This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059342, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier XXXXXXXXXXXXXX

Generation of Synthetic Data for Handwritten Word Alteration Detection

PRABHAT DANSENA[1] (Student Member, IEEE), SOUMEN BAG[2] (Member, IEEE), AND RAJARSHI PAL[3] (Member, IEEE)

[1]Department of Computer Science and Engineering, Indian Institute of Technology (ISM) Dhanbad, 826004, India (e-mail: p.dansena23@gmail.com)
[2]Department of Computer Science and Engineering, Indian Institute of Technology (ISM) Dhanbad, 826004, India (e-mail: soumen@iitism.ac.in)
[3]Institute for Development and Research in Banking Technology, Hyderabad, 500057 India (e-mail: iamrajarshi@gmail.com)

Corresponding author: Prabhat Dansena (e-mail: p.dansena23@gmail.com).

ABSTRACT Fraudsters often alter handwritten contents in a document in order to achieve illicit purposes. At times, this may result in financial and mental loss to an individual or an organization. Hence, ink analysis is necessary to identify such an alteration. Convolution Neural Network (CNN) can be used to identify such cases of alteration, as CNN has emerged as a monumental success in the field of computer vision for varieties of classification tasks. But, CNN requires large amount of labeled data for training. Hence, there is a need to generate a large dataset for the experiments relating to handwritten word alteration detection. Collection, digitization, and cropping of a large number of altered and unaltered handwritten words are tedious and time consuming. To overcome such an issue, an approach for synthetic word data generation is presented in this paper for handwritten word alteration detection experiments. This scheme is designed in such a way that the synthetically generated words are very similar to the original ones. In order to achieve this, handwritten character data set is prepared using 10 blue and 10 black pens. These handwritten characters are used for creating synthetic word alteration data set. The presented approach uses relatively less number of handwritten character images to create a huge word alteration data set. Further, deep learning models are trained on the synthetically generated data set for word alteration detection.

INDEX TERMS Convolution Neural Network, Document forensics, Handwritten, Ink analysis, Synthetic data.

I. INTRODUCTION

MOST of the traditionally considered powerful ink analysis techniques require physical copy of the document for alteration detection. These techniques are destructive in nature, such as thin layer chromatography [1], high performance liquid chromatography [2], [3], and infrared spectrum analysis of diffuse reflectance [4]. Several non-destructive techniques based on hyperspectral imaging [5]–[9], Raman spectroscopy [10], [11], and luminescence lifetime [12] are also introduced subsequently. But, these techniques, involving spectral response, require special imaging devices. These devices are costly as well as not available with every organization. Therefore, researchers have also focused on image processing based techniques [13]–[22] on scanned document images, which are acquired at visible light spectrum. Hence after, this paper will discuss only the later category of techniques.

Moreover, alteration of handwritten documents can be classified into two categories: (a) addition of new words and (b) alteration of existing words. Several methods have been devised to detect addition of a new word in a handwritten document [13], [14], [18], [19]. In [13], statistical features such as circular color moments, coarseness, contrast, saturation weighted mean, and variance of hue are extracted from HSI color space of ink pixels. Difference between these features have been used to detect addition of new word. In [14], mean intensity, local binary pattern, and Gabor filter responses are extracted from RGB color space representation of ink pixels to form histogram. Further, distance between these histograms of pair of words is computed to discriminate inks in pair of words. A threshold is used to decide whether two different pen inks have been used or not. Addition of new word has also been identified using machine learning in texture feature analysis [18], [19]. In [18], a set of statistical features (mean, variance, skewness, kurtosis, and mean absolute deviation) are computed for each color channel in RGB representation of ink pixels for each word. A feature vector is formed by concatenating the features of two words.

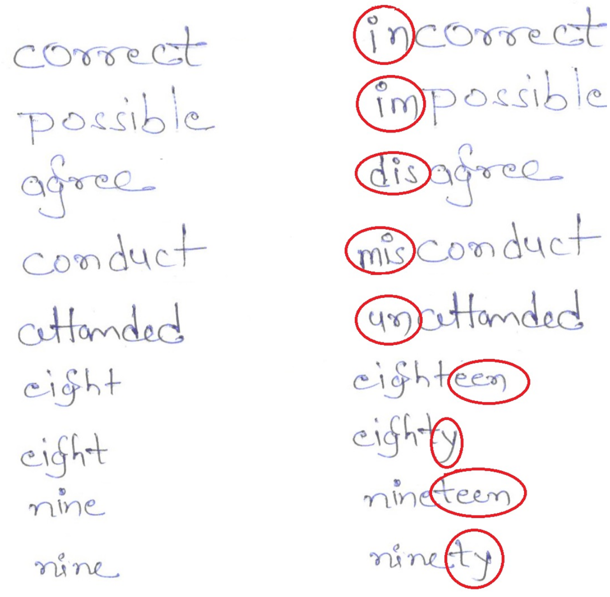
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059342, IEEE Access

This feature vector is fed to a multi-layer perceptron (MLP) for a two-class classification task. The MLP classifier is trained to determine whether a pair of words have been written using same ink or different inks. In [19], a set of features (mean, standard deviation, gabor filter, legendre, and geometric moments) is extracted from YC_bC_r representation of the word images. Further, MLP classifier is trained on the extracted feature set to identify whether pair of words are written using same pen or not. In [20], a quantitative assessment is presented to identify a suitable color model for detecting addition of new words. Further, a MLP classifier is trained on extracted statistical features for validating the selection of YC_bC_r color model.

On the other hand, for the second category of the problem, i.e., alteration in existing words (as illustrated in Fig. 1), a classification based approach is presented in [15]–[17] to detect alteration of existing handwritten word. These techniques use machine learning approaches (k-nearest neighbor, multi-layer perceptron, support vector machine) on hand-crafted texture features for alteration detection in existing words. But, key issue with these techniques is the requirement of manual selection of pen-stroke regions from a questioned word image for existing word alteration detection. Thus, the classification between altered and genuine words depends on correct selection of the pen strokes. In [21], another classification based method is proposed to detect alteration in existing word. In this method, a set of statistical features (harmonic mean, quartile deviation, variance, and skewness) has been extracted from pair of words. Then, MLP classifier has been used to identify whether any of the words has been altered or not. In [22], an automatic detection of possible word alterations is presented using Convolution Neural Network (CNN) based approach. This method skips the manual selection of pen stroke region. It uses CNN-based transfer learning to identify alteration in existing words.

Outcome of any machine learning model highly depends on the quality as well as the coverage of the data set. There is a readily available data set for word alteration detection system [23]. But size of the data set is very small (comprising of only 720 words). Though, data augmentation is found to be fruitful when data set size is small. Nevertheless, a robust recognition system, specifically a CNN-based method, demands a huge data set for better generalization of the problem. Thus, a need is felt to create a proper annotated data set for the problem. Collection, digitization, and segmentation of data set for the handwritten word alteration problem require a lot of time and effort.

In recent years, synthetic dataset creation methods have been found to be very fruitful in several contexts. Synthetic text dataset generation methods are used to create: (i) machine printed words and (ii) handwritten words. The machine printed text dataset generation schemes [24], [25], have been developed in order to build a good optical character recognition systems from printed texts. Therefore, these datasets are not suitable for ink analysis in handwritten texts. Moreover, handwritten synthetic dataset creation methods can be broadly classified into two categories: (i) on-line stroke movement based and (ii) off-line writing based. The stroke movement based technique [26] captures the movement and pressure while writing on a tablet screen. This dataset/technique is mainly used to study variability of handwriting patterns. Thus, this technique is also not suitable to create synthetic data set for ink analysis from off-line handwritten texts.



(a) (b)

FIGURE 1: Few samples of altered word images: (a) Genuine word images; (b) Corresponding altered word images marked with red circles indicating the altered portions.

Several off-line handwritten sample based techniques are also available for handwritten synthetic dataset generation [27]–[36]. In order to increase the variation in human writing samples while training recognition systems, various noise and deformations have been applied in [27]–[33]. In these methods, deformations have been introduced in writing samples to augment training samples. However, such deformations are not important from the perspective of ink analysis. Rather, these deformations have been used to create variability in the dataset in order to build a robust character recognition system. Moreover, in all these datasets [27]–[36], combinations of two different pens have not been considered to generate a single handwritten word. Thus, these datasets can not be used for handwritten word alteration detection experiments. This highlights the need for creating a new synthetic dataset for emulating handwritten words for document alteration detection experiments.

In [28], a character concatenation based method is introduced for synthetic word creation. In this method, different character sample features such as glyph, size, slant, etc. are considered while rendering to capture the writer-based features. This method concatenates the characters on the base line after geometric deformation. Further, polynomial interpolation is used to join adjacent characters. In [29], a combination of handwritten and on-line sample is used to create large Arabic handwritten word samples. This method collects multiple instances of each character on-line. Further, words are formed by concatenation of characters at appropriate position based on the start and end points of characters. In [32], characteristic of real Chinese handwriting words such as character spacing, slant, and aspect ratio are computed based on number of characters in a word. Further, these features are considered during synthetic word generation and characters are concatenated based on gravity center of segmented characters. In [33], a synthetic text generation method is proposed to increase training sample size for handwritten Japanese text recognition. In this method, segmented characters are deformed with local elastic distortion and distorted characters are concatenated to form synthetic words. Moreover, global distortion is performed to increase the variability in writing samples. A method for synthetic handwritten Arabic text dataset creation is based on concatenation of characters [34]. In this method, set of characters has been collected from available Arabic language handwriting dataset. Next, these characters are concatenated in appropriate position to synthesize words for handwriting analysis. Moreover, few neural network based techniques [35], [36] are also available. A method is proposed in [35], to create synthetic English and Farsi digits. In this method, neural network generative model is used to create synthetic digit images with sufficient fidelity and diversity. In [36], Generative Adversarial Network (GAN) is also successfully used to create synthetic handwritten French and Arabic word images. These neural network based techniques also focus on the variation in writing style and structure of handwritten words. These variations are not important from the perspective of pen ink analysis in handwritten word alteration detection.

In this paper, a novel method is proposed to synthetically generate handwritten word alteration data set. This data set is used for handwritten word alteration detection experiments using pen ink analysis. It has been observed that CNN-based approaches outperform other approaches for variety of classification tasks, like medical image classification [37], [38], plant disease classification [39], and optical character recognition (OCR) [40]–[42]. Training a CNN model requires huge amount of annotated data. Due to the unavailability of a proper data set for training a CNN model for handwritten word alteration detection task, a synthetic handwritten word alteration data set generation method is proposed in this paper. This is the first attempt to create a synthetic data set for handwritten word alteration detection task. Further, to validate the suitability of the synthetic word generation scheme, four CNN architectures are trained on the synthetically generated data set and related experimental results are reported in this paper. It is to be noted that this paper deals with words in English language only.

The main contributions of this paper are listed below:

- A scheme to generate synthetic word alteration data set is proposed to detect handwritten word alteration using pen ink analysis.
- The proposed scheme concatenates the most suitable character images to generate words. Hence, an effort is made to form the synthetic words with a realistic appearance. Thus, the generated words look more realistic.
- Further, this data set is used to train the CNN architectures for validation of the proposed scheme.

The rest of the paper is organized as following: The proposed methodology for synthetic word dataset generation scheme is presented in Section II. Section III discusses the experimental setup for data set creation and validation results. Finally, the concluding remarks are given in Section IV.

II. PROPOSED METHODOLOGY FOR SYNTHETIC WORD DATASET GENERATION

A word can be viewed as an ordered set of characters. Considering this fact, a large number of handwritten character samples are acquired. Segmented character samples are then used to create altered and genuine word images. This section presents the details of the proposed approach for synthetic word image dataset generation. The words in this dataset look like handwritten words.

A. COLLECTION OF CHARACTER SAMPLES

Characters are the building blocks for a word. Characters are concatenated to compose words. Ten blue and ten black ink pens are used here to write the characters in English alphabet. As this dataset is created for the experiments on pen ink differentiation for handwritten document forensics, twenty different pens are used for this task. Differentiating between similar color inks (e.g., blue versus blue) is tougher than differentiating between inks of two different colors (e.g., blue versus black). Hence, ten different pens are used for each of blue and black inks. Moreover, blue and black inks are heavily used in official documents as compared to any other colored inks (e.g., red or green). Hence, similar to other studies from literature, only blue and black colored pens are used for dataset creation. List of these pens is presented in Table 1.

In order to create character samples, 20 volunteers have actively participated. Each pen is associated with a volunteer. Each pen is used to write 200 samples of each character in a single plain A4 size paper. Each character sample is written in lower case. Variation in one's handwriting is natural phenomenon. Therefore, 200 samples are written for each character using a single pen by a single volunteer. This provides an option to select the most appropriate (in terms of size) character during a later concatenation stage. Thus, a total of 4000 samples (200 samples for each character per pen $\times$ 20 pens) for each character are collected. Half of these characters are written in blue ink and remaining half of these are written in black ink. Moreover, a collection of 4000 samples of each character facilitates the creation of a large dataset of handwritten words involving various pen

TABLE 1: List of pens used to create character samples.

Pen ink color	Pen_ID	Pen brand
Blue	P_1-P_{10}	Win Guide, Cello Pin Point, Link Celeb, Cello Papersoft, Link Sensor, Classmate Octane, Link Maestro, Flair Max Touch, Cello Butterflow, Montex Regal
Black	P_{11}-P_{20}	Rorito Jottex Classic, Flair Writo-Meter, Doms Trio-Matic, Cello Pin Point, Link Glycer L.V., Elkor Oxer, Cello Finegrip, Win-Duke, Cello Butterflow, Link Glycer 10x

inks. At the end of this stage, these handwritten characters are digitized using a normal scanner (Canon E560) at 300 dpi (dots per inch).

B. PRE-PROCESSING OF CHARACTER IMAGES

In order to separate foreground ink pixels from background, k-means clustering (k=2) has been performed in the gray scale version of the images. k-means clustering of intensities of the pixels in an image is also used for separating foreground and background pixels in several of the earlier works [18], [20]. Though, any good binarization technique can be adopted here. But this paper does not intend to study the effect of binarization on the proposed work. Hence, a simple k-means clustering (k=2) technique is adopted here. Further, each background pixel is assigned a value 255 for all three color channels. Foreground pixels retain their original color values. Retaining foreground pixel colors is necessary for ink analysis for handwritten word alteration detection. Moreover, background suppression is required to remove the effect of background colors in a CNN-based approach for word alteration detection. This approach is carried out for background suppression on each of the A4 size paper (containing 200 character samples using a single pen). Next, minimum bounding boxes for foreground ink pixels of character samples are identified automatically using connected component analysis. These identified bounding boxes are used to crop each character sample. This entire process is exhibited in Fig. 2. It is to be further noted that few character samples are inherently disconnected such as ‘i’ and ‘j’. Moreover, it has been observed that the character ‘k’ sometimes appears disconnected in the collected handwritten character set. Therefore, a dilation morphological operation is carried out in order to join the disconnected components of these characters as part of a single character. The structuring element for the dilation operation and its effect on a sample character are presented in Fig. 3. Once the disjoint components are joined using such a morphological operation, the connected component based approach can be applied to identify the minimum bounding box for the character. This morphological operation is carried out only to identify the coordinates of minimum bounding box. The actual character segment is extracted from the background suppressed image of the A4-sized page containing characters. This special treatment for one of these characters is pictorially depicted in Fig. 4.

C. GENERATION OF SYNTHETIC WORDS WITH COLLECTED CHARACTERS

As mentioned earlier, words are ordered set of characters. Similarly, word images can also be viewed as concatenation of character images. Based on this fact, synthetic word images are generated using segmented character images. It has been observed that direct joining of the randomly selected character images to construct synthetic words may have following issues: (i) inappropriate height of different character images (as in Fig. 5) and (ii) incorrect placement of different characters (as in Fig. 6). It can be seen in Fig. 5 that heights of the characters ‘n’ and ‘w’ are disproportionately more than the character ‘o’. The character images cannot be resized here, as the interpolation of pixel color values may create a new color value. The statistical properties of color values in foreground pixels of a character image will change. Hence, an appropriate character image selection process is proposed in this work. Another example can be seen in Fig. 6, where aligning all the characters in the word ‘ingress’ with a single base line created an incorrect formation of the word. A character placement procedure is proposed to handle this issue. Thicknesses of the strokes of characters are not considered because they are observed to be more or less similar. Therefore, height of a character is only considered. Both procedures of appropriate character selection and proper placement of it are based on the categorization of the characters as discussed next.

1) Categorization of characters based on zone coverage

Portion of any word can fall into three zones, i.e., upper, middle, and lower zone (as in Fig. 7). It has been observed that several characters cover single zone, whereas few characters require two zones. English alphabet (a-z) (presented in Fig. 7a), can be categorized into three groups based on the zones being covered by it: **(i)** Group-m (G_m): characters falling in middle zone only (Fig. 7b), **(ii)** Group-u (G_u): characters falling in middle and upper zones (Fig. 7c), and **(iii)** Group-l (G_l): characters falling in middle and lower zones (Fig. 7d). In the light of the categorization of alphabets, selection of appropriate characters as well as correct placement of these have been performed.

2) Selection of appropriate characters

Let a word be comprised of n characters. The corresponding word image W can be conceptualized as a concatenation of n character images, as $W = C_1^{k_1} \oplus C_2^{k_2} \oplus ... \oplus C_i^{k_i} \oplus ... \oplus C_n^{k_n}$. Here, $k_i \in \{1,2, ... , 200\}$ indicates the specific character

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059342, IEEE Access

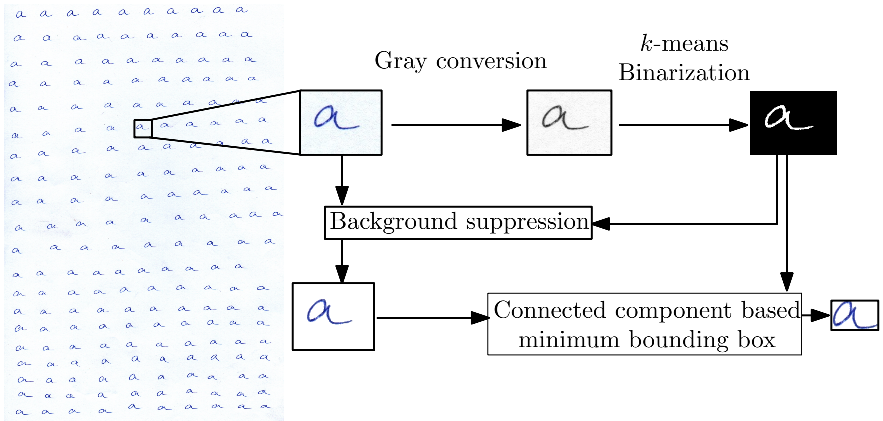


FIGURE 2: Illustrative example of background suppression on character image and its segmentation.

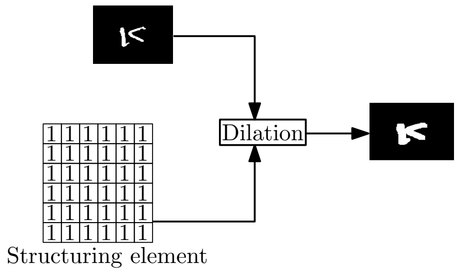


FIGURE 3: Illustrative example of dilation operation on a character sample.

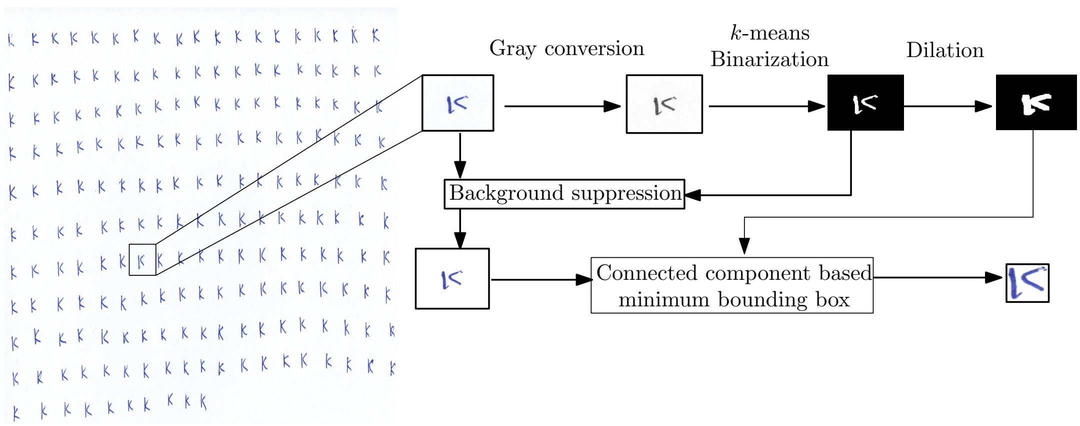


FIGURE 4: Illustrative example of background suppression on disconnected character image and its segmentation.

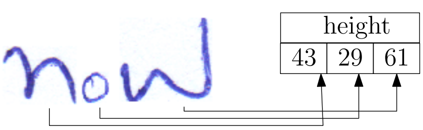


FIGURE 5: Height mismatch of different characters in a word ‘now’. (Heights are given as number of rows in the image as unit).

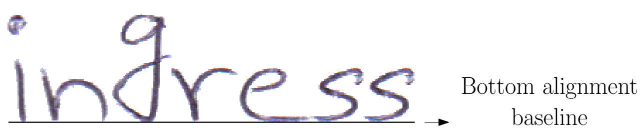


FIGURE 6: Incorrect placement of different characters in a word ‘ingress’.

image among 200 options of using a specific pen for the i^{th} character image of W. Initially, the first character image is randomly chosen to be any one of the 200 images using a specific pen with equal probability. If the first character of the word belongs to the group G_m, then the height of the first chosen character image, *height*$(C_1^{k_1})$, is considered as h_m (height of the middle zone for the word to be created) to select subsequent character images to form the word image. Otherwise, if the first character belongs to either group G_u or

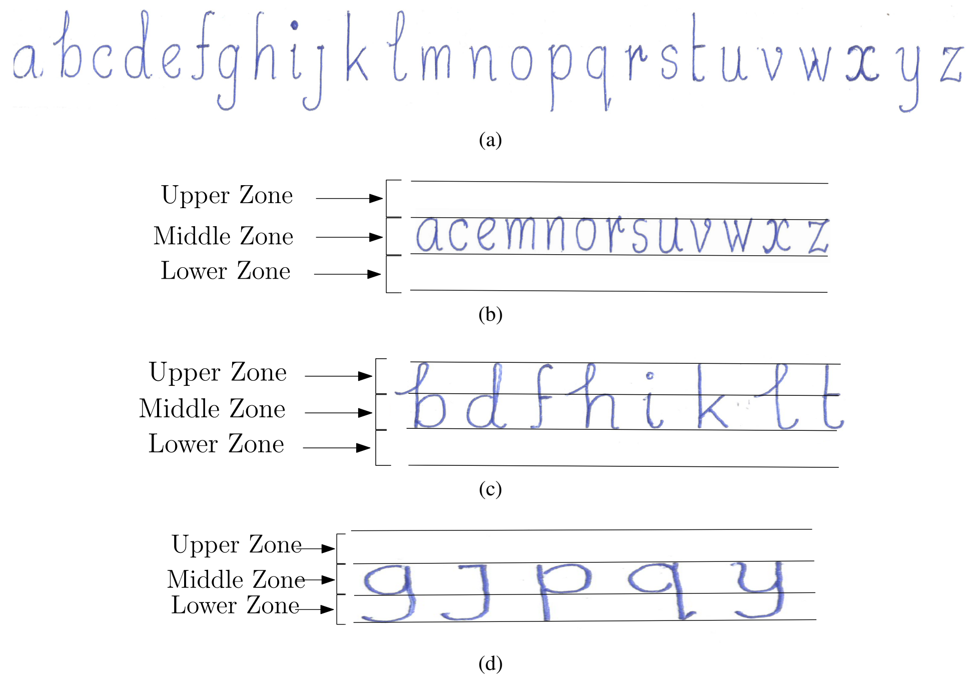


FIGURE 7: Grouping of characters in the English alphabet based on zone coverage: (a) Set of handwritten lower case alphabets; (b) Group-m (G_m) alphabet set; (c) Group-u (G_u) alphabet set; (d) Group-l (G_l) alphabet set.

group G_l, $\lfloor$*height*($C_1^{k_1}$)/2$\rfloor$ is considered as h_m. The symbol $\lfloor . \rfloor$ indicates the greatest integer which is lower than the argument. Further, any subsequent character image $C_i^{k_i}$ with height *height*($C_i^{k_i}$) is chosen from a pool of 200 images of the same character (using a specific pen) in the following manner:

- If the character is from G_m, then the character image with minimum difference between the height of the character image and the height of the middle zone, i.e., $|$ h_m - *height*($C_i^{k_i}$) $|$, is chosen for the next character. The symbol $|.|$ indicates the absolute value of its argument.
- Otherwise, the character image with minimum difference between the half of the height of the character image and the height of the middle zone, i.e., $|$ h_m- $\lfloor$*height*($C_i^{k_i}$)/2$\rfloor$ $|$, is chosen for the next character.

This process is opted for choosing the most appropriate character image sequence for the target word. Algorithm 1 depicts the steps of choosing appropriate character images for any word. Thereafter, next step is concatenation of these chosen character images by considering their correct placement. It is described in the following subsection.

Algorithm 1: CHARACTER IMAGE SELECTION: Identifies the appropriate character images for word creation

Input: sequence of n characters in a word
Output: sequence of selected character images {$C_1^{k_1}$, $C_2^{k_2}$,..., $C_n^{k_n}$}
Randomly select $C_1^{k_1}$
if $C_1^{k_1} \in G_m$ **then**
  h_m = *height*($C_1^{k_1}$)
else
  h_m = $\lfloor$*height*($C_1^{k_1}$)/2$\rfloor$
for $i \leftarrow 2$ *to* n **do**
  if $C_i^{k_i} \in G_m$ **then**
    Choose character image $C_i^{k_i}$ such that $|$ h_m - *height*($C_i^{k_i}$) $|$ is minimum
  else
    Choose character image $C_i^{k_i}$ such that $|$ h_m - $\lfloor$*height*($C_i^{k_i}$)/2$\rfloor$ $|$ is minimum
return {$C_1^{k_1}$, $C_2^{k_2}$,..., $C_n^{k_n}$}

3) Placement and adjustment of selected character images

Let the selected sequence of character images be {$C_1^{k_1}$, $C_2^{k_2}$,..., $C_n^{k_n}$}. As discussed above, direct placement of these selected character images side-by-side may result in visually incorrect word (as in Fig. 6). Thus, placements of these images are adjusted to eliminate this issue. In order to generate visually correct image, two types of adjustments are computed based on: (i) zone and (ii) height. Zone-based adjustments are computed to place each of the selected image appropriate to zone.

At first, bottom alignment-based concatenation of selected character images is performed for generation of artificial word images (as in Fig. 6). If n images of sizes $x_1 \times y_1$, $x_2 \times y_2$, ..., $x_n \times y_n$ are concatenated in this process, then

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059342, IEEE Access

the size of the concatenated image is $x_c \times y_c$. Here, x_c = max(x_1, x_2, ..., x_n) and y_c = (y_1 + y_2 + ... + y_n). Here, x_i and y_i indicate the number of rows and columns, respectively, in i^{th} image. Following steps describe the process of bottom alignment-based concatenation of character images.

Step 1: An image with white color (having value 255 at each color channel for each pixel) of size 1000 $\times$ 1500 is created which can accommodate the word to be created. This size is chosen to facilitate the formation of each word by considering the number of characters and the sizes of constituent character images.

Step 2: Row number 500 is considered as a base line to create the synthetic word. This row number 500 is chosen to accommodate the character images within the image created in Step 1.

Step 3: Each of the selected i^{th} character image of size $x_i \times y_i$ is placed in the white image with the coordinates of the top-left corner being (500-x_i+1, 1+$\sum_{j=1}^{i-1} y_j$).

At this stage, only character images belong to G_l are not concatenated appropriately. In order to handle this issue, zone-based adjustment computation is performed based on the zone-wise grouping of characters. Any character image belonging to G_l is shifted downwards by half of its height. This zone-based adjustment is carried out if any character from G_l exists in the word. The computational steps to carry out this zone-based adjustment are given here:

Step 1: Height of the i^{th} character image, which belongs to G_l, is computed as *height*($C_i^{k_i}$).

Step 2: To create a realistic synthetic word, the required downward shift of the i^{th} character is computed as $H_{C_i^{k_i}}$= $\lfloor$*height*($C_i^{k_i}$)/2$\rfloor$.

Step 3: Each character image belonging to G_l is shifted downwards and is placed at top-left coordinate being (500-x_i+1+$H_{C_i^{k_i}}$, 1+$\sum_{j=1}^{i-1} y_j$).

Moreover, in order to further fine tune the placement of the selected characters, height-based adjustments are also computed. Although, zone-based adjustment is performed to generate visually correct synthetic words, but due to the inappropriate height of the selected character images, generated synthetic word images may look visually incorrect. Therefore, to fine-tune the synthetically created words, selected character images are further aligned using height-based adjustments. The steps of this height-based adjustment process are stated here:

Step 1: Height of each of the character image is computed as *height*($C_i^{k_i}$).

Step 2: Effective middle zone height is computed for each of the character image belonging to either G_u or G_l as $H_{C_i^{k_i}}$= $\lfloor$*height*($C_i^{k_i}$)/2$\rfloor$. Moreover, for each of the character image belonging to G_m, $H_{C_i^{k_i}}$= *height*($C_i^{k_i}$) is considered as effective middle zone height.

Step 3: Based on the effective middle zone height of the 1^{st} character image, height based adjustments for subsequent characters are computed as $adj_{height}[i] = \lfloor (H_{C_1^{k_1}} - H_{C_i^{k_i}})/2 \rfloor$.

Step 4: If height based adjustment for i^{th} character image is positive ($adj_{height}[i]$>0), then upward shift of the character image is performed by adj_{height} rows. Otherwise, if height based adjustment for i^{th} character image is negative ($adj_{height}[i]$<0), then downward shift of the character image is performed by $|adj_{height}|$ rows.

Algorithm 2 depicts the steps to compute the zone and height-based adjustments as adj_{zone} and adj_{height} , respectively, to create synthetic word image. Finally, the foreground portion of the word image is cropped using a minimum bounding rectangle encompassing the ink pixels. For this, k-means clustering of color values at the pixels are carried out to identify the foreground pixels (as mentioned in Section II-B).

Algorithm 2: ADJUSTMENT COMPUTATION: Computes the zone and height-based adjustment for characters placement

```
Input: W={C_1^{k_1}, C_2^{k_2},..., C_n^{k_n}}, n
Output: adj_zone[n] adj_height[n]
adj_zone[1 : n] ← 0
adj_height[1 : n] ← 0
for i ← 1 to n do
    if C_i^{k_i} ∈ G_m then
        H_{C_i^{k_i}} ← height(C_i^{k_i})
    else
        H_{C_i^{k_i}} ← ⌊height(C_i^{k_i})/2⌋
    if C_i^{k_i} ∈ G_l then
        adj_zone[i] ← H_{C_i^{k_i}}
    if i >1 then
        adj_height[i] ← ⌊(H_{C_1^{k_1}} − H_{C_i^{k_i}})/2⌋
return {adj_zone[n], adj_height[n]}
```

Above mentioned selection of appropriate character images and corresponding placement adjustments (adj_{zone} and adj_{height}) are considered to achieve realistic creation of word images synthetically. It is to be noted that zone-based adjustment (adj_{zone}) will result in either positive integer value (for the characters in group G_l) or 0 (for the characters in other two groups). Moreover, each of the character images need to be shifted downwards by the number of rows computed as adj_{zone}. The height-based adjustment (adj_{height}) may result in either positive, zero, or negative integer values. The height-based upward or downward shifts are performed as per the positive or negative values of adj_{height}.

Finally, the effort of forming a word image using a sequence of character images is summarized using the following steps:

Step 1: A sequence of appropriate character images {$C_1^{k_1}$, $C_2^{k_2}$,..., $C_n^{k_n}$} is identified using Algorithm 1.

Step 2: The position adjustments $adj_{zone}[n]$ and $adj_{height}[n]$ are computed for each character of the word using Algorithm 2.

Step 3: The character images $C_1^{k_1}$, $C_2^{k_2}$,..., $C_n^{k_n}$ are concatenated with proper shifting based on $adj_{zone}[n]$ and $adj_{height}[n]$ to create the synthetic word image.
Pictorial representation of selected character images for example words ('unplanned' and 'misfit') and synthetically created words are presented in Fig. 8. This technique has been used to create both genuine and altered word image samples. A word image is considered as genuine, if each character image is selected from the pool of images using the same pen to form the word image. Moreover, if few character images are chosen from the pool of images of a single pen and remaining character images are selected from the pool of images for a different pen to create the word, then the word image is considered as an altered word sample. Creation of both types of word images is important to conduct experiments for handwritten word alteration detection.

III. EXPERIMENTAL SETUP AND RESULTS

This section delineates the experimental setup for word alteration data set creation. Further, the data set is validated for the task of handwritten word alteration detection using convolution neural network (CNN).

A. EXPERIMENTAL SETUP FOR WORD ALTERATION DATA SET CREATION

It has been observed that alteration in any word mostly requires addition of few pen strokes with perceptually similar color pen. Few examples of such possible alteration are already presented in Fig. 1. In order to simulate the word alteration problem, synthetic handwritten word image data set is created. The characteristic of this data set is that it contains two types of word images, namely genuine and altered. A genuine word image is an image where all characters are written using a single pen. An altered word image is an image where characters from two pens are considered. This data set is created with the help of character image samples written using 10 blue and 10 black pens. The set of words, that have been chosen to create synthetically in this data set for the problem in hand, is based on the probable set of prefixes/suffixes that can be added to any word to change the meaning of actual word.

It can be observed that few prefixes can be easily added to words to change their meanings. Examples of such prefixes are *im*, *in*, *dis*, *mis*, and *un*. Similarly, words can also be changed by addition of suffixes, like *teen*, *ty*, *een*, and *y*. Few base words that can be changed with these prefixes and suffixes, are presented in Table 2.

In order to create the data set for the word alteration problem, a word list [1] with prefixes *im*, *in*, *dis*, *mis*, and *un* is considered. Additionally, a word list with suffixes *teen*, *ty*, *een*, and *y* is also considered for creating the data set. In order to limit the size of the data set, the words having either a prefix or a suffix with lengths in between 5 to 9 characters have been considered for both genuine and altered word images creation. Word images for both genuine and altered classes are created for each considered word. If a synthetic word is created using character samples of a single pen, then the word image is considered to be in genuine class. If character images in prefix/suffix and in base word are chosen from the pool of images from two different pens, then the word image is considered in altered class. Keeping the length-related condition into account, 4387 genuine word images are created for each pen. Thus, 87740 (4387 word images per pen × 20 pens) genuine word images are synthetically created. In these word images, the character images for base word and prefix/suffix are considered from the same pen. Similarly, equal number of altered word images are generated for the same set of words as in the genuine class. In order to maintain the equal number of image samples for altered class (as in genuine class), synthetic word image samples for each pen P_i in this class are generated as following: For each pen P_i, a total of 4387 base word images are created. These are the same set of base words for the word images in the genuine set for pen P_i. If the pen P_i produces blue ink, then remaining nine pens with blue ink are used to add prefixes/suffixes to these 4387 base words. Similarly, if pen P_i produces black ink, then remaining nine pens with black ink are used to add prefixes/suffixes to these 4387 base words. Out of these 4387 samples, $\lfloor \frac{4387}{9} \rfloor$ are altered using each pen P_j of same color ink (either blue or black) such that i≠j. This process results in a total of 4383 ($\lfloor \frac{4387}{9} \rfloor \times 9$) number of altered word image samples. Further, in order to keep equal number of image samples, i.e., 4387 in both classes, 4 more altered image samples are created where one of those 9 pens are used to add suffixes/prefixes. Thus, 87740 (4387 altered word images per pen $P_i \times 20$ pens) word images are synthetically created for the altered class. This newly created dataset can be obtained on request using the URL as provided in [43].

[1] available at "https://www.morewords.com".

TABLE 2: Examples of prefixes and suffixes that can be used to change the meaning of a base word.

S. no.	Prefix	Base word	Suffix	Changed word
1.	*in*	complete	-	*in*complete
2.	*im*	possible	-	*im*possible
3.	*dis*	like	-	*dis*like
4.	*mis*	place	-	*mis*place
5.	*un*	defined	-	*un*defined
7.	-	six	*teen*	six*teen*
8.	-	seven	*ty*	seven*ty*
9.	-	eight	*een*	eight*een*
10.	-	eight	*y*	eight*y*

B. VALIDATION OF THE SYNTHETIC DATASET USING CNN

There is a need to experimentally validate the effectiveness of the newly created dataset. In an earlier work [22], AlexNet [44] and VGG-16 [45] pre-trained convolutional neural networks (CNNs) were used for the same problem of handwritten word alteration detection. In [22], experiments were reported using a small data set [23], where 360 genuine and 360 altered word images were considered.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3059342, IEEE Access

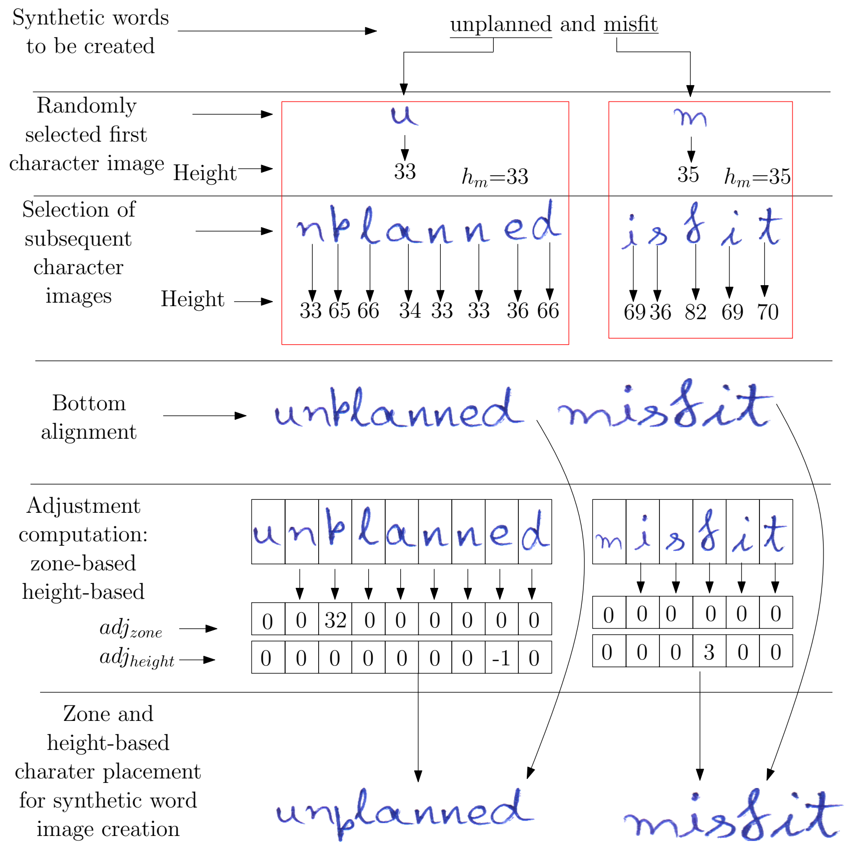


FIGURE 8: Illustrative examples of synthetic word image (‘unplanned’ and ‘misfit’) generation.

Comparatively fewer number of sample word images in [23] were not sufficient to train deep CNNs. Hence, the earlier work in [22] resorted to transfer learning, where pre-trained AlexNet [44] and VGG-16 [45] were used. The smaller data set in [23] was used to fine-tune these CNN models for the task of handwritten word alteration detection. Hence, a novel method has been proposed in this paper for synthetically generating a large collection of handwritten word images. In order to demonstrate the effectiveness of the synthetically created data set, four CNNs having similar architectures as AlexNet [44], VGG-16 [45], VGG-19 [45], and MobileNet [2] are used here. But instead of using a pre-trained model, these four CNN architectures have been trained from scratch using the newly created synthetic dataset. It is worth mentioning that pre-trained AlexNet, VGG-16, VGG-19, and MobileNet architectures are trained on ImageNet database [46] with 1000 classes; whereas our problem has 2 labeled classes (i.e., altered and genuine). Hence, similar architectures have been used in this study with 2 classification nodes in AlexNet, VGG-16, VGG-19, and MobileNet architectures.

Training of each of the CNN architectures is performed for blue and black pen synthetic data separately. The details in the training are mentioned here. It is to be noted that CNN works on predetermined input receptor size. The input word images are resized using bi-cubic interpolation to make them compatible with the input image receptor size of the CNN. It can be noted that interpolation changes the color values at the pixels in the resized image. The effect of different interpolation techniques on this specific classification task can be studied separately. But this is not considered within the scope of the current experiment. As long as the same interpolation technique is being used throughout the experiment, the result can be accepted. Further, data augmentation and dropouts are performed while training the above four CNN architectures. For data augmentation, horizontal and vertical flips of word images at 0.5 probability are performed during each iteration of training the architectures. Moreover, dropout with probability 0.5 has been used to avoid overfitting. Stochastic gradient descent with momentum solver [47] has been used during training the CNN architectures. Training of these CNNs are performed with mini-batch size of 20. These architectures are trained with 30 epochs with learning rate 0.00001 and L2 regularization of 0.004.

Next, the performances of this newly trained CNN models is compared with the performances of the pre-trained Alexnet, VGG-16, VGG-19, and MobileNet models to establish the effectiveness of the newly created synthetic dataset to train CNNs for handwritten word alteration detection. To make an effective performance comparison, the perfor-

[2]Howard *et al.* Searching for mobilenetv3. In Proceedings of the International Conference on Computer Vision 2019 (pp. 1314-1324).

mances of these trained CNNs (either from scratch or pre-trained) are evaluated using DIAL dataset [23]. It is worth mentioning that the work in [22] used pre-trained AlexNet and VGG-16 architectures for word alteration detection using transfer learning. To fit the pre-trained models to DIAL dataset, these models were fine-tuned using the handwritten word images of DIAL dataset. Moreover, as 20 different pens have been used in DIAL word alteration data set [23], 20-fold cross validation scheme was used for testing. In i-th fold of cross-validation, the words being either written or modified using pen P_i were used for testing and remaining words were used for fine-tuning the pre-trained architectures. Just to have a fare comparison between using the newly created synthetic dataset for training CNNs as against the training using ImageNet dataset in [22], the CNN architectures being trained on synthetically created new data set are assumed to be equivalent of pre-trained architectures. These trained models are fine-tuned and tested using DIAL dataset [23]. Similar to the work in [22], the 20-fold cross-validation is also performed. It is to be noted that detection of word alteration using blue pen is carried out using the models being trained using blue pen synthetic data. Similarly, the models being trained using the black pen synthetic data are used as pre-trained models for detecting word alteration using black pens.

It is to be noted that the main objective of this work is to create and validate the suitability of the synthetically created new data set. Thus, the same parameters and the testing environment have been opted as in [22]. Average classification accuracies of 10-folds per-training to blue and black pen cases are presented in Table 3. Experimental results are separately reported for blue and black pens. Here are crucial observations from Table 3:

- Comparison on words using blue pens: Use of Alex-Net looks alike architecture using blue pen synthetic data set achieves better average classification accuracy (87.03%) than pre-trained Alex-Net using ImageNet dataset (78.88%). Similarly, use of VGG-16, VGG-19, and MobileNet look alike architectures using blue pen synthetic data set achieve better average classification accuracies (81.30%, 79.55%, and 82.03%, respectively) than pre-trained VGG-16, VGG-19, and MobileNet using ImageNet dataset (76.67%,d 75.85%, and 77.22%, respectively).
- Comparison on words using black pens: Use of Alex-Net looks alike architecture using black pen synthetic data set achieves better average classification accuracy (88.70%) than pre-trained Alex-Net using ImageNet dataset (85.69%). Similarly, use of VGG-16, VGG-19, and MobileNet look alike architectures using blue pen synthetic data set achieve better average classification accuracies (83.15%, 83.82%, and 84.92%, respectively) than pre-trained VGG-16, VGG-19, and MobileNet using ImageNet dataset (79.13%, 80.86%, and 81.24%, respectively).

Experimental results confirm the suitability of synthetically created new dataset for handwritten word alteration detection task using ink color analysis.

TABLE 3: Classification accuracies being achieved on DIAL data set [23] using different CNN models.

Pen Set	Architecture Pre-Trained on	Architecture	Average Accuracy (%)
Blue Pens	ImageNet Data set [46]	AlexNet	78.88%
		VGG-16	76.67%
		VGG-19	75.85%
		MobileNet	77.22%
	Blue Pen Synthetic Data set	AlexNet	**87.03%**
		VGG-16	**81.30%**
		VGG-19	**79.55%**
		MobileNet	**82.03%**
Black Pens	ImageNet Data set [46]	AlexNet	85.69%
		VGG-16	79.13%
		VGG-19	80.86%
		MobileNet	81.24%
	Black Pen Synthetic Data set	AlexNet	**88.70%**
		VGG-16	**83.15%**
		VGG-19	**83.82%**
		MobileNet	**84.92%**

IV. CONCLUSION

Collection and annotation of handwritten text images for alteration detection are tedious and time consuming task. Therefore, a method of synthetically creating a large dataset for the said problem has been proposed in this paper. The attempt starts with collection and digitization of characters in English alphabet, which are written by a set of volunteers using different pen inks. Then, selection of appropriate character images for concatenation and subsequent zone-based and height-based adjustments of their placements are significant parts of this work. These steps ensure that the created synthetic word image samples maintain the realistic effect of the actual handwriting. The newly created handwritten word image dataset will be useful for the experiments on handwritten word alteration detection. It is the first of this kind of approach to create such a large dataset for the said problem. This dataset is also being made available to researchers for their experiments to enrich this field of study. It is the biggest contribution of this work. Several other handwriting datasets exist but those datasets are aimed to recognize the handwriting. Those data sets cannot be used for ink analysis based handwritten word alteration detection.

The suitability of the newly created dataset for the task of handwritten word alteration detection using ink color analysis is experimentally validated using CNN based classifiers. Four deep CNN architectures are trained for the said task using the newly created dataset. This training using the newly created dataset yields better performance than the case of training using a ImageNet dataset [46], which is very popular to train large CNNs for image classification tasks. The improvement in performance justifies the creation of the dataset for the said task.

As a last note, we acknowledge that the proposed method of synthetically creating a handwritten word dataset for word alteration detection task attempts to provide a realistic look to the created words. In future, an attempt can also be made to further improve the realistic look by considering the cursive style of handwriting.

REFERENCES

[1] Claude Roux, Michelle Novotny, Ian Evans, and Chris Lennard. A study to investigate the evidential value of blue and black ballpoint pen inks in australia. Forensic Science International, 101(3):167–176, 1999.

[2] Ashwini Kher, Mary Mulholland, Elinore Green, and Brian Reedy. Forensic classification of ballpoint pen inks using high performance liquid chromatography and infrared spectroscopy with principal components analysis and linear discriminant analysis. Vibrational Spectroscopy, 40(2):270–277, 2006.

[3] Xiang Feng Wang, Jing Yu, Meng Xia Xie, Ya Tong Yao, and Jie Han. Identification and dating of the fountain pen ink entries on documents by ion-pairing high-performance liquid chromatography. Forensic Science International, 180(1):43–49, 2008.

[4] Rena A Merrill and Edward G Bartick. Analysis of ballpoint pen inks by diffuse reflectance infrared spectrometry. Journal of Forensic Science, 37(2):528–541, 1992.

[5] Weili Wang, Liren Zhang, Dongmei Wei, Yuefeng Zhao, and Jingjing Wang. The principle and application of hyperspectral imaging technology in detection of handwriting. In Proceedings of International Conference on Advanced Infocomm Technology, pages 345–349, 2017.

[6] Zohaib Khan, Faisal Shafait, and Ajmal Mian. Automatic ink mismatch detection for forensic document analysis. Pattern Recognition, 48(11):3615–3626, 2015.

[7] Muhammad Jaleed Khan, Khurram Khurshid, and Faisal Shafait. A spatio-spectral hybrid convolutional architecture for hyperspectral document authentication. In Proceedings of International Conference on Document Analysis and Recognition, pages 1097–1102, 2019.

[8] Damian K Chlebda, Alicja Majda, Tomasz Łojewski, and Joanna Łojewska. Hyperspectral imaging coupled with chemometric analysis for non-invasive differentiation of black pens. Applied Physics A, 122(11):957, 2016.

[9] Zhipei Luo, Faisal Shafait, and Ajmal Mian. Localized forgery detection in hyperspectral document images. In Proceedings of International Conference on Document Analysis and Recognition, pages 496–500, 2015.

[10] Loong Chuen Lee, Muhammad Ismail Ab Samad, and Mohamad Azlan Mohd Ismail. Nondestructive classification and identification of ballpoint pen inks by raman spectroscopy for forensic document examinations. Journal of Analytical Chemistry, 71(7):723–729, 2016.

[11] Flávia de Souza Lins Borba, Ricardo Saldanha Honorato, and Anna de Juan. Use of raman spectroscopy and chemometrics to distinguish blue ballpoint pen inks. Forensic Science International, 249:73–82, 2015.

[12] Mototsugu Suzuki, Norimitsu Akiba, Kenji Kurosawa, Yoshinori Akao, and Yoshiyasu Higashikawa. Differentiation of black writing ink on paper using luminescence lifetime by time-resolved luminescence spectroscopy. Forensic Science International, 279:281–287, 2017.

[13] Haritha Dasari and Chakravarthy Bhagvati. Identification of non-black inks using HSV colour space. In Proceedings of International Conference on Document Analysis and Recognition, pages 486–490, 2007.

[14] Apurba Gorai, Rajarshi Pal, and Phalguni Gupta. Document fraud detection by ink analysis using texture features and histogram matching. In Proceedings of International Joint Conference on Neural Networks, pages 4512–4517, 2016.

[15] Rajesh Kumar, Nikhil R. Pal, J. D. Sharma, and Bhabatosh Chanda. A novel approach for detection of alteration in ball pen writings. In Proceedings of International Conference on Pattern Recognition and Machine Intelligence, pages 400–405, 2009.

[16] Rajesh Kumar, Nikhil R. Pal, Bhabatosh Chanda, and J. D. Sharma. Detection of fraudulent alterations in ball-point pen strokes using support vector machines. In Proceedings of Annual IEEE India Conference, pages 1–4, 2009.

[17] Rajesh Kumar, Nikhil R. Pal, Bhabatosh Chanda, and J. D. Sharma. Forensic detection of fraudulent alteration in ball-point pen strokes. IEEE Transactions on Information Forensics and Security, 7(2):809–820, 2012.

[18] Prabhat Dansena, Soumen Bag, and Rajarshi Pal. Differentiating pen inks in handwritten bank cheques using multi-layer perceptron. In Proceedings of International Conference on Pattern Recognition and Machine Intelligence, pages 655–663, 2017.

[19] Priyanka Roy and Soumen Bag. Identification of fraudulent alteration by similar pen ink in handwritten bank cheque. In Proceedings of International Conference on Computer Vision and Image Processing, pages 183–195, 2019.

[20] Prabhat Dansena, Rajarshi Pal, and Soumen Bag. Quantitative assessment of capabilities of colour models for pen ink discrimination in handwritten documents. IET Image Processing, 14(8):1594–1604, 2020.

[21] Priyanka Roy and Soumen Bag. Forensic performance on handwriting to identify forgery owing to word alteration. In Proceedings of IEEE International Conference on Identity, Security, and Behavior Analysis, pages 1–9, 2019.

[22] Prabhat Dansena, Rahul Pramanik, Soumen Bag, and Rajarshi Pal. Ink analysis using CNN-based transfer learning to detect alteration in handwritten words. In Proceedings of International Conference on Computer Vision and Image Processing, pages 223–232, 2019.

[23] DIAL word alteration dataset, 2019. https://sites.google.com/site/diafcse/resources/dial-word-alteration-dataset.

[24] Jiri Martınek, Ladislav Lenc, and Pavel Král. Building an efficient ocr system for historical documents with little training data. Neural Computing and Applications, pages 17209–17227, 2020.

[25] David Etter, Stephen Rawls, Cameron Carpenter, and Gregory Sell. A synthetic recipe for ocr. In International Conference on Document Analysis and Recognition, pages 864–869, 2019.

[26] Moussa Djioua and Réjean Plamondon. Studying the variability of handwriting patterns using the kinematic theory. Human Movement Science, 28(5):588–601, 2009.

[27] Gabriella Sanniti di Baja. Well-shaped, stable, and reversible skeletons from the (3, 4)-distance transform. Journal of Visual Communication and Image Representation, 5(1):107–115, 1994.

[28] Zhouchen Lin and Liang Wan. Style-preserving english handwriting synthesis. Pattern Recognition, 40(7):2097–2109, 2007.

[29] Raid M Saabni and Jihad A El-Sana. Comprehensive synthetic arabic database for on/off-line script recognition research. International Journal on Document Analysis and Recognition, 16(3):285–294, 2013.

[30] Tamas Varga and Horst Bunke. Off-line handwritten textline recognition using a mixture of natural and synthetic training data. In Proceedings of International Conference on Pattern Recognition, volume 2, pages 545–549, 2004.

[31] Shivansh Gaur, Siddhant Sonkar, and Partha Pratim Roy. Generation of synthetic training data for handwritten indic script recognition. In International Conference on Document Analysis and Recognition, pages 491–495, 2015.

[32] Gang Liu, Lianwen Jin, Kai Ding, and Hanyu Yan. A new approach for synthesis and recognition of large scale handwritten chinese words. In International Conference on Frontiers in Handwriting Recognition, pages 571–575, 2010.

[33] Nam Tuan Ly, Cuong Tuan Nguyen, and Masaki Nakagawa. Training an end-to-end model for offline handwritten japanese text recognition by generated synthetic patterns. In Proceedings of International Conference on Frontiers in Handwriting Recognition, pages 74–79, 2018.

[34] Yousef Elarian, Irfan Ahmad, Sameh Awaida, Wasfi G Al-Khatib, and Abdelmalek Zidouri. An arabic handwriting synthesis system. Pattern Recognition, 48(3):849–861, 2015.

[35] Amirhossein Ahmadian, Kazim Fouladi, and Babak Nadjar Araabi. Towards information theoretic measurement of fidelity and diversity in handwriting synthesis. In International Conference on Pattern Recognition and Image Analysis, pages 39–45, 2019.

[36] E. Alonso, B. Moysset, and R. Messina. Adversarial generation of handwritten text images conditioned on sequences. In Proceedings of International Conference on Document Analysis and Recognition, pages 481–486, 2019.

[37] Siyuan Lu, Shui-Hua Wang, and Yu-Dong Zhang. Detection of abnormal brain in MRI via improved alexnet and elm optimized by chaotic bat algorithm. Neural Computing and Applications, pages 1–13, 2020.

[38] Yuezhong Zhang, Shi Wang, Honghua Zhao, Zhenhua Guo, and Dianmin Sun. CT image classification based on convolutional neural network. Neural Computing and Applications, 2020.

[39] Edna Chebet Too, Li Yujian, Sam Njuki, and Liu Yingchun. A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, pages 272–279, 2019.

[40] Arun Madakannu and Arivazhagan Selvaraj. Digi-net: a deep convolutional neural network for multi-format digit recognition. Neural Computing and Applications, pages 11373–11383, 2019.

[41] Najwa Altwaijry and Isra Al-Turaiki. Arabic handwriting recognition system using convolutional neural network. Neural Computing and Applications, pages 1–13, 2020.

[42] Rahul Pramanik, Prabhat Dansena, and Soumen Bag. A study on the effect of CNN-based transfer learning on handwritten Indic and mixed numeral recognition. In Workshop on Document Analysis and Recognition, pages 41–51, 2018.

[43] Synthetic Word Alteration Dataset, 2020. https://sites.google.com/site/diafcse/resources/synthetic-word-alteration-dataset.
[44] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, pages 1097–1105, 2012.
[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
[46] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.
[47] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In Neural Networks: Tricks of the Trade, pages 9–48. 2012.

PRABHAT DANSENA is currently a Research Scholar at Indian Institute of Technology (ISM) Dhanbad, India. He received the B.E. degree in computer science and engineering from CSVTU, Bhilai, India, in 2012, and the M.Tech. degree from the School of Computer and Information Science, University of Hyderabad, Hyderabad, India, in 2015. His areas of interest include Image Processing, Ink Analysis, and Pattern Recognition.

SOUMEN BAG received his B.E. and M.Tech. degree in Computer Science and Engineering from NIT Durgapur in 2003 and 2008, respectively. He received his Ph.D. degree from IIT Kharagpur in 2013. Currently, he is working as an Assistant Professor in the department of Computer Science and Engineering at IIT (ISM) Dhanbad. Before this employment, he worked at BCET Durgapur, India and IIIT Bhubaneswar, India, respectively. He is the recipient of Institute Gold medal for holding the position of First-Class-First in his Master's degree. He is the recipient of different Fellowships/Scholarships from National and International Societies like IEEE Signal Processing Society, CSIR, Govt. of India, Microsoft India, MHRD, Govt. of India, etc. He is enlisted in the Marquis Who's Who in the World, USA, 32nd Ed., 2015. He is the recipient of Early Career Research Award (ECR) from SERB, Govt. of India, 2017. He is a member of different International societies like IEEE, IUPRAI, etc. He is the author of several reputed International Journals and Conferences. He acts as an Organizing and Programme Committee Members in different National and International Conferences. At present he is acting as an Associate Editor for the journal IET Image Processing. His research interests are in the areas of Document Image Analysis, OCR for Indian Scripts, Medical Image Analysis, Handwritten Forgery, and Digital Image Forgery.

RAJARSHI PAL is working as an Assistant Professor at Institute for Development and Research in Banking Technology (IDRBT), Hyderabad, India since 2011. His primary research interests include image processing, biometrics, and cyber security. He has obtained his Ph. D. form Indian Institute of Technology, Kharagpur, India in 2011. Prior to joining in IDRBT, he has served as a Research Associate at Center for Soft Computing Research, Indian Statistical Institute, Kolkata, India.