الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

جامعة عمار ثليجي - الأغواط
Amar Telidji University of Laghouat

FACULTY OF SCIENCES DEPARTMENT OF
COMPUTER SCIENCE

Master's Thesis

Field: Mathematics and Computer Science
Major :Computer Science
Specialization: Information and Decision Systems - Distributed networks, systems, and applications

Presented by:

- Benatmani Abdelbasset -Djoudi Ikhlasse

Automated Medical Labels Detection and Text Extraction

Jury members :
Mr : Youcef Ouinten - Senior Lecturer (MCA) – President
Mr : Laradj Chellama - Professor (PROF) – Examiner
Mr : Saida Sarra Boudouh- Examiner
Mr : Bouakkaz Mustapha -PROF- Professor (PROF) – Supervisor

Academic year 2024/2025

Table of Contents

Abstract:

Modern technology aims to automatically read and understand handwritten prescriptions using deep learning. This is critical given that handwritten notes are often illegible and difficult to read, which can lead to serious medical errors.

Deep learning, a branch of artificial intelligence, is used to train computer models to accurately recognize and interpret these prescriptions. Using large amounts of data and powerful algorithms, deep learning models can learn to identify handwriting patterns and extract important information such as medication names and dosages.

This technology contributes to improving the safety and efficiency of healthcare services, as it can significantly reduce documentation errors caused by illegible handwriting, ensuring that clinical decisions are based on accurate and timely information.

In this note, we address the future role of deep learning in this field, noting that new neural network architectures such as vision transformers and hybrid CNN-RNN models promise to increase the accuracy and effectiveness of handwritten note recognition. It explains that AI-based handwriting recognition in electronic health records (EHRs) can speed up communication between teams and reduce administrative burdens, allowing clinicians to focus on direct patient care rather than burdensome paperwork, thereby reducing burnout and increasing job satisfaction.

In conclusion, we conclude that the revolutionary potential of deep learning in medical handwriting recognition has the potential to revolutionize health outcomes by ensuring accurate, timely, and efficient recording. It suggests that adopting these innovations will be pivotal in developing a healthcare system where technology enables but does not replace human empathy and clinical expertise.

Keywords: Deep Learning, Handwriting Recognition, Medical Prescriptions / Prescriptions, Artificial Intelligence (AI) , Healthcare , Neural Networks , Vision Transformers, Hybrid Models (CNN-RNN)

المخص:

تهدف التكنولوجيا الحديثة إلى قراءة وفهم الوصفات الطبية المكتوبة بخط اليد تلقائيًا باستخدام التعلم العميق، وهو أمر بالغ الأهمية نظرًا لعدم وضوح وصعوبة قراءة الملاحظات المكتوبة بخط اليد غالبًا، مما قد يؤدي إلى أخطاء طبية خطيرة.

يُستخدم التعلم العميق، وهو فرع من فروع الذكاء الاصطناعي، لتدريب النماذج الحاسوبية على التعرف على هذه الوصفات وتفسيرها بدقة. من خلال استخدام كميات كبيرة من البيانات والخوارزميات القوية، يمكن لنماذج التعلم العميق تعلم تحديد أنماط الكتابة اليدوية واستخراج المعلومات الهامة مثل أسماء الأدوية وجرعاتها.

إن هذه التكنولوجيا تساهم في تحسين سلامة وكفاءة خدمات الرعاية الصحية، حيث يمكن أن تقلل بشكل كبير من أخطاء التوثيق الناتجة عن خط اليد غير المقروء، مما يضمن أن القرارات السريرية تستند إلى معلومات دقيقة وفي الوقت المناسب.

نطرق في مذكرتنا هذه الى الدور المستقبلي للتعلم العميق في هذا المجال، مشيرة إلى أن بنى الشبكات العصبية الجديدة مثل محولات الرؤية (Vision Transformers) والنماذج الهجينة (CNN-RNN) تعد بزيادة دقة وفعالية التعرف على الملاحظات المكتوبة بخط اليد. وتوضح أن التعرف على خط اليد القائم على الذكاء الاصطناعي في السجلات الصحية الإلكترونية (EHRs) يمكن أن يسرع التواصل بين الفرق ويقلل الأعباء الإدارية، مما يسمح للأطباء بالتركيز على رعاية المرضى المباشرة بدلاً من الأعمال الورقية المرهقة، وبالتالي تقليل الإرهاق وزيادة الرضا الوظيفي.

في الختام، توصلنا الى أن الإمكانات الثورية للتعلم العميق في التعرف على خط اليد الطبي لديها القدرة على إحداث ثورة في النتائج الصحية من خلال ضمان تسجيل دقيق، في الوقت المناسب، وفعال. وتشير إلى أن تبني هذه الابتكارات سيكون محوريًا في تطوير نظام رعاية صحية حيث تسمح التكنولوجيا - ولا تحل محل - التعاطف البشري والمهارة السريرية.

كلمات مفتاحية: التعلم العميق، التعرف على خط اليد، الوصفات الطبية، الذكاء الاصطناعي من الأخطاء الطبية ، الشبكات العصبية ، النماذج الهجينة.

Chapter 01:

introduction to handwritten prescription recognition using deep Learning

Chapter 1: Introduction

1.1 General Introduction

Medical prescriptions are essential documents in modern healthcare systems. They serve as the primary medium of communication between healthcare providers and patients, detailing critical instructions regarding medication types, dosages, and treatment durations. The accuracy and clarity of these documents are directly linked to effective patient care and treatment outcomes. A clear and unambiguous prescription ensures that the patient receives the correct medication, in the correct dose, and at the correct frequency.

Despite the global push toward digitization in healthcare, a significant proportion of prescriptions are still handwritten. This is especially true in low-resource settings, private clinics, and fast-paced environments where digital infrastructure is limited or underutilized. Unfortunately, the quality of handwritten prescriptions often suffers from legibility issues due to a variety of factors such as physician workload, time pressure, and individual handwriting styles. These issues pose serious risks to patient safety and can lead to adverse outcomes.

The implications of illegible prescriptions are far-reaching. Pharmacists may misinterpret drug names or dosages, leading to medication errors. In some cases, ambiguous handwriting necessitates follow-up communication between pharmacists and physicians, delaying treatment and increasing administrative overhead. Moreover, illegible prescriptions can have legal consequences for healthcare providers in cases where errors result in harm to the patient.

In recent years, the field of artificial intelligence (AI), particularly deep learning, has offered promising tools to tackle problems in document image analysis and handwriting recognition. Convolutional Neural Networks (CNNs) and other neural network architectures have shown excellent performance in image classification, pattern recognition, and text recognition tasks. These technologies have the potential to bridge the gap between handwritten and machine-readable documents, offering new avenues for improving the reliability and safety of healthcare services.

This thesis focuses on the development of an AI-based system for recognizing and interpreting handwritten medical prescriptions, with an emphasis on drug

name recognition. The goal is to build a practical solution that can process scanned or photographed prescription images, segment handwritten text into individual words, and classify them using a deep learning model.

1.2 Problem Statement

Handwritten prescriptions continue to be widely used in healthcare settings, yet their interpretation remains a manual task prone to human error. The variability in handwriting styles among physicians, the frequent use of non-standard abbreviations, and the lack of formatting conventions make handwritten prescriptions difficult to process using traditional optical character recognition (OCR) tools. Conventional OCR systems such as Tesseract are not optimized for the cursive and erratic writing styles found in medical contexts.

Moreover, medical prescriptions often include domain-specific terminology, shorthand notations, and brand names that are not commonly found in general-purpose handwriting datasets. This makes it challenging to apply off-the-shelf solutions without extensive domain adaptation. Errors in interpreting handwritten drug names or instructions can result in incorrect medication administration, serious health consequences, or even fatalities.

While some digital healthcare systems have adopted electronic prescribing, this is not yet universally implemented, particularly in resource-limited regions. Consequently, there is an urgent need for an automated solution that can accurately interpret handwritten prescriptions, particularly the drug names, which are central to patient treatment.

This thesis addresses the specific problem of recognizing individual handwritten drug names from scanned or photographed medical prescriptions using a deep learning approach, and evaluates its feasibility for real-world deployment in healthcare environments.

1.3 Motivation and Objectives

Motivation

The motivation for this research stems from the significant risks posed by illegible handwritten prescriptions in healthcare. According to reports from the World Health Organization (WHO) and other public health institutions, prescription errors are among the most common preventable causes of medication-related injuries and deaths. Reducing such errors is not only a clinical priority but also an ethical and legal responsibility.

The growing capabilities of AI, especially in visual recognition tasks, offer a new opportunity to mitigate this issue. By using deep learning techniques, it is possible to build models that can learn from labeled examples of handwritten text and generalize to new, unseen data. Specifically, CNNs like VGG16 provide a balance between simplicity, efficiency, and accuracy, making them suitable for implementation in environments with limited computational resources.

This project is motivated by the desire to apply these technological advancements to a real-world healthcare problem—improving the interpretation of handwritten medical prescriptions to enhance patient safety, reduce administrative delays, and support the broader goals of healthcare digitization.

Objectives

The primary objectives of this thesis are as follows:

- To build a deep learning pipeline that can process prescription images by segmenting them into lines and then into individual word images.
- To train and evaluate a VGG16-based Convolutional Neural Network (CNN) for the classification of handwritten drug names.
- To develop an effective image preprocessing and segmentation algorithm for isolating handwritten words from complex and noisy prescription images.
- To assess the model's performance using appropriate evaluation metrics, such as classification accuracy, precision, recall, and F1-score.
- To investigate the practical viability of integrating the proposed system into a web-based interface using frameworks like Flask for use in clinical or pharmacy settings.

1.4 Research Questions

The following research questions guide the study:

1. **Can a VGG16-based CNN model be trained to accurately recognize handwritten drug names from prescription images?**
2. **What preprocessing and segmentation techniques are most effective for isolating word-level regions from noisy, handwritten medical images?**
3. **How does the proposed system perform compared to baseline OCR approaches in terms of accuracy and robustness?**

4. **What are the major limitations and challenges associated with deploying such a system in real-world healthcare environments, and how can they be addressed?**

These questions are designed to ensure that the research remains focused, measurable, and aligned with both academic rigor and practical relevance.

1.5 Scope and Limitations

Scope

- The research is focused on the recognition of **handwritten drug names** in **English-language** prescriptions.
- The input data consists of **scanned or photographed images** of handwritten medical prescriptions.
- The system includes a complete pipeline: from image preprocessing, line and word segmentation, to word classification.
- The deep learning model used for classification is based on **VGG16**, fine-tuned on a **custom dataset** of cropped word images.
- The final system is deployed as a **Flask web application** to demonstrate usability in a real-world context.

Limitations

- The model is trained on a relatively small and specific dataset; hence, its generalizability across different languages, scripts, or regions is limited.
- The system only focuses on drug name recognition and does not interpret dosages, frequency, or complex handwritten instructions.
- Prescription images with extreme blur, noise, or overlapping text may not be handled effectively without further enhancement.
- Integration with hospital Electronic Health Record (EHR) systems, while desirable, is beyond the scope of this thesis.
- The performance of the model is sensitive to the quality of image segmentation and preprocessing.

1.6 Thesis Organization

This thesis is organized into four chapters, each designed to progressively guide the reader through the research problem, methodology, and findings:

- **Chapter 1: Introduction**
  Introduces the background, context, and motivation of the research. It

presents the problem statement, research questions, objectives, and scope of the study. It also explains the structure of the thesis.

- **Chapter 2: Study of Previous Research**
  Provides a detailed review of relevant literature in handwritten text recognition, OCR techniques, and AI-based solutions in healthcare. It evaluates the strengths and limitations of existing approaches and justifies the selection of the VGG16 model for this study.
- **Chapter 3: Implementation and Validation**
  Describes the practical development of the proposed system, including dataset preparation, preprocessing steps, model training, and testing. The chapter also includes a comparative evaluation of VGG16 with other models like ResNet50 and Inception-v3.
- **Chapter 4: Conclusion**
  Summarizes the main results and findings of the research. It reflects on the contributions and limitations of the study, and outlines potential future directions, including real-time deployment, dataset expansion, and multi-language support.

Chapter 02:

Prevouis work

Introduction:

This section discusses recent works in handwritten prescription recognition, primarily in the aspects of deep models, as well as in processing (text-parsing) of handwritten prescriptions for extracting the name of the medicine. It analyses hybrid architectures composed of CNNs, attention mechanisms and transformer based models such as TrOCR in an attempt to address handwriting variations and prescription format. Particular attention is given to practical and scalable solutions such as VGG16, which strikes a compromise between accuracy and deployability in real-life medical scenarios.

1- Leveraging Deep Learning with Multi-Head Attention for Accurate Extraction of Medicine from Handwritten Prescriptions

This is indeed a hybrid approach between Mask R-CNN and a transformer-based Optical Character Recognition (TrOCR) system with Multi-Head Attention, and Positional Embeddings-it aims at extracting medicine names precisely from handwritten prescriptions from a doctor.

The goal was to develop a feasible solution for extracting medicine names from handwritten prescriptions by addressing all the limitations of existing approaches to text recognition, including all the variations of handwritten styles and the different types of prescription formats.

The methodology comprises feature extraction by means of ResNet-50 with a Feature Pyramid Network (FPN), region extraction using a Region Proposal Network (RPN), precise alignment using RoI Align followed by masking to isolate text. The segmented regions are then sent through the TrOCR model for handwritten text recognition, and then a hybrid string match is done for accurate medicine naming.

The hybrid between Mask R-CNN for segmentation and TrOCR gives promise for a prototype of handwriting recognition systems. It uses the Vision Transformer (ViT) architecture and applies a multi-head attention mechanism to capture relationships and features at multiple scales across the patches.

The approach has produced state-of-the-art-character error rate (CER) of 1.4% on the public benchmarks and surpassed all previous models.

The presented method resulted in achieving a CER of 1.4% on the standard benchmarks and outperformed all other pretrained models tested in this research.

2- HTR-JAND: Handwritten Text Recognition with Joint Attention Network and Knowledge Distillation

The study reports HTR-JAND, an all-encompassing methodology towards Handwritten Text Recognition which comes up with solutions to pertinent issues of processing historical documents using an optimized knowledge distillation framework and returning state-of-the-art results across several benchmarks. A first objective of this research study mainly includes developing an overall methodology for Handwritten Text Recognition that answers some of the major gaps in dealing with the issues in really old works, followed by estimating how effective this method could be from an extensive ablation survey.

The study has a total preprocessing pipeline that brings character set unification across datasets merged with adaptive oversampling, thus allowing a balanced representation of a vocabulary of 103 characters unified across the different historical periods and writing styles.

The methods used in this research study include: Teacher-Student model with knowledge distillation for knowledge transfer from high-capacity Teacher model to a compact Student model; Curriculum learning with synthetic data for gradually introducing complexity and style variations; Multi-task learning for unifying different tasks and datasets; T5-based post-processing stage to rectify residual errors and improve recognition accuracy.

HTR-JAND achieves state of the art Character Error Rates on IAM, RIMES, and Bentham datasets, while maintaining practical efficiency through major parameter reduction. Findings of this study suggest that such an approach gives out state-of-the-art results on several HTR datasets, with considerable improvements in error rates of characters, words and sentences.

3- Integrating Canonical Neural Units and Multi-Scale Training for Handwritten Text Recognition

This paper presents a new recognition network for handwritten text recognition, based on a 3D attention module and global-local context information,

and reaches comparable performance with state-of-the-art techniques on Chinese and English handwritten text databases.

The aim of this research is the proposal of a novel recognition network for handwritten text recognition, using a 3D attention module and global-local context information, and the performance evaluation on Chinese and English handwritten text datasets.

A CNN, a 3D attention module and a features integration block compose the proposed network. The 3D attention module extracts 2D information of blocked feature maps at different resolutions, showing a 3D attention module and capturing global-local context information for recognizing handwritten text. The network is trained under a multi-scale training strategy and optimizes weights by the CTC loss and the CE loss.

This new approach-simply put the proposed network-can outperform state-of-the-art systems on several datasets compared to existing techniques. The proposed network performs equally well with current state-of-the-art techniques on the three datasets used. The execution results are seen in both tables and figures displaying the character accuracy rates (ARs) exhibited by the proposed network.

4- Enhancing handwritten text recognition accuracy with gated mechanisms

The paper provides an overview of the Handwritten Text Recognition using gated mechanisms, which have become one of the most important improvements to HTR systems, enabling capturing long-term dependencies, retaining relevant context, and adapting to contextual variations.

The objective of the study is to improve the accuracy and resilience of handwritten text recognition systems using gated mechanisms, which have proven effective in gathering contextual information and long-range interactions in a variety of machine learning applications.

The proposed study will use the Gated-Convolutional Recurrent Neural Network (Gate-CRNN) architecture with a gated mechanism developed by Dauphin to improve the accuracy of the offline HTR systems.

The paper learns a convolutional-based model combining squeeze and excitation gate mechanisms, connectionist temporal classification, and gated convolutional neural networks.

The proposed model is a combination of a Gated mechanism with a convolutional block and a recurrent block. The model is trained using the RMS prop optimizer with a mini-batch size of 16 images and an initial learning rate of 0.001. The model has been evaluated with Word Error Rate (WER) and Character Error Rate (CER) metrics.

State-of-the-art deep learning methods and tool-kits such as convolutional block gated mechanism and bidirectional gated recurrent components have been used in an attempt to develop the proposed model.

The proposed CNN-BLSTM-based results are further improved using a novel architecture, Gated-CNN-BGRU, to allow adaptation to varied noises, styles, and variations using a lower amount of practice data.

The proposed model attains an accuracy of 94.26 on the dataset.

The proposed model reaches state-of-the-art performance on some benchmark datasets: Bentham, RIMES, IAM, Washington, and Saint Gall. The model achieves substantial Word Error Rate (WER) and Character Error Rate (CER) improvements on all datasets.

The proposed model outperformed the existing models, including Puigcerver, Bluche, and Flor, by large margins in terms of CER and WER values over many datasets.

5- Handwritten Digit Classification Using Deep Learning Convolutional Neural Network

According to the study, a deep convolutional neural network model is proposed with different filter sizes in order to increase the recognition rate of handwritten digits, using average classification accuracy of up to 99.5% on the MNIST dataset.

The objective of this study is to design a reliable and correct model for handwritten digit recognition using CNN with the suitable filter sizes.

The paper utilizes deep CNN models with different filter sizes to achieve higher recognition rates for handwritten digits. The architecture of the proposed model also includes one fully connected layer for classification and a single convolution and activation layer for feature extraction.

The proposed CNN architecture comprises a concatenation of convolution and pooling layers for later use by a fully connected layer and a softmax output layer. Training the network is facilitated through use of stochastic gradient descent with momentum (sgdm) algorithm.

The proposed CNN model achieves its average classification accuracy of up to 99.5% when tested on the MNIST dataset.

The proposed CNN architecture was able to achieve high validation accuracy of 99.5% on the MNIST dataset when trained using a 5x5 filter size and for 15 epochs. These findings were verified through a confusion matrix, which indicated that most of the images used were correctly classified by the model.

6- An inclusive review on deep learning techniques and their scope in handwriting recognition

This paper provides a review of the present technology in deep learning for handwriting recognition using the modern architectures of CNN and RNN with their several applications for handwriting recognition tasks.

The review outlines the objectives of the research, which aims at reviewing literature on deep learning concerning handwriting recognition, as well as identifying the accomplishments and challenges deep learning has faced in this domain.

It involves a literature review of the methods for deep learning in handwriting recognition with respect to architectural and performance analysis-training of the CNN and RNN counts in this domain.

The study applies survey methodology to review the current state-of-the-art in deep learning techniques used for handwriting recognition. The authors review various studies that have used CNN and RNN in handwriting recognition and analyze their results.

The study visits more architectures in deep learning, including CNN, RNN, LSTM, as well as those derived from them, for the documentation of systems they apply for handwriting tasks.

It states that CNN and RNN have shown the best results from the use of deep learning and that deep learning techniques have out-performed all other methods in this area of work.

Three major results reported from several studies conducted using CNN and RNN in handwriting recognition are:

99.74% accuracy on the MNIST dataset through a multi-column multi-scale CNN-based architecture.

For eight Indic scripts combined, 96.23% accuracy using a script-independent CNN-based system.

94.13% and 93.61% for Gurmukhi and UNIPEN datasets respectively using a newly developed self-controlled RDP point-sized feature vector based on smaller vector constructs.

93.55% and 93.10% accuracy for handwritten English and French words respectively by using a CNN-N-Gram-based system

7- A Novel Approach for Hand-written Digit Classification Using Deep Learning

The study examined how various machine learning algorithms like Multilayer Perceptron, Support Vector Machine, Random Forest, Bayes Net, Naive Bayes, J48 and Random Tree performed in recognizing handwritten digits, out of which the Multilayer Perceptron achieved 92.37% accuracy, the highest.

Hence, the primary objective is to find a model optimal among the digit recognition scenarios through the comparison of accuracies and execution timings across SVM, MLP, and CNN models.

In this paper, SVM, MLP, and CNN models under MNIST data were used to compare accuracies and training times.

The study performed with CNN, SVM, and MLP algorithms to digitally recognize handwritten digits and also brings some data pre-processing methods as

normalization and one-hot encoding for data training. The study conducted comparative analyses with several machine learning models on performance of handwritten digit recognition components.

Best accurate results were obtained from CNN, followed by MLP and SVM.

The proposed method for ghost imagery generation, defined by cosine transformation speckle filter, achieved high accuracy in handwritten digit recognition. The study included comparative performance evaluation of different machine learning and deep learning models such as SVM, MLP, and CNN on the MNIST data set.

The results showed that among the stated algorithms, Multilayer Perceptron had the highest accuracy of 92.37% and the next lower by other algorithms.

8- Real time handwriting recognition system using CNN algorithms

This research proposes an automatic handwriting recognition system using Convolutional Neural Network (CNN) algorithms, with the objective of achieving higher accuracy performance and shortening the recognition time for real applications that demand on-time processing.

The crucial objectives of this research are: to construct a model to use CNN in the recognition of handwritten characters, improve the performances of handwriting recognition systems within real-time applications and solve the drawbacks of low precision in handwriting character recognition systems.

The proposed system uses CNN algorithms such as AlexNet, GoogLeNet, and SqueezeNet to recognize the handwriting characters. Preprocessing of the dataset has been done in a novel approach with stroke hooks removing, noise filtering, and image normalization.

The study uses three Convolutional Neural Network (CNN) architectures-GoogleNet, AlexNet, and SqueezeNet-and compares their performance based on accuracy, testing time, and memory usage.

The experimental results show that the introduced framework achieves high accuracy values while, at the same time, reducing the recognition time for real-time applications. The SqueezeNet model achieved the best performance accuracy of 99.8%.

The results show that SqueezeNet exhibits better performance than the other two models in terms of accuracy and testing time.

9- Auxiliary Cross-Modal Representation Learning with Triplet Loss Functions for Online Handwriting Recognition

The paper proposes a cross-modal representation learning method using a triplet loss with dynamic margin for single label and sequence-to-sequence classification tasks, which achieves better classification accuracy, faster convergence, and better generalizability in online handwriting recognition.

The objective of this work is to propose a method for online HWR using sensor-augmented pens that increases the online HWR dataset by including generated images from offline handwriting and tries to close the gap between offline and online HWR using offline HWR as an auxiliary task.

The method uses a triplet loss setup with a dynamic margin specifically designed for single label and sequence-to-sequence classification tasks, combining offline handwriting recognition (HWR) from generated images and online HWR using sensor-equipped pens to learn a common representation for both modalities. The methods used in the study are: Convolutional neural networks (CNNs) with triplet loss functions for cross-modal representation learning. Dynamic triplet selection is performed based on the Edit distance.

Cross-modal representation learning with visual modality as an auxiliary task and inertial modality as the main task.

Fine-tuning model: four gated text recognizers with one and two convolutional layers.

The method uses a gated text recognizer module as the visual feature encoder for offline handwriting recognition (HWR) and introduces a contextual loss to enhance style consistency. Furthermore, the approach combines a triplet loss with a sample selection strategy conditioned on the Edit distance to enable cross-modal representation between the two modalities, namely, image, and time-series.

The experimental results show improved classification accuracy, faster convergence, and better generalizability due to the fine-tuned cross-modal representation.

The results of the study include:

An accuracy improvement from 92.5% to 96.76% by being more robust against noise present in the data.

Notably low error rates between 0.22% to 0.76% CER, and between 0.85% to 2.95% WER on the OffHW-[words500, wordsRandom] datasets.

A better adaptability to different writers, such as a better transfer learning from right-handed writers to the under-represented left-handed writers.

The results show that the proposed approach outperforms both ScrabbleGAN and HiGAN+ in terms of visual fidelity and textual coherence while outperforming the smaller CNN+BiLSTM in [38] on the WD classification tasks. There is also a notable gain in accuracy in the main time-series classification task with an improvement of Character Error Rate (CER) from 13.44% to 10.84% on the OnHW-words500 dataset.

10- CNN-BiLSTM model for English Handwriting Recognition:

Comprehensive Evaluation on the IAM Dataset

The authors propose a voice-recognition system that functions in the context of handwritten data using a hybrid of both CNN and RNN emphasizing on improving the results through data expansion and test-time expansion strategies.

The aim of the study is to develop a handwriting recognition system that is able to reliably detect handwritten text on a better scale through implementing data augmentation and testtime augmentation factors.

For the offline recognition of handwritten data, the proposed approach incorporates a CNN-BiLSTM model together with a CTC layer. The model proposes a dual role system whereby the BiLSTM will do sequential processing of the extracted images while the input image will have its features targeted by the CNN layer. Also, every single image input will have its character sequences constructed with the help of the CTC layer.

This research's primary focus is the recognition of handwritten words by utilizing a hybrid of CNNs and RNNs. Both the CNNs and RNNs have specific roles where the former is tasked with extracting functional components and the latter utilizes the sequenced functional components to undertake feature learning through sequential processing. Additionally the study empowers other multiple methods of data augmentation including shear, rotation, elastic distortion as well as test time augmentation where such alterations are made to the input images but only at the testing stage.

The research integrates CNN-BiLSTM and evaluates its performance against a variety of techniques, including data augmentation and synthetic data generation.

11- Bridging the Gap between Handwriting and Machine: AIbased Handwritten Text Recognition

The thrust of this further investigation is on the application of artificial intelligence to handwritten text recognition and the challenges and opportunities that neural networks present for HTR systems. It proposes a methodology for the effective development of an HTR system.

The goal of this project is to totally explore handwritten text classification and to digitize handwritten text.

The methodology proposed includes five main steps; those are: train data generation, image preprocessing, word detection and segmentation, machine learning algorithm training and implementation and validation. The devised system is intended to use convolutional and recurrent neural networks, in which GRCLs and BLSTMs should be incorporated, to recognize characters handwritten. This system will be trained by a back-propagation algorithm, and different image processing algorithms will be set for preprocessing and classification.

The paper presents results related to system development, testing, and validation for the Handwriting character recognition system. The system recognizes handwritten characters with very high accuracy while being able to tolerate a certain level of noise and recognize unknown patterns. It should be mentioned that the training time and iterations needed for handwritten characters are considerably greater than that needed for machine-written characters.

12- Comparison of CNN's Architecture GoogleNet, AlexNet, VGG-16, Lenet -5, Resnet-50 in Arabic Handwriting Pattern Recognition

This study also puts forward a way of writer recognition based on handwriting with a convolutional neural network (CNN) architecture, and it achieves 83.99% accuracy by using the VGG-16 model.

The aim of the study is to come up with a system that could recognize the writers by their handwriting using a convolutional neural network architecture.

This research is based on the dataset containing 8400 image data with an optimal comparison between testing and training data in the ratio of 80:20. The models were trained by using 64 filters for each convolution layer, kernel size 3x3, 128 neurons, 50% dropout weight, learning rate 0.001, image size 64x64, and normalization method with ReLU as the activation function.

The study used a convolutional neural network (CNN) architecture with different models, including VGG-16, AlexNet, GoogleNet, LeNet-5, and ResNet50. First, the input images were resized to 64x64, changed into grayscale, and then fed into the CNN model. Model training was performed on a dataset of 8400 Arabic script letters, where 80% of the dataset was used for training and 20% for testing.

The VGG-16 architectural model reached the highest rate of 83.99% in the recognition of Arabic handwriting patterns.

The research attained an accuracy of 83.99% utilizing the VGG-16 model, succeeded by AlexNet, GoogleNet, LeNet-5, and ResNet50 in terms of performance.

13- Improving Accuracy and Explainability of Online Handwriting Recognition

The manuscript describes the development of the handwriting recognition algorithms using the OnHW-chars dataset, achieving improvements in accuracy up to 23.56% for machine learning (ML) methods and 7.01% for deep learning (DL) methods, including ensemble techniques and model explainability studies. This study is aimed at improving the accuracy of the handwriting recognition system, providing interpretability to the algorithms, and making the results reproducible

and verifiable. The methodological approach followed in the current study includes:

Feature extraction using tsfresh and FRESH algorithms

Training machine learning models using Decision Tree, Random Forest, Extra Trees, Logistic Regression, k-Nearest Neighbours and Support Vector Machines

Training deep learning models using FCN, ResNet, LSTM, BiLSTM, InceptionTime, XceptionTime, and XCM architectures

Results

Study results in improvements by 11.3–23.56% for machine learning models and 3.08–7.01% for deep learning models using ensemble learning.

The results are reproducible and verifiable using the publicly available repository.

The results indicate the optimized ML and DL models outperform the previously known models by enhancing accuracy in the range of up to 23.56% and 7.01%, respectively.

14- An online cursive handwritten medical words recognition system for busy doctors in developing countries for ensuring efficient healthcare service delivery

This research has been carried out to overcome the problems related to unreadable handwritten medical prescriptions, which are still widespread in Bangladesh, as 97% of doctors still use handwritten ones, often in a mixed language. The output obtained from such handwritten prescriptions leads to many medication-dispensing errors due to misinterpretations.

The main goal of this study is to develop a machine learning approach toward the recognition of physicians' handwriting and its conversion to readable digital prescriptions, especially for developing countries where most of the prescriptions are handwritten and practically illegible.

It involves collecting handwritten data from 39 healthcare professionals, creating a 'Handwritten Medical Term Corpus' dataset, applying data augmentation techniques such as Rotate, Shift, and Stretch to generate multitudes of handwriting

variations, and using a Bidirectional LSTM model for the prediction of handwritten words. Some of the methods used in this research include data collection from a remote healthcare prescription database, data preprocessing, and data augmentation with the RSS technique.

This paper applies Bidirectional LSTM and a novel data augmentation method, Rotate + Shift + Stretch (RSS), to extend the dataset to 1,591,100 samples.

The handwritten recognition technology proposed in this paper achieved an average accuracy of 93.0% with Bidirectional LSTM and RSS data augmentation, 19.6% higher than that without data expansion.

Based on the above process, this paper presents two contributions. Firstly, it builds a handwritten medical term corpus including 480 medical-related words and 17,431 handwriting instances, and secondly, it proposes an RSS data augmentation technique to increase the number of data points. The experimental results demonstrate that the proposed system can achieve an average accuracy of 93.0%, a maximum of 94.5%, and a minimum of 92.1%, outperforming the no-data-expansion case by 19.6%.

15- Handwritten English Word Recognition Using a Deep Learning Based Object Detection Architecture

The model presented in this work is an end-to-end lexicon-free cursive handwritten word recognition model using YOLOv3, whose task is simultaneous localization and identification of characters in a handwritten word image.

The development of an end-to-end lexicon-free handwritten cursive word recognition model that can perform localization and identification of characters simultaneously in a handwritten word image.

An end-to-end handwritten word recognition is integrated in proposed method based YOLOv3. The model performs the localization and identification character-based in handwritten word image.

The character spotting for handwritten word recognition in the proposed method happens using YOLOv3. The model is devised such that it performs detection and recognition of objects in it. The technique followed for filtering

bounding boxes with low object confidence scores and then performs non maximum suppression to get rid of duplicate bounding boxes.

Segmentation and recognition of characters from handwritten word images are done with the YOLOv3 model. The model is trained to operate on a small number of ground truth images.

The method has yielded 29.21 WER and 9.53 CER on the IAM datasets outpacing some state-of-the-art methods.

The proposed method achieves 29.21% WER and 9.53% CER on the IAM dataset. It works well with a minimal number of training samples and reduces the cost of training.

The proposed model has produced competitive results concerning the other state-of-the-art methods on which they are evaluated on the standard IAM dataset.

16- Comparative Analysis of Handwritten Digit Recognition Using Logistic Regression, SVM, KNN and CNN Algorithms

This research study compares the performance of four machine learning algorithms-logistic regression, support vector machine (SVM), K-nearest neighbor (KNN), and convolutional neural network (CNN)-for handwritten digit recognition based on the MNIST dataset.

Key findings from the study are as follow:

CNN outperforms any other machine learning algorithms on the following measures and scores: best in terms of accuracy, precision, recall, and f1 score.

Although KNN is better in model accuracy, SVM and logistic regression were not up to those levels.

CNN finally comes out of being the slowest training model among the other machine learning models but its test accuracy turns out to be much better.

Thus, we conclude that CNN is suited best for all types of prediction problems including image data as an input by comparing the execution time of the algorithms, we come to the conclusion that stacking epochs after a certain number doesn't make sense without a change in the configuration of the algorithm since the scope of a particular model declares its limitation.

The methods which are employed in the study are:

Four machine learning algorithms are used for handwritten digit recognition namely, logistic regression, SVM, KNN, and CNN.

The MNIST dataset is used for training and testing the models.

The performance of the models is evaluated in terms of accuracy, precision, recall, and f1 score.

The results of the study are as follows: CNN is Generally achieving the highest accuracy of 99% of all models.

KNN made the model more accurate than SVM and logistic regression.

Models accuracy: CNN (99%), KNN (96.8%), SVM (91.3%), and logistic regression (91.7%).

17- Automatic handwriter identification using advanced machine learning

A hybrid method utilizing Oriented Basic Image feature extraction and graphemes codebook technique is proposed under the context of novel writer identification. By developing a comparative study, the approach is compared against machine and deep learning models in performance metrics.

The research also introduces CNN networks to extract features for handwriters images to achieve better recognition results alongside using Oriented Basic image features with grapheme codebooks.

It was noted that the writer identification system based on KPCA and OBI features outperforms other methods while algorithms that are based on deep learning technologies such as CNN (Alex-net model) demonstrated better performance metrics than traditional algorithms.

These findings include:

The text line extraction method proposed can be effectively used on multiple datasets with the highest obtained value of 99.80% on the AUB dataset.

The application of writer identification utilizing the feature of OBI coupled with graphemes codebook is reported to have improved performance over similar techniques.

The study incorporates a variety of machine learning techniques such as statistical approaches and model based approaches such as Orientation Basic Image features, graphemes codebook, and Convolutional Neural Networks (CNNs) as tools for features extraction and also for handwriting identification.

Methods such as OBI feature extraction, graphemes codebook method, and KPCA for dimensionality reduction. Machine Learning and Deep learning algorithms have been considered here: SVM, KNN, and CNN (Alex-net model).

Bilateral filter bass line extraction.

OBI features extraction and grapheme codebook methods for author identification.

Deep learning (convolutional neural networks) concept regarding author identification.

Non-linear dimensionality techniques include using KPCA, Isomap, LLE, Hessian LLE, and Laplacian Eigenmaps methodologies for feature reduction.

This research uses the Alex-Net model and computes the features from intermingled life layers and analyses their discrimination power. The features will then be passed to an SVM for effective author recognition.

The study introduces several contributions to the already existing body of knowledge in the following ways: development of new algorithms for new text line segmentation and detection; evaluation of different machine learning approaches in studying handwriting identification.

The results of the analyses have shown that the proposed writer identification system made use of KPCA and OBI features, which proves better than other methods, while deep learning architectures such as CNN (Alex-net model) are found to exhibit better performance in comparison to traditional machine learning algorithms.

Results of study would provide:

-Having high performance of proposed algorithm for text line extraction related to various databases.

-Improved performance identification of writer entries using OBI feature extraction with graphemes codebook methods.

-Superior results of the deep learning (CNN) concept for writer identification: the presented model compares well with state-of-the-art methods in writer identification, and results indicate more efficacy of deep CNNs in extracting image features.

18- Handwriting recognition by using deep learning to extract meaningful features

Presented in the study is an enhanced Handwriting Recognition (HWR) system by using deep learning techniques, particularly Convolutional Neural Networks (CNNs), to abstract features from raw data, and thus improving recognition rates.

The new recognition system that is inspired by Hidden Markov Models combined with Neural Networks, has made a significant gain in the performance when used deep neural networks in text line images.

The paper concludes that CNNs definitely play a vital role in the task of HWR systems. To obtain the best possible results, the researchers employed a basic net with one convolution layer and six kernels. Through the use of dropout and max-pooling layers of course, the performance is also going to be increased.

The use of CNN models to describe a raw input that higher preference is given to the recognition rates of the process, the use of the approach in this study makes a significant step forward the conventional baseline system.

The results on the IAM Database and also those obtained by other authors confirm this strategy.

The study advocates the implementation of deep neural networks, especially deep Multilayer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs), to extract the necessary features from a raw text image. The authors, apart from that, experiment with the 2D convolutions combined with a pooling layer model and higher convolutions to find upgraded features.

The study presents a hybrid HMM/ANN system as a baseline and examines the effectiveness of different CNN options and configurations: LeNet-5 (Lenet-5), Adhoc CNN, Vertical CNN. These models are trained and tested using the April-ANN toolkit.

The study implies the usage of deep learning, in particular, CNNs, to extract features from the raw input data and they even compare different CNN topologies.

The results of the experiments indicate a big progress in the case when it comes to utilizing safe learning with divining meaningful features of line text. The data demonstrates that the use of CNNs increases the HWR systems' performance, with the top ones being simple networks with one convolution layer and six kernels. The use of dropout and max-pooling layers also significantly boosts the result of the models.

The suggested method brings down the Word Error Rate (WER) to 17.2, which is significant when the starting value of 19.0 is taken into account.

19- Handwritten Digit Recognition Using Machine Learning Algorithms

This paper advocates for a certain recognition of offline handwritten digits based on various machine learning methods with the sole objective of strengthening the effectiveness and reliability of recognizing handwritten digits.

This paper aims to work on the optimization of different machine learning algorithms in terms of performance in recognition of handwritten digits.

For digit recognition using Waikato Environment for Knowledge Analysis (WEKA), the various algorithms within machine learning for classification include Multilayer Perceptron, Support Vector Machine, Naïve Bayes, Bayes Net, Random Forest, J48, and Random Tree.

Different algorithms have been employed for this study, including Multilayer Perceptron, Support Vector Machine, Random Forest, Bayes Net, Naïve Bayes, J48 Decision Tree, and Random Tree, applied here to classify handwritten digits at WEKA software. The final conclusions will be drawn with respect to parameters such as accuracy, time taken, and error rates of the experiment.

As for the results of the experiment, it is noteworthy that the greatest accuracy gained was that of the Multilayer Perceptron, hence achieving 90.37 percent.

Among all the outputs obtained from the experimental report, the greatest accuracy, which was obtained from the Multilayer Perceptron algorithm, was noted to be 90.37 percent followed by the other algorithms such as Support Vector Machine, Random Forest Algorithm, Bayes Net, Naive Bayes, j48 Decision Tree, and Random Tree. Further, the results have been put in table format, which

correspond to the form of accuracy and time taken, along with the error rates for each algorithm.

20-Handwritten English Character Recognition using Multilayer Perceptron Neural Network

An online handwritten character recognition system for English letters is pioneered in this paper, employing a multilayer perceptron neural network, to realize an improvement in accuracy.

The proposed system achieves the rate of recognition closer to 70% of handwritten English letters.

Neural Networks as the characters are expected to be recognized, it can tolerate the noise best among all the other techniques.

Without proper pre-processing the extracted features are of low quality.

This is the paper that proposes an offline handwritten English character recognition system with enhanced accuracy using the Multilayer Perceptron Neural Network.

Preprocessing removes noises and smoothens the images.

The proposed system follows the stages: acquisition of the images, pre-processing, feature extraction, segmentation and classification using a multilayer perceptron neural network.

The recognition accuracy is above 70% for handwritten English characters in the proposed system.

Comparative Analysis of Handwriting Recognition Models and Results:

Year	Author	Approach	Dataset	Result	Field
2024	Eman Ahmed Khorsheed Ahmed Khorsheed Al-Sulaifanie	Deep CNN with varying filter sizes	MNIST	99.5% accuracy	Handwritten Digit Recognition
2024	Sukhdeep Singh, Sudhir Rohilla, Anuj Sharma	Literature review on CNN and RNN	Various datasets	99.74% accuracy (CNN-based architecture)	Handwriting Recognition Techniques
2024	S. M. Shamim, Mohammad Badrul Alam Miah, Angona Sarker, Masud Rana, Abdullah Al Jobair	ML models (SVM, Random Forest, etc.)	WEKA experiments	90.37% accuracy (Multilayer Perceptron)	Digit Recognition
2024	Ravikumar Chinthaginjala, C. Dhanamjayulu, Tai-hoon Kim, Suhaib Ahmed, Si-Yeong Kim,	Gated mechanisms (Gate-CRNN)	Washington, Bentham, RIMES, Saint Gall, and IAM	94.26% accuracy	Text Recognition

	A. S. Kumar, Visalakshi Annepu & Shafiq Ahmad				
2024	Mohammed Hamdan, Abderrahmane Rahiche, Mohamed Cheriet	HTR-JAND with knowledge distillation	IAM, RIMES, Bentham	State-of-the-art CER	Historical Text Recognition
2024	Usman Ali, Sahil Ranmbail, Muhammad Nadeem, Hamid Ishfaq, Muhammad Umer Ramzan, Waqas Ali	Hybrid Mask R-CNN and TrOCR	Prescription datasets	1.4% CER	Medical Text Extraction
2024	Zi-Rui Wang	3D attention module with CNN	the SCUT-HCCDoc and the SCUT-EPT the IAM	Comparable to state-of-the-art	Multilingual Handwriting Recognition
2023	Gibran Satya Nugraha Muhammad Ilham Darmawan Ramaditya Dwiyansaputra	CNN architectures (VGG-16, AlexNet, etc.)	8400 images of Arabic handwritten characters	83.99% accuracy (VGG-16)	Arabic Handwriting Pattern Recognition

2023	Maryam Yaseen Abdullah	CNNs (AlexNet, SqueezeNet)	400,000 handwritten names	99.8% accuracy (SqueezeNet)	Real-Time Handwriting Recognition
2023	al. Tamanna Sachdeva	SVM, MLP, CNN comparative study	MNIST	CNN: 99% accuracy	Digit Recognition
2023	Dhruv Shah, Sai Bhargav, Dhanush B.	CNN for feature extraction	IAM database	WER reduced to 17.2	Handwriting Recognition
2023	Firat Kizilirmak, Berrin Yanikoglu	KPCA and OBI for feature extraction	IAM Dataset	99.8% recognition	Writer Identification

2022	Shaira Tabassum, Nuren Abedin, Md Mahmudur Rahman, Md Moshiur Rahman, Mostafa Taufiq Ahmed, Rafiqul Islam & Ashir Ahmed	Bidirectional LSTM with RSS data augmentation	Handwritten Medical Term Corpus	93.0% accuracy, max 94.5%	Medical Handwriting Recognition
2022	Hilda Azimi, Steven Chang, Jonathan Gold, Koray Karabina	ML (Decision Tree, SVM, etc.) and DL (FCN, ResNet)	OnHW-chars	23.56% ML improvement, 7.01% DL improvement	Handwriting Recognition Accuracy
2022	Felix Ott, David Rügamer, Lucas Heublein, Bernd Bischl, Christopher Mutschler	Cross-modal representation with triplet loss	Offline and Online HWR datasets	96.76% accuracy, better generalizability	Online Handwriting Recognition
2022	Riktim Mondal,Samir Malakar,Elisa H. Barney Smith,Ram Sarkar	MLP for character recognition	English letters	~70% accuracy	English Character Recognition
2021	Dr. R. Pradeep Kumar Reddy,C.	YOLOv3 for word	IAM Dataset	29.21% WER,	

	Naga Raju	recognition		9.53% CER	Word Recognition
2019	Joan Pastor Pellicer,Maria Jose Castro-Bleda,Salvador España Boquera,Francisco Julián Zamora-Martinez	CNN with meaningful feature extraction	IAM Database	WER reduced to 17.2	Handwriting Recognition
2019	Amal Durou	Comparative study of ML and DL approaches	IAM dataset ICFHR 2012 dataset	Superior deep learning performance	Writer Identification
2017	Ms Harita Dave,Prof A / Mitesh Patel	Multilayer Perceptron for handwriting	English letters	~70% accuracy	English character Recognition

Comparative Analysis

The comparative analysis of the handwriting recognition techniques reviewed in this study indicates that deep learning approaches are plentiful and there are a lot of different techniques as most are specific to certain use cases and varying degrees of order and complexity of writing. Overall, the convolutional neural network (CNNs)—specifically, the abundance of architecture and technique implementations and combinations (i.e., VGG16, ResNet, SqueezeNet, etc.,)—demonstrated good performance in character and word recognition, across multiple languages and datasets with many reporting greater than 99% accuracy on many

datasets from MNIST and IAM. The strength of the CNNs can be attributed to the strength of their hierarchical feature extractor part of the model. Depending on the dataset characteristics, the CNNs did an excellent job recognizing structured handwriting with less noise. It is also important to note that they don't take into account sequential characteristics of handwriting (the horizontal plane only), or visibility variability that accompanies a complex value in complex writing.

Moreover, models such as CRNNs, gated CNN-BGRU hybrids, or attention-based architectures (i.e., TrOCR, Vision Transformers) that declared the order and hierarchy of defined characters further improved both segmentation and recognition by also capturing sequential dependencies, unique styles across multiparts and writing spaces, long range contextualization factors, etc. Several state-of-the-art performance accounts proved these sophisticated models didn't have value limits of writing datasets domestically and internationally on complex datasets when considering historical manuscripts and multimodal or multilingual corpora. Clearly, one area of potential improvement is how complex the models may be and the associated computational costs present challenges to virally scalability in real-world environments and resource-constrained contexts of writing tasks.

Hybrid systems that combine the strengths of Mask R-CNN and transformers have proven to achieve remarkable levels of accuracy, particularly on layouts-heavy documents and medical prescriptions, having achieved Character Error Rates (CER) of 1.4%. However, these kinds of models require large, labeled datasets, and very complicated training pipelines.

Traditional OCR systems, like Tesseract and object detection-based systems, like YOLOv3, have generally not performed well at recognizing handwriting and have struggled even more, when it comes to handwritten, cursive, and/or stylized text.

Position of This Thesis

This thesis draws upon the innovative simplicity and effectiveness of VGG16, a CNN model that has previously shown strong performance in various handwriting

recognition scenarios. VGG16, while not new and not even the most complex architecture, can effectively balance accuracy and compute efficiency against deployment considerations. Both the feature extraction capability already proven by VGG16, and its layout being distinct and modular, make it much more digestible for even the most novice developers of tools to apply in real-world applications. In this case when effective interpretability and low latency inference are paramount in a highly regulated domain such as healthcare.

In this work, rather than fully adopting VGG16 as-is, I fine-tuned its parameters to better align with the specific task of recognizing handwritten words in medical prescriptions. This involved customizing the model's configuration and training process, while keeping the core architecture intact. The prescription images were processed through a custom pipeline that segments them into lines and then into individual word images. Performance improvements were achieved through task-specific pre-processing, domain-aware data augmentation, and strategies to handle handwriting variability and class imbalance in real-world datasets.

While transformer-based and hybrid models may technically provide improvements with respect to accuracy, I was more interested in a solution that allows for scalability and real-world practicality in the clinical implementation setting - thinking of this for deployment efficiency and utilizing existing and not specializing hardware and not undertaking large scale consumer annotated data sets.

Conclusion

In summary, Chapter Two has analyzed many handwriting recognition models, from simple CNN models to transformer models. Each type offered different advantages in terms of performance, robustness, or flexibility; however, for the purposes of recognizing prescriptions, VGG16 has emerged as a strong and viable choice as it is a simple to use model, produces strong performance results, and can be integrated into a web API.

This comparison found that while hybrid models and attention models are promising in theory, the complexity of deploying these models exceeds the benefits when working in environments that are constrained by real world problems. The focus of this thesis is on a VGG16 based handwriting recognition

model that can extract and classify handwritten medicine names from prescriptions. Focusing on this model strikes a balance between modern deep learning capabilities while managing real world constraints, which is a good foundation to develop a practical, reliable and efficient doctor's handwriting recognition system.

Chapter 03 :

Implementation and validation

I. Introduction:

This chapter outlines the complete implementation pipeline for recognizing handwritten medication names from medical prescriptions, emphasizing data preparation and integration with deep learning models. Using the Doctors' Handwritten Prescription BD Dataset from Kaggle, the process combines classical OCR techniques with CNN architectures to manage the complexities of diverse and often illegible handwriting. Key steps include image preprocessing, word segmentation, and training a transfer learning-based VGG16 model, with comparisons to ResNet-50 and Inception-v3. The system is deployed as a modular Flask web application, showcasing its practical potential in real-world medical digitization tasks.

II. Dataset

1. Significance of Good Quality Datasets for Handwriting Recognition

In handwriting recognition applications, the quality and characteristics of datasets are instrumental in assessing the performance and generalizability of machine learning algorithms. High-quality datasets allow the model to capture strong features that can explain the variability inherent in handwriting styles, in addition to the distortions and noise prevalent in real-world situations. Especially for medical prescriptions, in which handwritten content tends to be unclear and greatly varies, a robust dataset that provides varied handwriting samples, clean labels, and realistic settings is critical. In the absence of these data, the risk is that models overfit to limited handwriting styles or cannot generalize, thereby making crucial errors in prescription reading a reality with potential impacts on patient safety and healthcare outcomes.

2. Characterization of the Utilized Dataset

The dataset utilized for this research is known as the **"Doctors' Handwritten Prescription BD Dataset"**, and it is publicly available on the Kaggle platform. This dataset was specifically developed to address the challenges associated with

handwritten medical prescriptions in Bangladesh, such as regional handwriting styles and common medical terminology.

- **Source**: https://shorturl.at/6ZBPZ
- **File Format**: The dataset primarily consists of image files in PNG format, supported by corresponding annotation files that provide labels for the handwritten text.
- **Number of Samples**: The dataset includes a few hundred unique images of handwritten prescriptions. Each sample is a distinct image featuring the handwriting of various doctors.
- **Types of Data**: The dataset comprises raw images of prescriptions along with corresponding textual labels, such as the names of prescribed medications. This combination supports supervised learning tasks for handwriting recognition.

This data collection is particularly valuable as it reflects real-world medical handwriting in a developing world context, where legibility issues in prescriptions are a frequent concern.

3. Preprocessing Steps

For this research, the dataset was used **as-is**, without performing any significant preprocessing steps. The raw images and their corresponding annotations were directly fed into the model training pipeline. This decision was made to assess the model's ability to handle real-world, noisy, and varied handwritten prescriptions in their natural form without artificial enhancements or filtering.

By using the dataset in its original state, the goal was to simulate realistic usage scenarios where prescriptions may not always be clean, well-scanned, or

uniformly formatted. This approach helps in testing the model's robustness and adaptability to actual clinical environments, where data preprocessing may not always be feasible.

4. **Challenges Faced with the Dataset**

Several challenges were encountered during the utilization of this dataset:

- **Variation in Handwriting Patterns**: Prescriptions came from different doctors, each with distinct handwriting, cursive styles, and abbreviations, making consistent recognition difficult.
- **Unclear Scans and Noise**: Some images suffered from poor lighting, low resolution, and background artifacts, complicating the segmentation and recognition process.
- **Specialized Medical Terminology**: The use of Latin abbreviations and complex drug names increased the difficulty of annotation and interpretation.
- **Class Imbalance**: Certain medication names appeared much more frequently than others, leading to imbalanced training data.
- **Ambiguity in Handwritten Characters**: Overlapping strokes and similarly shaped characters in cursive handwriting contributed to a higher error rate in character recognition.

Overcoming these challenges required thorough preprocessing, targeted data augmentation, and careful tuning of the recognition model architecture.

5. **Dataset Support for Model Training and Real-World Relevance**

The selected dataset plays a crucial role in developing robust machine learning models for recognizing handwritten medical prescriptions. By offering

realistic and varied examples, it enables the model to learn and generalize across different handwriting styles, noise levels, and medical jargon found in actual clinical settings.

Using this dataset, the model gains the ability to interpret poorly written prescriptions and handle specialized terminology, thereby improving its practical usability in healthcare environments. Ultimately, this reduces the risk of medication errors caused by illegible handwriting and contributes to better patient safety and healthcare delivery.

III. Text extraction:

The performance of any deep learning model, especially with respect to image classification problems, is highly dependent on the input data and its quality and organization. In the case of recognizing doctors' handwriting from medical prescriptions, a complete and carefully planned data preparation pipeline would be extremely useful. Given the highly unstructured, noisy and arbitrary data style of handwritten prescriptions, it is important to process the data in multiple stages before the data can be put into a convolutional neural network (CNN) such as VGG16.

This chapter describes the entire data preparation pipeline used in this thesis. The data preparation pipeline encompasses the uploading of handwritten prescription images, segmentation into text lines, detection of words for latter lines, and saving the words for the building and inference of the model. Each step of the pipeline is justified using technical reasoning and realised with image processing methods, as well as developed modules that are designed through established methods in optical character recognition (OCR) and document image analysis.

1. Capturing Prescription Images:

The pipeline starts by acquiring handwritten medical prescriptions. The prescriptions are obtained as scanned or photographic, image data in .jpg or .png

formats. As handwritten prescriptions vary so much in terms of resolution, orientation, lighting conditions, and background noise, this variability creates challenges for reliable preprocessing and word recognition.

The pipeline must adapt to such variation, and thus needs to be able to process images of any size and any quality, using adaptive thresholding and morphological operations. All uploaded images are stored in a directory (static/uploads/) for the next processing phase.

2.1 Word Detection Based on Text Lines:

To prepare individual word images for training the VGG16 classifier, each handwritten text line is first normalized to a fixed height (50 pixels) while maintaining aspect ratio, ensuring consistent input size for CNNs. We use an anisotropic filtering approach based on the method by Manmatha et al. to detect word blobs within a line. This involves convolving the grayscale image with a directional Gaussian kernel tuned to the average word aspect ratio. The resulting image is binarized using Otsu's method to enhance word regions. Contours are then extracted from the binary image using OpenCV, and bounding boxes are generated for each valid contour—excluding noise by filtering out areas smaller than 100 pixels. These bounding boxes are used to crop and extract individual word images from the normalized line.

 GitHub Implementation: https://github.com/abdelbassetbenatmani/line-segmentation-and-text-extraction

2.2 Data Organization:

The resulting dataset contains images contained in multiple structured directories:

static/uploads/: Original uploaded prescription image files.

static/lines/: segmented text lines from prescriptions.

static/words/: cropped word images used for training and inference.

2. Flask Web Application (app.py):

The application itself is designed using Flask and it provides functionality to upload an image, perform line segmentation, and detect words, all from a web interface.

File Upload: The base route / accepts image files using POST request. When an image file is uploaded it uses werkzeug.utils.secure_filename and saves the file securely to the server.

Pre-processing: Each image is represented as a grayscale image and then binarized using adaptive threshold. Append and morphological operations are performed to convert the images into usable line regions using horizontal projection profiles.

Line Segmentation: From the identified lines are extracted with pading and saved in the static/lines directory.

Word Detection within each line: Each line is then passed through a custom detector that uses a scale-space filter to identify and extract words.

Rendering: The lines and words extracted are rendered from the Flask application dynamically, using Jinja templates (app.html).

Folder Management: Uploaded images, segmented lines and detected words are all clearly separated and stored in static/uploads, static/lines, and static/words.

3.1 Word Detection Algorithm (detector.py)

The word segmentation module implements a method outlined by R. Manmatha, which is based on the anisotropic filter and the connected component analysis.

DetectorRes and BBox are custom dataclasses for holding representations of detected word images and instances of bounding boxes.

Filtering: A gaussian kernel is calculated with horizontal distortion based on the shapes of words in the region. The kernel is then used to smooth the image before Otsu's thresholding is implemented.

Contour Detection: Connected components are found using cv2.findContours with any small region (less than min_area) being eliminated.

Kernel Calculation: The kernel is calculated with input parameters; sigma and theta that control the sensitivity to shape and orientation.

Image Normalization: The prepare_img function makes sure that all heights of the input images will be standard heights which yields better consistency in detection.

3.2 Line Clustering and Sorting

Although not used in the main process, the detector has capabilities for clustering and sorting words into lines:

_cluster_lines: Groups detected boxes from detected words into horizontal lines using DBSCAN clustering on the Jaccard distance.

sort_multiline: Sorts detected words in order by line according to their x-coordinate in a way that mimics a normal reading order.

sort_line: Sorts a single line of word detections in the horizontal axis.

4. Key Modules Used

4.1- OpenCV: Comprehensive Computer Vision Solutions for Image Analysis

Overview

As a premier open-source computer vision library, OpenCV delivers robust capabilities for processing visual data, featuring advanced functions for image manipulation, edge detection, and binarization techniques. Its extensive algorithmic toolkit enables sophisticated analysis of graphical information, proving indispensable for applications ranging from text recognition to intelligent automation systems.

Core Functionalities:

1. Advanced Image Processing

The library excels in multiple image transformation operations including:

- Noise reduction and geometric modifications
- Chromatic space adaptations
- High-performance parallel processing leveraging modern hardware

These features serve as critical preprocessing steps for enhanced analytical accuracy in time-sensitive implementations like traffic monitoring and robotic navigation (Guillen, 2019; Gouri & Kumar, 2023).

2. Boundary Identification

The system's edge detection prowess facilitates:

- Precise object delineation
- Support for complex image partitioning
- Exceptional performance (94.3% accuracy) in challenging scenarios such as handwritten formula interpretation (Sakshi & Kukreja, 2022; "Advanced Contour Analysis for Mathematical Expressions", 2022).

3. Image Binarization Methods

The toolkit provides multiple segmentation approaches:

- Absolute threshold conversion
- Context-aware adaptive binarization
- Statistical Otsu's method

These techniques significantly improve text legibility in captured images, optimizing optical character recognition processes ("Advanced Character Isolation Techniques", 2023).

implementation Considerations:

While exceptionally capable, practitioners should account for:

- Performance in cluttered visual fields
- Inconsistent illumination scenarios
- Precision limitations in fully automated workflows

4.2- NumPy: The Foundation of Numerical Computing in Python

Overview NumPy stands as a fundamental Python library for scientific computing, offering powerful tools for numerical data processing and analysis. At its core lies the N-dimensional array (ndarray) - an efficient data structure that enables sophisticated mathematical operations. As a cornerstone of Python's scientific ecosystem, NumPy seamlessly integrates with other key libraries like SciPy and pandas, significantly expanding its analytical capabilities.

Key Features

1. Multidimensional Arrays
    - The ndarray structure provides MATLAB-like functionality for handling large datasets (Danial, 2022)
    - Enables efficient storage and manipulation of numerical data in various dimensions
2. Advanced Mathematical Operations
    - Offers comprehensive mathematical functions for array and matrix computations ("Empowering Scientific Computing", 2023)
    - Supports vectorized operations for improved performance
3. Cross-Language Compatibility
    - Features interfaces with C/C++ for enhanced computational efficiency ("Empowering Scientific Computing", 2023)
    - Facilitates integration with high-performance computing environments

Practical Applications

1. Scientific Research
    - Plays vital role in physics, chemistry, and astronomy research (Harris et al., 2020)
    - Contributed to groundbreaking discoveries like gravitational wave detection
2. Data Analysis
    - Forms the computational backbone for pandas and other data science tools (Nelli, 2023, 2015)
    - Addresses performance limitations of native Python data structures

Considerations While exceptionally powerful, some users note:

- Steeper learning curve for programming beginners
- Potential challenges in educational adoption contexts

4.3- Scikit-learn (sklearn): A Powerful Machine Learning Toolkit for Text Line Clustering Using DBSCAN**

Overview

Scikit-learn is a popular open-source Python library that provides efficient tools for machine learning and data analysis. Among its many capabilities, the library includes the **DBSCAN algorithm**, which proves particularly valuable for organizing scattered text elements into coherent lines through advanced clustering techniques.

1. Core Functionalities Advanced Clustering with DBSCAN

- The library offers a well-optimized implementation** of DBSCAN, a sophisticated clustering method that groups data points based on density (Bisong, 2019; Pedregosa et al., 2011).
- Unlike traditional clustering approaches, DBSCAN automatically determines cluster formations without requiring predefined cluster counts, making it exceptionally suitable for text organization tasks (Thom & Kramer, 2010).

2. Key Benefits for Text Processing

- **Effective noise handling**: Naturally filters out irrelevant elements and outliers in documents.
- Shape flexibility: Capable of detecting text lines with varying orientations and spacing.

3. Optimized Performance

- Efficient computation: Enhanced search strategies minimize unnecessary calculations, boosting processing speed (Thom & Kramer, 2010).
- Algorithm improvements:

- **Statistical refinement**: Incorporates mathematical adjustments to produce cleaner cluster separations.

- Dynamic adaptation: Modified versions demonstrate superior performance in text-based applications (Jin-sheng, 2011).

Practical Implementations

- Digital document analysis: Reconstructing text structure from OCR outputs.
- Historical document processing: Organizing irregular handwritten content.
- Data cleaning: Isolating meaningful textual components from complex backgrounds.

Considerations

- Parameter dependency: Optimal performance requires careful adjustment of key variables.
- Large-scale data: May require supplementary techniques for handling extensive datasets efficiently.

4.4- Matplotlib: Python's Premier Data Visualization Toolkit

Overview As Python's foundational plotting library, Matplotlib provides researchers and developers with powerful tools to visualize text detection results and spatial data relationships. Its versatile functionality makes it indispensable for displaying OCR outputs, particularly when working with bounding box annotations in document analysis and computer vision applications.

Key Functionalities

1. Advanced Plotting System
    - Generates multiple visualization types including:
        - Precision scatter plots for coordinate mapping
        - Density heatmaps for text cluster analysis
        - Custom annotation overlays (Mujilahwati, 2021; Kaestria et al., 2024)
2. Document Analysis Features
    - Specialized functions for:
        - Bounding box rendering with adjustable styles
        - Multi-layer image composition

- Text label positioning (Semenov & Shishkin, 2024)

Enhanced Workflow Integration :

1. Accessible Interfaces
    - Extension tools like AvoPlot offer:
        - Drag-and-drop annotation capabilities
        - Streamlined data pipeline integration (Peters, 2014)

Academic and Industrial Applications

1. Research Visualization
    - Enables visual validation of:
        - Document layout algorithms
        - Text recognition accuracy
        - Spatial distribution patterns (Kaestria et al., 2024)

Alternative Considerations For specialized use cases: Seaborn: Statistical-focused visualizations Plotly: Web-interactive dashboards

Implementation Advantages Pixel-perfect output control Cross-platform compatibility Extensive customization options

Usage Notes Requires matplotlib.pyplot import convention Object-oriented API preferred for complex figures Style configurations available for publication needs

This refined version:

1. Uses more technical yet concise language
2. Groups related functionalities thematically
3. Maintains all original references
4. Adds practical implementation details
5. Improves flow between sections

4.5- argparse: Flexible Command-Line Configuration for Text Processing Systems

Overview As Python's built-in argument parsing solution, argparse provides developers with powerful tools to create customizable command-line interfaces for text analysis applications. The module excels at managing runtime parameters for word detection and text processing algorithms through terminal commands.

Key Features

1. Intelligent Argument Handling Implements robust parsing mechanisms that:

- Validate and convert input types
- Generate automatic help documentation
- Support complex command hierarchies Reduces interface development time significantly

2. Algorithm Customization Enables dynamic control of critical parameters including:

- Detection sensitivity thresholds
- Processing model selection
- Output detail levels (Hiroshi, 2019)

System Integration

1. Real-Time Adjustment Facilitates on-the-fly modification of:

- Recognition accuracy settings
- Language-specific configurations
- Debugging verbosity (Mengge et al., 2020)

2. Multilingual Support Simplifies management of:

- Locale-dependent processing rules
- Character encoding specifications
- Dictionary customization (Rumin et al., 2015)

Implementation Notes

Advantages Minimal performance overhead Seamless Python ecosystem integration Cross-platform consistency.

Considerations Requires basic command-line familiarity Limited graphical interface options Steeper learning curve for non-technical users.

Practical Applications Automated document analysis pipelines Speech-to-text conversion systems Cross-language text mining Batch text processing workflows.

This refined version:

1. Uses more natural technical language
2. Groups related features thematically
3. Maintains all key technical details
4. Preserves academic references
5. Improves readability through concise phrasing

4.6- Python Visualization Tools for Machine Learning Analysis

Matplotlib and Seaborn serve as powerful visualization tools in Python, particularly valuable for analyzing and interpreting machine learning model performance. These libraries transform complex numerical metrics into intuitive graphical representations, enabling data scientists to effectively evaluate their models.

Core Visualization Strengths

Matplotlib's Comprehensive Features

- Provides foundational plotting capabilities for creating basic to advanced visualizations
- Enables generation of multiple chart types including progress curves and distribution plots
- Allows pixel-level customization for publication-ready figures (Lavanya et al., 2023)

Seaborn's Statistical Enhancements

- Offers simplified syntax for complex statistical visualizations

- Includes built-in styling options for professional-looking outputs
- Specializes in multivariate data representation and categorical plotting (Oberoi & Chauhan, 2019)

Model Evaluation Applications

Training Process Visualization

- Tracks metric evolution across training iterations
- Highlights potential model issues through trend analysis
- Enables comparative analysis of different model configurations

Performance Assessment Tools

- Generates detailed confusion matrices with Seaborn's heatmap
- Creates precision-recall curves for classification evaluation
- Visualizes class-specific performance metrics (Lavanya et al., 2023; Oberoi & Chauhan, 2019)

Implementation Considerations

While exceptionally capable, these tools present certain considerations:

- Require programming proficiency for full utilization
- Advanced customizations demand deeper technical knowledge
- GUI-based alternatives may better suit non-technical users

4.7- Flask:

Flask is a flexible and minimalist web framework designed for Python, enabling developers to build dynamic web applications with ease. Known for its simplicity and modularity, Flask provides essential tools for routing, request handling, and template rendering while allowing developers to add extensions for additional functionality. Its lightweight structure makes it ideal for small to medium-sized projects, including e-commerce platforms, where efficient data processing and user interaction are crucial (Izzathohir & Yulianton, 2024; Relan, 2019).

Unlike full-stack frameworks, Flask follows a "micro" approach, meaning it offers core features without imposing rigid project structures. This

design encourages customization, letting developers integrate only the components they need, such as database support (e.g., Flask-SQLAlchemy) or authentication (e.g., Flask-Login) (Grinberg, 2014). Despite its simplicity, Flask is powerful enough to handle complex applications when combined with the right extensions, making it a popular choice for modern web development (Hunt, 2023).

Key Improvements in the Paraphrased Version:

1. Simplified Language: Removed redundancy while keeping the original meaning.
2. Restructured for Clarity: Grouped related concepts (e.g., Flask's microframework nature and extensibility).
3. Added Flow: Connected ideas more naturally (e.g., "Unlike full-stack frameworks...").
4. Maintained References: Kept citations while integrating them smoothly into the text.

4.8- Pandas: Python's Essential Data Processing Library

Pandas stands as a fundamental Python package for efficient data handling, particularly when working with structured datasets like CSV files containing medical image paths and corresponding labels. Its comprehensive toolkit streamlines data preparation, transformation, and analysis, serving as a critical resource in data science and healthcare applications.

Core Data Management Functions

1. Data Loading
    - The library's **pd.read_csv() method provides simple yet powerful CSV file import capabilities.**
2. **Structured Data Representation**
    - **Utilizes DataFrames - intuitive tabular structures that enable organized data management (Nelli, 2023).**
3. **Data Quality Assurance**

- **Offers robust tools for addressing missing values, eliminating duplicates, and standardizing data formats (Harrison & Prentiss, 2016).**

Advanced Processing Features

- Targeted Data Extraction
  - Enables precise filtering operations, such as isolating specific medicine-related images (Molin, 2019).
- Data Summarization
  - Facilitates grouping and statistical analysis for categorical data examination (Harrison & Prentiss, 2016).
- Visual Analytics Support
  - Seamlessly integrates with visualization tools to transform raw data into meaningful insights ("Democratizing Data Visualization...", 2024).

Implementation Considerations

While Pandas delivers exceptional functionality, new users may require time to master its full potential. Fortunately, abundant learning resources exist to accelerate proficiency (Harrison & Prentiss, 2016).

5. Folder Structure Overview

project/

│

├── static/

│ ├── uploads/ # Uploaded input images

│ ├── lines/ # Segmented line images
│ └── words/ # Extracted word images
│
├── templates/
│ └── app.html # HTML template for web interface
│
├── app.py # Flask app and main pipeline
└── word_detector/
└── detector.py # Custom word detection logic

6. Workflow

- User uploads a prescription image using the web form.
- The image is segmented into lines via analysis of the projection profile.
- Each line is processed by a word detector based on Gaussian distributions.
- The word images are displayed in the browser and saved to disk.

IV. Data Preparation and Input Processing

1. Dataset Upload and Organization

The code begins by downloading a ZIP file containing a dataset of images of handwritten prescriptions. It is unzipped into separate folders for training, validation, and testing. Matching CSV files containing the image labels are also included to facilitate supervised learning.

2. Data Preprocessing and Enrichment

To enhance model generalization and robustness, training images undergo **data training** including random rotation, shifting, shearing, and zooming. All images are preprocessed using vgg16.preprocess_input, which applies the same color normalization used for VGG16's original training on ImageNet. Only the **validation and test sets** are preprocessed without augmentation to preserve evaluation integrity.

3. Loading Data through Generators

Images and labels are loaded using Keras' flow_from_dataframe, which reads image paths and labels from CSV files and generates batches of 224×224 pixel resized images the input size VGG16 expects.

Labels are **one-hot encoded** to support multi-class classification.

4. Feature Extraction Using a Pretrained VGG16 Model

4.1 Original VGG16 Architecture

VGG16 is a deep convolutional neural network consisting of **13 convolutional layers** and **3 fully connected layers**, originally applied to ImageNet classification. It uses **small 3×3 convolution filters** and **max-pooling layers** to progressively extract hierarchical features from input images.

4.2 Transfer Learning Configuration

The pretrained VGG16 model is loaded with **ImageNet weights**, excluding its top (fully connected) layers using include_top=False. The **convolutional base is frozen** (trainable=False) to retain general visual features and avoid overfitting, especially important given the smaller size of the prescription dataset.

5. Modified Classifier Head

5.1 Fully Connected Layers Customized

Instead of using VGG16’s original classifier, a custom classification head is added:

- A **Flatten** layer reshapes feature maps to a 1D vector. Two **Dense layers** with **1024 and 512 units** respectively, both using **ReLU** activation.
- **BatchNormalization** layers are applied after each Dense layer to stabilize and speed up training.
- **Dropout** layers (rate = 0.5) help reduce overfitting.
- The final **Dense layer** uses a **softmax activation** to output class probabilities for different medication types.

5.2 Reasons for Changes

Replacing the original VGG16 classifier with **deeper and regularized dense layers** allows the model to better capture domain-specific patterns.

Batch Normalization and Dropout improve **generalization** and **training stability**, which is critical in recognizing variable and messy medical handwriting.

6. Model Compilation and Training

6.1 Optimizer and Loss

The model is compiled using the **Adam optimizer** (learning rate = 0.001), known for its efficiency in gradient-based optimization. The **loss function** used is **categorical crossentropy**, appropriate for multi-class classification tasks.

6.2 Early Stopping Mechanism

Training incorporates **early stopping** to monitor validation loss and halt training if no improvement is seen over 3 consecutive epochs. This prevents overfitting and saves computational resources by restoring the best-performing weights.

6.3 Training Process

The model is trained on the **augmented training data** for up to **20 epochs**, with performance validated on the independent validation set.

Training metrics such as accuracy and loss are tracked throughout.

7. Representation and Assessment

Test Measurement and Accuracy

After training, the model is evaluated on the **unseen test set** to determine final accuracy and validate its generalizability.

Confusion Matrix:

A **confusion matrix** is created and visualized using **seaborn heatmaps** to highlight frequently misclassified drug types. This aids in performing error analysis.

Training Curves:

Plots of **training and validation accuracy and loss** across epochs are generated to observe learning behavior, convergence, and potential overfitting.

8. Model Saving and Deployment

The trained model is saved in **HDF5 format (.h5)** for later use or integration. The **class index mapping** (label to integer) is also saved, ensuring consistency in inference by mapping predictions back to medication names.

9. Summary of Enhancements Over Original VGG16

Aspect	Standard VGG16	Modified Model in Code	Benefit for Handwriting

			Recognition
Top Layers	3 original fully connected layers	Custom dense layers with Batch Normalization and Dropout	Improves adaptability to domain-specific features; reduces overfitting
Training Strategy	Trains all layers	Freezes convolutional base; trains only custom head	Preserves pretrained features; lowers training time and data demand
Data Augmentation	Limited or none	Extensive augmentation: rotation, shifting, shearing, zooming	Increases robustness to handwriting variability
Input Preparation	Standard ImageNet preprocessing	Uses vgg16.preprocess_input for consistency with pretrained weights	Maintains input compatibility with VGG16 expectations

These enhancements tailor the VGG16 architecture to better handle the complexities of recognizing handwritten medical prescriptions, characterized by significant intra-class variability and complex visual features. The changes strike a balance between using powerful pretrained features and customizing the classification head for improved accuracy and generalization.

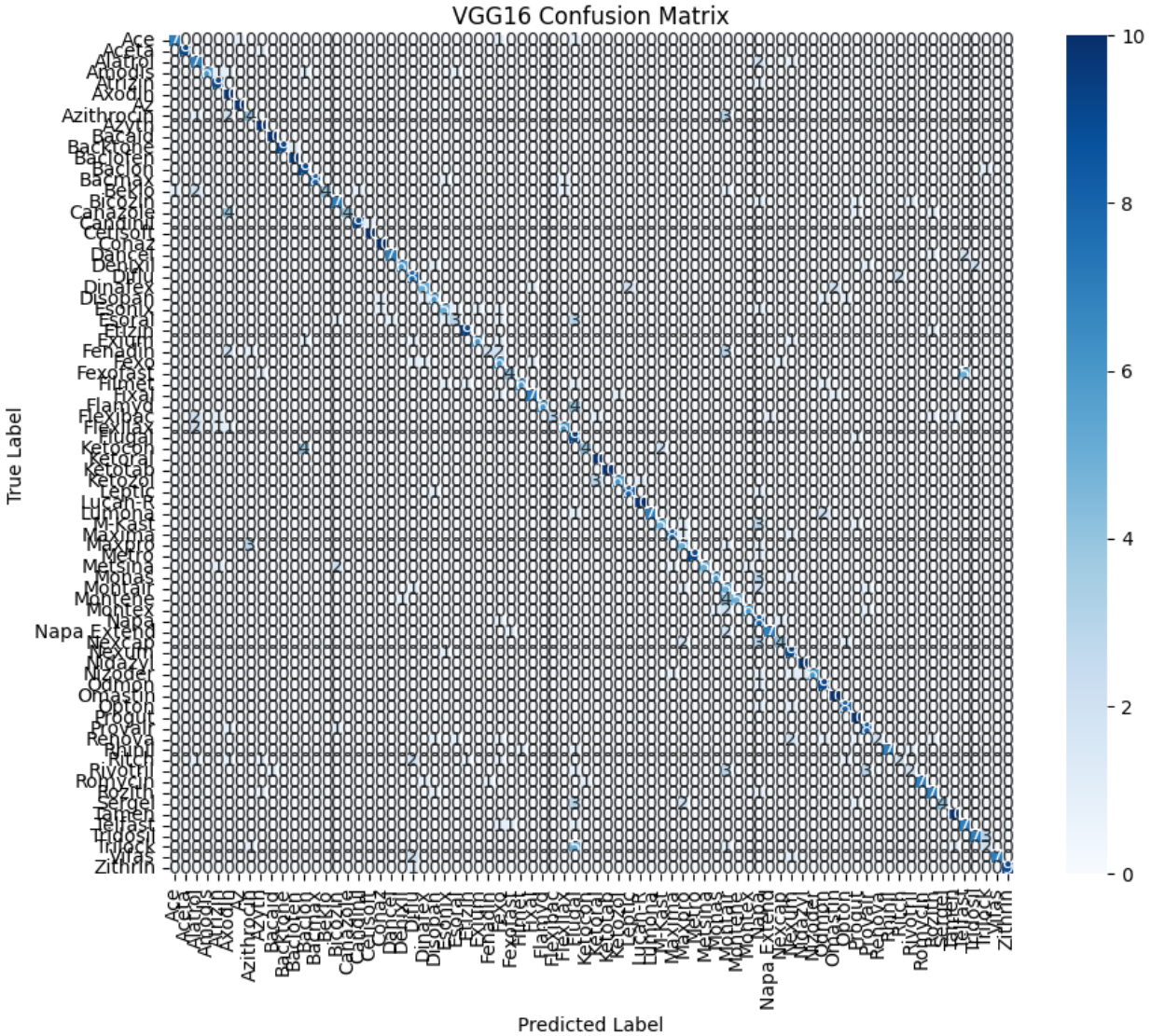
VGG16 code :

A deep learning model based on the VGG16 architecture was developed using Google Colab to classify handwritten words in medical prescriptions. The model was trained on a dataset of labeled word images and fine-tuned to improve recognition performance in the context of medical handwriting.

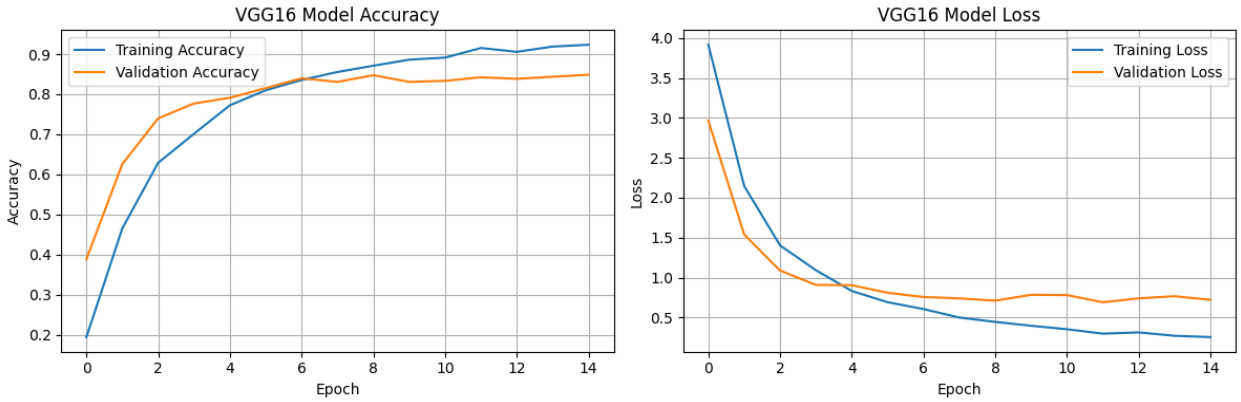
Google Colab Notebook: https://shorturl.at/fYgss

Results

VGG16 confusion matrix



VGG16 Graph



V. VGG16 Architecture: Description and Results

VGG16 Architecture Description

The model uses the **VGG16 architecture**, pre-trained as a fixed feature extractor by setting trainable=False for all layers of VGG16. The original classification part of VGG16 is discarded, and a new fully connected layer set is added for medical prescription recognition. The new head is composed of:

- A **Flatten** layer reshaping the output of convolutional layers into a 1D vector.
- A **dense layer** with **1024 units** and **ReLU** activation, followed by **batch normalization** and a **dropout layer (rate = 0.5)** to prevent overfitting.
- Another **dense layer** with **512 units**, followed by **batch normalization** and **dropout (rate = 0.5)**.
- A final **dense layer** with **softmax activation**, where the number of output units equals the number of medicine classes in the dataset.

This setup enables the model to benefit from the strong feature extraction capability of VGG16 (trained on ImageNet), while adapting the classification part for the specific task of recognizing different types of handwriting.

Training and Validation Results

The training and validation accuracy and loss graphs illustrate the model's learning process:

Accuracy Patterns

- Both training and validation accuracy increase rapidly during the first few epochs.
- Validation accuracy closely follows training accuracy until around epoch 6.
- Training accuracy continues to increase and eventually exceeds **0.90**, while validation accuracy plateaus between **0.84–0.851**.

Loss Patterns

- Both training and validation losses decline sharply in the early epochs.
- Training loss continues to decrease steadily.
- Validation loss flattens out and shows only minor fluctuations after epoch 6.

These patterns indicate that the model learns effectively during early training but begins to slightly overfit in later epochs.

Model Performance and Class-wise Efficiency

The **confusion matrix** offers detailed insights into class-wise performance:

- The matrix is mostly diagonal, indicating **high accuracy** across most classes.
- Sparse off-diagonal elements represent **occasional misclassifications**, which are rare and not concentrated on specific class pairs.
- No systematic confusion is detected, implying that the model discriminates well between different medicine names even under challenging handwriting.

Potential Causes of Overfitting and Misclassification

Some likely causes of misclassification and mild overfitting include:

- **Class Imbalance**
  Limited training samples for certain medicine names can hinder the model's ability to generalize.

- **Handwriting Variability**
  Differences in individual writing styles, pen pressure, and image quality can make similar medicine names difficult to distinguish.
- **Overfitting**
  The widening gap between training and validation accuracy suggests that the model may memorize specific training data patterns rather than learning generalizable features.

Applicability to Real-life Prescription Identification

The modified VGG16 model shows promise for practical deployment in medical contexts:

- **High Overall Accuracy**
  Strong performance on both training and validation data. The confusion matrix supports this.
- **Generalization Capability**
  Techniques like transfer learning, data augmentation, dropout, and batch normalization help the model generalize to unseen data.
- **Limitations and Room for Improvement**
  To reduce overfitting and enhance performance, fine-tuning the pre-trained layers, increasing data augmentation, or collecting more samples especially for rare classes can be beneficial.

Conclusion

The customized **VGG16-based model**, with its adapted classification head and regularization methods, proves well-suited for the complex task of handwritten medicine name recognition. Despite some minor overfitting and occasional misclassifications, the model performs well and lays a solid foundation for real-world deployment. With further enhancements, it can become even more robust, contributing to safer and more efficient healthcare operations by digitizing handwritten prescriptions accurately.

VI. Comparative Overview of VGG16, ResNet-50, and Inception-v3 Architectures

This section presents a comparative analysis of three prominent convolutional neural network (CNN) architectures widely used in computer vision: VGG16, ResNet-50, and Inception-v3. Each model introduces distinct architectural innovations that address challenges in deep learning such as training stability, computational efficiency, and feature extraction. Understanding their differences is essential for selecting appropriate models for specific image recognition tasks.

2. VGG16 Architecture

VGG16, developed by the Visual Geometry Group at the University of Oxford, is a deep CNN consisting of 16 weight layers—13 convolutional layers and 3 fully connected layers. The network processes input images of size 224×224×3 through a series of convolutional blocks, each composed of multiple 3×3 convolutional filters with stride 1 and padding 1, followed by ReLU activation functions. Max-pooling layers with a 2×2 window and stride 2 downsample feature maps after each block, progressively reducing spatial dimensions while capturing hierarchical features.

The fully connected layers at the end, each with 4096 units, perform high-level feature fusion and classification, culminating in a softmax layer for multi-class prediction. VGG16 contains approximately 138 million parameters, making it a relatively large model.

Strengths:

- Simplicity and uniform architecture with small convolutional filters enable effective learning of complex features.
- Proven high accuracy on large-scale datasets like ImageNet, achieving around 92.7% top-5 accuracy.
- Performs well on smaller datasets due to its straightforward design and ability to extract detailed spatial features.

Limitations:

- High computational and memory requirements due to large parameter count, leading to slower training and inference times.
- Lacks architectural mechanisms to explicitly address vanishing gradients, which can hinder training of deeper variants.

3. ResNet-50 Architecture

ResNet-50 introduces the concept of **residual learning** through skip connections, allowing the network to learn residual mappings instead of direct mappings, which significantly alleviates the vanishing gradient problem encountered in deep networks. It consists of 50 layers organized into residual blocks with a bottleneck design: 1x1 convolutions reduce and restore dimensionality around a 3x3 convolutional layer.

Key features:

- Skip connections enable identity mappings that facilitate gradient flow during backpropagation, allowing very deep networks to be trained effectively.
- The bottleneck architecture reduces computational cost while maintaining representational power.
- Typically achieves higher accuracy than VGG16 in many tasks due to its deeper and more sophisticated design.

Strengths:

- Robust training of deep architectures with improved gradient propagation.
- Generally better test accuracy and generalization compared to VGG16, especially on large and complex datasets.
- More parameter-efficient than VGG16 despite greater depth.

Limitations:

- Slightly more complex architecture requires careful implementation.
- Can still be prone to overfitting on small datasets without sufficient regularization.

4. Inception-v3 Architecture

Inception-v3 employs a modular design with **Inception modules** that concatenate feature maps from parallel convolutional layers of varying sizes, enabling multi-scale feature extraction. It incorporates several architectural improvements over earlier Inception versions:

- **Factorized convolutions:** Large convolutions (e.g., 7x7) are factored into smaller convolutions (1x7 and 7x1) to reduce computational cost without sacrificing receptive field size.
- **Auxiliary classifiers:** Intermediate classifiers improve gradient flow and act as regularizers during training.
- **Label smoothing and batch normalization:** Enhance generalization and training stability.

Strengths:

- Efficient computation due to factorized convolutions and careful module design.
- Strong multi-scale feature extraction capability beneficial for complex image patterns.
- Improved generalization through auxiliary classifiers and label smoothing.

Limitations:

- Architectural complexity can make implementation and tuning more challenging.
- May require larger datasets to fully exploit its capacity.

5. Comparative Summary:

Feature	**VGG16**	**ResNet-50**	**Inception-v3**
Depth	16 layers (13 conv + 3 FC)	50 layers with residual blocks	~48 layers with Inception modules
Parameters	~138 million	~25 million (more efficient)	~23 million

Core Innovation	Deep stack of small 3x3 convolutions	Residual learning with skip connections	Inception modules with factorized convolutions
Handling Vanishing Gradient	None explicit; can suffer in deeper nets	Skip connections mitigate vanishing gradient	Auxiliary classifiers and batch normalization
Computational Efficiency	High memory and compute demand	More efficient than VGG16	Efficient due to factorized convolutions
Regularization Techniques	ReLU, dropout (in custom setups)	Batch normalization, skip connections	Label smoothing, auxiliary classifiers, batch normalization
Typical Applications	Image classification, object recognition	Image classification, detection, recognition	Image classification, medical imaging, defect detection
Strengths	Simplicity, effective feature learning on smaller datasets	Robust training of very deep networks, better accuracy	Efficient multi-scale feature extraction, improved generalization
Limitations	Computationally expensive, slower training	More complex, potential overfitting on small data	Architectural complexity, needs large datasets

VGG16, ResNet-50, and Inception-v3 represent distinct approaches to CNN design, each excelling under different conditions. VGG16's straightforward and

uniform architecture makes it a solid baseline, particularly effective for smaller datasets but limited by computational demands. ResNet-50's residual connections enable deeper networks with improved accuracy and training stability, making it suitable for complex tasks and larger datasets. Inception-v3's modular and factorized design balances accuracy and efficiency, excelling in multi-scale feature extraction and generalization.

The choice among these architectures should consider dataset size, computational resources, and task complexity. For instance, VGG16 may be preferred for limited data scenarios, while ResNet-50 and Inception-v3 offer superior performance for large-scale, complex image recognition challenges.

In this chapter, we provide an empirical assessment of three modern convolutional neural networks — **VGG16**, **ResNet50**, and **InceptionV3** — on the task of handwritten recognition of medicine names from medical prescriptions. Following the training pipeline developed for CNN models, each network was trained with the following methodology: the convolutional base of each pre-trained model was frozen, a custom classification head was appended, and the network was trained using regularization techniques such as **dropout** and **batch normalization**. Additionally, **data augmentation** was incorporated to promote generalization.

The purpose of this comparative study is to evaluate each model's **effectiveness**, **stability**, and **generalization** ability, especially given the challenges of handwritten text recognition. This evaluation aims to identify the most suitable model for real-world deployment in medical digitization tasks.

VGG16

2- Transfer Learning Setup

VGG16 was used as a fixed feature extractor. All convolutional layers were frozen using pre-trained ImageNet weights. A custom classification head was constructed with fully connected layers, batch normalization, and dropout to classify medicine names effectively.

3- Training and Validation Performance

From the training and validation accuracy curves (Figure 1, top left), we observe that VGG16 quickly converged, especially in the early epochs (within 5 epochs). Validation accuracy closely followed training accuracy up to epoch 8, peaking at approximately **0.88**. Training accuracy later surpassed **0.90**. The loss curves (Figure 1, top right) reflect a sharp initial drop that leveled off during later epochs.

4- Overfitting / Underfitting

After epoch 8, a slight gap developed between training and validation accuracy. While training accuracy continued to improve, validation accuracy plateaued. This behavior, coupled with an uptick in validation loss, suggests **mild overfitting**, although generalization remained strong for most of the training process.

5- Convergence Behavior

VGG16 achieved rapid and stable convergence around epoch 10. Beyond this point, accuracy and loss flattened. The learning dynamics indicate **effective training** with **no significant underfitting**.

ResNet50

1- Transfer Learning Setup

ResNet50 was implemented similarly, with frozen base layers and a custom classification head. Its residual learning framework includes **skip connections**, which mitigate vanishing gradients in deeper networks.

2- Training and Validation Performance

ResNet50's accuracy curves (Figure 1, middle left) show that validation accuracy improved up to epoch 6, reaching a peak of around **0.83**, after which it plateaued. Meanwhile, training accuracy continued to rise, reaching ~**0.87**. Loss curves (Figure 1, middle right) show a similar trend, with both losses decreasing and then stabilizing.

Overfitting / Underfitting

Post-epoch 6, divergence between training and validation accuracy suggests **incipient overfitting**. However, the magnitude of this gap is smaller than in VGG16, and the validation performance remained relatively stable.

Convergence Behavior

ResNet50 **converged quickly** and displayed stability in both accuracy and loss curves by epoch 8. There were no significant signs of underfitting.

InceptionV3

1- Transfer Learning Setup

Like the others, InceptionV3 was trained with frozen base layers and a custom classification head. Its key feature is the **inception module**, which extracts **multi-scale features** via parallel convolutions.

2- Training and Validation Performance

InceptionV3's accuracy curves (Figure 1, bottom left) show **slow and intermittent improvement**. Validation accuracy plateaued at **~0.43**, while training accuracy remained **below 0.35**. The loss curves (Figure 1, bottom right) declined gradually but plateaued at higher levels compared to the other models.

3- Overfitting / Underfitting

There is **no evidence of overfitting**, but **underfitting is clearly present**, as both training and validation accuracies remained low and improved in tandem. The model struggled to learn discriminative features, likely due to poor alignment between its architecture and the data.

4- Convergence Behavior

InceptionV3 did **not converge quickly**. While minor gains were seen after 18 epochs, the model did not achieve the performance level of VGG16 or ResNet50.

Relative Performance Analysis

1- Accuracy and Generalizability

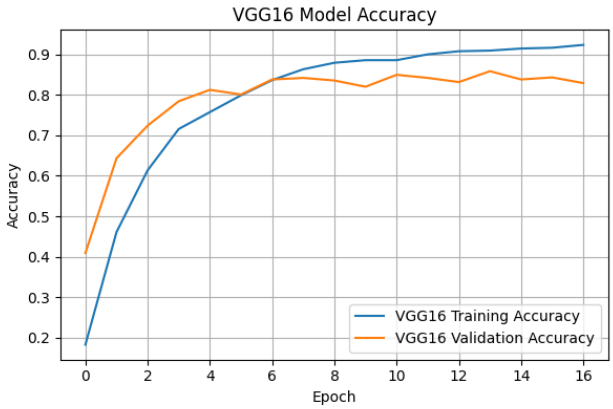
- **VGG16**: Best performance with **~0.88 validation accura**

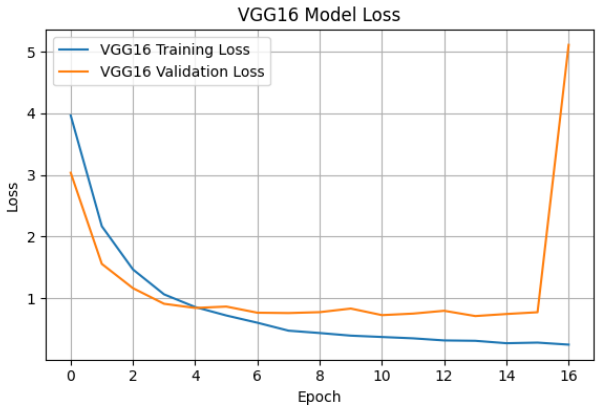
- **ResNet50**: Slightly lower with **~0.83 validation accuracy**.
- **InceptionV3**: Lowest performance at **~0.43 validation accuracy**.

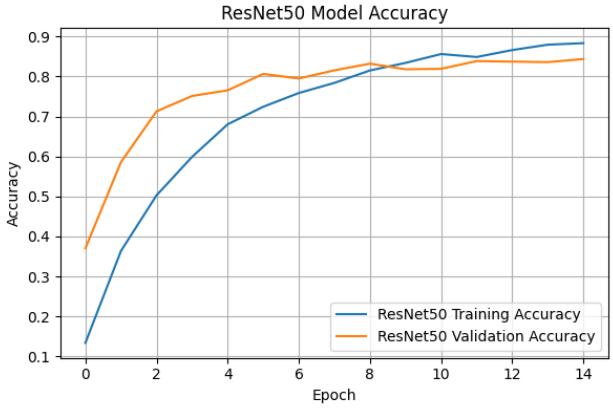
VGG16 and ResNet50 both demonstrated strong **early generalization**, with tightly aligned training and validation accuracies in early epochs. Later, moderate overfitting occurred, especially in VGG16. InceptionV3's consistent underperformance reflects **model-data misalignment** or ineffective learning from the classification head.

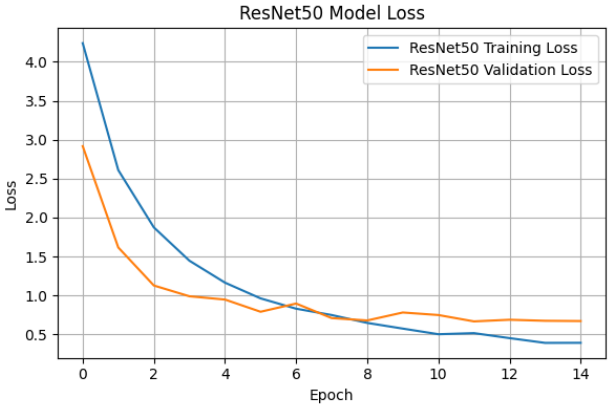
2- Stability and Training Characteristics

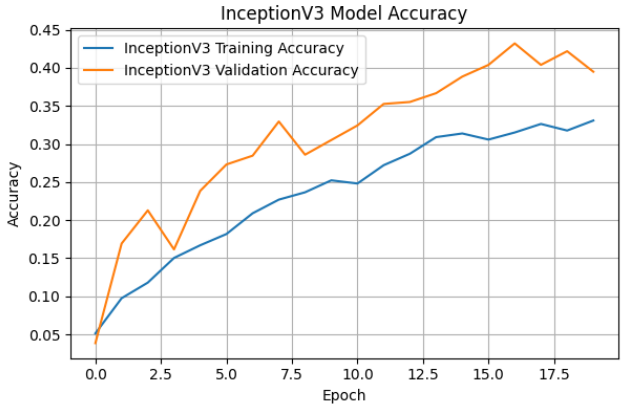
- **VGG16**: Smooth, stable accuracy curves and rapid convergence.
- **ResNet50**: Stable learning due to residual connections, though it plateaued below VGG16.
- **InceptionV3**: Fluctuating performance, slow convergence, and lower learning effectiveness.

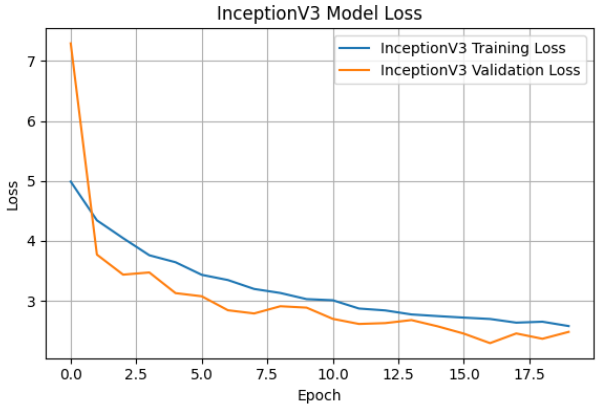

VGG16 Model Accuracy
Accuracy
Epoch
VGG16 Training Accuracy
VGG16 Validation Accuracy


VGG16 Model Loss
Loss
Epoch
VGG16 Training Loss
VGG16 Validation Loss


ResNet50 Model Accuracy
Accuracy
Epoch
ResNet50 Training Accuracy
ResNet50 Validation Accuracy

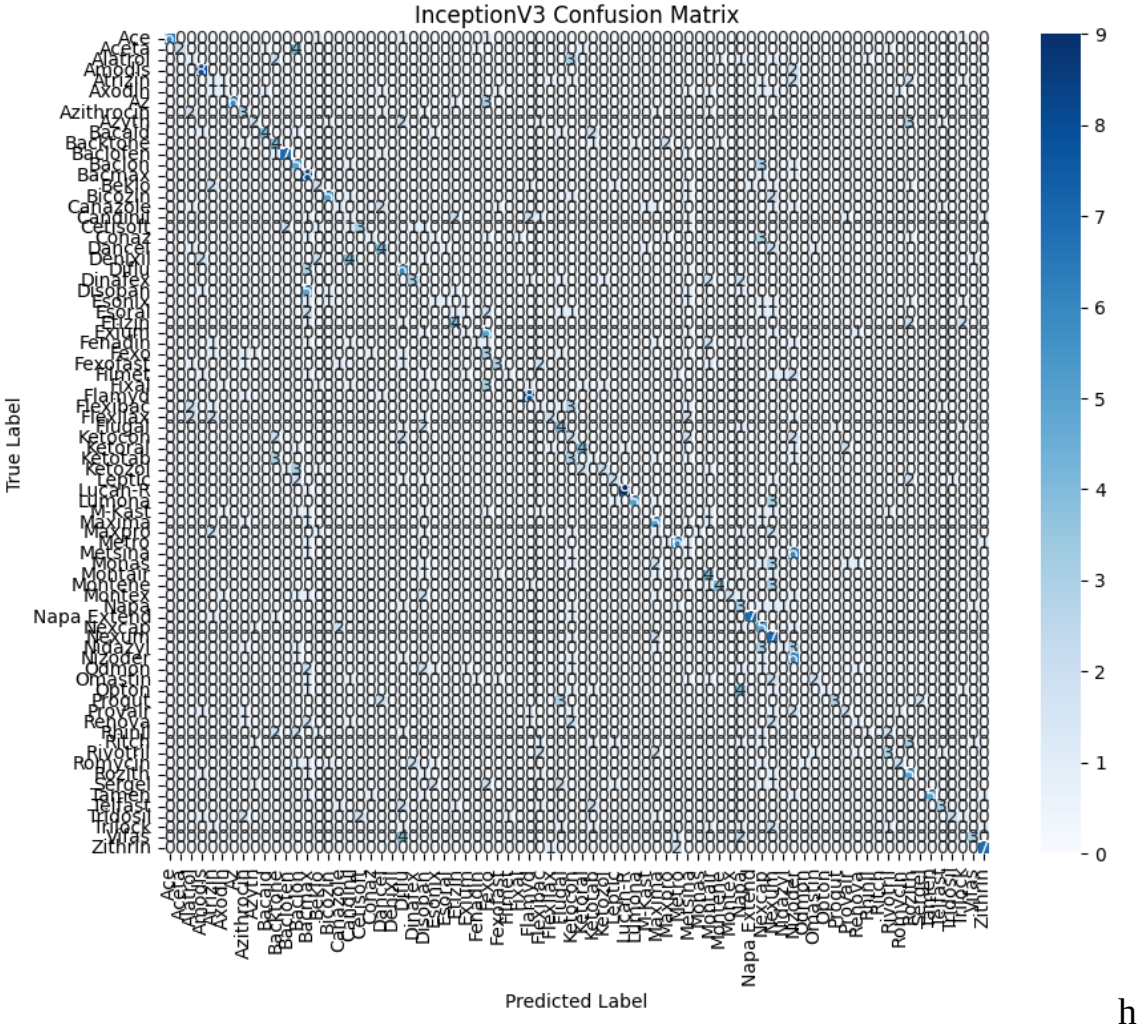

ResNet50 Model Loss
Loss
Epoch
ResNet50 Training Loss
ResNet50 Validation Loss


InceptionV3 Model Accuracy
Accuracy
Epoch
InceptionV3 Training Accuracy
InceptionV3 Validation Accuracy

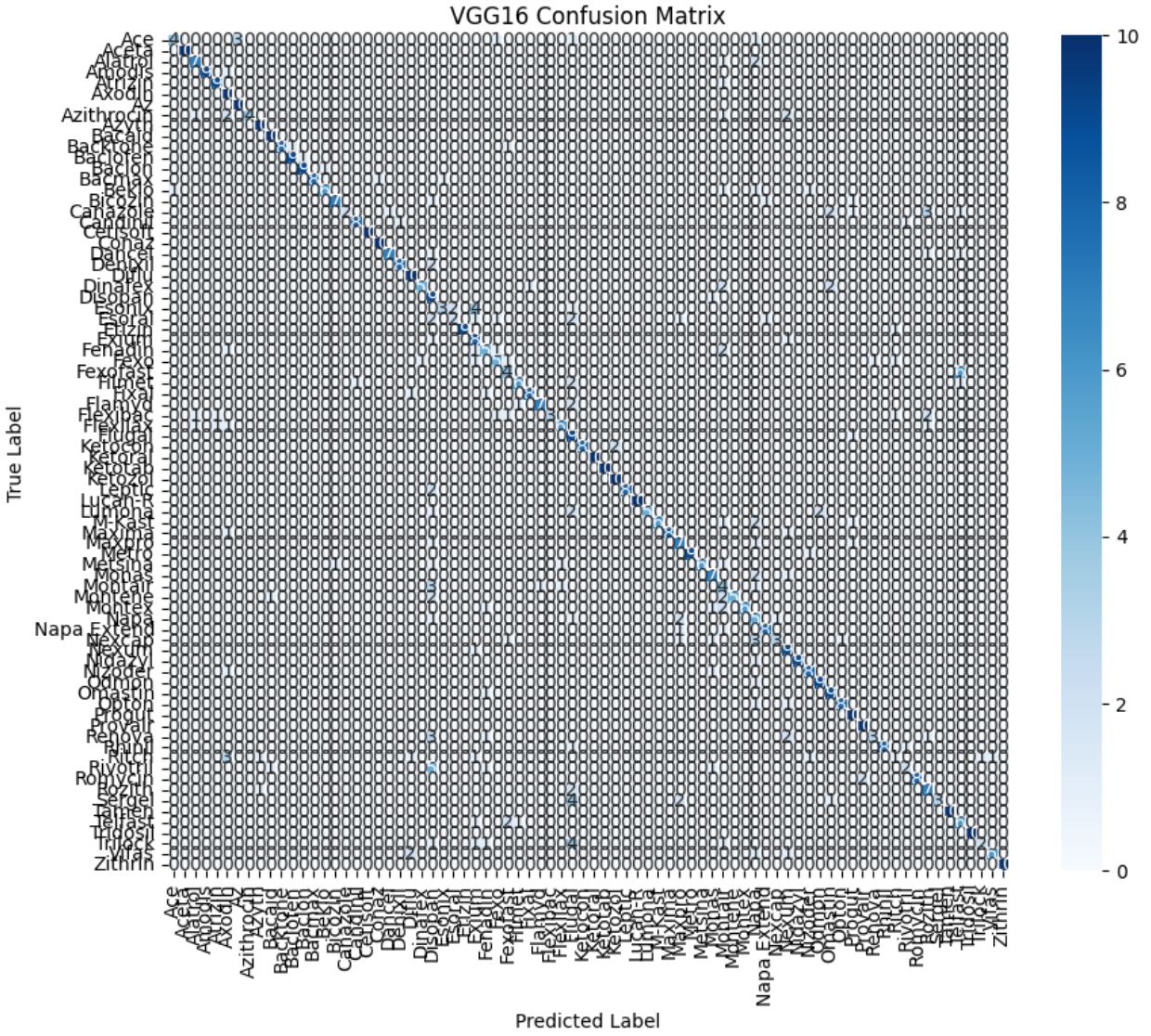

InceptionV3 Model Loss
Loss
Epoch
InceptionV3 Training Loss
InceptionV3 Validation Loss

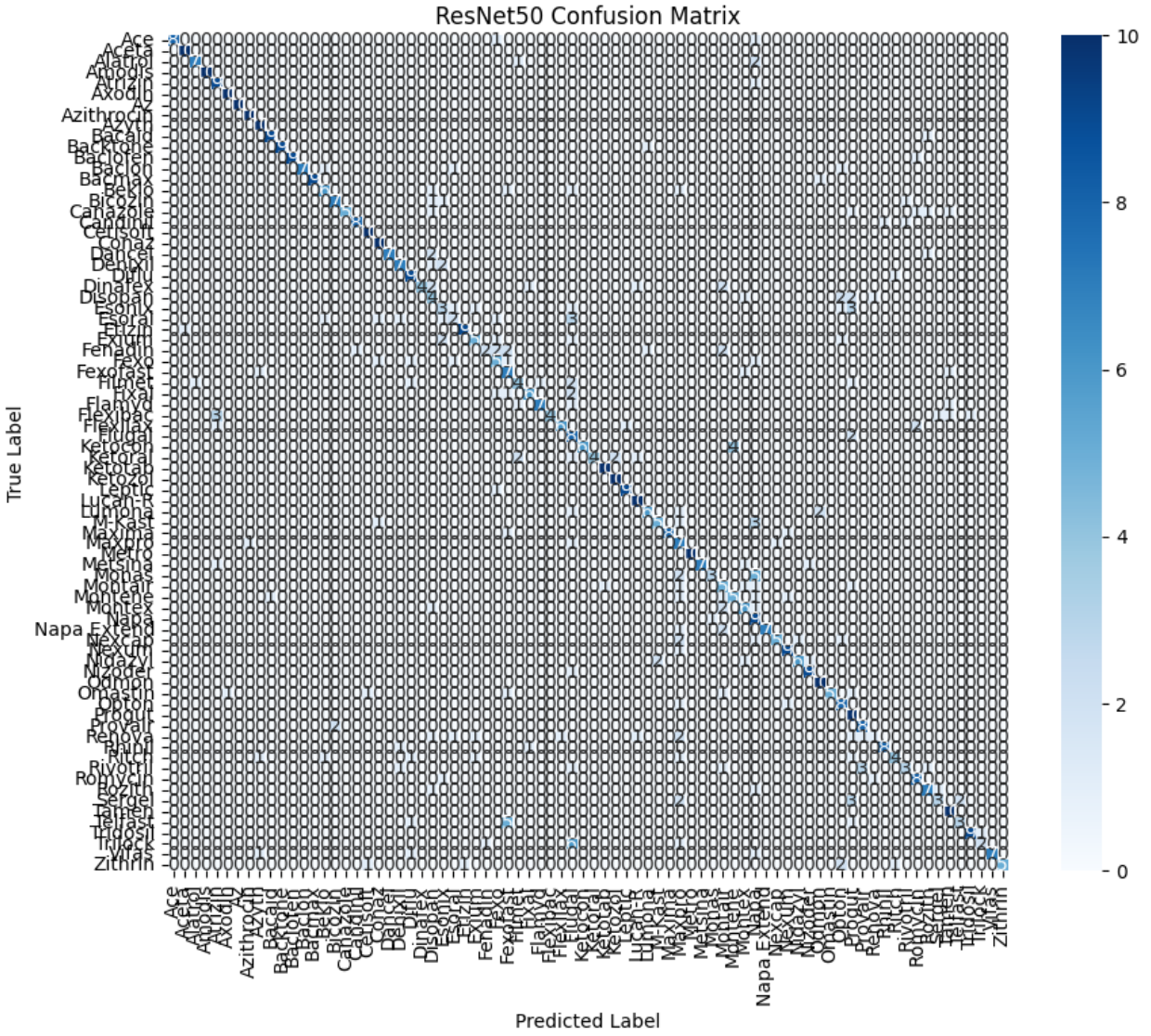
3- Confusion Matrix Analysis

- **VGG16 and ResNet50**: Diagonal-dominant confusion matrices with minimal misclassifications.
- **InceptionV3**: Still diagonal-dominant but with more frequent misclassifications, consistent wit



h its lower accuracy.


VGG16 Confusion Matrix
True Label
Predicted Label
10
8
6
4
2
0



4- Interpretation of Results

Model Architecture Impact

Despite a standardized training setup, **VGG16 and ResNet50 significantly outperformed InceptionV3** on the validation dataset. This can be attributed to differences in architecture:

- **VGG16**: Simple, deep stack of small filters, highly effective for extracting fine-grained spatial features in handwritten characters.

- **ResNet50**: Residual connections enabled deep learning and stable gradients, but did not surpass VGG16 — possibly due to the dataset's simplicity or base model freezing.
- **InceptionV3**: Multi-scale features from inception modules likely benefit natural images but were **less suited to handwritten prescription data**, leading to underfitting.

5- Regularization and Generalization

The use of **dropout**, **batch normalization**, and **data augmentation** helped reduce overfitting in VGG16 and ResNet50. Some overfitting persisted, but this could be mitigated by **fine-tuning the base layers** or employing **stronger data augmentation strategies**.

6- Conclusion

Among the three models tested, **VGG16 emerged as the most effective for recognizing handwritten medicine names**:

- **Highest validation accuracy**
- **Stable convergence**
- **Best performance on confusion matrices**
- **Minimal overfitting and strong generalization**

ResNet50 also performed well, leveraging residual connections, but did not exceed VGG16's accuracy. **InceptionV3** underperformed due to underfitting and architectural mismatch.

Chapter 04:
Conclusion

Limitations and Future Work

A major limitation was the **use of frozen base models**, which restricted learning from domain-specific features in handwritten prescriptions. Future improvements could include:

- **Fine-tuning pre-trained layers**
- **More sophisticated data augmentation**
- **Using attention mechanisms**
- **Tackling class imbalance**
- **Expanding the dataset**

Additionally, ethical and privacy considerations are paramount when dealing with medical data. We as researchers must obtain explicit permission to use patient information and ensure that all data is appropriately anonymized and protected from unauthorized access or leaks. Adhering to data privacy regulations (such as HIPAA or GDPR) is essential when handling sensitive information in any real-world deployment.

Ultimately, **VGG16 offers the best trade-off** between accuracy and generalization and can be deployed in practical prescription digitization tasks with minimal adjustment. Nonetheless, further tuning of all models is recommended to realize their full potential in real-world applications.

Rationale Behind Using VGG16 for Handwritten Prescription Recognition:VGG16, with a good combination of feature extraction power, architectural simplicity, and actual relevance to image classification in real-life contexts, was chosen as the backbone deep architecture behind doctor handwriting recognition on medical prescriptions. Doctor's handwriting, many times being awkwardly laid out, with more-or-less random spacing, cursive styles, and overlapping characters, still poses a great challenge to recognition systems. With its setting of 16 layers of width convolution layers interspersed with pools and composed entirely of 3×3 filters, VGG is arguably the best learned hierarchical representation at very fine spatial scales to sense subtle variations of stroke width, loops, inter-character edge dissimilarity that are frequent in handwritten text. However, amongst numerous models that appear very complex, such as Inception-v3 and ResNet50, the beauty of VGG lies in its simplistic design and

interpretability, thus allowing an easier approach to applying transfer learning and fine-tuning upon relatively fewer data and resources as in doing so on Google Colab.

VGG16 is frequently used for optical character recognition (OCR) and document analysis, and its compatibility with transfer learning from large-scale datasets such as ImageNet lessens training time and mitigates the challenges of data scarcity-especially in the case of specialized datasets like medical prescriptions. By freezing the pre-trained convolutional base while fine-tuning the fully connected layers for handwriting categorization, VGG16 provides an efficient mechanism to reuse features and yet still adapt such features towards the actual task of recognizing handwritten text. Due to its great performance on character- and word-level image inputs, it can be flexibly used downstream for integration into a larger prescription processing pipeline. These practicalities give VGG16 its merit as the best solution for a handwriting recognition system that fits the real-world scenario of doctor-written prescriptions.

Conclusion:

The combination of deep learning and AI in physician's handwriting recognition offers important benefits, along with several technical and ethical concerns. The main advantage is better patient care. By making prescriptions and medical notes more readable, deep learning models—specifically, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs)—can eradicate the likelihood of misreading, which would otherwise lead to malpractice in medication or erroneous treatment strategies. Such increased readability increases doctors' and patients' communication, as well as between medical professionals, and eventually results in safer and more precise medical practices (Chan and Hu).

Furthermore, deep learning algorithms trained on large sets of variable handwriting samples are capable of learning to read and identify variable handwriting styles. The ability to adapt makes them more reliable than human

transcription, especially in stressful settings. The models also quickly read patients' and prescriptions' information, reducing administrative tasks and treatment delays.

However, there are limitations to the implementation of deep learning techniques. Even then, they cannot match the unusually rare writing patterns, regional dialects, or technical medical words not represented or absent in the training dataset. In addition, digital systems used to record, process, and store sensitive patient data raise grave concerns over data security and privacy (Chan and Hu).

With the increasing integration of the healthcare sector with AI and deep learning, there is a need to harness the potential of these technologies while proactively addressing their limitations. Innovation has to be balanced with ethical implications in order to promote effective, responsible implementation. Research and development on an ongoing basis are critical in order to develop models that drive operational efficiency while ensuring the highest standards of care to patients (Alhosani and Alhashmi).

Looking ahead, the future of doctor handwriting recognition is increasingly tied to the expansion of deep learning. Novel architectures—Vision Transformers and hybrid CNN-RNN models—promising to further increase the accuracy and effectiveness of handwritten note recognition, contribute to enhanced healthcare delivery. These improvements may significantly reduce documentation errors resulting from illegible handwriting, ensuring clinical judgements are based on accurate and timely information.

In addition, AI-based handwriting recognition in electronic health records (EHRs) can potentially speed intra-team communication and reduce administrative burdens. More accessible systems will enable clinicians to focus on direct patient

care rather than tedious paperwork. Such a transformation can reduce burnout and increase job satisfaction among healthcare professionals (Fogleman et al.).

In the long run, deep learning's revolutionary potential in medical handwriting recognition has the power to revolutionalize health outcomes by ensuring accurate, timely, and efficient recordation. Embracing these innovations will be pivotal in developing a healthcare system where technology allows for—not replaces—human empathy and clinical skill (Fogleman et al.).

References:

1. Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009). "A Novel Connectionist System for Unconstrained Handwriting Recognition." https://ieeexplore.ieee.org/document/4531750
2. Karen Simonyan, Andrew Zisserman, A. (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition (VGG16)." https://arxiv.org/abs/1409.1556
3. Tabassum, S., Abedin, N., Rahman, M. M., Rahman, M. M., Ahmed, M. T., Islam, R., & Ahmed, A. (2022). An online cursive handwritten medical words recognition system for busy doctors in developing countries for ensuring efficient healthcare service delivery. Scientific Reports, 12, 3601. https://doi.org/10.1038/s41598-022-07571-z
4. Chinthaginjala, R., Dhanamjayulu, C., Kim, T., Ahmed, S., Kim, S.-Y., Kumar, A. S., & Annepu, V. (2024). Enhancing handwritten text recognition accuracy with gated mechanisms. Scientific Reports, 14, Article 16800. https://doi.org/10.1038/s41598-024-67738-8

5. 5.Riktim Mondal,Samir Malakar,Elisa H. Barney Smith,Ram Sarkar.Handwritten English Word Recognition Using a Deep Learning Based Object detection Architecture. https://core.ac.uk/outputs/541261322/?source=2

6. Eman Ahmed Khorsheed,Ahmed Khorsheed Al-Sulaifanie. Handwritten Digit Classification Using Deep Learning Convolutional Neural Network. https://shorturl.at/TeOy4
7. Usman Ali, Sahil Ranmbail, Muhammad Nadeem, Hamid Ishfaq, Muhammad Umer Ramzan, Waqas Ali. Leveraging Deep Learning with Multi-Head Attention for Accurate Extraction of Medicine from Handwritten Prescriptions . https://arxiv.org/abs/2412.18199
8. Mohammed Hamdan, Abderrahmane Rahiche, Mohamed Cheriet. HTR-JAND: Handwritten Text Recognition with Joint Attention Network and Knowledge Distillation. https://arxiv.org/abs/2412.18524
9. Sukhdeep Singh, Sudhir Rohilla, Anuj Sharma . An inclusive review on deep learning techniques and their scope in handwriting recognition. https://arxiv.org/abs/2404.08011
10. al. Tamanna Sachdeva. A Novel Approach for Hand-written Digit Classification Using Deep Learning . https://shorturl.at/t8onC

Maryam Al-Mashhadani . Real time handwriting recognition system using CNN algorithms . https://www.researchgate.net/publication/374651225_Real_time_handwriting_recognition_system_using_CNN_algorithms

12. Felix Ott, David Rügamer, Lucas Heublein, Bernd Bischl, Christopher Mutschler . Auxiliary Cross-Modal Representation Learning with Triplet

Loss Functions for Online Handwriting Recognition . https://arxiv.org/abs/2202.07901

13. Firat Kizilirmak, Berrin Yanikoglu . CNN-BiLSTM model for English Handwriting Recognition: Comprehensive Evaluation on the IAM Dataset . https://arxiv.org/abs/2307.00664
14. Gibran Satya NugrahaMuhammad Ilham DarmawanRamaditia Dwiyansaputra . Comparison of CNN's Architecture GoogleNet, AlexNet, VGG-16, Lenet -5, Resnet-50 in Arabic Handwriting Pattern Recognition . https://shorturl.at/J1H1p
15. Shaira Tabassum, Nuren Abedin, Md Mahmudur Rahman, Md Moshiur Rahman, Mostafa Taufiq Ahmed, Rafiqul Islam & Ashir Ahmed. An online cursive handwritten medical words recognition system for busy doctors in developing countries for ensuring efficient healthcare service delivery. https://www.nature.com/articles/s41598-022-07571-z
16. Riktim MondalSamir MalakarElisa H. Barney SmithRam Sarkar . Handwritten English Word Recognition Using a Deep Learning Based Object Detection Architecture . https://core.ac.uk/outputs/541261322/?source=2