



Smart Prescription Reader: Enhancing Accuracy in Medical Prescriptions

Ragil Yulianto

Faculty of Computer Science, Universitas Nusa Mandiri, Indonesia

E-Mail: 14240007@nusamandiri.ac.id

Received Jan 25th 2025; Revised Mar 22th 2025; Accepted Apr 12th 2025; Available Online Jun 19th 2025, Published Jun 22th 2025

Corresponding Author: Ragil Yulianto

Copyright ©2025 by Authors, Published by Institut Riset dan Publikasi Indonesia (IRPI)

Abstract

Reading a doctor's handwritten prescription is a challenge faced by most patients and some pharmacists, which in some cases can lead to negative consequences due to misinterpretation of the prescription. The "Doctor's Handwritten Prescription BD Dataset" on Kaggle contains segmented images of handwritten prescription words from BD (Bangladesh) doctors. This dataset, intended for machine learning applications, includes 4,680 individual words segmented from prescription images. This study introduces a Handwriting Recognition System using Convolutional Neural Network (CNN) developed to identify text in prescription images written by doctors and convert the cursive handwriting into readable text. Two models were evaluated in this study: CNN and MobileNet. Based on the experiments, MobileNet showed better results compared to CNN alone. From the dataset of 4,680 words, 3,120 were used for training, 780 for testing, and 780 for validation. The study achieved a training accuracy of 97%, a testing accuracy of 88%, and a validation accuracy of 83%. The developed model was successfully implemented in a web application.

Keyword: Convolutional Neural Network, Machine Learning, MobileNet, Prescription Images

1. INTRODUCTION

Medication errors are a significant public health issue, with an estimated 44,000 to 98,000 emergency hospital patients dying each year due to such errors, making them one of the leading causes of death in the United States [1]. Some studies even suggest higher figures, potentially making medical errors the third leading cause of death in the country [2], [3]. The healthcare system faces significant challenges in patient safety and quality improvement. Despite extensive efforts, sustainable progress in reducing preventable harm has been elusive [4]. Systemic weaknesses, including lack of standardization, inadequate staff wellbeing, and organizational culture issues, contribute to persistent safety concerns [5], [6].

These systemic weaknesses manifest in various types of medication errors, compromising patient safety. A report from King's College Hospital NHS Foundation Trust, London, highlighted that, out of 12,006 reported medication incidents, 1,568 involved wrong-dose errors, with 44.8% due to prescribing errors, 14.2% due to dispensing errors, and 41% due to administration errors [7]. The findings further noted that overdoses accounted for 59% of wrong-dose cases, emphasizing the critical nature of accurate prescription handling [7]. However, this report is merely a snapshot and cannot be used to generalize facts, as the data only covers one hospital and may not reflect the situation elsewhere. In the context of the King's College Hospital NHS Foundation Trust report, most errors occurred due to systemic failures rather than individual negligence [8], [9]. Individual negligence, such as errors in writing prescriptions or misreading dosages, can indeed occur, but they are often influenced by systemic factors such as high workload, lack of training, or non-standard procedures. These human factors contribute to individual negligence, which can lead to medication errors. A systems-based approach, including pharmacist involvement in prescription review, can significantly reduce error rates [10].

In addition to systemic weaknesses, Prescription errors often stem from human factors such as fatigue, high workload, and inadequate system support. Handwritten prescriptions often present challenges in legibility and completeness, leading to potential medication errors and the need for clarification calls to prescribers. Illegible handwriting, lack of pharmacological knowledge, distractions, performance deficits, and failure to follow protocols further exacerbate the risk [1]. To address this issue, interventions such as improved education and training, automated systems, and periodic audits are recommended [11], [12].

Technological advancements in handwritten prescription recognition offer a promising avenue to address these challenges. Automated tools using Optical Character Recognition (OCR) and artificial intelligence (AI) have been shown to streamline healthcare processes by converting handwritten text into readable and verifiable digital information [11]. These tools can extract data from medical forms, synthesize research evidence, and assist in clinical documentation [12], [13], [14], [15], [16]. However, many existing solutions face limitations. Traditional OCR methods often struggle with the variability and complexity of handwritten medical scripts. Additionally, most approaches rely on structured, predefined forms that may not be adaptable to the unstructured and diverse nature of real-world handwritten prescriptions [17].

To bridge these gaps, this study proposes the development of a robust, AI-based handwritten prescription reader. Leveraging CNNs for image classification and integrated with a user-friendly web application built using Flask, this project aims to improve the accuracy of prescription interpretation. By automating the reading process, this system seeks to enhance medical accuracy, minimize human error, and contribute to overall patient safety.

The objectives of this study are to develop, train, and evaluate a CNN-based model that can accurately read handwritten prescriptions and to create a web interface that facilitates seamless user interaction with the AI. This approach not only addresses the technological shortcomings of traditional OCR systems but also provides a scalable solution tailored for healthcare applications. With this approach, it is hoped that a more reliable and efficient system for reading handwritten prescriptions can be created, thereby making a significant contribution to improving the quality of healthcare services.

2. MATERIALS AND METHOD

2.1 Data Pipeline

We used the publicly available "Doctor's Handwritten Prescription BD" dataset from Kaggle for this paper [35]. We registered on the website and downloaded images of handwritten lines along with their annotations, which were available in CSV format. The total image data consists of 4680 images, with 60% used for training, 20% for validation, and 20% for testing, ensuring stratified data to maintain an equal amount of data in each class. The training directory contains 60% of the total data, the testing directory contains 20%, and the validation directory also contains 20%, each including Excel and CSV files indexing word names. We loaded and examined the dataset by reading the CSV file containing labels for the images and ensuring the necessary columns were present in the CSV file.

The images were processed by loading them from the extracted folder, resizing them to 64x64 pixels, converting them to arrays, and normalizing the pixel values. This was done for several important reasons in data processing for machine learning, including ensuring consistent image sizes, as machine learning models, especially neural networks, require input with consistent dimensions. By resizing all images to 64x64 pixels, we ensure that each image has the same dimensions, allowing the model to process them efficiently. Secondly, reducing the image size to 64x64 pixels decreases the total number of pixels the model needs to process, which helps reduce computational complexity and speeds up the model training process without losing important information from the images. Thirdly, converting images to arrays transforms them from a visual format to a numerical format that the model can understand, with each pixel represented by a numerical value indicating color intensity. Finally, normalizing the pixel values by dividing them by 255 to range between 0 and 1 helps improve model performance, as neural networks work better with input values that have a uniform and controlled range. Normalization also aids in faster convergence during training.

Labels were converted to numerical form using LabelEncoder and then transformed into a format usable by the model (one-hot encoding). This was done because machine learning models, especially neural networks, cannot work directly with categorical data (such as drug names in text form). They require data in numerical form to perform mathematical calculations. If categorical labels were directly converted to numbers (e.g., "Paracetamol" to 1, "Ibuprofen" to 2), the model might assume an ordinal relationship between the labels, which is incorrect. One-hot encoding avoids this issue by converting each category into a separate binary vector, treating all categories as equal. One-hot encoding produces a binary representation where each class is represented by a vector with a value of 1 at the index corresponding to that class and 0 at other indices. This is very useful for classification tasks as it allows the model to predict probabilities for each class independently. Additionally, many loss functions used in classification, such as categorical_crossentropy, require labels in one-hot encoded form to correctly calculate prediction errors.

2.2 Model Design

CNN are deep learning algorithms used for image analysis. A neural network qualifies as a CNN even with just one convolutional layer. CNNs are effective in extracting deep semantic features from images and are widely used in computer vision for object classification and identification. The convolution operation makes CNNs robust to local variations and image modifications, allowing them to recognize objects regardless of position, size, or orientation. CNNs play a significant role in image analysis, especially in the medical field where data sensitivity is high. Researchers use CNNs for feature extraction, transfer learning,

and abnormality detection. Figure 1 shows a CNN's basic structure, including convolution, pooling, and dense layers. Every CNN has these essential components with some transformations or adaptations.

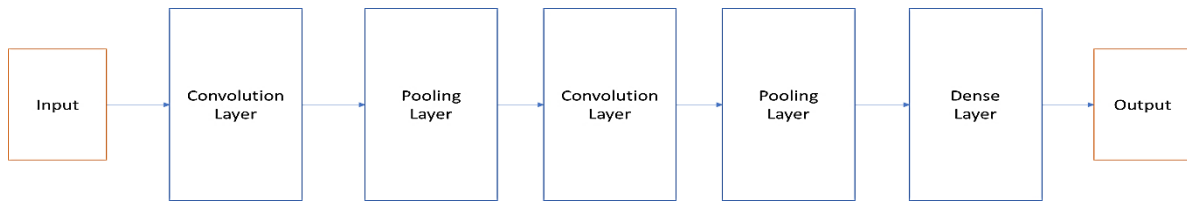


Figure 1. The basic architecture of CNN

The convolutional layer is a key component in CNNs used for processing image data. This layer applies a kernel (filter) to the input image to extract important features. Features in this context refer to specific patterns or characteristics in the image, such as edges, textures, or shapes, which aid in object recognition or image classification. The filter size can vary depending on the architecture. Kernels can be applied both pointwise and depthwise. Pointwise convolution uses a 1x1 filter applied to each point in the image, allowing each input channel to interact with each output channel. Depthwise convolution applies a filter to each input channel separately, which are then combined to produce the final feature map. After applying the kernel to the image, a feature map is obtained that can be further processed.

The feature map size is larger than the input image, so another pooling layer is added to the CNN to reduce the feature map size. Pooling layers function to reduce the dimensions of the feature map, which helps decrease overall computational time and cost. The most popular pooling technique is called max pooling. As the name implies, the max pooling procedure selects the maximum value from a patch of the feature map.

Dense layers have neurons that are densely connected, meaning each neuron in this layer is connected to every neuron in the previous layer. This is also known as a fully connected layer. Its function is to receive input (learned features) from the previous layers and produce the output. The dense layer is crucial in the final stage of the neural network as it completes the data classification task, transforming the learned features into the final predictions.

A CNN can solve problems by either building a network from scratch or retraining an already-developed model for new types of problems. This study uses a basic CNN model and a model with MobileNet. The basic CNN model provides a strong foundation for feature extraction, while MobileNet, being a lightweight model, is designed to work efficiently on devices with limited resources. The combination of these two models allows for accurate and efficient image classification under various conditions and on different devices.

This study focuses on solving image classification problems and categorizing images into specific classes. The procedure adopted for this purpose uses well-developed and well-trained models on diverse image datasets for classification. The methodology involves: (i) collecting and preparing the image dataset; (ii) normalizing and processing the images; (iii) using basic CNN and MobileNet models to classify the images; and (iv) evaluating the results from both models.

2.2.1 Basic CNN

The CNN model used consists of several convolutional layers followed by dense and dropout layers. We use the Rectified Linear Unit (ReLU) activation function after each convolutional layer to introduce non-linearity. ReLU is a commonly used activation function in neural networks that converts negative values to zero and keeps positive values unchanged. This helps the model learn more complex patterns and speeds up the training process. The CNN model built includes convolutional, pooling, flatten, dense, and dropout layers. This model is implemented using the Sequential Model, which is a linear stack of neural network layers. The Sequential Model is created by adding an input layer that specifies the input shape of 64x64 pixels with 3 color channels (RGB).

The first convolutional layer has 32 filters with a kernel size of 3x3 and uses the ReLU activation function to introduce non-linearity. This is followed by the first pooling layer that performs max pooling with a pool size of 2x2 to reduce the spatial dimensions of the feature maps. The second convolutional layer has 64 filters with a kernel size of 3x3 and also uses the ReLU activation function, followed by the second pooling layer with a pool size of 2x2. The third convolutional layer has 128 filters with a kernel size of 3x3 and uses the ReLU activation function, followed by the third pooling layer with a pool size of 2x2.

The flatten layer converts the output from the last convolutional layer into a one-dimensional vector, which is then fed into the first dense layer with 512 units and the ReLU activation function. To prevent overfitting, a dropout layer with a rate of 0.5 is used after the first dense layer. Dropout is a regularization technique used to reduce overfitting in neural networks. During training, dropout randomly "disables" a number of units in the neural network layer with a certain probability (in this case, 0.5), thus preventing the

network from becoming too reliant on specific units and improving the model's generalization ability. Finally, the second dense layer has 78 units with the softmax activation function, which produces probabilities for each of the 78 classes. The model is trained using preprocessed data split into training and testing sets. After training, the model is saved and used to predict labels from test images. The learning rate used is consistently set at 0.001 throughout the entire training process.

2.2.2 MobiliNet

MobileNet was developed by Howard et al. [18]. As the name suggests, MobileNet was created to offer a computationally light and useful model that can be used on mobile devices. This model improves accuracy while reducing computational latency. MobileNet employs depth-wise separable convolution, where each channel of the input image is first subjected to a depth-wise filter before being combined by a pointwise filter. MobileNet also uses width and resolution multipliers, two hyperparameters that help MobileNet be smaller than conventional convolutional networks. Figure 2 illustrates how the standard convolution filter (image a) is replaced by two layers in MobileNet V1. These layers use depthwise convolution (image b) and pointwise convolution (image c) to form a depthwise separable filter.

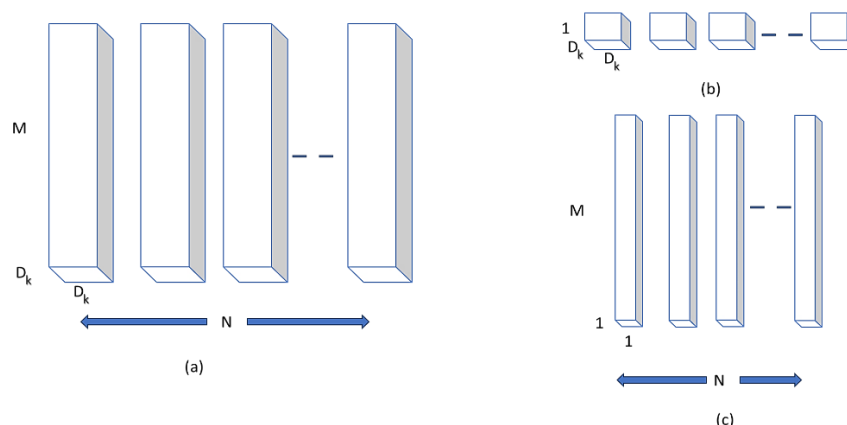


Figure 2. Convolutional Mechanisms in MobileNet

This study uses MobileNet as its base, with pre-trained weights from ImageNet and without the top layer. The model is built sequentially using the Keras Sequential API, which allows for layer-by-layer construction. First, the GlobalAveragePooling2D layer converts the 2D feature maps into a 1D vector by averaging each feature map. Next, a dense layer with 512 neurons and ReLU activation is added to provide non-linearity. Following this, a dropout layer with a 50% dropout rate helps prevent overfitting by randomly ignoring half of the neurons during training. Finally, a dense layer with the number of neurons corresponding to the number of classes (78) and softmax activation produces probabilities for each class.

The model accepts input images with a shape of (224, 224, 3), meaning images of 224x224 pixels with 3 color channels (RGB). It is designed to classify images into 78 different classes. The model is compiled using the Adam optimizer with a learning rate of 0.001, the categorical_crossentropy loss function, and accuracy as a metric. By leveraging the efficient and powerful MobileNet architecture, this model aims to perform well in image classification tasks.

2.3 Front-End Integration

In this study, the Python Flask package is utilized. Flask is a lightweight web framework for Python used to build web applications. In this context, Flask serves as the backend framework that handles server-side logic, routes, and API endpoints. The machine learning model developed using Keras is integrated with Flask to process input data (prescriptions) and generate predictions. The web-based user interface (UI) allows users, such as medical staff, to input handwritten prescriptions into the system. The UI is designed to be intuitive and user-friendly, ensuring that users can easily upload images or enter text data. Once the data is submitted, the UI interacts with Flask to send the data to the backend, where it is processed by the Keras model and predictions are generated. The results are then displayed back on the UI, which can be accessed through a browser, for the user to review.

2.4 Model Training dan Evaluation

2.4.1 CNN Model Training

The training process of the model was conducted over 50 epochs, where each epoch represents a full cycle through the entire training dataset. The training dataset consists of 3,120 preprocessed images. This

dataset was divided into small batches, each containing 32 samples, allowing the model to update its weights after processing each batch. A batch size of 32 was chosen because it is a commonly used size that balances training speed and weight update stability. The loss function used is `categorical_crossentropy`, which is suitable for multi-class classification problems. This loss function measures the difference between the model's predictions and the actual labels, aiming to minimize this error during training. We chose `categorical_crossentropy` because it is more effective for handling multi-class classification problems compared to other loss functions like `mean_squared_error`, which is more suitable for regression, or `binary_crossentropy`, which is used for binary classification.

The optimizer used is Adam with a learning rate of 0.001. An optimizer is an algorithm used to adjust the neural network's attributes, such as weights and biases, to reduce errors. Adam was chosen because it combines the advantages of two other optimization methods, AdaGrad and RMSProp, making it efficient and capable of performing well without much parameter tuning. Adam is also known for its ability to adjust the learning rate during training, helping to achieve faster and more stable convergence. A learning rate of 0.001 was chosen because it is a commonly used value that provides a balance between convergence speed and training stability. A learning rate that is too high can cause the model not to converge, while a learning rate that is too low can make training very slow. This training process aims to minimize the loss function and improve the model's accuracy on the validation data. Validation was performed using 780 images from the dataset, which were used to evaluate the model's performance during training.

2.4.2 Mobilenet Model Training

The training process of the model in the provided code involves several important steps. The model is trained for 50 epochs, with a batch size of 32, meaning the model is updated after seeing 32 training samples. The loss function used is `categorical_crossentropy`, which is suitable for multi-class classification problems. The optimizer used is Adam with a learning rate of 0.001, which combines the advantages of the AdaGrad and RMSProp optimization methods.

The model uses the MobileNet architecture as its base, which has been previously trained on the ImageNet dataset, but without the top layers. The input to the model has dimensions of 224 x 224 x 3, meaning the input images are resized to 224 pixels by 224 pixels with 3 color channels (red, green, and blue). Therefore, the training data is preprocessed to match these dimensions, such as resizing the images to 224 x 224 pixels and ensuring they have 3 color channels.

After that, several additional layers are added, including GlobalAveragePooling2D, Dense with 512 units and ReLU activation, Dropout with a rate of 0.5, and Dense with the number of units corresponding to the number of classes (78) and softmax activation. GlobalAveragePooling2D is a layer that simplifies the data by averaging each feature map over its spatial dimensions, effectively reducing the data's dimensionality.

2.4.3 Evaluation

After the training is completed, the model is evaluated using metrics such as accuracy, precision, recall, and F1-score. Accuracy measures how often the model's predictions are correct overall (1). Precision measures the proportion of true positives (correctly predicted positive cases) out of all positive predictions, thus accounting for false positives (incorrectly predicted positive cases) (2). Recall measures the proportion of true positives out of all actual positive cases, thus accounting for false negatives (actual positive cases that were incorrectly predicted as negative) (3). The F1-score is the harmonic mean of precision and recall, providing a balanced view of both metrics (4). This evaluation provides a more comprehensive understanding of the model's performance across various aspects.

2.5 User Interface Design and UX Goals

The UI of the Prescription Detector is designed with a minimalistic layout to ensure ease of use and avoid overwhelming users (see Figure 3). Key action areas, such as the file upload zone and the "Upload" button, are prominently displayed for intuitive interaction. The UI supports both drag-and-drop functionality and a file picker, catering to diverse user preferences. Feedback mechanisms are prominently integrated, providing immediate results and confidence levels after uploading, ensuring clarity without requiring additional navigation. Additionally, a "Help" button at the top offers quick access to documentation or FAQs, supporting users who may need guidance during the process.

The process flow of the Prescription Detector begins with users selecting or dragging their prescription image into the designated upload area, ensuring a user-friendly and straightforward interaction. Once uploaded, the system validates the file's format and suitability for prediction, checking aspects such as size and type. The validated image is then processed by a backend AI-powered model, which analyzes the prescription details and generates a prediction result. The output, including the prediction and a confidence level, is displayed in a clear, readable format. For example, a confidence level of 0% may indicate an issue with the uploaded file or data, prompting users to retry or seek assistance.

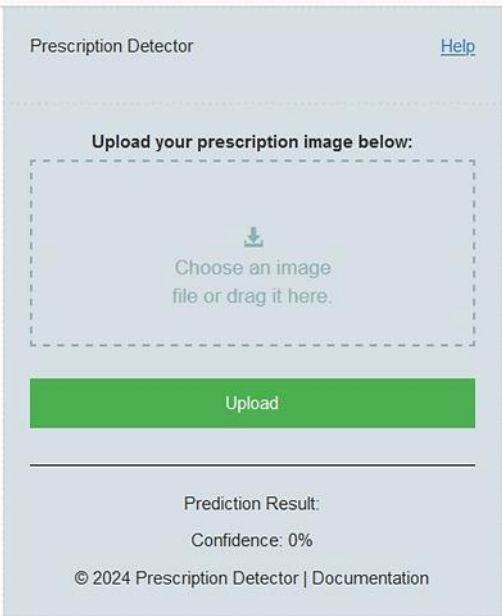


Figure 3. Prescription Detector Upload Form

The system provides comparative information on whether prescribed medication meets the patient's needs and offers alternative medications based on the patient's condition and medical history. This helps users make informed decisions regarding treatment. The classification results are clearly displayed for easy viewing and saving. The primary goal is to minimize medication errors and enhance healthcare quality.

3. RESULT AND ANALYSIS

The performance of the Handwriting Recognition System was evaluated using a test dataset consisting of 780 prescription images. Table 1 shows the performance comparison between the CNN and MobileNet models. The CNN model achieved an accuracy of 63%, with precision, recall, and F1-score of approximately 65%, 62%, and 63% respectively. On the other hand, the MobileNet model achieved an accuracy of 83%, with precision, recall, and F1-score of approximately 84%, 82%, and 83% respectively.

Table 1. Performance Comparison of CNN and MobileNet Models

Matric	CNN	MobiliNet
F1	0,63	0,83
Precision	0,65	0,84
Recall	0,62	0,82
Accuracy	0,63	0,83

The evaluation process was conducted through several steps. First, a dataset of prescription images was collected and prepared for model training and testing. These images were then processed and labeled to ensure that each image had the correct annotations. Following this, the CNN and MobileNet models were trained using the training dataset, and their performance was tested using the test dataset. In terms of case studies, the model showed notable success in identifying certain words. For instance, it accurately identified the word "Aceta" from a prescription image with a high confidence score of 85%, thanks to the clear and distinct handwriting. Similarly, the word "Amodis" was correctly recognized with a confidence score of 80%, attributed to the relatively simple and legible handwriting style.

However, the model also encountered challenges. For example, it struggled with the word "Azithrocin" due to the cursive and overlapping handwriting style, resulting in a low confidence score of 45% and incorrect identification. Another difficult case was the word "Baclon," which was misinterpreted as "Baclofen" due to the similarity in character shapes and the presence of noise in the image.

The test results suggest that while MobileNet model performs well on clear and distinct handwriting, it faces difficulties with cursive and overlapping styles. These challenges can be attributed to the variability in handwriting styles, noise and artifacts in the prescription images, and the complexity of the model. Despite these challenges, the study demonstrates the potential of using MobileNet for handwriting recognition in medical prescriptions, with opportunities for further improvements to enhance accuracy and reliability.

4. DISCUSSION

The Handwriting Recognition System shows potential in handling real-world variabilities, particularly in the legibility of handwritten prescriptions. The use of CNN with MobileNet is effective in recognizing clear text, but its performance decreases with more complex handwriting. Additionally, the model is limited in handling multilingual prescriptions because the dataset used primarily consists of English words written by Bangladeshi doctors. To address this, future research could involve training the model on a more diverse and multilingual dataset, thereby enhancing its robustness and versatility.

Additionally, the system's performance on extremely illegible text remains a significant challenge. Noise and artifacts in the prescription images can obscure the text, leading to misinterpretation. Enhancing the pre-processing steps to better clean and segment the images could help mitigate this issue. Moreover, incorporating advanced techniques such as attention mechanisms or transformer-based models could improve the system's ability to focus on relevant parts of the image and better handle complex handwriting styles.

In conclusion, while the Handwriting Recognition System shows potential in converting doctors' handwritten prescriptions into readable text, there are several areas for improvement. Addressing the variability in handwriting styles, expanding the model's capability to handle multilingual prescriptions, and enhancing its performance on extremely illegible text are crucial steps towards developing a more robust and reliable system. Future research and development efforts should focus on these aspects to further enhance the accuracy and applicability of the system in real-world scenarios.

5. CONCLUSION AND FUTURE DIRECTIONS

This study has made significant contributions to enhancing healthcare efficiency and accuracy by developing a Handwriting Recognition System using CNN to interpret doctors' handwritten prescriptions. The system, particularly the model with MobileNet, demonstrated a notable improvement in recognizing and converting cursive handwriting into readable text. With a validation accuracy of 88% and a test accuracy of 83% on the test set, the model shows promise in reducing the risk of misinterpretation of prescriptions, which can lead to negative consequences for patients. The successful implementation of the model in a web application further underscores its practical applicability. Despite the challenges faced with more complex and less legible handwriting, the system's ability to correctly identify a significant portion of the test samples highlights its potential utility in real-world healthcare settings.

For future research, several avenues can be explored to enhance the system's performance and applicability. One promising direction is the integration of natural language processing (NLP) techniques to not only recognize handwritten text but also understand the content of prescriptions. This could enable the system to provide additional context and ensure the correct interpretation of medical instructions. Another important area for future work is the collaboration with healthcare facilities to access larger and more diverse datasets. This would allow the model to be trained on a wider variety of handwriting styles and languages, improving its robustness and generalizability. Additionally, incorporating advanced machine learning techniques, such as transformer-based models, could further enhance the system's ability to handle extremely illegible text and complex handwriting patterns.

In conclusion, while the current study has laid a strong foundation for the development of a handwriting recognition system for medical prescriptions, ongoing research and development are essential to address its limitations and expand its capabilities. By leveraging advancements in machine learning and collaborating with healthcare professionals, the system can be refined to provide even greater accuracy and reliability, ultimately contributing to safer and more efficient healthcare delivery.

REFERENCES

- [1] N. Alrabadi *et al.*, "Medication errors: a focus on nursing practice," *Journal of Pharmaceutical Health Services Research*, vol. 12, no. 1, pp. 78–86, Feb. 2021, doi: 10.1093/jphsr/rmaa025.
- [2] L. C., "New Questions on the Road to Safer Health Care," *Pediatrics*, doi: 10.1542/peds.2013-0647.
- [3] H. Carole, H. M., E. S. Susan, and D. U. Beth, "Who killed patient safety?," *J Patient Saf Risk Manag*, doi: 10.1177/25160435221077778.
- [4] R. Mannion and J. Braithwaite, "False Dawns and New Horizons in Patient Safety Research and Practice," *Int J Health Policy Manag*, vol. 6, no. 12, pp. 685–689, doi: 10.15171/ijhpm.2017.115.
- [5] L. J. Donaldson, C. Lemer, D. J. Noble, F. Greaves, and M. Fletcher, "Finding the Achilles' heel in healthcare," *J R Soc Med*, vol. 103, no. 2, pp. 40–41, doi: 10.1258/jrsm.2009.090417.
- [6] H. H. Louise, J. Judith, W. Ian, T. A., and B. O. Daryl, "Healthcare Staff Wellbeing, Burnout, and Patient Safety: A Systematic Review," *PLoS One*, doi: 10.1371/journal.pone.0159015.
- [7] G. F. Cavell and D. Mandaliya, "Magnitude of error: a review of wrong dose medication incidents reported to a UK hospital voluntary incident reporting system," *European Journal of Hospital Pharmacy*, vol. 28, no. 5, pp. 260–265, Sep. 2021, doi: 10.1136/ejhpharm-2019-001987.
- [8] L. L., "A systems analysis approach to medical error," *J Eval Clin Pract*, doi: 10.1046/J.1365-2753.1997.00006.X.

- [9] G. A. James, "A systems approach to preventing adverse drug events.," *Stud Health Technol Inform.*
- [10] C. Michelle *et al.*, "Influence of a systems-based approach to prescribing errors in a pediatric resident clinic.," *Acad Pediatr*, doi: 10.1016/j.acap.2014.03.018.
- [11] S. Mohsin-Shaikh *et al.*, "The impact of electronic prescribing systems on healthcare professionals' working practices in the hospital setting: A systematic review and narrative synthesis," *BMC Health Serv Res*, vol. 19, no. 1, pp. 1–8, 2019, doi: 10.1186/s12913-019-4554-7.
- [12] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, doi: 10.1109/access.2020.3012542.
- [13] S. Karthikeyan, A. G. S. de Herrera, F. Doctor, and A. Mirza, "An OCR Post-Correction Approach Using Deep Learning for Processing Medical Reports," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 2574–2581, doi: 10.1109/tcsvt.2021.3087641.
- [14] L. V Rasmussen, P. L. Peissig, C. A. McCarty, and J. Starren, "Development of an optical character recognition pipeline for handwritten form fields from an electronic health record," *Journal of the American Medical Informatics Association*, vol. 19, no. e1, pp. e90–e95, doi: 10.1136/amiajnl-2011-000182.
- [15] N. Gal-Nadasan, V. Stoicu-Tivadar, E. Gal-Nadasan, and A. R. Dinu, "Robotic Process Automation Based Data Extraction from Handwritten Medical Forms," in *Studies in Health Technology and Informatics*, IOS Press, 2023. doi: 10.3233/shti230741.
- [16] E. Coiera and S. Liu, "Evidence synthesis, digital scribes, and translational challenges for artificial intelligence in healthcare," *Cell Rep Med*, vol. 3, no. 12, p. 100860, doi: 10.1016/j.xcrm.2022.100860.
- [17] L. V. Rasmussen, P. L. Peissig, C. A. McCarty, and J. Starren, "Development of an optical character recognition pipeline for handwritten form fields from an electronic health record," *Journal of the American Medical Informatics Association*, vol. 19, no. e1, pp. e90–e95, Jun. 2012, doi: 10.1136/amiajnl-2011-000182.
- [18] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017, [Online]. Available: <http://arxiv.org/abs/1704.04861>