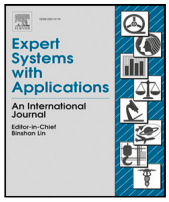
Expert Systems With Applications 231 (2023) 120740

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



CNN feature and classifier fusion on novel transformed image dataset for dysgraphia diagnosis in children

Jayakanth Kunhoth *, Somaya Al Maadeed, Moutaz Saleh, Younes Akbari

Department of Computer Science and Engineering, Qatar University, Doha, Qatar

ARTICLE INFO

Keywords:
Learning disabilities
Dysgraphia diagnosis
Handwriting
Machine learning
CNN ensembles
CNN feature fusion

ABSTRACT

Dysgraphia is a neurological disorder that hinders the acquisition process of normal writing skills in children, resulting in poor writing abilities. Poor or underdeveloped writing skills in children can negatively impact their self-confidence and academic growth. This work proposes various machine learning methods, including transfer learning via fine-tuning, transfer learning via feature extraction, ensembles of deep convolutional neural network (CNN) models, and fusion of CNN features, to develop a preliminary dysgraphia diagnosis system based on handwritten images. In this work, an existing online dysgraphia dataset is converted into images, encompassing various writing tasks. Transfer learning is applied using a pre-trained DenseNet201 network to develop four distinct CNN models separately trained on word, pseudoword, difficult word, and sentence images. Soft voting and hard voting strategies are employed to ensemble these CNN models. The pre-trained DenseNet201 network is used for CNN feature extraction from each task-specific handwritten image data. The extracted CNN features are then fused in different combinations. Three machine learning algorithms support vector machine (SVM), AdaBoost, and Random forest are employed to assess the performance of the CNN features and fused CNN features. Among the task-specific models, the SVM trained on word data achieved the highest accuracy of 91.7%. In the case of ensemble learning, soft voting ensembles of task-specific CNNs achieved an accuracy of 90.4%. The feature fusion approach substantially improved the classification accuracy, with the SVM trained on fused features from the task specific-data achieving an accuracy of 97.3%. This accuracy surpasses the performance of state-of-the-art methods by 16%.

1. Introduction

Dysgraphia is a learning disability that primarily affects a person's ability to express themselves in writing. It can impact not only handwriting, but also spelling, grammar, and organization of words and letters (Deuel, 1995). Studies have shown that between 10% and 30% of children worldwide struggle with handwriting difficulties. Accurately diagnosing a learning disorder, such as dysgraphia, poses significant challenges due to the consideration of multiple cues. The symptoms of dysgraphia are diverse and vary according to the child's age and developmental stage. Moreover, these indicators need to persist for a minimum of six months, alongside parallel intervention actions (American Psychiatric Association. & American Psychiatric Association. DSM-5 Task Force, 2013). Dysgraphia is a complex condition that can manifest independently or coexist with other disorders, such as autism spectrum disorder (ASD), developmental coordination disorder (DCD), or attention deficit hyperactivity disorder (ADHD), further complicating the assessment process (Lopez, Hemimou, Golse, & Vaivre-Douret, 2018). Consequently, early diagnosis and intervention play a crucial role in addressing dysgraphia as they contribute significantly to reducing the effort and time required for treatment.

The diagnosis of dysgraphia in children involves a collaborative effort among specialists from diverse fields, including education, psychology, and medicine. These professionals, such as teachers, occupational therapists, speech therapists, and ophthalmologists, work together to assess the student's handwriting ability and identify potential factors that may impact writing performance. Prior to conducting a comprehensive dysgraphia assessment, it is crucial to exclude other conditions that could contribute to handwriting impairments, such as hearing loss, visual impairments, or inadequate training. During the evaluation process, various factors need to be considered to effectively diagnose dysgraphia. These factors encompass aspects such as writing speed, legibility, spelling consistency, and pencil grip. While there is currently no universally standardized medical assessment method for dysgraphia

* Corresponding author.
E-mail addresses: jayakanth.k.chandran@gmail.com (J. Kunhoth), s_alali@qu.edu.qa (S. Al Maadeed), Moutaz.saleh@qu.edu.qa (M. Saleh), akbari_younes@yahoo.com (Y. Akbari).

https://doi.org/10.1016/j.eswa.2023.120740
Received 10 April 2023; Received in revised form 18 May 2023; Accepted 5 June 2023
Available online 8 June 2023

diagnosis, certain widely used assessments can provide valuable insights. Examples include the Concise Evaluation Scale for Children's Handwriting (BHK) in French (Hamstra-Bletz & de Bie J, 1987) and the Detailed Assessment of Speed of Handwriting (DASH) in Latin (Barnett, Henderson, Scheib, & Schulz, 2009).

It is important to acknowledge that manual assessment conducted by experienced specialists heavily relies on the evaluation of the handwritten product. However, it is crucial to recognize that these assessments can be susceptible to both positive and negative influences stemming from human bias and the specialists' level of expertise. Furthermore, the manual assessment process is time-consuming and demands a significant allocation of human resources. Thus, considering these factors becomes essential when determining the most appropriate assessment approach for dysgraphia diagnosis in research and clinical settings.

To overcome the aforementioned limitations, researchers have developed automated systems for dysgraphia diagnosis. These systems primarily focus on statistically analyzing handwriting characteristics obtained through digitizing tablets. Handwriting analysis has been extensively explored in the literature for diagnosing various neurological and elderly disease conditions (Ammour et al., 2020; Chai et al., 2023; Drotár et al., 2014; Kawa, Bednorz, Stepien, Derejczyk, & Bugdol, 2017; Ribeiro, Afonso, & Papa, 2019). Recent advancements in the mobile industry have facilitated the creation of tablets equipped with built-in capabilities for extracting a wide range of handwriting features and raw data. These features include the position of the pen tip, On-Surface/In-Air pen position, pen tip pressure, azimuth angle of the pen to the tablet surface, tilt of the pen, and timestamp (Faundez-Zanuy et al., 2020). Numerous studies have leveraged digitizing tablets and the wealth of information they provide to identify dysgraphia, employing machine learning algorithms for assistance. In addition to the digitizing tablet-based approaches, researchers have also proposed image/offline handwritten data analysis-based methods (Devi & Kavya, 2022) for dysgraphia screening. These methods offer an alternative avenue for diagnosis, complementing the traditional focus on online handwritten features.

The existing literature on dysgraphia diagnosis in children has predominantly focused on online data analysis-based approaches, leaving offline data analysis approaches relatively unexplored. This research work aims to address this gap by introducing a novel approach centered around offline data analysis for dysgraphia diagnosis. The primary contribution of this study lies in the development of a transformed image/offline handwritten dataset specifically designed for the dysgraphia diagnosis problem. This dataset serves as a valuable resource for conducting comprehensive investigations and analyses. Furthermore, novel methods are proposed, focusing on ensemble learning and feature fusion techniques, with the aim of enhancing the diagnosis performance. Specifically, the fusion of features extracted from different writing tasks, such as word, pseudoword, and sentence, is considered. Additionally, an ensemble of classifiers trained on task-specific data are considered to improve the overall diagnostic accuracy. To the best of our knowledge, our approach is novel, and no work in the literature has considered the concept of feature fusion and ensemble learning using task-specific data for dysgraphia diagnosis. The main objectives of this study can be summarized as follows:

- Develop and publish a novel image dataset for automated dysgraphia diagnosis problem by extending publicly available online handwritten data and evaluate the same using deep learning and machine learning methods. To the best of our knowledge, the developed novel image dataset is the first-ever publicly available image dataset for dysgraphia diagnosis problems.
- Apply transfer learning methodology, specifically transfer learning via fine-tuning and transfer learning via feature extraction, for dysgraphia diagnosis using handwritten image data.
- Apply an ensemble learning approach by creating an ensemble of handwriting task-specific deep CNN classifiers. This ensemble will consist of two or more deep convolutional neural network models trained with task-specific data to improve dysgraphia classification.
- Apply a feature fusion approach where handwriting task-specific features are combined. This involves extracting features from two or more handwritten tasks and developing classifiers using traditional machine learning algorithms to effectively classify normal and dysgraphia handwritten images.
- Analyze the effectiveness of three supervised machine learning algorithms, namely SVM, AdaBoost, and Random forest, for distinguishing the image features extracted from the handwritten images.

The remainder this article is structured as follows: In Section 2, we give an overview of previous research on the topic. Section 3 explains the dataset we used for this study and how we created it. In Section 4, we provide detailed information about the materials and methods we used in our work, including the algorithm we developed. The results and findings are presented in Section 5, followed by a discussion of the results, limitations, and future directions in Section 6. Lastly, Section 7 concludes the paper.

2. Related works

Despite the inherent challenges in diagnosing dysgraphia, numerous automated dysgraphia diagnosis systems leveraging machine learning techniques have been proposed in the literature. The majority of these studies have focused on analyzing online handwritten data captured using digitizing tablets to differentiate between normally developing handwriting and dysgraphia. In 2017, Mekyska et al. (2017) proposed methods for classifying normally developing and dysgraphic handwriting. Their approach involved using a Wacom Intuos tablet to collect data from 54 school students. Various characteristics such as kinematics, dynamics, and non-linear dynamics attributes were explored to distinguish normal and dysgraphic writing. Random forest and linear discriminant algorithms were employed to develop classifiers trained on the extracted attributes of online handwritten data. The developed classifier achieved a sensitivity of 96% in handwriting classification.

Richard and Serrurier (2020) analyzed the performance of different machine learning algorithms for classifying online handwritten features to detect the presence of dysgraphia. This involved utilizing features such as pen tip pressure, letter and word characteristics including shape and spacing. The Random forest algorithm, logistic regression algorithm, and naïve Bayes algorithm were employed as classifiers. Asselborn et al. (2018) proposed an automated dysgraphia diagnosis tool using a consumer-level tablet. The study involved 298 primary school students, including 56 with dysgraphia. Participants wrote on a Wacom Intous tablet for 5 min using the Ductus software. 54 handwriting features were extracted, including static, kinematic, and dynamic characteristics. A Random forest (RF) classifier was trained on these features, achieving excellent accuracy for dysgraphia diagnosis. Drotár and Dobeš (2020b) proposed a machine learning-based system for dysgraphia detection. They collected a new dataset comprising handwriting samples from 120 school students, including those with dysgraphia. Trained professionals gathered the data using the WACOM Intuos Pro Large tablet, capturing pen movement, pressure, azimuth, and altitude during writing. A total of 22 types of spatiotemporal and kinematic features were extracted from the collected data. Multiple machine learning algorithms were employed for classification, with the AdaBoost algorithm achieving the highest accuracy of 80%. Among the extracted features, pressure and pen lifts showed high discriminatory potential.

Dimauro, Bevilacqua, Colizzi, and Di Pierro (2020) introduced a software system designed to partially automate the evaluation of the

Concise Evaluation Scale for Children's Handwriting (BHK) test. The BHK test involves evaluating thirteen handwriting characteristics and assigning scores based on their quality. The proposed software system automatically generates scores for nine of the thirteen characteristics by modifying multiple document analysis algorithms.

For the online handwriting analysis-based dysgraphia diagnosis methods (Asselborn, Chapatte, & Dillenbourg, 2020; Asselborn et al., 2018; Drotár & Dobeš, 2020b; Dui et al., 2020; Gargot et al., 2020; Kunhoth, Al Maadeed, Saleh & Akbari, 2022b; Kunhoth, Al Maadeed, & Akbari, 2023; Mekyska et al., 2019, 2017; Zvoncak, Mekyska, Safarova, Smekal, & Brezany, 2019), spatial characteristics of writing, including stroke dimensions and spacing; temporal characteristics of writing, including time taken for writing and idle time in between writing; dynamic characteristics of writing, including pressure, tilt, and azimuth; and kinematic characteristics of writing, including velocity, acceleration, and jerk, have equal or lesser significance in distinguishing normally developing handwriting from dysgraphia.

On the other hand, offline image-based methods focus on the extraction of different types of image features from the handwritten product. Devi and Kavya (2022) proposed an end-to-end CNN neural network architecture for classifying the images into normal and dysgraphia classes. This research employed a combination of handwriting and geometric features, obtained using the Kekre-Discrete Cosine mathematical model, to identify dysgraphia. The acquired features were effectively utilized in the feature learning stage of deep transfer learning for dysgraphia detection. The Kekre-Discrete Cosine Transform with Deep Transfer Learning (K-DCT-DTL) approach outperformed existing methods. Notably, the proposed K-DCT-DTL approach achieved the highest accuracy of 99.75%, indicating the effectiveness and efficiency of the proposed method. Sharmila et al. (2023) presented a research study that introduced a transfer learning-based approach for discriminating between normal and abnormal handwriting. Specifically, their work focused on analyzing images of handwritten letters. The authors employed various pre-trained deep neural network architectures to leverage the advantages of transfer learning in their investigation. By leveraging the knowledge and learned representations from these pre-trained models, they aimed to enhance the accuracy and performance of their handwriting classification system. Comparative analysis of few related works is provided in Table 1.

Among the existing methods proposed in the literature for the diagnosis of dysgraphia, only a limited number are based on the analysis of handwritten images or offline handwriting. The majority of studies have focused on the analysis of online handwritten data (Kunhoth, Al-Maadeed, Kunhoth & Akbari, 2022a). The online handwriting approach involves the use of a digitized tablet and a stylus pen, with participants writing directly on the tablet surface or on a blank paper placed on the tablet. Due to its ability to capture different characteristics of writing compared to offline image data, the online handwriting analysis-based approach has gained popularity for diagnosing dysgraphia. However, the lower friction surface of tablet computers can alter graphomotor execution, which contradicts the intended purpose (Guilbert, Alamargot, & Morin, 2019). Additionally, the pressure sensitivity of these tablets may vary depending on the model (Prunty, Pratt, Raman, Simmons, & Steele-Bobat, 2020). Therefore, the analysis of offline images or the final output of handwriting is crucial for dysgraphia diagnosis. The online data acquired using a digital tablet can be transformed into images, as it provides coordinated information for any writing activity. Moreover, none of the existing literature has explored the application of feature fusion or ensemble learning approaches to distinguish between normally developing handwriting and dysgraphia handwriting.

3. Dataset

The proposed work focuses solely on analyzing images for diagnosing dysgraphia in children. To the best of our knowledge, only a few image databases have been proposed in the literature for diagnosing dysgraphia in children (Devi & Kavya, 2022; Ghouse et al., 2022; Sharmila et al., 2023). However, none of these databases are publicly available or accessible to researchers interested in studying the same problem. On the other hand, the literature presents numerous online handwriting datasets for diagnosing dysgraphia. Among these datasets, only one is publicly available. Typically, online handwriting data considers various attributes of writing, including dynamics and kinematics specific to each individual. In contrast, offline or image datasets capture various static and spatial characteristics, such as the shape of the written output and stroke size. These datasets offer valuable resources for exploring the diagnosis of dysgraphia in children.

To develop an image dataset, we started with the only publicly available online dataset for the dysgraphia diagnosis problem (Drotár & Dobeš, 2020a, 2020b). The dataset consists of online handwritten data acquired for six different writing activities in Slovak orthography. It includes writing the letter "l", the syllable "le", the simple word "leto", the pseudo word "lamoken", the difficult word "hračkárstvo", and the sentence "V lete bude teplo a sucho". A total of 120 students completed the handwriting task, with 63 exhibiting normally developing handwriting and the remaining 57 having dysgraphia.

The handwriting samples in the public dataset were acquired using a Wacom Intuos digitizing tablet. This tablet can capture the x and y positions of the pen tip on the tablet's surface, along with additional modalities such as time, the pressure exerted by the pen, altitude, azimuth angle of the pen to the writing surface, and a flag value indicating whether the pen is on or away from the tablet's surface. The publicly available online dataset is not provided in a task-specific format. Instead, it consists of a single data file for each individual, containing the x, y, and other writing modalities for all tasks in a continuous manner.

To separate the task-specific data for each individual, we considered the x and y positions, as well as the flag value. We then plotted the available online handwritten data as a single image for each individual. By manually analyzing the plotted single image and online handwritten data, we estimated the x and y positions for each writing activity and extracted the respective data. From the separated data, we generated images for each task, storing them in RGB format with a resolution of 400 pixels × 400 pixels.

Out of the six writing tasks, we excluded the images generated from letter writing and syllable writing data due to the variability in writing speeds. Thus, we considered images from four tasks. The resulting image dataset consists of 120 images for each writing task, totaling 480 images for the four tasks. This new transformed handwritten image dataset is publicly available, and the link is provided in the data availability section.

Considering the limited number of samples for each writing task, there is a risk of overfitting the trained machine learning models, particularly deep neural networks. To address this issue, we artificially augmented the dataset by applying three different transformations to each original image: zooming (20%), pixel shifting (0.1 fractions of width and height), and shearing (5 degrees). These transformations are sufficient for capturing the variations in human handwriting.

The structure of the dataset, including the number of images for each task, is illustrated in Fig. 1 (Drotár & Dobeš, 2020a, 2020b). Additionally, sample images from the extended image dataset are provided in Fig. 2.

4. Materials and methods

This section focuses on explaining the proposed CNN-based transfer learning methodologies, CNN-based ensembles, and feature fusion methods for automated dysgraphia diagnosis.

Table 1
Comparative analysis of state of the art dysgraphia diagnosis approaches, (LDA: linear discriminant analysis, SVM: support vector machine, ANN: artificial neural network, RF: Random forest, DT: decision tree, CNN: convolutional neural network).

Ref.	Data type	Subjects	Features	Classifiers	Performance
Mekyska et al. (2017)	Online	54	Kinematic and nonlinear dynamic	LDA , RF	Recall : 96%
Asselborn et al. (2018)	Online	298	Static, Kinematic, Pressure, Tilt	RF	Recall : 96.5%
Isa, Syazwani Rahimi, Ramlan, and Sulaiman (2019)	Offline	–	OCR,MSER	ANN	Accuracy : 71%
Mekyska et al. (2019)	Online	76	Spatial, temporal, kinematic, dynamic, other – pen elevations and relative number of interruptions	XG-Boost	Specificity : 90%
Zvoncak et al. (2019)	Online	65	Kinematic, temporal, spatial, and dynamic	SVM and RF	Recall : 88%
Dui et al. (2020)	Online	104	Gesture smoothness, pressure(mean value), drawing kinematics	Logistic regression	Area under curve : 0.82
Gargot et al. (2020)	Online	280	Static, kinematic, pressure, and tilt	linear regression	–
Drotár and Dobeš (2020b)	Online	120	Dynamic, Spatiotemporal and kinematic features	AdaBoost	Accuracy : 79.5%
Rosen-blum and Dror (2016)	Online	90	Spatiotemporal , dynamic, kinematic and other features	SVM	Accuracy : 90%
Sihwi, Fikri, and Aziz (2019)	Online	32	Spatial, temporal , dynamic and other features	SVM	Accuracy : 82.51%
Dankovi-cova, Hurtuk, and Fecilak (2019)	Online	72	Spatial,temporal , dynamic, kinematic and other features	SVM	Sensitivity : 75.5%
Devi, Kavya, Therese, and Gayathri (2021)	Online	40	Not explicitly mentioned	DT	Not mentioned
Kedar et al. (2021)	Online	60	Spatiotemporal, dynamic and kinematic features	RF	Recall: 92.85%
Skunda, Nerusil, and Polec (2022)	Online	120	CNN features	CNN	Accuracy: 79.7%
Devi and Kavya (2022)	Offline	–	CNN features, Geometric features	CNN	Accuracy: 99.75%
Sharmila et al. (2023)	Offline	–	CNN features	ResNet (transfer learning)	Accuracy: 98.22%
Ghouse, Paran-jothi, and Vaithiyanathan (2022)	Offline	–	CNN features	CNN	Accuracy: 98.16%

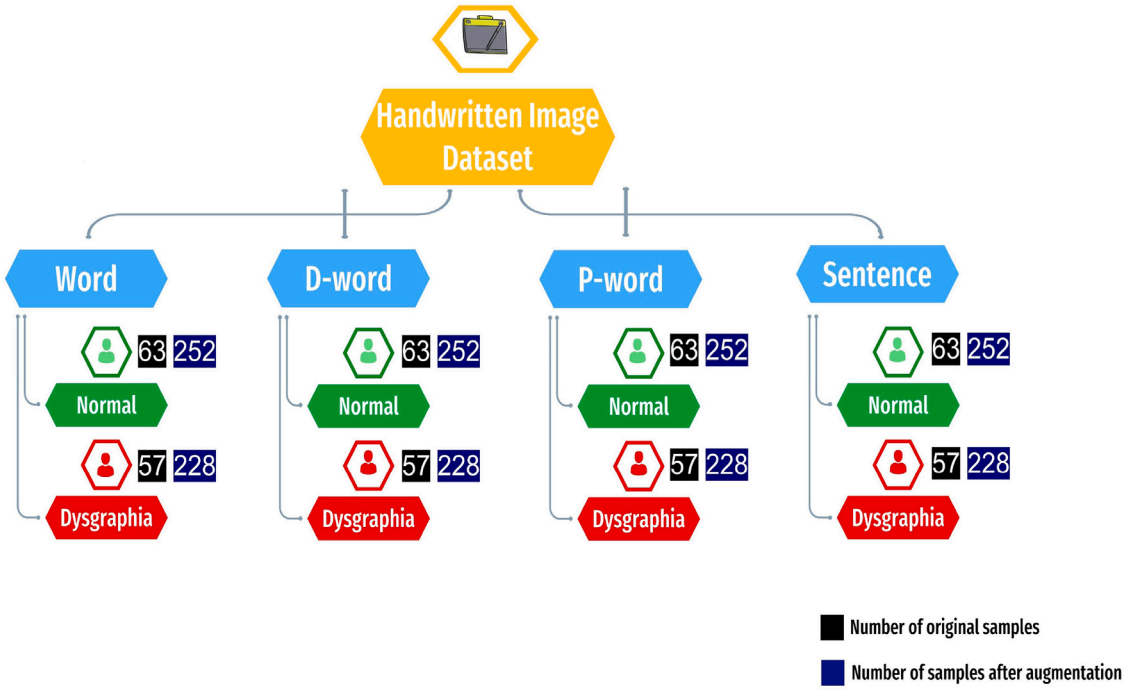


Fig. 1. Structure of the novel transformed handwritten image dataset. The number of images/samples in each class is specified in the diagram.

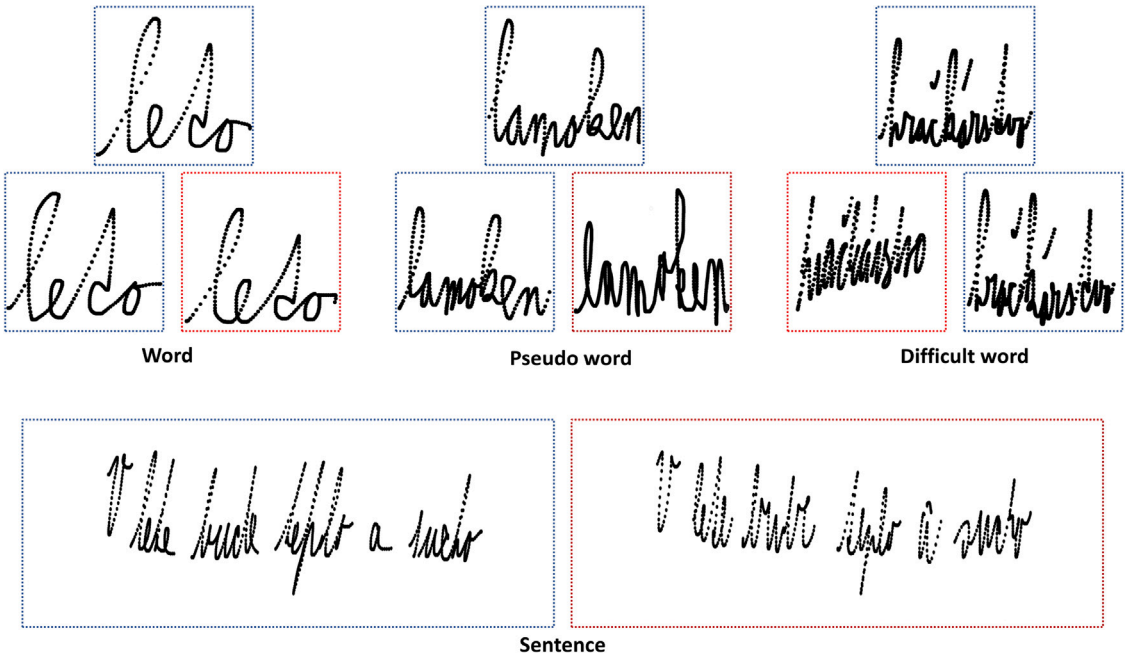


Fig. 2. Sample images from the dataset. Handwritten images obtained from different writing task for a normally developing student and dysgraphia student are shown. Handwritten samples in blue border indicate normally developing and red border indicate dysgraphia.

4.1. Materials

4.1.1. Convolutional neural networks

The convolutional neural network (CNN) is a class of neural networks suitable for various problems where the input data consists of images or time series information presented as multi-dimensional arrays. The main computational components or layers of a CNN include convolutional layers, pooling layers, fully connected dense layers, and batch normalization layers. Additionally, a CNN consists of an input layer and an output layer. The convolutional layers serve as the foundation of any deep CNN architecture. They comprise a set of learnable kernels, feature detectors, or filters with a small receptive field. These

layers are responsible for generating feature maps from the input data through basic convolution operations.

Let I be an input image. For all available local patches i in I, the convolutional operation is performed using the learnable kernels when the image I is forwarded through a convolutional layer. The learnable kernel/filter glides over each patch i in I to produce the feature map. The convolutional operation is usually applied to the raw image as well as the subsequent feature maps. Stacking multiple CNN layers enables prediction models to generate and learn hierarchical features from raw input data.

Let M_i^{l-1} be a feature map produced by a previous convolutional layer in a CNN model, and M_i^l be the feature map produced by the current convolutional layer. M_i^l is defined as Kunhoth, Karkar, Al-Maadeed, and Al-Attiyah (2019),

$$M_i^l = f\Big(\sum_{i \in N_K} M_i^{l-1} * w_k^l + b_k^l\Big) \tag{1}$$

Where N_K is the number of kernels, b_k^l is the neural network bias value, w_k^l is the pre-assigned weight matrix of the current layer, and f is the activation function.

The convolutional operation is a linear transformation. To introduce non-linearity, activation functions are incorporated in deep CNN models. After the convolutional operation, the dot product of the input feature map and the neurons' components in the current convolutional layers is passed through an activation function f to introduce non-linearity. These activation functions are often referred to as transfer functions, as they transform the output of the convolutional operation into a specific interval such as [0, 1] or [−1, 1]. Sigmoid, tanh, and rectified linear activation functions (ReLU) are commonly used activation functions in the literature. Among them, ReLU is very popular and has displayed better performance in the literature. ReLU (Agarap, 2018) is defined as

$$f(x) = \begin{cases} 0, & if\, x < 0 \\ x, & if\, x \geq 0 \end{cases} \tag{2}$$

The batch normalization layer is responsible for speeding up the learning process of a neural network. It achieves this by normalizing the output from the previous layers. In deep CNN architectures, pooling layers are utilized to implement dimensionality reduction. This means that pooling layers are responsible for reducing the spatial size of the feature map. Dimensionality reduction aids the network in learning important features from the raw data and attaining translation invariance.

The multidimensional feature maps generated by the convolutional layer are transformed into a one-dimensional array before being fed into the fully connected layer. The fully connected layer, also known as the dense layer, is a basic artificial feed-forward neural network. The output layer is placed at the end of the fully connected layer. In a classification problem, the output layer estimates the probabilities for each input to belong to each class, serving as the network's prediction function.

Neural network learning involves an optimization problem. Within a neural network, the responsibility of the optimizer algorithm is to adjust the weights and learning rate in order to minimize the prediction loss. Optimization algorithms work towards minimizing the objective function, which is the average loss over all training samples (Goodfellow, Bengio, & Courville, 2016). Let $O(\boldsymbol{\theta})$ be the objective function and it is defined as Goodfellow et al. (2016),

$$O(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim\hat{p}_{\text{data}}} L(f(\mathbf{x};\boldsymbol{\theta}), y) \tag{3}$$

where L is the function for each sample, $\hat{p}_{\text{data}}$ is the empirical distribution, f is the prediction function, x is the input and y is the ground-truth label of input data x.

For a binary classification problem, the binary cross entropy/ log loss is used as the loss function. log loss is defined as Vovk (2015),

$$L = -(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)) \tag{4}$$

where y_i is the ground truth label of the current sample, p_i is the probability of class label 1.

For an input sample i, the p_i is computed using the softmax function for binary classification as follows (Fakhrou, Kunhoth, & Al Maadeed, 2021),

$$p_i = P(y = y_1 | i : \boldsymbol{\theta}) = \frac{e^{\boldsymbol{\theta}_{y_1}^T i}}{\sum_{c=0}^{1} e^{\boldsymbol{\theta}_{y_c}^T i}} \tag{5}$$

Where $\boldsymbol{\theta}$ is the parameters of the network, c number of classes (for binary classification two classes, ie, 0 and 1)

The objective function is minimized by altering the network parameters $\boldsymbol{\theta}$. The network parameter updation is accomplished by updating them in the opposite direction of their gradient. There are multiple variants of gradient optimization approaches available for neural networks (Ruder, 2016). The difference between them mainly lies in the amount of data processed at a time for parameter updation. Batch gradient descent processes the full data available for training to update the parameters. On the other hand, stochastic gradient descent (SGD) and mini-batch gradient descent process a single train data and a mini-batch of train data at a time, respectively, to update the parameters. In this work, we utilized the Adam optimization algorithm. Adam is fast and converges quicker compared to other optimization algorithms.

4.1.2. Transfer learning

Transfer learning is a commonly used approach in machine learning, where knowledge acquired from solving a generic problem is reused to tackle other related problems. In transfer learning, an existing pre-trained machine learning model is employed as the initial starting point to learn from a new dataset. Technically, this involves initializing the weights of the new machine learning models using the weights from the pre-trained models. In order to mitigate overfitting and enhance the generalization capability of prediction models, machine learning algorithms often necessitate a sufficient amount of data to learn the underlying patterns. Unlike traditional machine learning algorithms, deep learning algorithms require a large volume of data to develop prediction models with satisfactory generalization ability. The transfer learning approach empowers users to construct effective deep learning-based prediction models, even in situations where the dataset is not sufficiently large. To initiate transfer learning, a pre-trained deep learning model trained with a substantial amount of data is essential.

Transfer learning can be employed in two different ways. The first approach is 'transfer learning via feature extraction,' where a pre-trained neural network is utilized to extract meaningful features from the new dataset. Subsequently, these extracted features are used to develop a machine learning model by feeding them into supervised learning algorithms such as SVM or k-nearest neighbors (KNN). In the second approach, known as 'transfer learning via fine-tuning,' a pre-trained neural network model is directly trained on the new dataset. In most cases, the hidden layers of the pre-trained model are kept frozen before initiating training on the new data. To adapt to the new dataset, some layers and parameters of the pre-trained network are modified.

4.1.3. Ensemble learning

Ensemble learning is a widely used approach in machine learning to enhance performance by leveraging the decision-making abilities of multiple trained models. In ensemble learning, for a given problem, multiple machine learning models, either with identical underlying algorithms or different ones, are trained separately using either subsets or the entire dataset. Ensemble learning can effectively reduce variance. Once the individual training of each base model in the ensemble is completed, they are combined to make predictions on test samples. Different decision-making strategies can be applied to generate the final prediction from multiple base models. Two popular decision-making strategies are hard voting and soft voting. Hard voting, also known

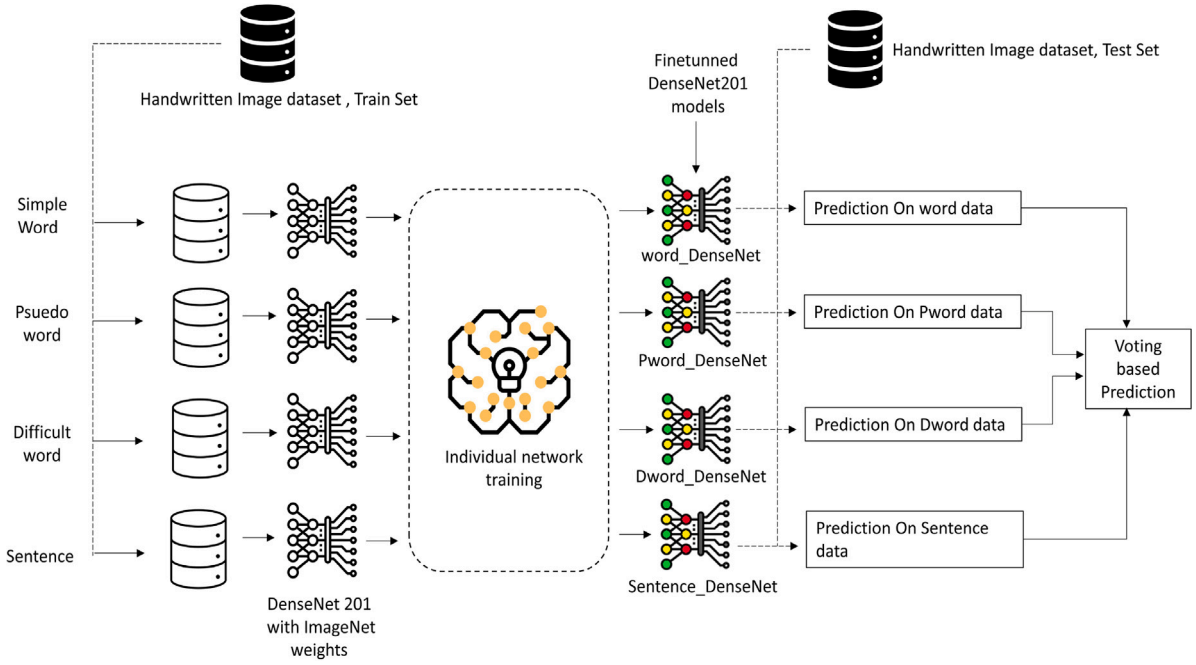


Fig. 3. Ensemble of deep CNN models trained on task specific data: An overview.

as majority voting, involves each classifier making its own prediction. For input data 'I', the final prediction is determined by selecting the most frequently occurring label among all the predictions made by the classifiers. Soft voting, or average voting, calculates the average prediction confidence or probability for each class across all base classifiers. The class with the highest average prediction confidence is considered the final predicted label for the input data.

4.1.4. Feature fusion

Feature fusion is a technique utilized to enhance the performance of a machine learning model by combining multiple sources of information or features. It involves integrating features extracted from various modalities or sources, such as visual and audio information, to create a new and more comprehensive feature representation that captures information from all sources. In the context of image classification, feature fusion can be performed at different levels, including the pixel level or feature level. At the pixel level, individual pixels from different modalities are combined to generate a new image. At the feature level, features extracted from different modalities are merged using techniques such as concatenation or weighted averaging. The objective of feature fusion is to leverage the complementary information provided by different sources, aiming to improve the performance of a model, particularly when no single modality offers sufficient information on its own.

4.2. Methods

In this work, we utilized the popular neural network architecture called DenseNet201 (Huang, Liu, Van Der Maaten, & Weinberger, 2017), which was pretrained on the ImageNet dataset (Deng et al., 2009), for implementing transfer learning. The ImageNet dataset is a vast collection of images specifically designed for image classification and object recognition tasks. It comprises images of 1000 different objects, including stationary objects, fruits, vegetables, animals, electronic devices, and more. We employed the pretrained DenseNet201 model to fine-tune it on task-specific handwritten image data. These fine-tuned DenseNet models were then utilized as base classifiers for ensemble learning. Additionally, the pretrained DenseNet201 model was used as a feature extractor to extract CNN features from the task-specific handwritten image data. These extracted features were subsequently employed in the CNN feature fusion approach.

4.2.1. Ensemble of fine-tuned CNNs

The overview of the proposed ensemble approach for fine-tuned CNNs in this work is provided in Fig. 3. The handwritten dataset, generated from the online handwritten data, consists of images of words, pseudowords, difficult words, and sentences written by each individual. From this image dataset, multiple subsets of data are generated based on the tasks. Specifically, the word images from each individual are grouped as the word dataset. Similarly, the pseudoword dataset, difficult word dataset, and sentence dataset are created.

In our ensemble learning approach, a DenseNet201 architecture pretrained on the ImageNet dataset is employed as a base classifier for fine-tuning on each subset of data. The pre-trained DenseNet201 is fine-tuned on each subset of the dataset using the transfer learning approach, resulting in the generation of four DenseNet201 models. During the prediction phase, the input data is separated into images corresponding to each task, which are then fed into their respective prediction model. The predictions from each fine-tuned DenseNet201 model are combined using either a soft voting or hard voting strategy to obtain the final prediction.

The proposed ensemble approach for making predictions using multiple base models trained or fine-tuned on subsets of the dataset can be mathematically explained as follows:

Let T be the DenseNet201 network pre-trained on the ImageNet dataset. And D is the actual image dataset, $D = \{(d_{1w}, d_{1p}, d_{1d}, d_{1s}), (d_{2w}, d_{2p}, d_{2d}, d_{2s}), \ldots, (d_{nw}, d_{np}, d_{nd}, d_{ns})\}$, where d_{1w} indicate the image

of word from the first sample, similarly d_{1p}, d_{1d}, d_{1s} indicates images of pseudoword, difficult word and sentence from first sample respectively.

$T_{Ensemble} = \{T_{word}, T_{dword}, T_{pword}, T_{sentence}\}$, is the set of four base classifiers fine-tuned in word, difficult word, pseudoword, and sentence images respectively. $C = [c_0, c_1]$ is the list of classes in this problem. There are only two classes, c_0 indicates the negative class and c_1 indicates the positive class.

Let $d = \{d_w, d_p, d_d, d_s\}$ be the input data available for prediction. Each sub-data inside the input data is forwarded to its respective prediction network to generate the independent prediction.

Prediction on the word data, $P_{word} = T_{word}(d_w)$

Prediction on the pseudoword data, $P_{pword} = T_{pword}(d_p)$

Prediction on the difficult word data, $P_{dword} = T_{dword}(d_{dw})$

Prediction on the sentence data, $P_{sentence} = T_{sentence}(d_s)$

Combined prediction $P = (P_{word}, P_{pword}, P_{dword}, P_{sentence})$

Case 1: Majority voting/hard voting

If the final decision-making strategy for the ensemble classifier is majority voting, then in $P = (P_{word}, P_{pword}, P_{dword}, P_{sentence})$, each P_i will be the predicted class label. Then final prediction,

$$P_{final}^{Majority} = mode(P) = mode(P_{word}, P_{pword}, P_{dword}, P_{sentence}) \tag{6}$$

Case 2: Average voting/soft voting

When the final decision-making strategy for the ensemble classifier is average voting then each P_i where $i = \{word, pword, dword, sentence\}$ in P will be the pair of predicted probabilities for each class from each base prediction model.

Ie, $P_i = [p(y = c_0|d_i : \theta), p(y = c_1|d_i : \theta)]$

where $p(y = c_0|d_i : \theta)$ is the probability that the given sample d_i falls in class 0/negative class. Similarly $p(y = c_1|d_i : \theta)$ probability for class 1/positive class. And θ is the parameter of the prediction model.

Here,

$$p(y = c_0|d_i : \theta) = \frac{e^{\theta_{c_0}^T d_i}}{e^{\theta_{c_0}^T d_i} + e^{\theta_{c_1}^T d_i}} \tag{7}$$

$$p(y = c_1|d_i : \theta) = \frac{e^{\theta_{c_1}^T d_i}}{e^{\theta_{c_0}^T d_i} + e^{\theta_{c_1}^T d_i}} \tag{8}$$

Then final prediction,

$$P_{final}^{Average} = argmax(p_{averages}) \tag{9}$$

where, $p_{averages} = [p_{average}^{c_0}, p_{average}^{c_1}]$. $p_{average}^{c_0}$ and $p_{average}^{c_1}$ are the average of probabilities obtained for class 0 and class 1 respectively for a given test input in each base classifier. The formal definition of the ensemble learning algorithms is provided as Algorithm 1

For Algorithm 1, the time complexity is analyzed separately for each phase. Algorithm 1 consists of three phases: training the task-specific classifiers, prediction and hard voting based aggregation, and prediction and soft voting based prediction. The first phase consists of four major operations including, loading the DenseNet201 model, replacing the fully connected layer, initializing the weights of the new layer, and finetuning the CNN_i. Among those first three have constant time complexity. The time complexity of finetuning depends on the number of epochs, and the number of training subsets of data.

Assume that fine tunning process takes $O(f)$ time per epoch, where f represents the time complexity of the fine-tuning process for a single epoch. Then the time complexity of fine-tuning for 'n' epochs is $O(n \cdot f)$.

The overall time complexity for training four task-specific classifiers is $O(4 \cdot n \cdot f)$.

The second phase "Prediction and Hard Voting Based Aggregation" starts by initializing empty lists and variables. It then iterates over each test sample set in the given test data. For every test sample set, an empty list called "Labels" is created to store the predicted labels from each model. The algorithm further iterates over each model in the list of trained models. Within each iteration, a prediction is made on the current sample using the corresponding model. The time complexity of this prediction step is dependent on the sample size and the complexity of the model's forward pass. The computed predicted label for the sample is then added to the "Labels" list.

After collecting predictions from all models for a specific sample set, the algorithm proceeds to calculate the mode of the "Labels" list. This step identifies the most frequently occurring predicted label among the models, which serves as the aggregated prediction for that sample set. The time complexity of computing the mode is determined by the size of the "Labels" list.

Algorithm 1: Ensemble learning using hard voting and soft voting approach.

Require: Deep CNN architecture (Densenet 201) pre-trained on ImageNet dataset, Four task-specific datasets: D_1, D_2, D_3, D_4, Train set: T_1, T_2, T_3, T_4, Test set: $Test$, where $Test = \{\{x_{11}, ...x_{1j}\}, \{x_{21}, ...x_{2j}\}, \{x_{31}, ...x_{3j}\}, \{x_{41}, ...x_{4j}\}\}$

Ensure: Predicted labels, Y

Training task specific classifier

Load Densenet 201 as $CNN_{\text{pretrained}}$
Initialize empty list $Trained_{Models}$
for each dataset D_i in D_1, D_2, D_3, D_4 **do**
    Load $CNN_{\text{pretrained}}$ as CNN_i
    Replace the last fully connected layer of CNN_i with a new layer for dataset D_i
    Initialize weights of the new layer
    Fine-tune CNN_i using training data T_i for n number of epochs
    Append CNN_i to $Trained_{Models}$
end for

Prediction and hard voting based aggregation

Initialize empty list $Aggregated\_Predictions$
for each test sample set x in $Test$ **do**
    Initialize empty list $Labels$ of size equal to the number of classes
    for each model CNN_i in $Trained_{Models}$ **do**
        run prediction on sample x_{ij} using CNN_i
        Compute the predicted label for sample x_{ij} using CNN_i, denoted as y_{ij}
        Append y_{ij} to $Labels$
    end for
    Compute the mode of $Labels$ to obtain the aggregated predicted label for sample x
    Append the aggregated predicted label to $Aggregated\_Predictions$
end for

Prediction and soft voting based aggregation

Initialize empty list $Aggregated\_Predictions$
for each test sample set x in $Test$ **do**
    Initialize empty list $Probabilities$ of size equal to the number of classes
    for each model CNN_i in $Trained_{Models}$ **do**
        run prediction on sample x_{ij} using CNN_i
        Compute the predicted probability vector $\mathbf{p}_{ij}$ for sample x_{ij} using CNN_i
        Append $\mathbf{p}_{ij}$ to $Probabilities$
    end for
    Compute the aggregated predicted probability vector $\mathbf{p}$ for test sample set x using soft voting on $Probabilities$
    Predict the label for sample x based on the highest probability in $\mathbf{p}$
    Append the predicted label to $Aggregated\_Predictions$
end for
return $Aggregated\_Predictions$ as Y

The aggregated predicted label is then added to the "Aggregated_Predictions" list. This process is repeated for each test

sample set in the dataset. Finally, the algorithm returns the "Aggregated_Predictions" list as the output denoted by "Y".

The time complexity of the algorithm can be estimated by considering the number of test sample sets, the number of models, and the time complexity of performing a single prediction and computing the mode. It can be represented as $O(t \cdot c \cdot (p+1)+m)$. Where p represents the time complexity of a single prediction and m represents the time complexity of computing the mode, t represents number of test sample set, c represents number of classifiers or models.

The prediction and soft voting based aggregation step is almost similar to prediction and hard voting aggregation step considering the time complexity. After individual predictions, instead of computing the mode, averaging of probabilities (soft voting) is implemented. This is the only difference. If soft voting takes $O(v)$ time, where v represents the time complexity of the soft voting process, then the overall time complexity of the second phase of Algorithm 1 can be represented as $O(t \cdot c \cdot (p+1)+v)$.

In this work, in addition to simply ensembling the four classifiers, we explored different ensembling combination scenarios, including ensembles of all possible pairs of base classifiers and ensembles of all possible triads of base classifiers.

4.2.2. CNN feature fusion

In this work, feature fusion is implemented to combine the information extracted from the outputs of different writing tasks. The overview of the feature fusion approach implemented in this work is provided in Fig. 4.

The handwritten dataset generated from the online handwritten data consists of images of words, pseudowords, difficult words, and sentences written by each individual. From this image dataset, multiple subsets of the dataset are created based on the tasks. For example, the word images from each individual are grouped as the word dataset. Similarly, the pseudoword dataset, difficult word dataset, and sentence dataset are also created.

A DenseNet201 network pretrained on the ImageNet dataset is used as the feature extractor. To transform the end-to-end prediction network DenseNet201 into a feature extractor, the top classification layer is removed, and a global max pooling layer is attached to the top of the network. The global max pooling layer is responsible for converting the multidimensional features from the DenseNet201 network into one-dimensional features for each input data.

Each subset, including simple word, pseudoword, difficult word, and sentence, from the dataset is passed through the pre-trained DenseNet201 architecture to generate CNN features for each task. Subsequently, for each sample in the main dataset, their respective CNN features from each subset of data are horizontally concatenated to form the final feature vector. Similarly, the final feature vector is generated for all the samples in the main dataset.

In addition to fusing the features from the four subsets of data, this work explores different fusion combinations, such as the fusion of all possible pairs of CNN features and the fusion of all possible triads of CNN features from the subsets of the dataset.

Machine learning algorithms are necessary for training and evaluating the performance of fused features. In this work, multiple machine learning algorithms are employed to train and assess the performance of various feature fusion combinations. The data analysis task addressed in this work involves binary classification.

Algorithm 2: CNN feature fusion based ML algorithm.

Require: Deep CNN architecture (Densenet 201) pre-trained on ImageNet dataset, Four task-specific datasets: D_1, D_2, D_3, D_4, Machine learning algorithm M

Ensure: Predicted labels, Y

Feature extraction

Load Densenet 201 as $CNN_{\text{pretrained}}$
Initialize empty list $Feature_{vector}$
for each dataset D_i in D_1, D_2, D_3, D_4 **do**
    Load $CNN_{\text{pretrained}}$ as CNN_i
    Replace the last fully connected layer of CNN_i with a global max pooling layer
    Run one forward pass on CNN_i using training data T_i
    Append output of the forward pass $features_i$ to $Feature_{vector}$
end for
Concatenate respective features of each $features_i$ in $Feature_{vector}$ to form $Fused_{vector}$
Split the $Fused_{vector}$ and create $Fused_{trainvector}$ and $Fused_{testvector}$

Training the classifier using fused features

Select a machine learning algorithm, M.
Train the model M using the training feature set $Feature_{trainvector}$:
    Initialize the algorithm's parameters and hyperparameters.
    Iterate through the training features:
        Update the model's parameters using the optimization algorithm:
            $M \leftarrow \text{optimize}(M, Feature_{trainvector})$
    Repeat until convergence or a maximum number of iterations

Prediction on test data

Initialize an empty list Y to store the predicted labels.
for each testing sample x_i in $Fused_{testvector}$ **do**
    Feed the sample x_i into the trained model M:
        $y_i \leftarrow \text{predict}(M, x_i)$
    Append y_i to the list Y.
end for
return Y as the predicted labels for the testing samples.

A significant number of supervised algorithms are available in the literature for binary classification tasks, including simple algorithms like KNN and decision trees, as well as more complex algorithms such as SVM and deep neural networks. SVM and deep neural networks, in particular, are widely used and extensively studied in the literature for analyzing various types of data, including audio and imagery (Akbal, Barua, Dogan, Tuncer, & Acharya, 2022; Karadal, Kaya, Tuncer, Dogan, & Acharya, 2021; Yildiz et al., 2023).

Given that the horizontal concatenation-based feature fusion significantly increases the dimensionality of the feature, we considered complex algorithms capable of handling nonlinear data points. In this work, we employed SVM (Pisner & Schnyer, 2020), Random forest (RF) (Biau, 2012), and AdaBoost (AB) (Schapire, 2013) algorithms to train and evaluate the performance of the CNN features and fused CNN features.

The CNN feature fusion-based machine-learning method for classifying handwritten images is formally defined as Algorithm 2. Algorithm 2 constitutes three phases: feature extraction phase, classifier training phase, and prediction on test data phase. Let us assume the time complexity of a forward pass on a single dataset is $O(f)$. Since there are four types of handwritten data, the total time complexity of the feature extraction phase is $O(4 \cdot f)$.

The training process involves iterating through the training feature set and updating the model's parameters using an optimization algorithm. The number of iterations and the complexity of the optimization algorithm can impact the overall time complexity. The complexity of the training classifier will change according to the underlying algorithm. Suppose $O(g)$ is the time complexity of training the classifier for

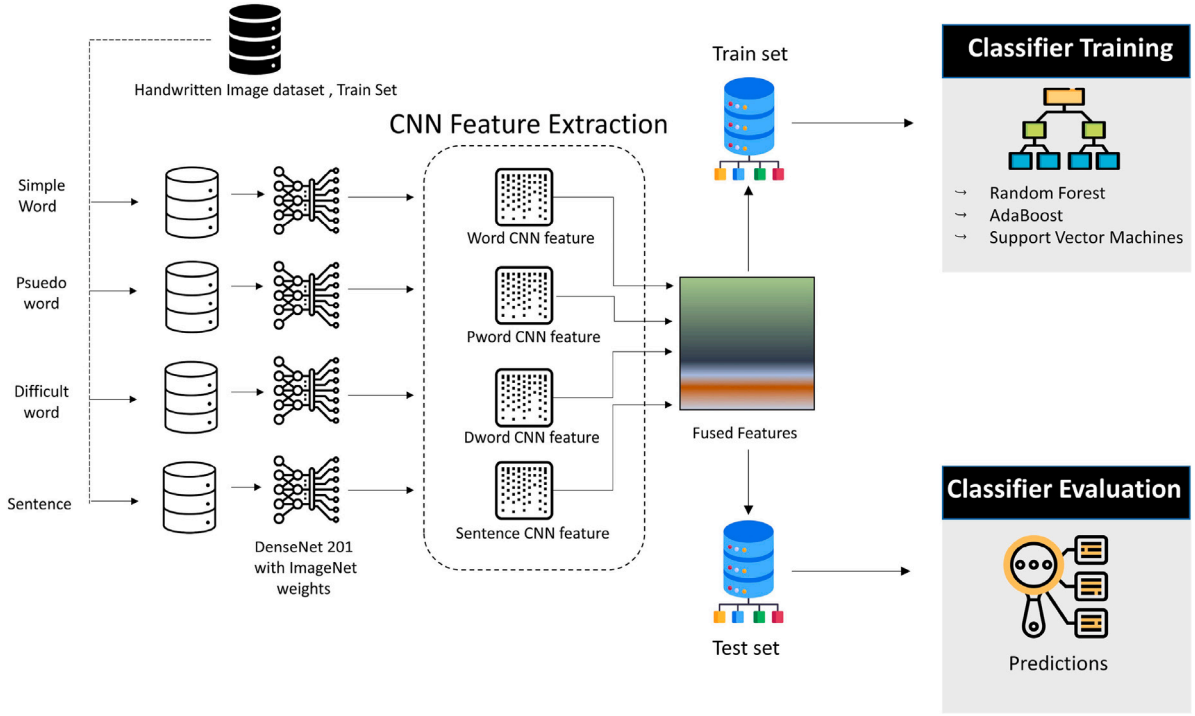


Fig. 4. CNN feature fusion from task specific handwritten images: An overview.

one iteration then the overall complexity of training the classifier for N iteration is $O(N \cdot g)$. Similarly the prediction time complexity of prediction changes with the algorithm. In general, if the time complexity of one prediction is $O(h)$, then the time complexity for predicting M samples is $O(M \cdot h)$.

5. Evaluation and results

Evaluation or assessment of proposed methods is crucial to examine their efficacy in addressing the problem at hand. In this work, multiple experiments were conducted to effectively examine and analyze the performance of the proposed methods. A total of 45 traditional ML classifiers (including SVM, Random forest, and AdaBoost trained on 15 different feature sets) and 20 deep learning classifiers (comprising four fine-tuned deep CNNs and 15 possible ensemble combinations of fine-tuned deep CNNs) were trained and evaluated using stratified ten-fold cross-validation.

All experiments were implemented in the Python language. The training and evaluation were performed on a machine equipped with an Intel(R) Core(TM) i7-7820HK CPU operating at 2.90 GHz (2901 MHz) with four cores and an Nvidia GTX 1060 GPU. To implement traditional machine learning algorithms, the popular machine learning framework SciKit was utilized, while TensorFlow was used for implementing deep learning algorithms.

Multiple evaluation metrics are considered to analyze the performance of the proposed methods. The evaluation metrics used in this work are as follows,

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

Where TP, true positives indicate the number of actual positive samples which are classified as positives; TN, true negatives indicate the number of actual negative samples which are classified as negatives; FP, false positives indicate the number of actual negative samples which are misclassified as positives; FN, false negatives indicate the number of actual positive samples which are misclassified as negatives.

$$F1score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \tag{13}$$

Although accuracy is a popular evaluation metric used for classification problems, precision, recall, and F1-score are better suited for handling imbalanced datasets. In addition to these metrics, this work considered the receiver operating characteristic (ROC) plot and the area under the ROC curve (AUC_ROC) to delve deeper into the performance of the classifiers. AUC measures provide a broader view of the classifier's performance compared to other metrics. Furthermore, the confusion matrix of the classification is provided for each classifier.

The experiment section is divided into six subsections. In the first subsection, we present the evaluation results of the finetuned DenseNet201 in each task subset of the dataset. This allows us to assess the performance of the model on individual subsets and understand its effectiveness in handling specific tasks. Moving on to the second subsection, our focus shifts to evaluating ensemble models generated from the finetuned DenseNet201 networks in each task subset of the dataset. Ensemble models have the potential to improve classification performance by combining the predictions of multiple models, and we examine their effectiveness in this context. The third subsection delves into the classification performance of CNN features extracted from each task subset and trained on multiple traditional machine learning classifiers. By utilizing the extracted features and applying traditional classifiers, we explore alternative approaches to classification and evaluate their performance. In the fourth subsection, we report the classification performance of fused CNN features trained on traditional machine learning classifiers. Fusing the features obtained from different task subsets may provide a comprehensive representation of the data, and we assess the impact of this fusion on the classification results. The fifth subsection offers a comparative analysis of the proposed methods with state-of-the-art methods. This allows us to understand how our approach measures up against existing techniques and provides insights into its strengths and weaknesses. Lastly, the sixth subsection is dedicated to reporting the results regarding the image resizing approach and the performance of the classifiers. Here, we examine the impact of

Table 2
Accuracy, precision, recall, f1-score and ROC_AUC scores of DenseNet201 network finetuned on task specific subset of the data. Confusion matrix is in the order: TP, TN, FP, FN.

Classifier	Accuracy	Precision	Recall	F1-Score	AUC	Confusion matrix
Dense_dword	81.45 ± 7.42	86.54 ± 10.9	75.37 ± 17.33	0.79 ± 0.1	0.92 ± 0.04	172/219/33/56
Dense_word	84.79 ± 4.93	89.13 ± 9.08	79.03 ± 9.89	0.83 ± 0.05	0.93 ± 0.05	180/227/25/48
Dense_pword	81.04 ± 4.7	92.01 ± 8.96	68.06 ± 14.35	0.77 ± 0.07	0.93 ± 0.03	155/234/18/77
Dense_sentence	79.16 ± 6.58	79.64 ± 10. 23	80.15 ± 19.66	0.77 ± 0.12	0.92 ± 0.03	183/197/55/45

image resizing on the classification task and provide an assessment of the classifiers' performance under two image resizing approach.

5.1. Transfer learning via fine-tuning

A DenseNet201 network pretrained on the ImageNet dataset is utilized and fine-tuned on subsets of the data. Each subset contains handwriting images acquired from a specific writing task. Four DenseNet201 models are constructed and evaluated: *Dense_word*, which is finetuned on word data from the subset; *Dense_dword*, finetuned on difficult word data; *Dense_pword*, finetuned on pseudo word data; and *Dense_sentence*, finetuned on sentence data.

For training, the models except *Dense_word* are trained for 18 epochs, while *Dense_word* is trained for 12 epochs since it reached convergence earlier. Early stopping is implemented to prevent overfitting. The Adam optimizer with a constant learning rate of 0.003 is used for training all four models. The input size of each model is adjusted to 400 × 400 pixels to match our image size.

To effectively analyze the classifiers' performance and mitigate the impact of random selection bias, we employ stratified ten-fold cross-validation for both training and evaluation. This technique ensures that each fold of the cross-validation maintains a similar distribution of class labels as the original dataset.

The classification performance of the four finetuned DenseNet201 models is presented in Table 2. The reported accuracy, precision, recall, F1-score, and AUC values in Table 2 represent the averages obtained from each fold of cross-validation. Additionally, the standard deviation (SD) of the values obtained in each fold is provided, along with the average performance metric value.

In Table 2, the confusion matrix is presented in the following order: true positives, true negatives, false positives, and false negatives. The confusion matrix provides valuable information about the model's classification performance for each class, enabling a comprehensive evaluation of its effectiveness.

Among the four classifiers that were fine-tuned on task-specific data, the DenseNet201 model fine-tuned on word data, referred to as $Dense_{word}$, achieved the highest classification accuracy of 84.79% (standard deviation: 4.93). The classifiers $Dense_{dword}$ and $Dense_{pword}$ achieved accuracies of 81.45% (standard deviation: 7.42) and 81.% (standard deviation: 4.7) respectively. Although the accuracies of $Dense_{dword}$ and $Dense_{pword}$ are comparable, $Dense_{pword}$ is preferred over $Dense_{dword}$ in terms of accuracy due to its lower standard deviation. The lowest classification accuracy among the four fine-tuned classifiers is 79.16% (standard deviation: 6.58), which is obtained by the $Dense_{sentence}$ classifier.

When evaluating the performance of classifiers for a medical diagnosis problem, relying solely on accuracy can be insufficient, especially when dealing with imbalanced datasets. In such cases, other metrics, such as recall, become more important. Recall values take into account the number of false negatives in the predictions, providing an indication of how well the classifiers identify positive cases.

In the context of diagnosing dysgraphia, recall is a crucial metric to consider over accuracy and precision. A higher recall value suggests a lower number of false negatives, which is desirable for accurate diagnosis. Evaluating the classifiers based on precision metric, both *Dense_sentence* and *Dense_word* outperformed the other classifiers. This observation is supported by the reported false negative values in the confusion matrix.

To provide a visual representation of the classifiers' performance, the ROC curve plot of the four fine-tuned classifiers is presented in Fig. 5. This curve provides insights into the trade-off between the true positive rate and the false positive rate, aiding in the assessment of classifier performance across different decision thresholds.

5.2. Ensemble of fine-tuned DenseNet201 models

This section presents the results obtained for the ensembles of fine-tuned DenseNet201 models. The DenseNet201 models, which were fine-tuned on task-specific image data, are combined in an ensemble. The objective of this experiment is to investigate the performance enhancement in classification when two or more classifiers fine-tuned on task-specific data are ensembled during prediction.

The base classifiers, including $Dense_{word}$ (DenseNet201 fine-tuned on word data from the subset), $Dense_{dword}$ (DenseNet201 fine-tuned on difficult word data from the subset), $Dense_{pword}$ (DenseNet201 fine-tuned on pseudo word data from the subset), and $Dense_{sentence}$ (DenseNet201 fine-tuned on sentence data from the subset), are developed by following the same methodology explained in sub Section 4.1.1.

During the prediction phase, multiple combinations of ensembles are considered. Two different prediction strategies for the ensemble model are explored in this experiment: hard voting and soft voting. For hard voting, ensemble classifiers are developed for all possible combinations of three or more classifiers. On the other hand, for soft voting, all possible combinations of two or more classifiers are considered to develop ensemble classifiers.

The classification performance of the eleven soft voting-based ensemble classifiers and five hard voting-based ensemble classifiers is presented in Table 3. The accuracy, precision, recall, F1-score, and AUC values reported in Table 3 represent the averages obtained from each fold of cross-validation.

The classification performance showed a significant improvement when multiple fine-tuned DenseNet201 models were ensembled. The soft voting ensemble of *Dense_word*, *Dense_dword*, *Dense_pword*, and *Dense_sentence* base models achieved the highest classification performance, with an accuracy of 90.41% (standard deviation: 2.66). The soft voting ensembles of *Dense_word*, *Dense_pword*, and *Dense_sentence* delivered the second-best classification performance, achieving a classification accuracy of 90.2% (standard deviation: 4.06).

Furthermore, the soft voting ensemble of *Dense_sentence* and *Dense_word* base models, as well as the soft voting ensembles of *Dense_word*, *Dense_dword*, and *Dense_sentence* base models, demonstrated similar classification accuracies, both achieving 89.58% (standard deviation: 4.46 and 3.84, respectively). This indicates that combining *Dense_dword* with the ensemble of *Dense_sentence* and *Dense_word* did not lead to a significant improvement in classification performance. However, it did result in a decrease in the standard deviation of accuracy.

These findings highlight that while ensembling multiple networks can enhance classification performance in certain cases, it is not always guaranteed. Moreover, the decision to employ ensemble models should consider the potential increase in computational cost. Careful consideration is necessary to strike a balance between improved performance and increased computational complexity, ensuring practicality and efficiency in decision-making processes.

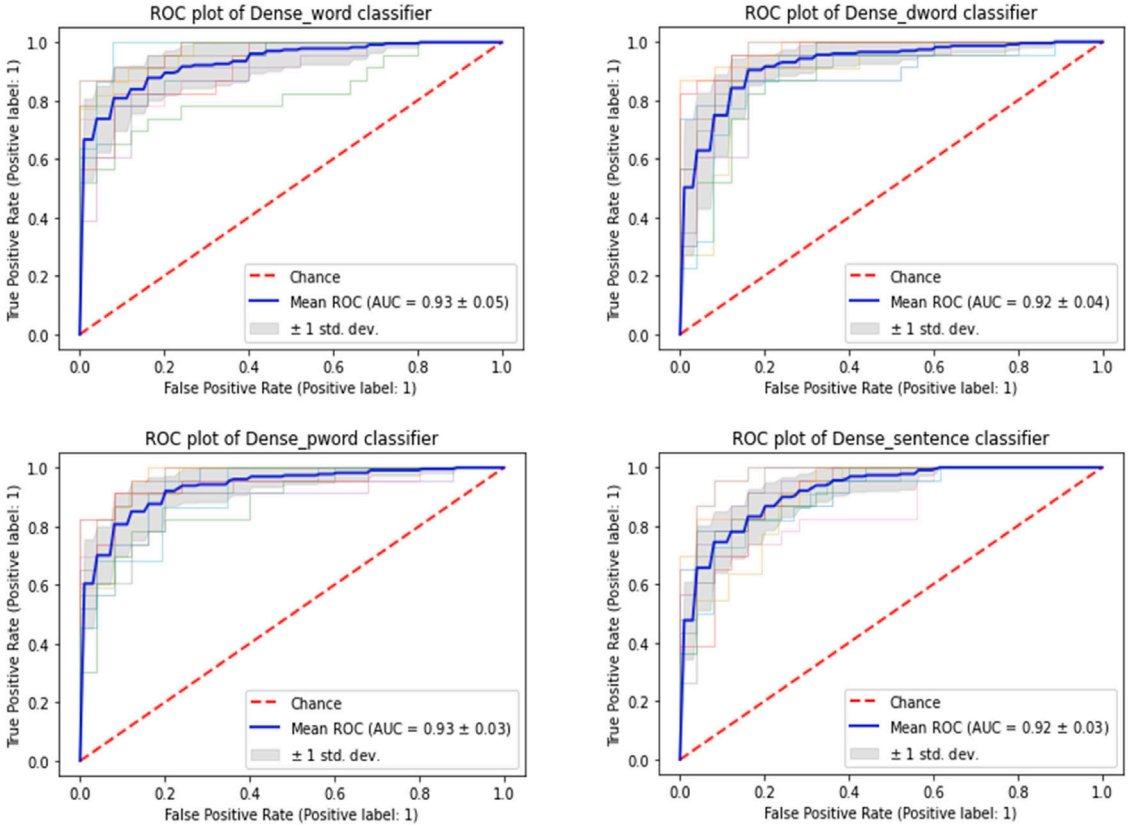


Fig. 5. ROC plot of fine-tuned classifiers; *Dense_word*:top left, *Dense_dword*:top right, *Dense_pword*:bottom left, *Dense_sentence*:bottom right.

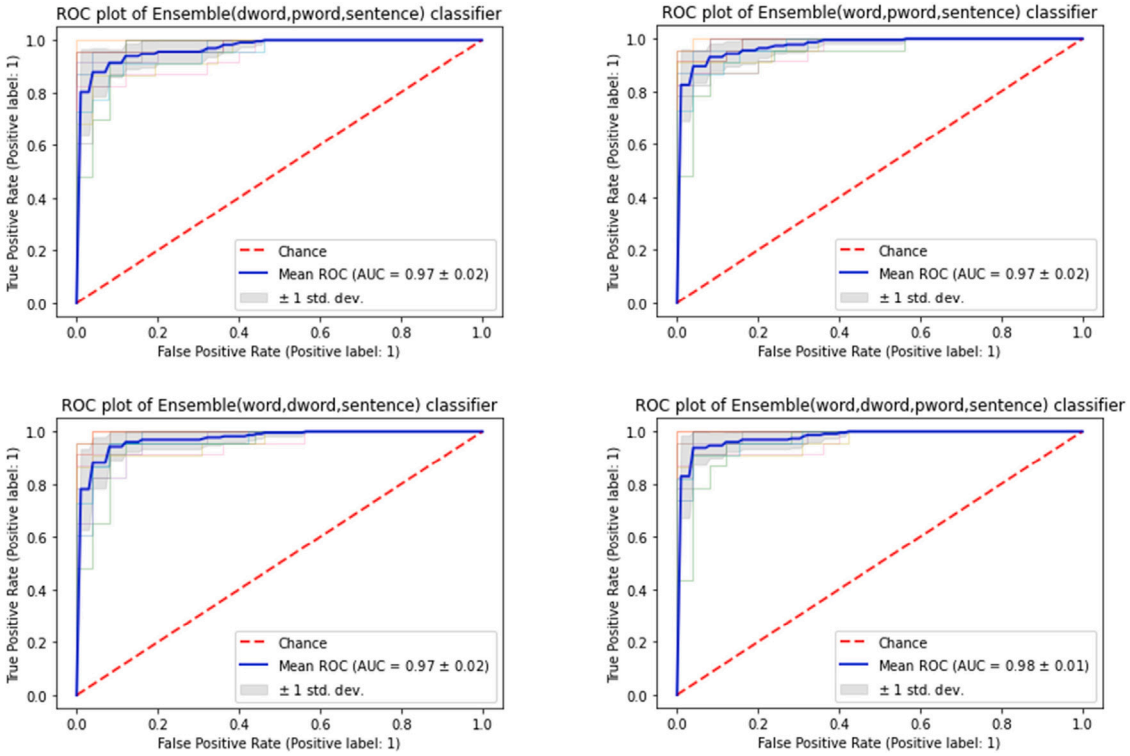


Fig. 6. ROC plot of top performing ensemble classifiers, all four are soft voting based ensembles.

In comparison to the soft voting approach, the effectiveness of hard voting is relatively limited in this problem. This observation is supported by the classification performance presented in Table 3. The ensemble of $Dense_{word}$, $Dense_{pword}$, and $Dense_{sentence}$ base models achieved the highest classification performance using hard voting. However, upon comparing it with the soft voting-based ensembles, it

Table 3
Accuracy, precision, recall, f1-score and ROC_AUC scores of Ensembles of DenseNet201 networks fine-tuned on task specific subset of the data. Confusion matrix is in the order: TP, TN, FP, FN.

Voting strategy	Classifier combination	Accuracy	Precision	Recall	F1-Score	AUC	Confusion matrix
Soft voting	*Dense_word*, *Dense_dword*	87.91 ± 5.0	91.18 ± 8.77	84.22 ± 11.07	0.86 ± 0.06	0.96 ± 0.03	192/230/22/36
	Dense_dword , *Dense_pword*	86.66 ± 4.28	94.46 ± 8.66	78.04 ± 11.69	0.84 ± 0.06	0.96 ± 0.02	178/238/14/50
	Dense_word, *Dense_pword*	88.12 ± 3.23	94.89 ± 8.16	80.75 ± 8.84	0.86 ± 0.03	0.96 ± 0.04	184/239/13/44
	Dense_sentence, *Dense_word*	89.58 ± 4.46	91.83 ± 6.82	86.08 ± 11.35	0.88 ± 0.06	0.96 ± 0.02	198/232/20/30
	Dense_sentence, *Dense_dword*	86.25 ± 5.03	90.06 ± 8.50	81.48 ± 12.95	0.85 ± 0.07	0.95 ± 0.03	186/228/24/42
	Dense_sentence, *Dense_pword*	87.08 ± 6.44	92.57 ± 9.94	80.67 ± 11.31	0.85 ± 0.07	0.96 ± 0.02	184/234/18/44
	Dense_word, *Dense_dword*, *Dense_pword*	88.54 ± 3.26	95.07 ± 7.21	81.14 ± 9.48	0.87 ± 0.04	0.97 ± 0.02	185/240/12/43
	Dense_dword, *Dense_pword*, *Dense_sentence*	88.74 ± 3.97	93.92 ± 7.93	82.82 ± 9.80	0.87 ± 0.05	0.97 ± 0.02	189/237/15/39
	Dense_word, *Dense_pword*, *Dense_sentence*	90.2 ± 4.06	94.81 ± 7.42	85.09 ± 9.21	0.89 ± 0.05	0.97 ± 0.02	194/239/13/34
	Dense_word, *Dense_dword*, *Dense_sentence*	89.58 ± 3.84	93.23 ± 8.96	85.98 ± 9.69	0.88 ± 0.04	0.97 ± 0.02	196/234/18/32
	Dense_word, *Dense_dword*, *Dense_pword*, *Dense_sentence*	90.41 ± 2.66	94.81 ± 7.45	85.55 ± 6.76	0.89 ± 0.02	0.98 ± 0.01	195/239/13/33
Hard voting	*Dense_word*, *Dense_dword*, *Dense_pword*	86.45 ± 3.26	94.23 ± 7.97	77.66 ± 10.92	0.84 ± 0.05	N/A	177/238/14/51
	Dense_dword, *Dense_pword*, *Dense_sentence*	86.04 ± 5.27	92.03 ± 8.74	78.87 ± 12.86	0.83 ± 0.07	N/A	180/233/19/48
	Dense_word, *Dense_pword*, *Dense_sentence*	87.50 ± 3.22	92.79 ± 7.65	81.16 ± 8.99	0.86 ± 0.04	N/A	185/235/17/43
	Dense_word, *Dense_dword*, *Dense_sentence*	86.66 ± 5.90	90.13 ± 10.64	83.32 ± 11.62	0.85 ± 0.06	N/A	190/226/26/38
	Dense_word, *Dense_dword*, *Dense_pword*, *Dense_sentence*	87.29 ± 3.54	92.22 ± 8.74	81.64 ± 9.32	0.86 ± 0.04	N/A	186/233/19/42

becomes evident that multiple soft voting ensembles outperformed the hard voting ensemble, even with a smaller number of base classifiers.

To provide a visual representation of the top-performing ensemble models, the ROC curve plot of the four ensembles is presented in Fig. 6. This plot illustrates the trade-off between the true positive rate and the false positive rate for each ensemble, aiding in the assessment of their performance across different decision thresholds.

5.3. Transfer learning via feature extraction

A DenseNet201 network pretrained on the ImageNet dataset is employed as a feature extractor to generate CNN features from each task-specific subset of the data. RGB images with dimensions of 400 × 400 pixels are provided as input to the pretrained CNN model to obtain the features. The one-dimensional features of each image are extracted from the final layer of the network, specifically the global max pool layer.

To examine the performance of the CNN features from each task-specific image subset individually, multiple machine learning models, including SVM, Random forest, and AdaBoost, are trained and evaluated. Tenfold cross-validation is employed for tuning the hyperparameters of the machine learning algorithms as well as for the final training and evaluation process.The hyperparameters used in each classifier for training task-specific features are detailed in Table 4.

The classification performance of the three classification algorithms trained with four different sets of features (word, dword, pword, and sentence) separately is presented in Table 5. The reported accuracy, precision, recall, F1-score, and AUC values in Table 5 represent the averages obtained from each fold of cross-validation.

Among all the extracted CNN features, the CNN features obtained from word images and pseudoword images demonstrated the highest classification performance. The SVM classifier trained with CNN features from word images achieved a classification accuracy of 91.7% (standard deviation: 3.5), while the SVM classifier trained with CNN

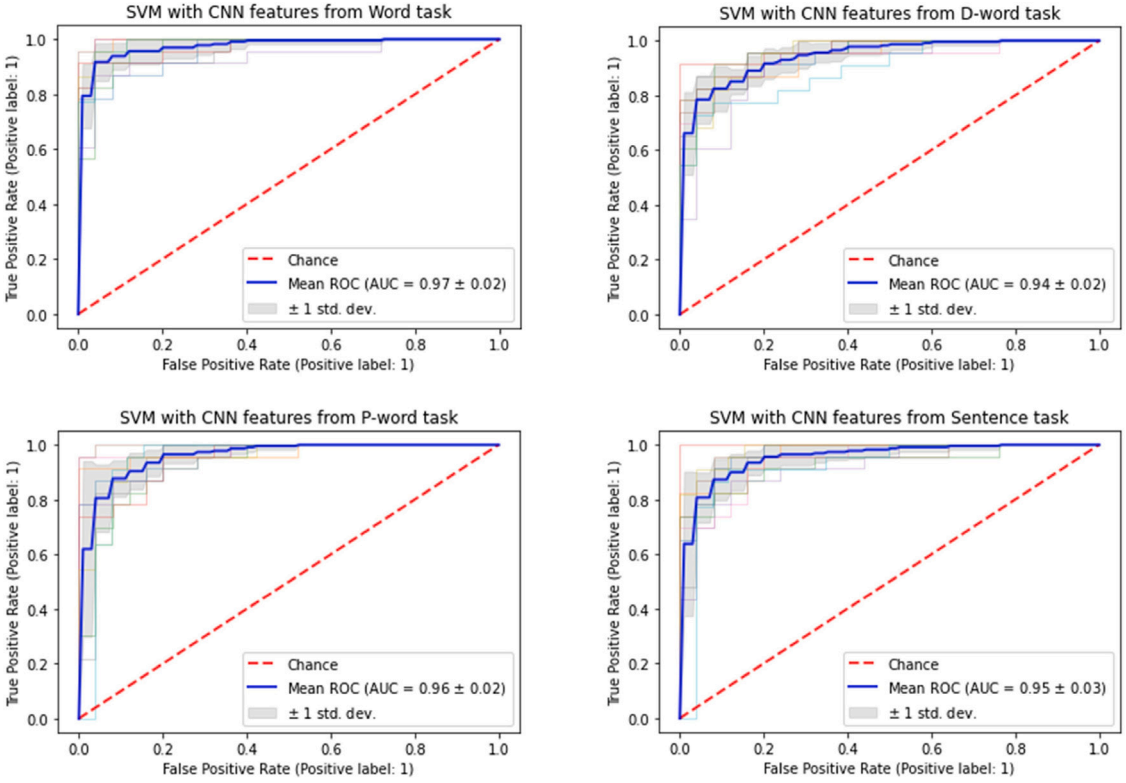


Fig. 7. ROC plot of top performing machine learning classifiers trained on task specific CNN features.

Table 4
Hyperparameter configuration of classifiers for task specific features.

Algorithm	Hyper parameters	Values
SVM	C	0.1
	gamma	1
	kernel	Polynomial
Random Forest	Splitting criterion	Gini
	No. of esitmators	150
	Maximum depth	10
	Minimum samples leaf	5
	Minimum samples split	5
AdaBoost	Learning rate	0.5
	No. of esitmators	150

features from pseudoword images achieved an accuracy of 90.0% (standard deviation: 4.1).

Similarly, in the Random forest classifier, the CNN features from word images yielded a classification accuracy of 79.4% (standard deviation: 6.1), and the CNN features from pseudoword images achieved an accuracy of 80.4% (standard deviation: 4.9). Furthermore, in the AdaBoost classifier, the CNN features extracted from word images achieved a classification accuracy of 82.5% (standard deviation: 3.3), while the CNN features from pseudoword images achieved an accuracy of 79.0% (standard deviation: 4.4).

Although the classification performance of the Random forest classifier and AdaBoost classifier using CNN features from word and pseudoword images might not be considered satisfactory, they still outperformed other types of features in terms of classification accuracy. These findings indicate that CNN features derived from word and pseudoword images hold significant discriminative information for the task at hand, as they consistently exhibited better performance across multiple classifiers.

Among all the classifiers utilized for classifying the CNN features extracted from various task-specific images, the Support Vector Machine (SVM) demonstrated superior performance. It outperformed other classifiers in effectively classifying these features. The SVM achieved a minimum classification accuracy of 87.1% (standard deviation: 2.0) when classifying difficult word images. Notably, this minimum accuracy value in SVM surpassed the accuracy achieved by any classifiers developed using either the Random forest or the AdaBoost algorithm.

Furthermore, when considering the recall value, SVM exhibited its effectiveness in classifying handwritten images for dysgraphia diagnosis. The SVM classifier trained with CNN features from word images achieved a recall value of 90.4%, while the SVM classifier trained with CNN features from pseudoword images achieved a recall value of 88.2%. These recall values indicate the SVM's ability to accurately identify positive cases, making it well-suited for dysgraphia diagnosis.

In contrast, the Random forest algorithm did not demonstrate satisfactory performance for a medical diagnosis problem. Although the Random forest classifier achieved a classification accuracy of approximately 80%, the poor recall values obtained render it less favorable for the dysgraphia diagnosis problem.

To provide a visual representation of the top-performing machine learning classifiers trained on task-specific CNN features, the ROC plot of the four classifiers is presented in Fig. 7. This plot illustrates the trade-off between the true positive rate and the false positive rate for each classifier, enabling a comprehensive assessment of their performance across different decision thresholds.

5.4. CNN feature fusion

This section presents the classification performance of machine learning algorithms trained on fused CNN features. The basic approach for CNN feature extraction involves generating CNN features separately for each task subset (word, difficult word, pseudoword, sentence) using all samples in the dataset. Subsequently, these CNN features are fused for each sample to implement feature fusion-based classifiers.

In this work, all possible combinations of feature fusion from the available CNN features generated for the independent tasks are examined. Once the features are fused, their performance is evaluated

Table 5
Accuracy, precision, recall, f1-score, and ROC_AUC scores of CNN features from the task-specific subset of the data trained on SVM, Random forest, and AdaBoost classifiers. The confusion matrix is in the order: TP, TN, FP, FN.

Classifiers	CNN features	Accuracy	Precision	Recall	F1-Score	AUC	Confusion matrix
SVM	CNN_dword	87.1 ± 2.0	87.9 ± 3.6	84.6 ± 3.7	0.861 ± 0.23	0.94 ± 0.02	193/225/27/35
	CNN_word	91.7 ± 3.5	92.3 ± 4.5	90.4 ± 7.2	0.92 ± 0.04	0.97 ± 0.02	206/234/18/22
	CNN_pword	90.0 ± 4.1	90.8 ± 5.8	88.2 ± 6.4	0.89 ± 0.05	0.96 ± 0.02	201/231/21/27
	$CNN\_sentence$	88.7 ± 4.2	89.5 ± 5.1	86.8 ± 7.0	0.88 ± 0.05	0.95 ± 0.03	198/228/24/30
Random forest	CNN_dword	78.3 ± 4.9	83.5 ± 7.4	68.3 ± 8.8	0.71 ± 0.08	0.88 ± 0.04	156/219/33/72
	CNN_word	79.4 ± 6.1	82.8 ± 8.3	68.0 ± 7.8	0.75 ± 0.07	0.89 ± 0.07	151/221/31/77
	CNN_pword	80.4 ± 4.9	85.0 ± 8.7	64.9 ± 7.7	0.77 ± 0.04	0.88 ± 0.04	160/227/25/68
	$CNN\_sentence$	75.8 ± 6.8	79.0 ± 10.2	63.5 ± 9.5	0.74 ± 0.07	0.86 ± 0.07	153/220/32/75
AdaBoost	CNN_dword	80.2 ± 4.4	81.2 ± 5.1	75.9 ± 6.6	0.78 ± 0.52	0.88 ± 0.04	173/212/40/55
	CNN_word	82.5 ± 3.3	82.5 ± 6.2	81.2 ± 7.5	0.81 ± 0.04	0.90 ± 0.03	185/211/41/43
	CNN_pword	79.0 ± 4.4	78.4 ± 6.4	77.6 ± 9.4	0.78 ± 0.06	0.85 ± 0.06	177/202/50/51
	$CNN\_sentence$	77.7 ± 7.0	78.8 ± 6.4	72.3 ± 12	0.75 ± 0.08	0.84 ± 0.06	165/208/44/63

Table 6
Hyperparameter configuration of classifiers for fused features.

Algorithm	Hyper Parameters	Values
SVM	C	0.1
	gamma	1
	kernel	Polynomial
Random Forest	Splitting criterion	Gini
	No. of esitmators	200
	Maximum depth	10
	Minimum samples leaf	5
	Minimum samples split	10
AdaBoost	Learning rate	0.5
	No. of esitmators	200

using SVM, Random forest, and AdaBoost classifiers. The details of hyperparameters used in each classifier for training fused features are shown in Table 6.

By exploring different combinations of fused CNN features and utilizing various machine learning algorithms, we aim to assess the effectiveness of feature fusion in enhancing the classification performance. The performance evaluation of these fused features provides insights into the potential benefits of combining task-specific CNN features for improving the accuracy and robustness of the classification system.

The feature fusion process is carried out incrementally, starting with the generation of all possible pairwise combinations from the available four independent feature vectors: CNN_{word}, CNN_{dword}, CNN_{pword}, and $CNN_{sentence}$. This results in a total of six possible feature combinations for fusion. Once a pair of feature vectors is horizontally fused or concatenated, they are evaluated using various machine learning algorithms.

Table 7 presents the classification performance of these six possible feature combinations when employed with three different machine learning algorithms. The reported accuracy, precision, recall, F1-score, and AUC values in Table 7 represent the averages obtained from each fold of cross-validation.

The fusion of features extracted from handwritten images belonging to two independent writing tasks has led to a substantial improvement in classification performance. Specifically, when training Support Vector Machine (SVM) classifiers using the CNN_{dword} features and CNN_{word} features independently, the classification accuracies achieved were 87.1% (standard deviation: 2.0) and 91.% (standard deviation: 3.5), respectively.

However, upon fusing the CNN_{dword} and CNN_{word} features and training an SVM classifier with the resulting feature vectors, a significant improvement in classification accuracy was observed, reaching 95.4% (standard deviation: 2.6). This improvement highlights the effectiveness of feature fusion in enhancing classification performance.

Moreover, the recall value of the SVM classifier trained with the fusion of CNN_{dword} and CNN_{word} features was found to be 94.8% (standard deviation: 5.1). This means that out of 100 positive samples in the test set, the proposed method can correctly classify approximately 95 of them as positive. The high recall value demonstrates the superior performance of feature fusion methods in dysgraphia classification from handwritten images.

Similarly, in all other fusion combinations of two feature sets trained on SVM, the classification performance exhibited improvement. This further emphasizes the effectiveness of feature fusion in enhancing the classification capabilities of the SVM classifier.

The classification performance of the five possible feature combinations (fusion of three or more independent CNN features) with three machine learning algorithms is presented in Table 8. The reported accuracy, precision, recall, F1-score, and AUC values in Table 8 represent the averages obtained from each fold of cross-validation.

The fusion of CNN_{word}, CNN_{dword}, and CNN_{pword} features has demonstrated remarkable classification performance when trained using the Support Vector Machine (SVM) classifier. This fusion approach achieved a classification accuracy of 97.3% (standard deviation: 1.6) and a recall value of 97.% (standard deviation: 3.9), which are the highest among all the listed classification performance metrics in Table 8.

Among all the proposed methods in this study, the SVM classifier trained on the fusion of CNN_{word}, CNN_{dword}, and CNN_{pword} features exhibited the best classification performance across multiple evaluation metrics, including accuracy, recall, and F1-score. In a 10-fold cross-validation setup, out of 228 positive samples in the test set, the SVM classifier correctly predicted 221 samples as positives, showcasing the efficacy of the proposed method for dysgraphia diagnosis from handwritten images.

In the Random forest and AdaBoost classifiers, slight improvements in classification performance were observed when three or more features were fused. Moreover, the best performance in these classifiers was achieved when all four features (CNN_{word}, CNN_{dword}, CNN_{pword}, and $CNN_{sentence}$) were fused. To visually illustrate the performance of the top-performing machine learning classifiers trained on the fusion of task-specific CNN features, the ROC curve plot of these classifiers is presented in Fig. 8.

5.5. Comparison with state of the art methods

The effectiveness of the proposed methods is demonstrated by comparing their performance with state-of-the-art dysgraphia diagnosis methods evaluated on the same dataset. Table 9 presents a performance comparison between the proposed method and the state-of-the-art methods. The proposed methods are highlighted in bold, and the type of data analysis (online/offline) is provided in the second column of Table 9.

To compare the performance of our work, we considered previous studies in the literature that utilized the same dataset for evaluation. Specifically, three works (Drotár & Dobeš, 2020b; Kunhoth et al.,

Table 7
Accuracy, precision, recall, f1-score, and ROC_AUC scores of SVM, Random forest, and AdaBoost classifiers trained on fused pair of CNN features.

Classifiers	Fused features	Accuracy	Precision	Recall	F1-Score	AUC	Confusion matrix
SVM	CNN_word, CNN_dword	95.4 ± 2.6	95.7 ± 3.0	94.8 ± 5.1	0.95 ± 0.03	0.99 ± 0.01	216/242/10/12
	CNN_dword, CNN_pword	91.9 ± 3.0	93.0 ± 4.7	90.0 ± 4.7	0.91 ± 0.03	0.97 ± 0.02	205/236/16/23
	CNN_pword, CNN_word	95.2 ± 2.5	96.1 ± 0.30	93.9 ± 5.6	0.95 ± 0.03	0.99 ± 0.01	214/243/9/14
	$CNN\_Sentence$, CNN_word	93.3 ± 3.1	94.3 ± 4.5	91.7 ± 4.2	0.93 ± 0.03	0.98 ± 0.01	209/239/13/19
	$CNN\_Sentence$, CNN_dword	91.2 ± 5.4	93.0 ± 7.6	88.5 ± 5.0	0.91 ± 0.06	0.97 ± 0.02	202/236/16/26
	$CNN\_Sentence$, CNN_pword	90.0 ± 5.2	90.6 ± 7.7	88.6 ± 5.6	0.89 ± 0.05	0.97 ± 0.03	202/230/22/26
Random forest	CNN_word, CNN_dword	81.0 ± 4.5	87.5 ± 8.8	72.0 ± 9.1	0.8 ± 0.07	0.92 ± 0.04	169/230/22/59
	CNN_dword, CNN_pword	80.2 ± 5.0	88.6 ± 8.3	68.8 ± 5.9	0.77 ± 0.05	0.87 ± 0.04	159/231/21/69
	CNN_pword, CNN_word	82.7 ± 6.3	90.0 ± 8.2	69.3 ± 9.3	0.8 ± 0.07	0.93 ± 0.04	171/232/20/57
	$CNN\_Sentence$, CNN_word	79.0 ± 6.7	88.2 ± 9.9	68.4 ± 10.1	0.77 ± 0.09	0.91 ± 0.05	150/228/24/78
	$CNN\_Sentence$, CNN_dword	79.8 ± 4.8	87.5 ± 8.1	67.5 ± 9.5	0.78 ± 0.07	0.89 ± 0.04	159/230/22/69
	$CNN\_Sentence$, CNN_pword	79.8 ± 4.6	86.4 ± 10.1	69.3 ± 9.1	0.75 ± 0.05	0.88 ± 0.03	160/216/36/68
AdaBoost	CNN_word, CNN_dword	84.8 ± 5.0	86.1 ± 7.1	81.6 ± 6.0	0.84 ± 0.05	0.92 ± 0.04	186/221/31/42
	CNN_dword, CNN_pword	84.4 ± 3.9	85.9 ± 7.4	81.1 ± 3.5	0.83 ± 0.04	0.90 ± 0.04	185/220/32/43
	CNN_pword, CNN_word	87.3 ± 4.8	87.4 ± 6.4	86.0 ± 5.1	0.87 ± 0.05	0.94 ± 0.03	196/223/29/32
	$CNN\_Sentence$, CNN_word	84.6 ± 5.3	85.4 ± 7.2	82.0 ± 5.7	0.84 ± 0.05	0.93 ± 0.03	187/219/33/41
	$CNN\_Sentence$, CNN_dword	83.3 ± 4.2	85.5 ± 5.6	78.5 ± 5.9	0.82 ± 0.05	0.91 ± 0.04	179/221/31/49
	$CNN\_Sentence$, CNN_pword	82.5 ± 4.4	83.6 ± 7.4	79.4 ± 6.9	0.81 ± 0.05	0.90 ± 0.03	181/215/37/47

2023; Skunda et al., 2022) utilized this dataset, but all of them focused on analyzing online handwritten data. In contrast, our proposed approach transformed the online data into offline images and further analyzed them for dysgraphia diagnosis. Among the available methods evaluated on this dataset, Drotár and Dobeš (2020b), Kunhoth et al. (2023) achieved the highest performance. However, the maximum classification accuracy reported in the literature was approximately 81%. In comparison, our proposed approach significantly improved the classification performance, achieving an accuracy of 97.3%. This substantial improvement demonstrates the efficacy of our proposed methods. Moreover, our approach had an advantage in terms of the number of data samples. We employed data augmentation techniques to increase the number of training samples, thereby enhancing performance. Additionally, the adoption of feature fusion and ensemble learning techniques played a significant role in developing an intelligent decision-making algorithm with an accuracy of 97%.

5.6. Effect of image resizing

This section focuses on investigating the effects of image resizing on the analysis of handwritten images. Since handwritten images are typically rectangular, resizing them to a square shape can result in information loss if the aspect ratio is not preserved. Two types of resizing approaches mentioned in the literature were considered in this work: normal resizing without preserving the aspect ratio and aspect ratio-preserving resizing with padding (Cho et al., 2020; Hashemi, 2019).

The first approach involved normal resizing, where rectangular images were resized to 400 × 400 pixels using the inter-area interpolation method. The second approach extended the height of the images by padding the rectangular images with white pixels at the top and bottom, and the resulting square image was then resized to 400 × 400 pixels. The resizing approaches are visualized in Fig. 9. To evaluate the effectiveness of these resizing approaches, the resulting images were subjected to transfer learning using the feature extraction method. The classification performance of SVM, Random forest, and AdaBoost classifiers was assessed based on task-specific features extracted from two sets of data: one resized while preserving the aspect ratio, and the other resized without preserving the aspect ratio. The classification performance results are provided in Table 10.

The results presented in Table 10 indicate that, in most cases, there were no significant changes in the classification performance between the two considered data resizing approaches. However, in some cases, particularly with the SVM classifier, features extracted from data resized using the normal resizing approach demonstrated better classification performance compared to the features extracted from data resized while preserving the aspect ratio. The most notable changes in classification performance were observed in SVM trained with features extracted from word data and SVM trained with features extracted from sentence data. In both cases, the data resized using the normal resizing approach yielded superior results compared to the data resized while preserving the aspect ratio.

Table 8
Accuracy, precision, recall, f1-score, and ROC_AUC scores of SVM, Random forest, and AdaBoost classifiers trained on fusion of three or more independent CNN features. The confusion matrix is in the order: TP, TN, FP, FN.

Classifiers	Fused features	Accuracy	Precision	Recall	F1-Score	AUC	Confusion matrix
SVM	CNN_word, CNN_dword, CNN_pword	97.3 ± 1.6	97.4 ± 2.1	97.0 ± 3.9	0.97 ± 0.02	0.99 ± 0.01	221/246/6/7
	CNN_dword, CNN_pword, $CNN\_sentence$	94.4 ± 2.6	96.2 ± 4.9	92.2 ± 4.7	0.94 ± 0.03	0.99 ± 0.01	210/243/9/18
	CNN_word, CNN_pword, $CNN\_sentence$	96.7 ± 2.5	97.5 ± 4.3	95.6 ± 2.8	0.97 ± 0.03	0.99 ± 0.01	218/246/6/10
	CNN_word, CNN_dword, $CNN\_sentence$	96.5 ± 2.3	97.8 ± 2.9	94.8 ± 3.8	0.96 ± 0.02	0.99 ± 0.01	216/247/5/12
	CNN_word, CNN_dword, CNN_pword, $CNN\_sentence$	96.9 ± 2.1	97.4 ± 4.5	96.1 ± 4.5	0.97 ± 0.02	0.99 ± 0.00	219/246/6/9
Random forest	CNN_word, CNN_dword, CNN_pword	82.5 ± 4.7	91.8 ± 4.1	72.4 ± 8.8	0.81 ± 0.05	0.92 ± 0.03	165/233/19/63
	CNN_dword, CNN_pword, $CNN\_sentence$	80.8 ± 4.4	89.0 ± 8.3	71.1 ± 6.7	0.78 ± 0.06	0.91 ± 0.04	164/229/23/64
	CNN_word, CNN_pword, $CNN\_sentence$	80.6 ± 5.6	88.6 ± 8.8	67.5 ± 12.4	0.79 ± 0.09	0.92 ± 0.03	163/234/18/65
	CNN_word, CNN_dword, $CNN\_sentence$	81.5 ± 6.2	88.1 ± 6.3	71.1 ± 9.2	0.80 ± 0.07	0.92 ± 0.03	163/229/23/65
	CNN_word, CNN_dword, CNN_pword, $CNN\_sentence$	83.1 ± 5.5	89.4 ± 5.9	72.4 ± 10.9	0.81 ± 0.04	0.93 ± 0.04	168/240/12/60
AdaBoost	CNN_word, CNN_dword, CNN_pword	85.8 ± 6.0	87.3 ± 7.8	82.9 ± 7.4	0.85 ± 0.06	0.93 ± 0.03	189/223/29/39
	CNN_dword, CNN_pword, $CNN\_sentence$	83.3 ± 3.4	84.3 ± 6.2	80.2 ± 4.7	0.82 ± 0.04	0.91 ± 0.02	183/217/35/45
	CNN_word, CNN_pword, $CNN\_sentence$	85.8 ± 3.7	87.0 ± 8.0	83.8 ± 6.4	0.85 ± 0.04	0.92 ± 0.04	191/221/31/37
	CNN_word, CNN_dword, $CNN\_sentence$	85.4 ± 4.6	86.1 ± 6.5	83.4 ± 7.2	0.84 ± 0.05	0.93 ± 0.03	190/220/32/38
	CNN_word, CNN_dword, CNN_pword, $CNN\_sentence$	86.7 ± 5.3	87.9 ± 7.4	84.3 ± 8.3	0.86 ± 0.06	0.94 ± 0.04	192/224/28/36

Table 9
Comparison with state-of-the-art methods.

Methods	Data type	Accuracy
AdaBoost (Drotár & Dobeš, 2020b)	Online	79.5%
AdaBoost (Kunhoth et al., 2023)	Online	80.8%
CNN (Skunda et al., 2022)	Online	79.7%
SVM with word, dword and sentence features	**Offline**	**97.3%**

6. Discussion

This research article aims to assess the effectiveness of handwritten image analysis, specifically offline handwritten data analysis, for diagnosing dysgraphia in children. The study begins by transforming an online handwritten dataset into offline images, which are further categorized into word images, pseudo-word images, difficult word images, and sentence word images. The primary investigation revolves around evaluating the performance of transfer learning through fine-tuning and feature extraction on the transformed image data. Task-specific convolutional neural networks (CNNs) are developed using transfer learning via fine-tuning, with the DenseNet201 architecture fine-tuned on the word dataset exhibiting the highest classification performance (accuracy: 84.79% ±4.93%) among all the task-specific CNNs. The remaining fine-tuned task-specific CNNs achieve a minimum classification accuracy of approximately 80%. However, the recall values of all four fine-tuned task-specific CNN classifiers do not meet the desired standards.

On the contrary, CNN features are extracted from the task-specific data using a pre-trained DenseNet201 network. These CNN features are utilized to develop classifiers employing three machine-learning algorithms: Support Vector Machines (SVM), Random forest, and AdaBoost. The performance of these algorithms is analyzed for classifying the CNN features derived from the handwritten data. The results reveal the superiority of the SVM algorithm over the other approaches. SVM classifiers

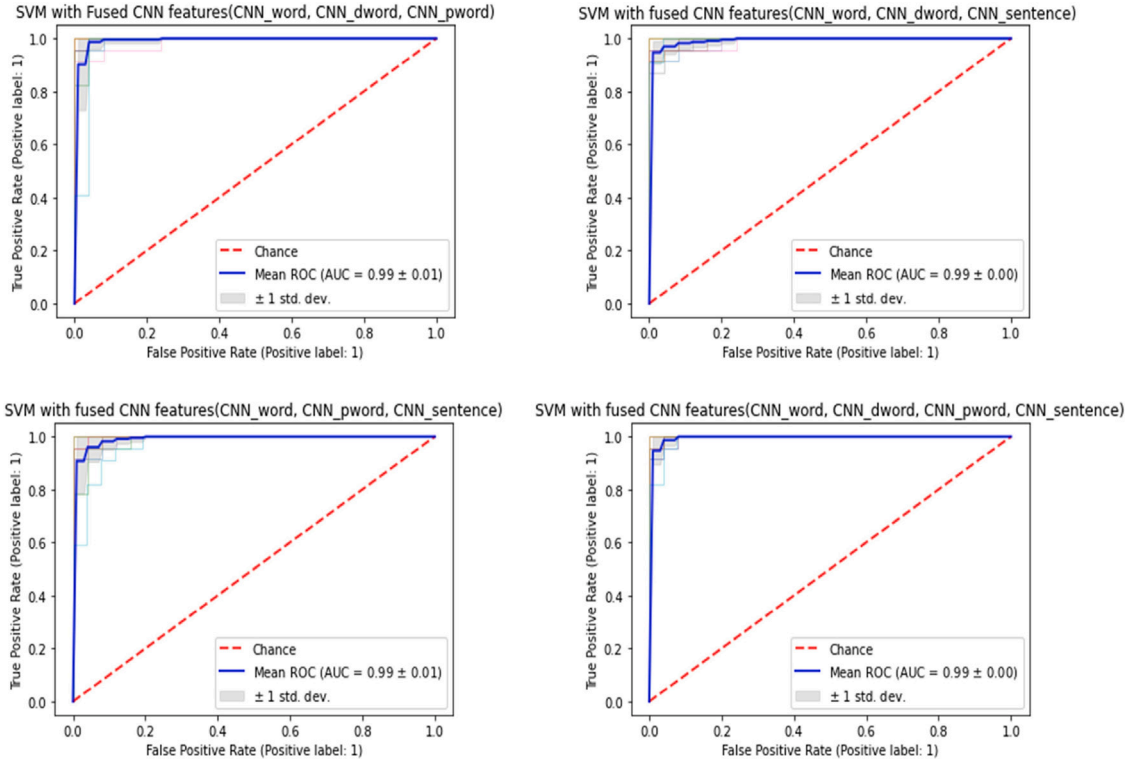


Fig. 8. ROC plot of top performing machine learning classifiers trained on fusion of task specific CNN features.

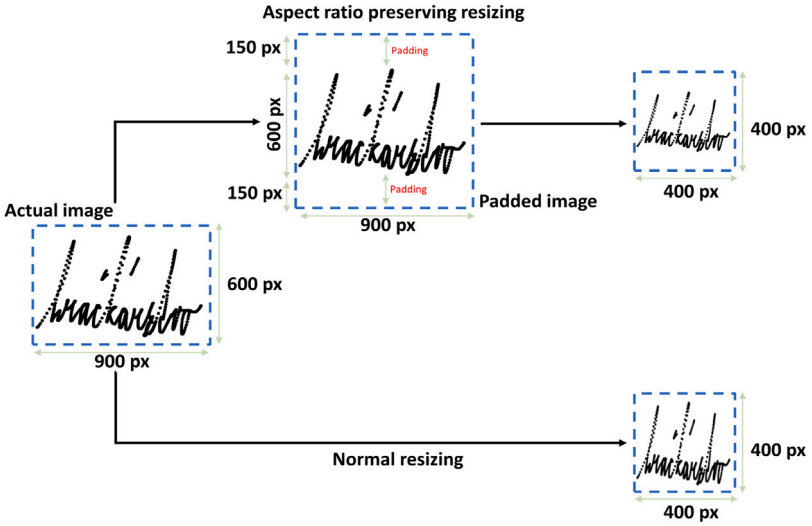


Fig. 9. Image resizing approach. The dimension of the original image is representative.

trained on the CNN features from each task-specific dataset exhibit promising outcomes. Notably, the CNN features extracted from difficult word images achieve a minimum classification accuracy of 87.1%, while the word data yields the maximum accuracy of 90.0%. Moreover, the SVM classifier trained on the CNN features from the word data demonstrates an acceptable recall value of 90.4%, the highest among all classifiers trained on task-specific data. The varying classification performances obtained by classifiers trained on different subcategories of data indicate that certain writing tasks possess greater discriminatory power in identifying the presence of dysgraphia.

Subsequently, the investigation aimed to enhance the diagnostic/classification performance by leveraging the integration of information from independent task-specific data or classifiers trained on such data. To achieve this, an ensemble of fine-tuned task-specific CNNs was developed, and traditional machine learning classifiers were trained on the fusion of features extracted from task-specific data. Two distinct ensemble learning strategies, namely soft voting and hard voting, were employed to aggregate predictions from multiple independent classifiers. However, the hard voting ensemble did not yield a significant improvement in classification performance compared to the soft voting ensembles.

Among the ensemble models developed, the soft voting ensemble of four fine-tuned CNNs, each trained on word data, pseudoword data, difficult word data, and sentence data separately, demonstrated the most favorable classification performance. Remarkably, the soft voting ensemble of two fine-tuned CNNs, trained on word data and sentence data individually, exhibited a very similar performance to the top-performing ensemble, which comprised four fine-tuned CNNs.

Table 10
Performance comparison of classifiers with data resizing approaches. Here, A.R.P means aspect ratio preserving resizing.

Classifiers	CNN features	Accuracy	
		Normal resizing	A.R.P resizing
SVM	CNN_dword	87.1 ± 2.0	87.9 ± 4.8
	CNN_word	91.7 ± 3.5	85.6 ± 3.5
	CNN_pword	90.0 ± 4.1	89.4 ± 3.2
	$CNN\_sentence$	88.7 ± 4.2	85.4 ± 3.5
Random forest	CNN_dword	78.3 ± 4.9	81.2 ± 6.6
	CNN_word	79.4 ± 6.1	77.9 ± 8.2
	CNN_pword	80.4 ± 4.9	81.2 ± 7.0
	$CNN\_sentence$	75.8 ± 6.8	75.4 ± 7.9
AdaBoost	CNN_dword	80.2 ± 4.4	79.2 ± 3.7
	CNN_word	82.5 ± 3.3	83.8 ± 4.0
	CNN_pword	79.0 ± 4.4	82.9 ± 3.6
	$CNN\_sentence$	77.7 ± 7.0	77.9 ± 4.8

These findings suggest that increasing the number of base classifiers does not necessarily lead to a substantial performance improvement. Specifically, the ensemble of two base classifiers and the ensemble of four base classifiers achieved comparable performances, with the latter being computationally more demanding for prediction purposes.

The SVM classifiers trained on the fused CNN features derived from task-specific data exhibited a significant enhancement in classification performance. The fusion of CNN features from word and sentence data yielded a minimum classification accuracy of 90%, while the fusion of CNN features from word, pseudoword, and difficult word data achieved a maximum accuracy of 97.3%. These results underscore the superior performance achieved through the feature fusion approach. Notably, when utilizing the SVM classifier on the fused features from word, pseudoword, and difficult word data, the recall value improved to an excellent 97.0%. Comparative analysis demonstrated that both the fusion of four task-specific feature sets and the fusion of three task-specific feature sets yielded similar performance levels in SVM classification. Consequently, it is advisable to employ classifiers trained on the fusion of three feature sets, as expanding the feature set further increases the dimensionality of the fused features, thereby augmenting computational complexity.

Compared to the state of the art methods, our proposed approach yielded a substantial improvement in classification performance, attaining an accuracy of 97.3%. This significant enhancement highlights the effectiveness of our proposed methods. However, our approach have an advantage in terms of data sample size. We employed data augmentation techniques to augment the number of training samples, thereby bolstering performance. But further results shows that the incorporation of feature fusion and ensemble learning techniques played a pivotal role in significantly improving the classification performance and achieving a recall value of 97%

The scalability of the dysgraphia diagnosis classifiers developed in this study is limited due to the dataset's reliance on Slovak orthography. Consequently, these classifiers would not be suitable for distinguishing handwriting in English orthography. To address this limitation, the development of new classifiers specifically trained on English orthographic data is necessary. However, there is potential to leverage the existing classifiers for fine-tuning or initializing the weights of classifiers designed for English orthography.

The proposed methodologies are primarily capable of differentiating between normally developing handwriting and dysgraphia. It is important to note that dysgraphia severity varies among individuals, particularly in children. Identifying the level of severity can provide valuable insights to occupational therapists for tailoring specific treatments.

Future research directions encompass exploring the utilization of multimodal data for training intelligent decision-making classifiers in the context of dysgraphia diagnosis. This involves fusing features extracted from online handwritten data with those obtained from corresponding offline images to train classification models. Additionally, investigating the potential of weighted ensembles comprising task-specific deep CNN classifiers could be explored. Furthermore, novel methods for implementing feature fusion strategies warrant investigation.

Machine learning-based dysgraphia screening systems has practical applications in educational institutions, clinics, and diagnostic centers. It offers early identification and intervention support, reducing the need for subjective evaluations. By analyzing data using machine learning techniques, it enables educators, healthcare professionals, and parents to identify children with dysgraphia and deliver targeted interventions. Additionally, machine learning techniques allows for longitudinal tracking of a child's progress, facilitating ongoing assessment and adjustment of intervention strategies based on individual needs.

7. Conclusion

This work proposed machine learning-based methods for diagnosing dysgraphia using handwritten images/offline handwritten data. An online handwritten dataset has been transformed into offline images, and multiple experiments were conducted on this transformed dataset. The study explores the potential of transfer learning via feature extraction and transfer learning via fine-tuning in developing machine learning and deep CNN classifiers for dysgraphia diagnosis. The results obtained from task-specific deep CNN and traditional ML classifiers suggest that offline data from specific tasks, particularly word writing, play a crucial role in achieving accurate diagnoses. Additionally, this work enhances the classification performance of the diagnosis models through the introduction of ensemble learning and feature fusion approaches. Ensemble learning leverages soft voting and hard voting-based prediction strategies to aggregate the independent predictions from task-specific CNN classifiers. The experimental results demonstrate that soft voting-based ensembles consistently outperform hard voting-based ensembles across all possible combinations of task-specific CNN classifiers. The feature fusion approach adopts a straightforward technique of horizontally concatenating the features extracted from multiple task-specific datasets. This approach significantly improves the classification performance, resulting in a remarkable classification accuracy of 97.3%. Notably, this accuracy is approximately 7% higher than the maximum accuracy achieved by the ensemble learning classifiers. By proposing ensemble learning and feature fusion techniques, this work successfully enhances the accuracy and effectiveness of dysgraphia diagnosis models, providing a valuable contribution to the field of dysgraphia assessment and offering promising avenues for future research in this domain.

Ethics approval

This paper complies with the ethical standards of research and methodology

CRediT authorship contribution statement

Jayakanth Kunhoth: Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Visualization. **Somaya Al Maadeed:** Supervision, Conceptualization, Writing – review & editing, Project administration, Funding acquisition. **Moutaz Saleh:** Supervision, Writing – review & editing. **Younes Akbari:** Conceptualization, Writing – review & editing, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data is available online and link is shared in the manuscript. The code of methods will be shared up on request.

Funding

This publication was supported by Qatar University Graduate Assistant Grant. The contents of this publication are solely the responsibility of the authors and do not necessarily represent the official views of Qatar University. Open Access funding provided by the Qatar National Library.

References

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

Akbal, E., Barua, P. D., Dogan, S., Tuncer, T., & Acharya, U. R. (2022). Despatnet25: Data encryption standard cipher model for accurate automated construction site monitoring with sound signals. *Expert Systems with Applications*, *193*, Article 116447.

American Psychiatric Association. & American Psychiatric Association. DSM-5 Task Force (2013). *Diagnostic and statistical manual of mental disorders : DSM-5*. American Psychiatric Association.

Ammour, A., Aouraghe, I., Khaissidi, G., Mrabti, M., Aboulem, G., & Belahsen, F. (2020). A new semi-supervised approach for characterizing the arabic on-line handwriting of parkinson's disease patients. *Computer Methods and Programs in Biomedicine*, *183*, Article 104979.

Asselborn, T., Chapatte, M., & Dillenbourg, P. (2020). Extending the spectrum of dysgraphia: A data driven strategy to estimate handwriting quality. *Scientific Reports*, *10*, 1–11.

Asselborn, T., Gargot, T., Kidziński, W., Cohen, D., Jolly, C., & Dillenbourg, P. (2018). Automated human-level diagnosis of dysgraphia using a consumer tablet. *Npj Digital Medicine*, *1*.

Barnett, A. L., Henderson, S. E., Scheib, B., & Schulz, J. (2009). Development and standardization of a new handwriting speed test: The detailed assessment of speed of handwriting. *British Journal of Educational Psychology*.

Biau, G. (2012). Analysis of a random forests model.

Chai, J., Wu, R., Li, A., Xue, C., Qiang, Y., Zhao, J., et al. (2023). Classification of mild cognitive impairment based on handwriting dynamics and qeeg. *Computers in Biology and Medicine*, *152*, Article 106418.

Cho, B. H., Lee, D. Y., Park, K.-A., Oh, S. Y., Moon, J. H., Lee, G.-I., et al. (2020). Computer-aided recognition of myopic tilted optic disc using deep learning algorithms in fundus photography. *BMC Ophthalmology*, *20*, 1–9.

Dankovicova, Z., Hurtuk, J., & Fecilak, P. (2019). Evaluation of digitalized handwriting for dysgraphia detection using random forest classification method. In *SISY 2019 - IEEE 17th international symposium on intelligent systems and informatics, proceedings*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). Ieee.

Deuel, R. K. (1995). Developmental dysgraphia and motor skills disorders. *Journal of Child Neurology*.

Devi, A., & Kavya, G. (2022). Dysgraphia disorder forecasting and classification technique using intelligent deep learning approaches. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, Article 110647.

Devi, A., Kavya, G., Therese, M. J., & Gayathri, R. (2021). Early diagnosing and identifying tool for specific learning disability using decision tree algorithm. In *2021 Third international conference on inventive research in computing applications* (pp. 1445–1450). IEEE.

Dimauro, G., Bevilacqua, V., Colizzi, L., & Di Pierro, D. (2020). TestGraphia, a software system for the early diagnosis of dysgraphia. *IEEE Access*, *8*, 19564–19575.

Drotár, P., & Dobeš, M. (2020a). Dysgraphia detection dataset. https://github.com/peet292929/Dysgraphia-detection-through-machine-learning.

Drotár, P., & Dobeš, M. (2020b). Dysgraphia detection through machine learning. *Scientific Reports*, *10*, 1–11.

Drotár, P., Mekyska, J., Rektorová, I., Masarová, Z., & Faundez-Zanuy, M. (2014). Analysis of in-air movement in handwriting: A novel marker for parkinson's disease. *Computer Methods and Programs in Biomedicine*, *117*, 405–411.

Dui, L. G., Lunardini, F., Termine, C., Matteucci, M., Stucchi, N. A., Borghese, N. A., et al. (2020). A tablet app for handwriting skill screening at the preliteracy stage: Instrument validation study. *JMIR Serious Games*, *8*, Article e20126.

Fakhrou, A., Kunhoth, J., & Al Maadeed, S. (2021). Smartphone-based food recognition system using multiple deep cnn models. *Multimedia Tools and Applications*, *80*, 33011–33032.

Faundez-Zanuy, M., Fierrez, J., Ferrer, M. A., Diaz, M., Tolosana, R., & Plamondon, R. (2020). Handwriting biometrics: Applications and future trends in e-security and e-health. *Cognitive Computation*.

Gargot, T., Asselborn, T., Pellerin, H., Zammouri, I., Anzalone, S. M., Casteran, L., et al. (2020). Acquisition of handwriting in children with and without dysgraphia: A computational approach. *PLoS ONE*, *15*, 1–22.

Ghouse, F., Paranjothi, K., & Vaithiyanathan, R. (2022). Dysgraphia classification based on the non-discrimination regularization in rotational region convolutional neural network. *International Journal of Intelligent Engineering & Systems*, *15*.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Guilbert, J., Alamargot, D., & Morin, M. F. (2019). Handwriting on a tablet screen: Role of visual and proprioceptive feedback in the control of movement by children and adults. *Human Movement Science*.

Hamstra-Bletz, L., & de Bie J, d. B. B. (1987). Concise evaluation scale for children's handwriting. Swets 1 zeitlinger ed.Lisse.

Hashemi, M. (2019). Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation. *Journal of Big Data*, *6*, 1–13.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).

Isa, I. S., Syazwani Rahimi, W. N., Ramlan, S. A., & Sulaiman, S. N. (2019). Automated detection of dyslexia symptom based on handwriting image for primary school children. *Procedia Computer Science*, *163*, 440–449.

Karadal, C. H., Kaya, M. C., Tuncer, T., Dogan, S., & Acharya, U. R. (2021). Automated classification of remote sensing images using multileveled mobilenetv2 and dwt techniques. *Expert Systems with Applications*, *185*, Article 115659.

Kawa, J., Bednorz, A., Stepien, P., Derejczyk, J., & Bugdol, M. (2017). Spatial and dynamical handwriting analysis in mild cognitive impairment. *Computers in Biology and Medicine*, *82*, 21–28.

Kedar, S., et al. (2021). Identifying learning disability through digital handwriting analysis. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*, 46–56.

Kunhoth, J., Al-Maadeed, S., Kunhoth, S., & Akbari, Y. (2022a). Automated systems for diagnosis of dysgraphia in children: A survey and novel framework. arXiv preprint arXiv:2206.13043.

Kunhoth, J., Al Maadeed, S., Saleh, M., & Akbari, Y. (2022b). Machine learning methods for dysgraphia screening with online handwriting features. In *2022 International conference on computer and applications* (pp. 1–6).

Kunhoth, J., Al Maadeed, M., & Akbari, Y. (2023). Exploration and analysis of on-surface and in-air handwriting attributes to improve dysgraphia disorder diagnosis in children based on machine learning methods. *Biomedical Signal Processing and Control*, *83*, Article 104715.

Kunhoth, J., Karkar, A., Al-Maadeed, S., & Al-Attiyah, A. (2019). Comparative analysis of computer-vision and ble technology based indoor navigation systems for people with visual impairments. *International Journal of Health Geographics*, *18*, 1–18.

Lopez, C., Hemimou, C., Golse, B., & Vaivre-Douret, L. (2018). Developmental dysgraphia is often associated with minor neurological dysfunction in children with developmental coordination disorder (DCD). *Neurophysiologie Clinique*.

Mekyska, J., Bednarova, J., Faundez-Zanuy, M., Galaz, Z., Safarova, K., Zvoncak, V., et al. (2019). Computerised assessment of graphomotor difficulties in a cohort of school-aged children. In *International congress on ultra modern telecommunications and control systems and workshops, 2019-Octob*.

Mekyska, J., Faundez-Zanuy, M., Mzourek, Z., Galaz, Z., Smekal, Z., & Rosenblum, S. (2017). Identification and rating of developmental dysgraphia by handwriting analysis. *IEEE Transactions on Human–Machine Systems*, *47*, 235–248.

Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. In *Machine learning* (pp. 101–121). Elsevier.

Prunty, M. M., Pratt, A., Raman, E., Simmons, L., & Steele-Bobat, F. (2020). Grip strength and pen pressure are not key contributors to handwriting difficulties in children with developmental coordination disorder. *British Journal of Occupational Therapy*, *83*, 387–396.

Ribeiro, L. C., Afonso, L. C., & Papa, J. P. (2019). Bag of samplings for computer-assisted parkinson's disease diagnosis based on recurrent neural networks. *Computers in Biology and Medicine*, *115*, Article 103477.

Richard, G., & Serrurier, M. (2020). Dyslexia and dysgraphia prediction: A new machine learning approach. arXiv.

Rosenblum, S., & Dror, G. (2016). Identifying developmental dysgraphia characteristics utilizing handwriting classification methods. *IEEE Transactions on Human–Machine Systems*, *47*, 293–298.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Schapire, R. E. (2013). Explaining adaboost. In *Empirical inference* (pp. 37–52). Springer.

Sharmila, C., Shanthi, N., Santhiya, S., Saran, E., Sri Rakesh, K., & Sruthi, R. (2023). An automated system for the early detection of dysgraphia using deep learning algorithms. In *2023 International conference on sustainable computing and data communication systems* (pp. 251–257).

Sihwi, S. W., Fikri, K., & Aziz, A. (2019). Dysgraphia identification from handwriting with support vector machine method. *vol. 1201*, In *Journal of Physics: Conference Series*.

Skunda, J., Nerusil, B., & Polec, J. (2022). Method for dysgraphia disorder detection using convolutional neural network. In *Proceedings - WSCG 2022: 30th International conference in central europe on computer graphics, visualization and computer vision*. Václav Skala-UNION Agency.

Vovk, V. (2015). The fundamental nature of the log loss function. In *Fields of logic and computation II: Essays dedicated To Yuri Gurevich on the Occasion of His 75th Birthday* (pp. 307–318).

Yildiz, A. M., Barua, P. D., Dogan, S., Baygin, M., Tuncer, T., Ooi, C. P., et al. (2023). A novel tree pattern-based violence detection model using audio signals. *Expert Systems with Applications*, *224*, Article 120031.

Zvoncak, V., Mekyska, J., Safarova, K., Smekal, Z., & Brezany, P. (2019). New approach of dysgraphic handwriting analysis based on the tunable Q-factor wavelet transform. In *2019 42nd International Convention on information and communication technology, electronics and microelectronics, MIPRO 2019 - Proceedings* (pp. 289–294).