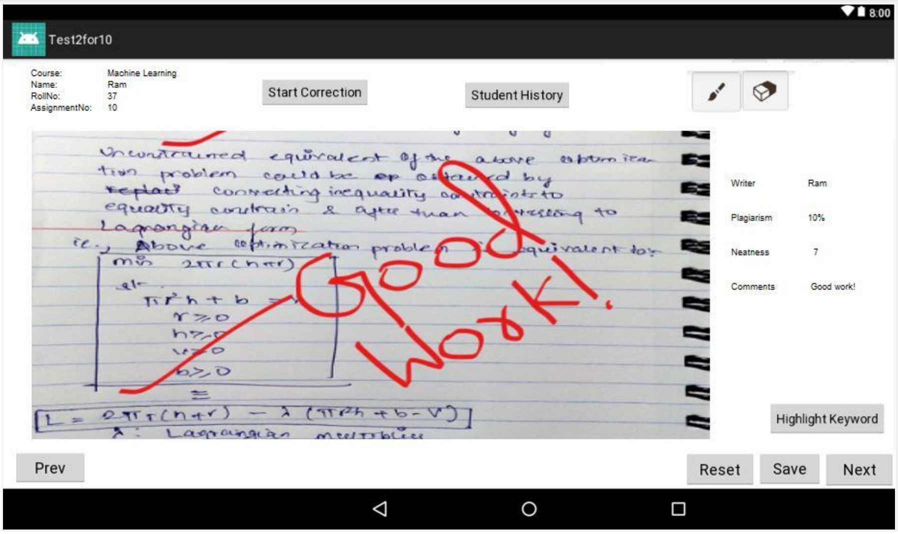
Fig 9 Result of sliding window.

We take an image as the input and output bounding boxes corresponding to all patches in an image that are most likely to be to be text. These region proposals can be noisy, overlapping and may not contain the object perfectly but amongst these region proposals, there will be a proposal which will be very close to the actual object in the image.

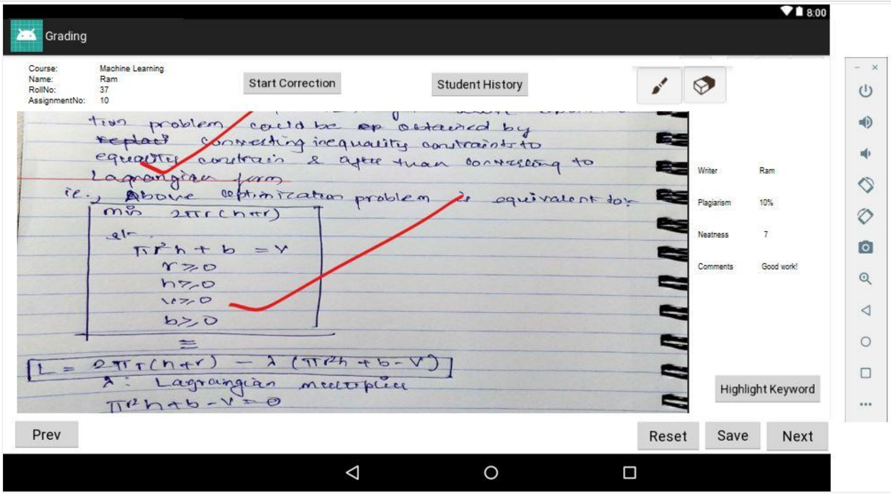
These texts are then used to give the recommendation of comments that show up in the Grading App as suggestions of comments that the teacher can use.

4 Results

After the completion of all the metrics and the Grading App, the app was tested on few answer sheets. The result were pretty accurate for all the metrics that are displayed on the app. All the metrics are calculated and are sent to the app as JSON objects and then they are loaded into the app. The comments that a teacher can write are also shown as a recommendation.



a) Result when the app was run on an emulator.



b) Result when the app was run on an emulator.

Fig 10 Result

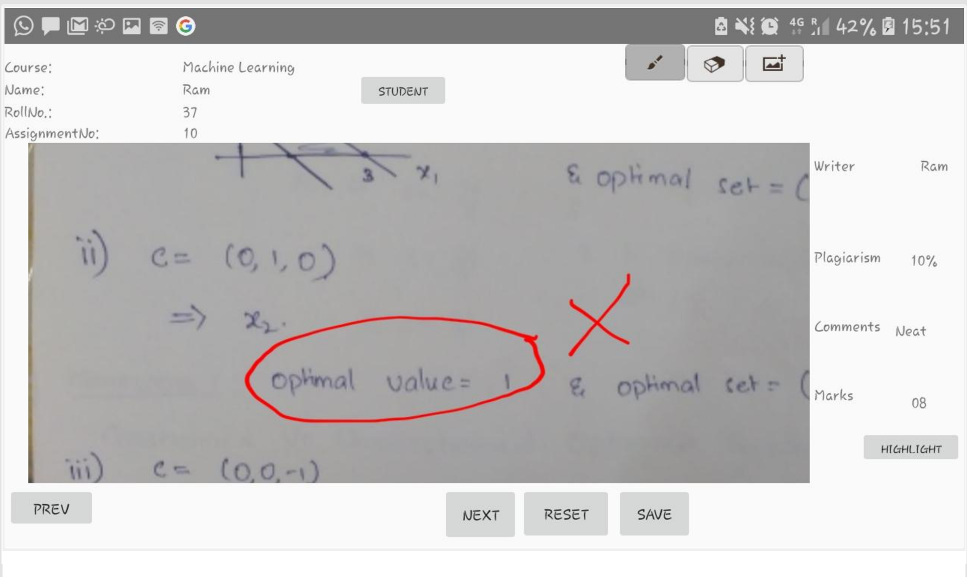


Fig 11 Grading App when run on a device.

We have simplified the evaluating process by reducing the movement of the answer sheet, making the process more transparent between teachers and students and also by providing feedbacks for every answer script.

Parallel and distributed assessment[2] by multiple instructors is straightforward in an electronic workflow system. At the heart of our solution, we have - a customizable android application for high through-put, image processing algorithms that improve the quality and readability, image annotation module that process the evaluations/feedbacks as a separate layer.

Various top research papers in handwritten and programming assessment space are evaluated and converted into research modules. These modules are first evaluated on smaller test sets and are finally plugged into the system. We have focused specifically on handwriting and programming space to assist the evaluators dealing with courses containing handwriting and programming assessments. Various in-house research projects are also integrated into the system. Our document workflow system also saves considerable time (average assessment time for class) when compared to manual handwritten paper based assessments. This is because few tasks can be skipped while using online assessments.

We make the overall student assessment process efficient by removing paper movement, paper arrangements (eg. sorting pile of papers by student IDs) and additional data entry (manual entry of scores into a database explicitly). We bring correction/evaluation electronically as an extra annotation layer. This enabled parallel, distributed and multiple grading of the same student assessment. We also incorporated a set of computer vision methods required to meet the immediate goal.

Our solution allows on-screen evaluation of uploaded handwritten assessments as an app is available where the teachers can login and correct the uploaded answer sheets. The instructor can highlight, annotate and comment on document images. These annotations are saved separately along with its image coordinate details. Since these annotations are immediately available to the students, they can immediately start a discussion with the instructors.

5 Future Work and Conclusion

Our solution allows on-screen evaluation of uploaded handwritten assessments as an app is available where the teachers can login and correct the uploaded answer sheets. The instructor can highlight, annotate and comment on document images. These annotations are saved separately along with its image coordinate details. Since these annotations are immediately available to the students, they can immediately start a discussion with the instructors.

But still we need a teacher to be present physically for correcting the answer sheets even though a lot of other metrics are provided for ease of correction.

Semi-automated evaluation: In a university setting, tutors are required to evaluate several students and thousands of answers at a time. This can be cumbersome and any assistance provided to the instructors which can increase the throughput of evaluations will be a value-add which is what our solution has proposed to do. Clustering based assessment techniques are available for text based assessments [3]. The method first trains a model on similarity metric between student responses, but then go on to use this metric to group responses into clusters and sub clusters. A similar method can be implemented for handwritten evaluations where segmented words can be clustered based on semantic similarity between students response and reference answer given by the instructor. Student responses can be queued from the clusters based on the similarity metric which can increase the throughput of evaluations. We call this semi-automated evaluation of handwritten assessments. We can make sure that keywords from an answer are detected and then they are highlighted in the answer sheet to make it easier for the instuctor to correct as they don't have to spend a lot of time bybjust reading the whole answer and then search for keywords in the answer. When keywords are highlighted, the grading process becomes easier and less time-consuming.

Fully Automated evaluation: Automated evaluation of handwritten assessments[7] can be seen as an extension to the our mentioned method, where assistance was restricted to clustering answers, queuing them and highlighting the keywords in assessments. This can be further enhanced provided that the reference answer is available. A regression model can be trained on a set of semantic word features [10] in visual space, which can predict an evaluation score similar to that of an instructor. The score may not be necessarily accurate but we feel that a nearest score with a confidence metric can boost the throughput of evaluations enormously. Just like in Online Judges where the answers or codes just need to be submitted and then the other algorithms like grading along with plagiarism detection begins, in fully automated correction too the handwritten answer sheets are meant to be corrected without any human intervention.

This paper proposes to make the grading of handwritten answer sheets easier and less time consuming at the university level where students are expected to write answer as well as submit codes online.We make the overall student assessment process efficient by removing paper movement, paper arrangements (eg. sorting pile of papers by student IDs) and additional data entry (manual entry of scores into a database explicitly). We bring correction/evaluation electronically as an extra annotation layer. This enabled parallel, distributed and multiple grading of the same student assessment. We also incorporated a set of computer vision methods required to meet the immediate goal. We have simplified the evaluating process by reducing the movement of the answer sheet, making the process more transparent between teachers and students.

Handwriting recognition has not reached a state that can directly help with the scalability of automated evaluations. However, we argue that our work flow system can enhance the efficiency and quality of the assessments without the need of Optical Handwriting Recognition (OHR). Our system presented in this paper addresses the need for a tool to computerize the existing handwritten assessments / answer sheets at all levels of our education system. Through this paper we tried to showcase the capabilities of our workflow system. To summarize, it has a useful set of tools which encompass existing technologies for text, code and handwritten assessments, which can enhance the tutors and students experience alike by minimizing the time required for the whole assessment management process. Our document workflow system also saves considerable time (average assessment time for class) when compared to manual handwritten paper based assessments. This is because few tasks can be skipped while using online assessments. Though the process is not yet perfect, the platform is open for future enhancements not only in text and handwritten work space but also in integrating research output from audio and video space.

6 Acknowledgment

I would like to express my deep sense of gratitude to Indian Academy of Sciences' (IASc-INSA-NASI) Summer Research Fellowship Programme for providing me an oppotunity to carry out this project. I would like to thank Prof. C.V. Jawahar for the constant guidance and encouragement and for the lab facilities he has provided throughout the research work. I extend my special thanks to Vijay Rowtula, a Master's student here at IIIT Hyderabad who mentored and guided me throughout the project. I would also like to thank all the managing staff here at the CVIT lab for their help. I would like to extend my deepest sense of gratitude towards my parents who have supported me with their valuable guidance.

7 References

[1] Samsung SPen SDK https://developer.samsung.com/galaxy/pen.

[2] http://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w47/Rowtula_Scaling_Handwritten_Student_CVPR_2018_paper.pdf.

[3] Praveen Krishnan, C. V. Jawahar, 2016, Matching Handwritten Document Images, Computer Vision – ECCV 2016,Lecture Notes in Computer Science, pp. 766-782.

4] D. Bertolini, L.S. Oliveira, E. Justino, R. Sabourin, 2013, Texture-based descriptors for writer identification and verification, Expert Systems with Applications, vol. 40, no. 6, pp. 2069-2080.

[5] R. Milewski, V. Govindaraju, Handwriting analysis of pre-hospital care reports, Proceedings. 17th IEEE Symposium on Computer-Based Medical Systems.

[6] SARGUR N. SRIHARI, 2000, HANDWRITTEN ADDRESS INTERPRETATION: A TASK OF MANY PATTERN RECOGNITION PROBLEMS, International Journal of Pattern Recognition and Artificial Intelligence, vol. 14, no. 05, pp. 663-674.

[7] Sargur Srihari, Jim Collins, Rohini Srihari, Pavithra Babu, Harish Srinivasan, 2006, Automated Scoring of Handwritten Essays Based on Latent Semantic Analysis, Document Analysis Systems VII,Lecture Notes in Computer Science, pp. 71-83.

8] Sargur Srihari, Jim Collins, Rohini Srihari, Harish Srinivasan, Shravya Shetty, Janina Brutt-Griffler, 2008, Automatic scoring of short handwritten essays in reading comprehension tests, Artificial Intelligence, vol. 172, no. 2-3, pp. 300-324.

[9] L. Schomaker, M. Bulacu, 2004, Automatic writer identification using connected-component contours and edge-based features of uppercase Western script, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 6, pp. 787-798.

[10] Soumajit Adhya, Setua S.K, 2016, Text Plagiarism Checker Using Friendship Graphs, International Journal of Computer Science and Information Technology, vol. 8, no. 4, pp. 13-21.

[11] 2016, Plagiarism checker for Persian (PCP) texts using hash-based tree representative fingerprinting, Journal of Artificial Intelligence and Data Mining, vol. 4, no. 2.

[12] Heinz Dreher, 2007, Automatic Conceptual Analysis for Plagiarism Detection, Proceedings of the 2007 InSITE Conference.

[13] Hao Feng, Chan Choong Wah, 2002, Private key generation from on-line handwritten signatures, Information Management & Computer Security, vol. 10, no. 4, pp. 159-164.

[14] Sanaul Hoque, Michael Fairhurst, Gareth Howells, 2008, Evaluating Biometric Encryption Key Generation Using Handwritten Signatures, 2008 Bio-inspired, Learning and Intelligent Systems for Security.

[15] Golnaz Ghiasi, Reza Safabakhsh, 2013, Offline text-independent writer identification using codebook and efficient code extraction methods, Image and Vision Computing, vol. 31, no. 5, pp. 379-391.

[16] S. Guberman, 1998, Off-line and online handwriting recognition - common approach, IEE Third European Workshop on Handwriting Analysis and Recognition.

[17] Jill Burstein, Susanne Wolff, Chi Lu, Randy M. Kaplan, 1997, An automatic scoring system for advanced placement biology essays, Proceedings of the fifth conference on Applied natural language processing.

[18] Chao Liu, Chen Chen, Jiawei Han, Philip S. Yu, 2006, GPLAG, Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06.

[19] G. Canfora, A. Cimitile, Algorithms for program dependence graph production, Proceedings of International Conference on Software Maintenance.

[20] Patrick M. ScanlonDavid R. NeumannInternet Plagiarism Among College StudentsI Patrick M. Scanlon David R. Neumann.

[21] CHRIS PARK, 2003, In Other (People's) Words: Plagiarism by university students--literature and lessons, Assessment & Evaluation in Higher Education, vol. 28, no. 5, pp. 471-488.

[22] Indrajit Bhattacharya, Prabir Ghosh, Swarup Biswas, 2013, Offline Signature Verification Using Pixel Matching Technique, Procedia Technology, vol. 10, pp. 970-977.

[23] Jill Burstein, Susanne Wolff, Chi Lu, Randy M. Kaplan, 1997, An automatic scoring system for advanced placement biology essays, Proceedings of the fifth conference on Applied natural language processing.

[24] Praveen Krishnan, Kartik Dutta, C.V. Jawahar, 2018, Word Spotting and Recognition Using Deep Embedding, 2018 13th IAPR International Workshop on Document Analysis Systems (DAS).

[25] Syed Saqib Bukhari, Faisal Shafait, Thomas M. Breuel, 2010, Performance evaluation of curled textline segmentation algorithms on CBDAR 2007 dewarping contest dataset, 2010 IEEE International Conference on Image Processing.

[26] F. Shafait and T. M. Breuel. Document image dewarping contest. In 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brazil, pages 181–188, 2007.Document image dewarping contest.

[27] F. Shafait and T. M. Breuel. Document image dewarping contest. In 2nd Int. Workshop on Camera-Based Document Analysis and Recognition, Curitiba, Brazil, pages 181–188, 2007.Document image dewarping contest.

[28] Sebastian Sudholt, Gernot A. Fink, 2016, PHOCNet: A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents, 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR).

[29] Ankit Gandhi, C.V. Jawahar, 2013, Detection of Cut-and-Paste in Document Images, 2013 12th International Conference on Document Analysis and Recognition.

[30] Chao Liu, Chen Chen, Jiawei Han, Philip S. Yu, 2006, GPLAG, Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06.

[31] A. Aiken. Measure of software similarity. URL http://www.cs. berkeley. edu/-aiken/moss. html, 1994.

[32] Jiaru Song, 2017, Sliding window filter based unknown object pose estimation, 2017 IEEE International Conference on Image Processing (ICIP).

[33] Naoyuki Gunji, Hitoshi Niigaki, Ken Tsutsuguchi, Takayuki Kurozumi, Tetsuya Kinebuchi, 2016, 3D object recognition from large-scale point clouds with global descriptor and sliding window, 2016 23rd International Conference on Pattern Recognition (ICPR).

[34] Yoshinobu Sato, Shin Nakajima, Nobuyuki Shiraga, Hideki Atsumi, Shigeyuki Yoshida, Thomas Koller, Guido Gerig, Ron Kikinis, 1998, Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images, Medical Image Analysis, vol. 2, no. 2, pp. 143-168.

[35] Du William Shen Margaret Siamese CNN for Authorship Verification

[36] JANE BROMLEY, JAMES W. BENTZ, LÉON BOTTOU, ISABELLE GUYON, YANN LECUN, CLIFF MOORE, EDUARD SÄCKINGER, ROOPAK SHAH, 1994, SIGNATURE VERIFICATION USING A “SIAMESE” TIME DELAY NEURAL NETWORK, Series in Machine Perception and Artificial Intelligence,Advances in Pattern Recognition Systems Using Neural Network Technologies, pp. 25-44.

[37] Anshuman Majumdar, Praveen Krishnan, C.V. Jawahar, 2016, Visual Aesthetic Analysis for Handwritten Document Images, 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR).