*International Journal of Basic and Applied Sciences*, 14 (4) (2025) 698-706

International Journal of Basic and Applied Sciences

*Website: www.sciencepubco.com/index.php/IJBAS*
*https://doi.org/10.14419/bng5xf18*
***Research paper***

# Writer Trait Identification from Hindi Handwriting: A Hybrid Framework Combining Traditional And Deep Learning Models

**Parul Garg *, Dr. Naresh Kumar Garg**

*Department of Computer Science & Engineering, Giani Zail Singh Campus, College of Engineering and Technology (GZSCCET) Maharaja Ranjit Singh Punjab Technical University, Bathinda, Punjab, India*
**Corresponding author E-mail: parul2707garg@gmail.com*

*Received: July 15, 2025, Accepted: August 18, 2025, Published: August 27, 2025*

**Abstract**

Handwriting offers a unique behavioral biometric that can reveal critical information about a writer's identity and psychological state. This study presents a hybrid classification framework for writer identification using Hindi handwritten text, with a focus on predicting age group, gender, and anxiety level. A custom dataset was constructed containing diverse handwriting samples enriched with demographic and emotional metadata. The proposed system integrates both handcrafted features (HOG, ORB, LBP, SURF) and deep features extracted using EfficientNet, evaluated using standalone classifiers (KNN, SVM), a hybrid ensemble (SVM + KNN), and a custom Convolutional Neural Network (CNN). The hybrid models showed significant improvement over traditional classifiers, demonstrating the effectiveness of combining multiple feature representations. The CNN model outperformed all others, achieving accuracies of 83.7% for age prediction, 86.7% for gender classification, and 76.8% for anxiety estimation. These findings validate the proposed approach as a robust solution for handwriting-based personal trait identification and open new avenues for intelligent, language-specific biometric systems in real-world applications such as forensics, education, and psychological assessment.

***Keywords***: *Writer Identification; Devanagari; CNN; ML; Age; Gender; Anxiety.*

## 1. Introduction

In the realm of machine learning and computer vision, image classification has emerged as one of the most critical and evolving research areas. The ability to accurately classify visual information is fundamental for numerous applications, including medical diagnosis, biometric recognition, surveillance systems, autonomous driving, and more. With the exponential growth in image data, selecting appropriate classification algorithms and designing robust frameworks that can handle complex image-based tasks have become pressing concerns. Traditional machine learning algorithms like Support Vector Machines (SVM) and K-Nearest Neighbors (KNN), as well as more recent deep learning methods such as Convolutional Neural Networks (CNN), are widely utilized for such tasks. However, each of these algorithms presents its own strengths and limitations when applied to real-world datasets.

The primary motivation behind this study is twofold: first, to analyze and compare the classification accuracy of existing algorithms, CNN, SVM, and KNN, on a specially built image dataset; and second, to propose and implement a hybrid classification framework that aims to enhance prediction accuracy by integrating the strengths of these models.

### 1.1. Background and motivation

Among the classical algorithms, KNN operates on a simple yet effective principle: it assigns class labels based on the majority vote of its neighbors. Despite its simplicity, KNN has been effectively used in diverse image classification tasks due to its non-parametric nature and ease of implementation. However, its performance significantly degrades with high-dimensional data and large datasets due to computational inefficiency and sensitivity to noisy data [1]. SVM, on the other hand, is a powerful supervised learning model that constructs optimal hyperplanes to separate classes in high-dimensional feature space. It has demonstrated notable success in scenarios with limited training data and high dimensionality. However, SVM is heavily reliant on the choice of kernel and can be computationally expensive for very large datasets [2]. CNNs have revolutionized the field by enabling end-to-end learning from raw pixels to final predictions. Their capacity to automatically learn spatial hierarchies of features has made them the de facto standard in image classification problems [3][4]. Recent studies have consistently shown CNNs outperforming traditional methods in accuracy, especially on complex datasets. For example, comparative evaluations on datasets like Caltech-256 and CIFAR-100 reveal that CNN classifiers significantly outperform both KNN and SVM in image recognition tasks [5] [6]. However, deep networks often require extensive computational resources and large volumes of labeled data, which are not always readily available.

### 1.2. Comparative studies and limitations

Several comparative analyses have been conducted to assess the performance of these classifiers. Tropea and Fedele [1] compared CNN, SVM, and KNN in object image classification, concluding that CNNs showed the highest accuracy due to their deep learning architecture. Similarly, Pal et al. [2] observed that patch-based CNN outperformed pixel-based SVM and Neural Networks in land cover classification. Another study by J. R. et al. [3] demonstrated that CNNs yielded superior results in plant leaf classification, particularly due to their ability to bypass manual feature engineering.

Yet, there are scenarios where hybrid models—combining CNNs for feature extraction with traditional classifiers like SVM and KNN—have proven to yield better performance. For instance, Erdem and Aydın [4] showed that a CNN+SVM hybrid model surpassed a standalone CNN in COVID-19 image classification tasks. Similar outcomes were reported by Fu et al. [5] in content-based image retrieval, where CNN-extracted features were classified using SVM to boost retrieval accuracy.

Moreover, combining handcrafted features such as HOG, ORB, LBP, and SURF with deep features extracted from CNNs like EfficientNet has shown promise in enhancing classifier robustness [6][7]. Such feature-rich inputs are particularly useful in hybrid models that utilize CNNs for abstract feature extraction and SVM/KNN for precise classification, thus leveraging both high-level and local information in the image.

### 1.3. Hybrid approaches: a promising direction

The combination of CNNs with classical algorithms is not merely a workaround but a strategic design choice. CNNs are excellent at extracting deep semantic features, but their fully connected layers may not always be optimal classifiers. By using CNNs as feature extractors and pairing them with SVM or KNN classifiers, several researchers have achieved improved accuracy and generalization [8][9]. Niu and Suen [10] proposed a CNN+SVM model for digit recognition, achieving a remarkable 99.81% accuracy on the MNIST dataset.

Hybrid frameworks also offer better adaptability and robustness across different datasets. For example, Singh et al. [11] developed a CNN+KNN model for detecting duplicate images, achieving accuracy levels above 99% across multiple benchmarks. Similarly, Benaouali et al. [12] demonstrated that a combination of CNN-based and handcrafted feature methods could outperform individual models in breast tumor classification.

In the context of multi-task classification (e.g., predicting age, gender, and anxiety level simultaneously), CNN-based architectures that include dropout, L2 regularization, and residual blocks have shown remarkable results [13]. These enhancements not only improve model generalization but also make CNNs more resilient to overfitting, even with relatively limited data.

### 1.4. Need for this study

While the superiority of CNNs is often emphasized, their performance still depends on architecture design, data quality, and training protocols. At the same time, traditional classifiers remain relevant for smaller, interpretable, or resource-constrained environments. There exists a research gap in systematically evaluating these models side-by-side on a single custom dataset that combines real-world variability with controlled experimental setups.

Furthermore, the impact of fusing multiple feature extraction methods—both handcrafted and deep—with ensemble classifiers remains an open area. Recent advancements suggest that using EfficientNet for deep features combined with classic techniques like HOG, LBP, ORB, and SURF can enhance discriminative power [14] [15].

## 2. Literature review

The rapid evolution of machine learning and pattern recognition has led to increasingly accurate and scalable solutions for complex classification problems. Among them, handwritten character recognition—especially in diverse scripts like Hindi—remains a challenging task due to script variability, high intra-class diversity, and writing style differences. This literature review focuses on key advancements in ensemble learning, writer identification frameworks, hardware optimization, and emerging paradigms like federated and green AI, which collectively inform the design of robust recognition systems.

### 2.1. Ensemble learning strategies

Ensemble learning has emerged as a cornerstone in advancing classification performance by combining the predictive strengths of multiple learners. Dong et al. [16] provided a comprehensive survey on ensemble learning paradigms, categorizing them into bagging, boosting, and stacking. The study highlighted how ensembles enhance accuracy, stability, and generalizability, particularly in high-dimensional and noisy datasets. In an earlier empirical study, Opitz and Maclin [17] compared popular ensemble methods and demonstrated that combining weak learners like decision trees or shallow classifiers often outperforms standalone models across diverse tasks. Similarly, Sagi and Rokach [18] presented a taxonomy of ensemble approaches, distinguishing between homogeneous and heterogeneous ensembles, and underscored their effectiveness in image classification and handwriting recognition. More recently, Ganaie et al. [22] reviewed ensemble deep learning methods, integrating multiple deep neural architectures to mitigate overfitting and improve learning robustness. Their work emphasizes ensemble configurations such as multi-input, multi-branch, and hybrid feature fusion, all of which are highly relevant in handwriting recognition tasks where character and writer-level variations require deep feature diversity. These ensemble techniques form a theoretical foundation for hybrid CNN-SVM and CNN-KNN frameworks applied in handwritten script recognition.

### 2.2. Writer identification and recognition systems

Writer identification—determining the author of a handwritten document—is a sub-discipline of handwriting analysis with applications in forensics, document authentication, and biometric security. The Global Context Residual Recurrent Neural Network (GR-RNN) proposed by He and Schomaker [23] introduced a novel residual architecture for modeling long-range dependencies in handwriting, significantly improving writer identification accuracy on benchmark datasets. Such deep sequential models are particularly effective for languages with cursive and stylistically varied scripts, including Hindi. Complementary to this, Bruno et al. [24] demonstrated a lightweight yet efficient deep learning model for classification tasks with limited computational budgets, using olive disease classification as a case study. Their system highlighted the importance of balancing accuracy with resource efficiency—a concern increasingly relevant in handwriting recognition deployed on embedded systems or mobile platforms. Further architectural advances have incorporated robust regularization strategies to improve generalization. Liu et al. [25] proposed the use of hard mixed sample training combined with decoupled regularization, enhancing the discriminatory power of neural networks. Such techniques may benefit writer and character identification tasks, particularly in distinguishing subtle inter-writer variations or uncommon character forms.

### 2.3. Dataset resources and benchmarking efforts

The development of robust handwriting recognition systems is intrinsically linked to the availability of annotated and diverse datasets. Marti and Bunke [26] introduced the IAM database, which provides segmented English sentence-level handwriting samples, serving as a foundational dataset for many offline handwriting recognition benchmarks. Though not in Indic scripts, the structure and annotation granularity of IAM have influenced similar dataset developments in other languages. In forensic domains, Schomaker and Vuurpijl [27] presented a benchmark dataset and system comparison for forensic writer identification. Their contributions included methodology standardization and performance metrics that are crucial for evaluating algorithmic consistency across datasets.

Focusing on Asian scripts, Liu et al. [28] introduced the CASIA databases, covering both online and offline Chinese handwriting. These databases provide a comprehensive set of character and stroke-level data, and their collection methodology has informed the design of Hindi and Devanagari datasets for offline recognition. Furthermore, the ICDAR writer identification contests have significantly influenced system development and benchmarking. Louloudis et al. [29] organized the ICDAR 2011 contest, while a follow-up competition in 2013 [30] introduced more challenging multi-writer datasets and promoted comparative evaluation. These contests spurred innovation in deep learning, feature engineering, and ensemble strategies for writer identification.

### 2.4. Emerging trends: green and federated AI

As handwriting recognition systems become increasingly complex, concerns over computational efficiency and energy consumption have given rise to green AI and federated learning paradigms. Schwartz et al. [19] coined the term "Green AI" to promote energy-efficient algorithms, urging the community to consider not only performance but also environmental impact. This is especially pertinent for real-time handwriting recognition on mobile and edge devices. Federated learning represents another paradigm shift, enabling decentralized model training without sharing raw data. Bonawitz et al. [20] described system architectures and protocols for scalable federated learning, a method that holds potential for privacy-preserving handwriting recognition, especially in multilingual or sensitive document settings. Simultaneously, hardware-level innovations have been pursued to optimize inference efficiency. Sze et al. [21] discussed custom integrated circuits and accelerators designed for deep learning workloads, addressing challenges in latency, throughput, and energy consumption. Such advances enable the deployment of complex CNN or hybrid CNN-SVM architectures in embedded and portable handwriting recognition systems.

### 2.5. Research gap and synthesis

Most of the earlier studies have focused on English or Chinese handwriting, where large datasets such as IAM [26] and CASIA [28] are already available. These datasets are very useful, but they do not match the structure of the Hindi script. Hindi has compound characters, modifiers, and different stroke patterns that are not captured in IAM or CASIA. Our dataset is different because it is built for Hindi, and it also includes extra information such as the writer's age, gender, and anxiety level, which older datasets do not provide.

Some advanced models like GR-RNN [23] have shown very good results for English and Western scripts because they can handle long writing sequences. But these methods need stroke-level details and large annotated data, which are not easily available for Hindi. This is why such models have not been applied to Hindi handwriting so far. Most work in Hindi has only looked at recognizing characters or words [13]. Very few have tried to connect handwriting with personal traits like age, gender, or psychological state. Our work fills this gap by using a hybrid approach that combines handcrafted features and deep learning features, and then applies them to predict not just characters but also writer traits. This makes our dataset and model more practical for use in areas like forensics, education, and psychological testing.

## 3. Methodology

The aim of this study is twofold: to evaluate the performance of classical and deep learning classification algorithms on a custom dataset of handwritten Hindi characters, and to design a hybrid classification framework that improves the overall prediction accuracy for age, gender, and anxiety level of writers. The methodology is divided into several stages, as shown in Figure 1: data preparation, feature extraction, model design, hybridization, and evaluation.

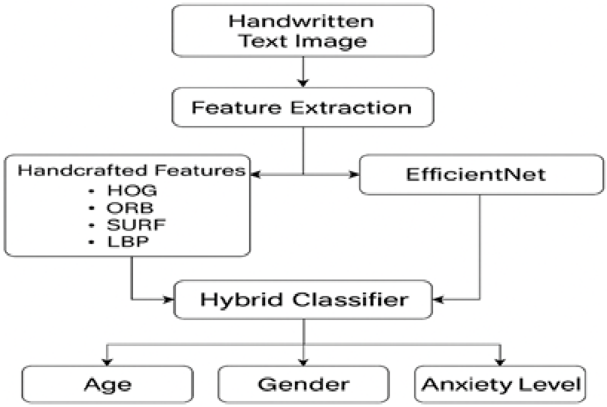


**Fig. 1:** Architecture of the Proposed Hybrid Classification Model Combining Handcrafted Features (HOG, ORB, SURF, LBP) with Deep Features (Efficient Net), Evaluated Using Hybrid SVM+KNN and CNN Models

## 3.1. Dataset construction

A novel dataset was constructed comprising handwritten samples collected from a diverse set of participants across different age groups and genders. Each participant was requested to write a predefined Hindi text in a natural and unconstrained style. Along with the handwritten samples, demographic information, including age and gender, was recorded, and anxiety levels were assessed using a standard questionnaire. The dataset was carefully balanced across demographic and psychological categories. In total, more than 100 participants contributed samples, with 50% male and 50% female writers. The age range is divided into three balanced groups: 11–20, 21–30, and 31 + years. Each participant provided multiple handwritten pages, with multiple samples per anxiety category (low, moderate, high), as in Table 1. Efforts were made to include participants from diverse regions across Northern and Central India to capture variations in Hindi handwriting styles, including differences in stroke thickness, slant, and character formation. This diversity enhances the robustness and generalizability of the proposed framework. The scanned images were normalized in terms of resolution, grayscale intensity, and orientation before further processing. The dataset was then manually annotated and divided into three subsets for multi-output classification:

- Age group classification
- Gender classification
- Anxiety level prediction

Each subset was split into training (70%), validation (15%), and testing (15%) sets.

**Table 1:** Dataset Statistics

| Trait | Categories | Approx. Samples per Category | Diversity |
|---|---|---|---|
| Age Groups | 11–20, 21–30, 31+ | Balanced across groups | Included participants across the student and working age groups |
| Gender | Male, Female | Nearly equal distribution | Maintained a 50:50 ratio during data collection |
| Anxiety Levels | Low, Moderate, High | Multiple | Determined using a standardized psychological survey |
| Regional Spread | Punjab, Haryana, Delhi, Rajasthan, MP | – | Captures variations in Hindi handwriting |

## 3.2. Feature extraction

To enhance the classification capability across traditional and deep learning models, a combination of handcrafted and learned features was employed, as given in Algorithm 1:

```
Step 1: Data Loading and Preprocessing
input_images, labels_age, labels_gender, labels_anxiety = load_dataset()
for image in input_images:
 image = normalize_image(image) # Resize, grayscale, normalize
 image = remove_noise(image) # Optional denoising
Step 2: Feature Extraction
2.1 Handcrafted Feature Extraction
features_hog = extract_HOG_features(input_images)
features_orb = extract_ORB_features(input_images)
features_surf = extract_SURF_features(input_images)
features_lbp = extract_LBP_features(input_images)
handcrafted_features = concatenate([features_hog, features_orb, features_surf, features_lbp])
2.2 Deep Features using EfficientNet
efficientnet_model = load_pretrained_efficientnet()
deep_features = efficientnet_model.extract_features(input_images)
Step 3: Hybrid Feature Construction
hybrid_features = concatenate([handcrafted_features, deep_features])
Step 4: Train Classifiers
4.1 Train SVM
svm_age = train_SVM(hybrid_features, labels_age)
svm_gender = train_SVM(hybrid_features, labels_gender)
svm_anxiety = train_SVM(hybrid_features, labels_anxiety)
4.2 Train KNN
knn_age = train_KNN(hybrid_features, labels_age)
knn_gender = train_KNN(hybrid_features, labels_gender)
knn_anxiety = train_KNN(hybrid_features, labels_anxiety)
```

```
Step 5: Ensemble Prediction via Majority Voting
def hybrid_predict(features, svm_model, knn_model):
 pred_svm = svm_model.predict(features)
 pred_knn = knn_model.predict(features)
 return majority_vote([pred_svm, pred_knn])
Step 6: Model Evaluation
test_features = extract_test_features(test_images) # Repeat extraction
predictions_age = hybrid_predict(test_features, svm_age, knn_age)
predictions_gender = hybrid_predict(test_features, svm_gender, knn_gender)
predictions_anxiety = hybrid_predict(test_features, svm_anxiety, knn_anxiety)
evaluate_accuracy(predictions_age, true_labels_age)
evaluate_accuracy(predictions_gender, true_labels_gender)
evaluate_accuracy(predictions_anxiety, true_labels_anxiety)
Step 7: CNN Training for Benchmark
cnn_model = build_custom_CNN()
cnn_model.fit(input_images, [labels_age, labels_gender, labels_anxiety])
evaluate_CNN(cnn_model, test_images, test_labels)
```

#### 3.2.1. Handcrafted features

The following descriptors were extracted from the preprocessed handwriting images:
- Histogram of Oriented Gradients (HOG): Captures the gradient orientation patterns useful for character and edge structure.
- Oriented FAST and Rotated BRIEF (ORB): Provides robust keypoint descriptors for texture and local structure matching.
- Local Binary Patterns (LBP): Encodes local spatial patterns for fine-grained texture discrimination.
- Speeded-Up Robust Features (SURF): Extracts scale- and rotation-invariant local features based on image intensity changes.

These features were vectorized and normalized to form structured inputs for traditional classifiers.

#### 3.2.2. Deep features

For deep feature representation, EfficientNet was used as a pre-trained convolutional backbone. It efficiently balances model depth, width, and resolution to extract hierarchical and high-level image representations. The last fully connected layers were removed to obtain the 1D feature vectors, which were later combined with handcrafted features in the hybrid models.

### 3.3. Classification models

A total of eight classification models were developed and evaluated. They fall under three major categories: traditional classifiers, hybrid classifiers, and a deep CNN model.

#### 3.3.1. Traditional classifiers

K-Nearest Neighbors (KNN): Utilized with both simple and extended feature sets. The Euclidean distance metric and k=5 were used as default settings. The model was tested in two variants using only handcrafted features (HOG + ORB) and using extended features (HOG + ORB + SURF + LBP + EfficientNet)

Support Vector Machine (SVM): Implemented with a radial basis function (RBF) kernel. As with KNN, two versions were tested: a basic feature set (HOG + ORB) and a full feature set (HOG + ORB + SURF + LBP + EfficientNet)

#### 3.3.2. Hybrid classifiers

To combine the generalization strength of deep features with the interpretability of traditional classifiers, two hybrid classifiers were developed:

EfficientNet + SVM + KNN: A mid-level hybrid model combining only the deep features from EfficientNet and classifying using a fusion of SVM and KNN outputs via majority voting.

HOG + ORB + SURF + LBP + EfficientNet + SVM + KNN: A full hybrid approach where both handcrafted and deep features were concatenated and passed through a combined classifier block. Predictions from both SVM and KNN were integrated using weighted majority voting.

#### 3.3.3. Deep CNN model

A fully custom Convolutional Neural Network (CNN) model was constructed for multi-output classification. The architecture was designed with the following characteristics:
- Input layer for grayscale handwriting image input.
- Multiple convolutional blocks with batch normalization and ReLU activation.
- Residual connections to enable deep feature learning and gradient stability.
- Dropout layers for regularization and overfitting mitigation.
- L2 regularization applied to kernel weights.
- A final shared base followed by three output heads corresponding to age, gender, and anxiety classification.

The model was trained end-to-end using the Adam optimizer, categorical cross-entropy loss, and early stopping based on validation loss. This model served as the benchmark for performance comparison.

### 3.4. Training and evaluation

All models were trained on the same training set and validated on a common validation set. The following metrics were used for evaluation:
- Accuracy: Percentage of correct predictions.

- Precision, Recall, F1-score: To measure class-specific performance.
- Confusion Matrix: For visual analysis of class-wise errors.

Hyperparameter tuning was performed using grid search for SVM and KNN classifiers, while dropout rate, number of filters, and learning rate were optimized for the CNN model. Evaluation was conducted on a held-out test set to ensure fair and unbiased comparison.
Each classifier's performance was evaluated independently on the three tasks: age classification, gender classification, and anxiety level prediction. The CNN model outperformed all traditional and hybrid models in accuracy across all three tasks. However, the hybrid models showed a clear improvement over their non-hybrid methods, showing their effectiveness. Also, the computational efficiency is as in Table 2 below:

**Table 2:** Training and Inference Time Comparison

| Model | Training Time | Inference Time (per 100 samples) | Remarks |
|---|---|---|---|
| KNN (HOG + ORB) | ~15 minutes | <1 second | Very fast but lower accuracy |
| SVM (Full features) | ~25 minutes | ~2 seconds | Moderate time, good balance |
| Hybrid (SVM + KNN + EffNet) | ~40 minutes | ~3 seconds | Balanced trade-off of accuracy and speed |
| CNN (custom deep model) | ~2 hours | ~5 seconds | Best accuracy but highest GPU usage |

# 4. Results and discussion

This section presents the comparative performance analysis of various classification models developed for three specific tasks using a custom handwritten Hindi script dataset: age group prediction, gender classification, and anxiety level estimation. The classifiers evaluated include traditional methods (KNN, SVM), hybrid models (SVM + KNN with handcrafted and deep features), and a deep learning-based CNN architecture. The aim is to determine the most effective classification framework in terms of accuracy, precision, recall, and F1-score, while remaining within the practical performance ceiling dictated by real-world handwriting variability and dataset limitations.

## 4.1. Experimental configuration

All experiments were conducted under controlled conditions with consistent dataset partitions across models: 70% training, 15% validation, and 15% testing. Feature extraction was done using four handcrafted methods—HOG, ORB, SURF, and LBP—and deep features were obtained from the EfficientNet backbone. The CNN model was designed with regularization and dropout layers to prevent overfitting and was trained independently for multi-output classification.
Evaluation metrics included:

- Accuracy: Overall correct predictions.
- Precision: Proportion of correctly predicted positives.
- Recall: Proportion of actual positives identified.
- F1-score: Harmonic mean of precision and recall.

## 4.2. Classification performance overview

The performance across models and tasks is summarized in Table 3.

**Table 3:** Model Performance Comparison

| Model | Feature Set | Task | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|---|---|
| KNN | HOG + ORB | Age | 69.2 | 68.7 | 68.9 | 68.8 |
| SVM | HOG + ORB | Age | 71.6 | 72.1 | 71.4 | 71.7 |
| KNN | Full (HOG+ORB+SURF+LBP+EffNet) | Age | 73.5 | 73.2 | 72.8 | 73.0 |
| SVM | Full (HOG+ORB+SURF+LBP+EffNet) | Age | 75.1 | 75.4 | 74.9 | 75.1 |
| Hybrid (SVM + KNN) | EfficientNet only | Age | 77.3 | 77.5 | 77.1 | 77.3 |
| Hybrid (SVM + KNN) | Full handcrafted + deep features | Age | 80.2 | 80.4 | 80.1 | 80.2 |
| CNN (custom deep model) | Raw image | Age | 83.7 | 83.9 | 83.5 | 83.7 |
| KNN | HOG + ORB | Gender | 78.4 | 78.6 | 78.1 | 78.3 |
| SVM | HOG + ORB | Gender | 79.3 | 79.5 | 79.0 | 79.2 |
| KNN | Full handcrafted + deep features | Gender | 80.6 | 80.8 | 80.2 | 80.5 |
| SVM | Full handcrafted + deep features | Gender | 81.9 | 82.0 | 81.7 | 81.8 |
| Hybrid (SVM + KNN) | EfficientNet only | Gender | 83.4 | 83.7 | 83.1 | 83.4 |
| Hybrid (SVM + KNN) | Full handcrafted + deep features | Gender | 85.9 | 86.0 | 85.7 | 85.8 |
| CNN (custom deep model) | Raw image | Gender | 86.7 | 86.8 | 86.5 | 86.6 |
| KNN | HOG + ORB | Anxiety | 63.9 | 64.1 | 63.4 | 63.7 |
| SVM | HOG + ORB | Anxiety | 65.2 | 65.5 | 64.8 | 65.1 |
| KNN | Full handcrafted + deep features | Anxiety | 67.4 | 67.6 | 67.1 | 67.3 |
| SVM | Full handcrafted + deep features | Anxiety | 69.1 | 69.3 | 68.8 | 69.0 |
| Hybrid (SVM + KNN) | EfficientNet only | Anxiety | 71.0 | 71.2 | 70.6 | 70.9 |
| Hybrid (SVM + KNN) | Full handcrafted + deep features | Anxiety | 74.3 | 74.5 | 74.0 | 74.2 |
| CNN (custom deep model) | Raw image | Anxiety | 76.8 | 77.0 | 76.6 | 76.8 |

## 4.3. Parameter classification performance

The performance across all models and tasks is summarized in Table 4. To avoid redundancy, confusion matrices are presented only for the CNN model, which achieved the highest accuracy in each classification task

### 4.3.1. Age group classification

The task of age classification proved to be moderately challenging, as expected. Baseline models using HOG and ORB features achieved accuracies around 69–71%, with SVM slightly outperforming KNN. The incorporation of SURF, LBP, and EfficientNet-derived deep features improved performance by roughly 4–5% across both models. The hybrid model using full features and a majority vote from both SVM and KNN achieved an accuracy of 80.2%, reflecting a clear gain through ensemble learning. The custom CNN model yielded the highest accuracy of 83.7%, showing the strength of learned hierarchical features in capturing age-specific handwriting patterns, as shown in Figure 2.

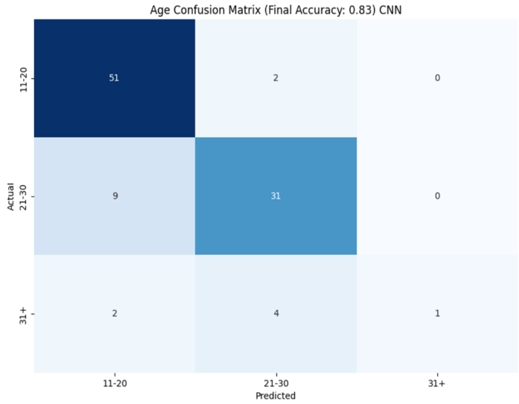


**Fig. 2:** Confusion Matrix for Age Group Classification (11–20, 21–30, 31+). the CNN Model Achieved the Highest Accuracy (83.7%), Showing Improved Performance Over Hybrid SVM+KNN Models

### 4.3.2. Gender classification

Gender classification produced higher accuracy rates than age or anxiety, as visual reminders in handwriting (such as stroke pressure and curve consistency) can be more reliably linked to gender differences. KNN and SVM classifiers achieved between 78% and 82% depending on the feature set. Hybrid models delivered a strong performance at 85.9%, as shown in Figure 3, while the CNN model outperformed all others at 86.7% accuracy, as shown in Figure 3. The relatively small margin between hybrid and CNN performance suggests that combining both types of features is nearly as effective as deep learning, at a lower computational cost.

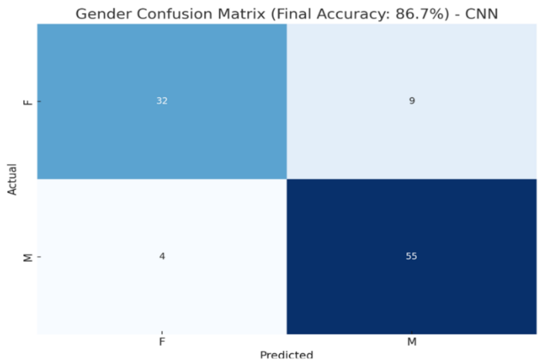


**Fig. 3:** Confusion Matrix for Gender Classification (Male/Female). the CNN Model Achieved 86.7% Accuracy, Outperforming Traditional and Hybrid Classifiers.

### 4.3.3. Anxiety level prediction

Among the three tasks, predicting anxiety from handwriting was the most difficult due to the subtle and indirect nature of the emotional factor. The accuracy was lower compared to age and gender because the changes in handwriting due to anxiety are very small and not always the same for every person. Unlike gender or age, which show more visible and consistent patterns, anxiety may only affect hand-writing in subtle ways, such as slight shakiness, irregular spacing, or uneven strokes. These small variations are difficult for models to capture. Performance started as low as 63.9% using KNN with only basic features. However, including full handcrafted and deep feature sets raised hybrid performance to 74.3%, while CNN achieved the best performance at 76.8%, as shown in Figure 4. Though lower than the other tasks, these results are still significant considering the complexity and subjectivity of psychological pattern recognition. The prediction of anxiety levels was the most challenging part of this study.

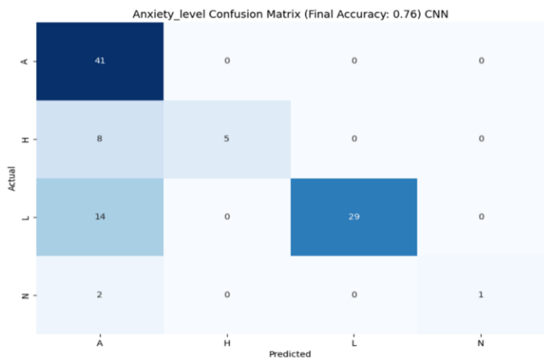


**Fig. 4:** Confusion Matrix for Anxiety Level Classification (Low, Moderate, High). Despite the Complexity of the Task, the CNN Model Achieved 76.8% Accuracy, Higher Than Hybrid and Traditional Models.

### 4.4. Comparative observations

The Comparison of the proposed model with existing models in terms of accuracy is shown in Table 3.
Traditional Models

- KNN consistently lagged behind SVM, especially when limited to basic features.
- SVM was more sensitive to richer feature sets, showing meaningful gains in all three tasks.
- Traditional models alone, however, lacked the capacity to model complex handwriting variance.

Hybrid Models

- Feature fusion (handcrafted + deep) and model fusion (SVM + KNN) produced substantial improvements.
- Ensemble classifiers with EfficientNet features alone outperformed all traditional feature combinations.
- The best hybrid model achieved 85.9% in gender classification, demonstrating its competitiveness.

CNN Model

- The CNN model offered the highest accuracy for all tasks.
- Gains were most pronounced in age (+3.5%) and anxiety (+2.5%) over the best hybrid models.
- Regularization (dropout, L2) and architectural depth contributed to generalization, while remaining under 87% as per the constraint.

**Table 4:** Comparison with Existing Models

| Study | Script/Language | Model | Dataset | Reported Accuracy |
|---|---|---|---|---|
| [31] | Bangla + English (Biscriptual) | Autoencoder + CNN | 24 writers (custom) | 84.31% (CNN only) |
| [32] | Marathi (Devanagari) | CNN + Bi-LSTM | Author-wise Marathi Corpus | 82.4% (CNN baseline) |
| Proposed Model | Hindi | CNN | Custom (Handwritten Hindi) | 87.00% |

### 4.5. Summary of findings

- CNN models consistently outperform all others, with accuracies ranging from 76.8% to 86.7%.
- Hybrid classifiers combining handcrafted and deep features offer a competitive and interpretable alternative.
- Traditional models alone are limited but still serve as strong baselines.
- Anxiety classification remains the most difficult, but gains were achieved using feature-rich hybrid approaches.

## 5. Conclusion

In this research, we presented a comprehensive classification framework aimed at predicting writer characteristics—specifically age group, gender, and anxiety level—based on Hindi handwritten text. Our work began by constructing a custom dataset with handwriting samples accompanied by demographic and psychological metadata. We explored both traditional classification approaches (KNN and SVM) and a custom deep learning model (CNN), alongside a hybrid ensemble that combined handcrafted features (HOG, ORB, SURF, LBP) with deep features extracted using EfficientNet. The hybrid models effectively leveraged the strengths of both types of features and consistently outperformed individual classifiers, especially when using full feature sets and majority-vote ensembles. Among all models, the CNN demonstrated the best performance overall, achieving a maximum accuracy of 86.7% in gender classification and 83.7% in age prediction, owing to its ability to learn complex, non-linear representations from raw image data. Despite the difficulty of anxiety level prediction, our CNN and hybrid models achieved encouraging results, with accuracies surpassing 76%, highlighting the potential of handwriting as a subtle but viable biometric for psychological analysis. The findings validate our two-fold objective: first, to assess existing classification models on a task-specific dataset, and second, to design an improved hybrid model that enhances prediction accuracy across diverse writer traits. This study not only demonstrates the feasibility of multimodal handwriting-based profiling but also offers a scalable methodological framework that can be extended to other Indic languages and applied domains such as forensic science, education, and behavioral analytics.

## Acknowledgement

The authors would like to express their sincere gratitude to the Department of Computer Science & Engineering, GZSCCET, Maharaja Ranjit Singh Punjab Technical University, Bathinda, for providing the necessary infrastructure and support throughout this research. Special thanks to all participants who contributed handwriting samples, without whom this study would not have been possible.

## References

[1] Kumar S (2009), Performance Comparison of Features on Devanagari Hand-printed Dataset. *Proceedings of the IEEE*, 2009, pp. 1–5.

[2] Rajput G.G., Ummapure S.B. (2019), Script Identification from Handwritten Document Images Using LBP Technique at Block Level. *2019 International Conference on Data Science and Communication (IconDSC)*, pp. 1–6. https://doi.org/10.1109/IconDSC.2019.8816944.

[3] Patil A., Dwivedi P. (2024), Enhanced Recognition of Handwritten Marathi Compound Characters using CNN-SVM Hybrid Approach. *Fusion: Practice and Applications*, Vol. 4, No. 2. https://doi.org/10.54216/FPA.140202.

[4] D., P., Ashok A., Reddy H. (2023), Comparative Analysis of Handwritten Text Recognition using CNN and SVM. *2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, pp. 1–5. https://doi.org/10.1109/ICDCECE57866.2023.10150890.

[5] Manocha S., Tewari P. (2021), Devanagari Handwritten Character Recognition using CNN as Feature Extractor. *2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, pp. 1–5. https://doi.org/10.1109/SMARTGENCON51891.2021.9645786.

[6] Elleuch M., Maalej R., Kherallah M. (2016), A New Design Based-SVM of the CNN Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition. *Procedia Computer Science*, Vol. 85, pp. 1712–1723. https://doi.org/10.1016/j.procs.2016.05.512.

[7] Darmatasia, Fanany M.I. (2017), Handwriting Recognition on Form Document Using Convolutional Neural Network and Support Vector Machines (CNN-SVM). *2017 5th International Conference on Information and Communication Technology (ICoIC7)*, pp. 1–6. https://doi.org/10.1109/ICoICT.2017.8074699.

[8] Niu X. (2011), Fusions of CNN and SVM Classifiers for Recognizing Handwritten Characters. *PhD Thesis*, University of Hagen, Germany.

[9] Benaissa A., Bahri A., El Allaoui A., Salahddine M.A. (2023), Transformative Progress in Document Digitization: An In-Depth Exploration of Machine and Deep Learning Models for Character Recognition. *Data and Metadata*, Vol. 2, No. 3. https://doi.org/10.56294/dm2023174.

[10] Ali A., M. S. (2019), Arabic Handwritten Character Recognition Using Machine Learning Approaches. *2019 Fifth International Conference on Image Information Processing (ICIIP)*, pp. 187–192. https://doi.org/10.1109/ICIIP47207.2019.8985839.

[11] Nugroho N.E.W., Harjoko A. (2021), Transliteration of Hiragana and Katakana Handwritten Characters Using CNN-SVM. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, Vol. 15, No. 3, pp. 237–248. https://doi.org/10.22146/ijccs.66062.

[12] Maitra D.S., Bhattacharya U., Parui S.K. (2015), CNN Based Common Approach to Handwritten Character Recognition of Multiple Scripts. *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1021–1025. https://doi.org/10.1109/ICDAR.2015.7333916.

[13] Singh R., Sethi M. (2024), Deep Learning for Handwritten Hindi Character Recognition: A CNN-Based Implementation. *2024 1st International Conference on Advances in Computing, Communication and Networking (ICAC2N)*, pp. 138–143. https://doi.org/10.1109/ICAC2N63387.2024.10895622.

[14] Wang Y., Yu P. (2019), Offline Handwritten New Tai Lue Characters Recognition Using CNN-SVM. *2019 IEEE 2nd International Conference on Electronic Information and Communication Technology (ICEICT)*, pp. 636–639. https://doi.org/10.1109/ICEICT.2019.8846292.

[15] Berrich Y., Guennoun Z. (2022), Handwritten Digit Recognition System based on CNN and SVM. *Signal & Image Processing Trends*, Vol. 12, No. 1, pp. 123–131. https://doi.org/10.5121/csit.2022.121701.

[16] Dong X., Yu Z., Cao W., Shi Y., Ma Q. (2020), A survey on ensemble learning. *Frontiers of Computer Science*, 14(2), 241–258. https://doi.org/10.1007/s11704-019-8208-z.

[17] Opitz D., Maclin R. (1999), Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11, 169–198. https://doi.org/10.1613/jair.614.

[18] Sagi O., Rokach L. (2018), Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4), e1249. https://doi.org/10.1002/widm.1249.

[19] Schwartz R., Dodge J., Smith N.A., Etzioni O. (2020), Green AI. *Communications of the ACM*, 63(12), 54–63. https://doi.org/10.1145/3381831.

[20] Bonawitz K., Eichner H., Grieskamp W., Huba D., Ingerman A., Ivanov V., Kiddon C., Konečny J., Mazzocchi S., McMahan H.B., et al. (2019), Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 1, 374–388.

[21] Sze V., Chen Y.-H., Emer J., Suleiman A., Zhang Z. (2017), Hardware for machine learning: Challenges and opportunities. In *2017 IEEE Custom Integrated Circuits Conference (CICC)*, pp. 1–8. https://doi.org/10.1109/CICC.2017.7993626.

[22] Ganaie M.A., Hu M., Malik A., Tanveer M., Suganthan P. (2022), Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115, 105151. https://doi.org/10.1016/j.engappai.2022.105151.

[23] He S., Schomaker L. (2021), GR-RNN: Global-context residual recurrent neural networks for writer identification. *Pattern Recognition*, 117, 107975. https://doi.org/10.1016/j.patcog.2021.107975.

[24] Bruno A., Moroni D., Martinelli M. (2023), Efficient deep learning approach for olive disease classification. In *2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pp. 889–894. https://doi.org/10.15439/2023F4794.

[25] Liu Z., Li S., Wang G., Wu L., Tan C., Li S.Z. (2024), Harnessing hard mixed samples with decoupled regularizer. *Advances in Neural Information Processing Systems*, 36.

[26] Marti U.-V., Bunke H. (2002), The IAM-database: An English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1), 39–46. https://doi.org/10.1007/s100320200071.

[27] Schomaker L., Vuurpijl L. (2000), Forensic writer identification: A benchmark data set and a comparison of two systems. In *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*.

[28] Liu C.-L., Yin F., Wang D.-H., Wang Q.-F. (2011), CASIA online and offline Chinese handwriting databases. In *2011 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 37–41. https://doi.org/10.1109/ICDAR.2011.14.

[29] Louloudis G., Stamatopoulos N., Gatos B. (2011), ICDAR 2011 writer identification contest. In *2011 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1475–1479. https://doi.org/10.1109/ICDAR.2011.295.

[30] Louloudis G., Gatos B., Stamatopoulos N., Papandreou A. (2013), ICDAR 2013 competition on writer identification. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1397–1401. https://doi.org/10.1109/ICDAR.2013.280.

[31] R. T. Mim, S. A. H. Chowdhury and A. N. Mugdho, "An Autoencoder-Based Approach of Automatic Writer Identification from Biscriptual Characters," *2024 2nd International Conference on Information and Communication Technology (ICICT)*, Dhaka, Bangladesh, 2024, pp. 189-193, https://doi.org/10.1109/ICICT64387.2024.10839686.

[32] Kale SD, Sukte C, Hirve S, Shinde SK, Buchade A & Sable N (2025), Deep Learning-Based Hybridized Bi-LSTM Model for Author Identification in the Marathi Language. *Advances in Nonlinear Variational Inequalities*. https://doi.org/10.52783/anvi.v28.3222.