AUTOMATIC E-GOVERNMENT SERVICES WITH ARTIFICIAL INTELLIGENCE

[1]M.Tech Student, DEPARTMENT OF COMPUTER SCIENCE ENGINEERING, A1 GLOBAL INSTITUTE OF ENGINEERING AND TECHNOLOGY, MARKAPUR.523316, ANDHRA PRADESH
[2]ASSISTANT PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE ENGINEERING, A1 GLOBAL INSTITUTE OF ENGINEERING AND TECHNOLOGY, MARKAPUR.523316, ANDHRA PRADESH
[1]P. V. SRAVANI [2]SK. MD. RAFI

ABSTRACT

Artificial Intelligence (AI) has made remarkable strides across a growing range of fields in recent years. Despite these advancements, its integration into e-government systems still presents a number of challenges. These obstacles limit AI's potential in enhancing both the internal efficiency of e-government infrastructures and the quality of interactions between governments and citizens.In this study, we examine these key challenges and introduce a comprehensive framework designed to harness the power of AI for modernizing and streamlining e-government services. Our approach involves three major contributions. We present a structured framework focused on the effective management of e-government information resources.We design a series of deep learning models intended to automate core public service operations.We propose a smart architecture for an AI-enabled e-government platform that supports the creation and deployment of intelligent solutions across various administrative domains.The primary objective of our framework is to implement reliable and transparent AI methodologies that can transform existing e-government services. This transformation aims to accelerate service delivery, lower operational costs, and enhance citizen satisfaction through intelligent, automated processes.

INTRODUCTION

Artificial Intelligence (AI) has existed in theoretical concepts and complex systems for decades. However, it is only with the recent surge in computational capabilities and the availability of massive datasets that AI has begun to deliver exceptional performance across a wide range of disciplines. Notable progress has been witnessed in fields such as computer vision [1], healthcare technologies [2], natural language processing [3], reinforcement learning [4], among others.

At its core, AI refers to a system's capability to mimic human intelligence while continuously enhancing its performance through experience. Contrary to common misconceptions, AI extends far beyond robotics — it represents the cognitive intelligence of machines, enabling them to carry out tasks such as autonomous driving, strategic gameplay, and the execution of complex problem-solving operations.

AI is inherently multidisciplinary, drawing from and contributing to several overlapping domains. These include Machine Learning (ML) [5], Deep Learning (DL) [6], Natural Language Understanding [3], Context-Aware Computing [7], and Data Security and Privacy [8]. The interconnections between AI and these related fields are visually summarized in Figure 1, highlighting how AI functions as a nexus of modern intelligent technologies.

Machine Learning (ML) refers to the capability of algorithms to learn from existing data and make intelligent decisions in unfamiliar scenarios. Instead of being explicitly programmed for every task, ML systems develop patterns and insights from historical datasets—for instance, using demographic data to forecast trends like employment rates. This learning process, where algorithms are trained using labeled datasets, is known as supervised learning.

In contrast to traditional ML techniques, Deep Learning (DL)—a specialized branch of ML—has evolved to overcome many of their limitations. Deep learning is essentially a computational approach that maps raw input data (such as a medical scan) to a specific output (like a diagnosis) by minimizing prediction errors through optimization methods like stochastic gradient descent (SGD) [9].

Deep learning models are inspired by the structure of the human brain and are composed of multi-layered artificial neural networks. These networks process inputs through multiple hidden layers, which are responsible

for transforming data using a sequence of nonlinear operations (typically dot products followed by activation functions). The term “deep” reflects the presence of these many layers through which data flows.

A significant benefit of deep learning is that it automatically learns relevant features from raw data, eliminating the need for manual feature extraction, which is often required in traditional machine learning approaches.

While deep learning has significantly advanced the performance of intelligent systems across various fields, its application within e-government continues to face numerous obstacles [10]. One major challenge is the shortage of skilled professionals proficient in deep learning technologies, especially in developing nations, where the talent pool for designing robust and efficient AI solutions remains limited despite rapid progress in the field.

CNN WORKING PROCEDURE:

To illustrate the process of building a Convolutional Neural Network (CNN) for image classification, we will develop a simple 6-layer CNN model capable of distinguishing between different images. This lightweight network is intentionally designed to be efficient enough to run on a standard CPU, making it suitable for learning and experimentation purposes.

Unlike more complex image classifiers that require powerful GPUs and extensive training time due to their large number of parameters, our goal is to provide a practical introduction to CNNs using TensorFlow—one of the most widely used frameworks for deep learning.

At its core, a neural network is a mathematical system designed to solve optimization problems. These networks are constructed using units called neurons, each performing a basic computation. A single neuron accepts an input (denoted as x), applies a transformation such as a weighted sum ($z = wx + b$), and passes the result through a non-linear activation function to generate the output or activation.

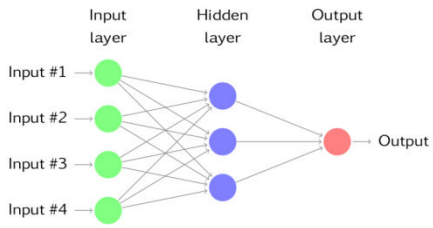
Activation functions play a critical role in introducing non-linearity to the model, enabling the network to learn complex patterns. One of the commonly used activation functions is the Sigmoid function, which maps the input into a range between 0 and 1. This behavior is especially useful for binary classification and helps the model make probabilistic predictions.

Through this example, we aim to walk through the complete process of constructing, training, and testing a real-world CNN model using TensorFlow, providing both conceptual understanding and hands-on experience.

When a neuron uses the sigmoid function as its activation mechanism, it is commonly referred to as a sigmoid neuron. The choice of activation function plays a crucial role in defining the behavior of the neuron, and accordingly, neurons are often named after the functions they use. Besides sigmoid, other widely used activation functions include ReLU (Rectified Linear Unit) and TanH (Hyperbolic Tangent), each with its own mathematical properties and use cases in neural networks.

When multiple neurons are arranged side by side in a sequence, this structure is known as a layer—a fundamental component of any neural network. Layers serve as the foundation for building complex architectures by stacking one over another, allowing the network to learn deeper and more abstract representations of data as information passes through each layer.

The illustration below typically shows how neurons are organized within layers, highlighting how inputs flow through the network—from the input layer, through one or more hidden layers, to the output layer.



In order to **predict the class of an image**, the neural network processes it through **multiple interconnected layers**. Each layer extracts and refines features from the image, gradually improving the model's understanding. This iterative process continues until the network reaches a point where **no further improvements in prediction accuracy** can be achieved — effectively converging on the best possible output.

IMAGE UPLOAD INSTRUCTIONS:

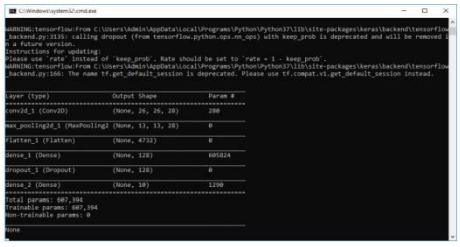
You can upload all the required facial expression images from the folder named 'expression_images_to_upload'. This directory contains a variety of emotion-based images used to train or test the sentiment detection model.

RUNNING THE PROJECT:

To launch the application, simply double-click the 'run.bat' file located in the project directory. This will initiate the program and display the main interface screen, from where you can begin interacting with the system.

In above screen click on 'Generate Hand Written Digits Recognition Deep Learning Model' button to generate CNN digits recognition model

In above screen we can see digits model generated and CNN layer details you can see black console

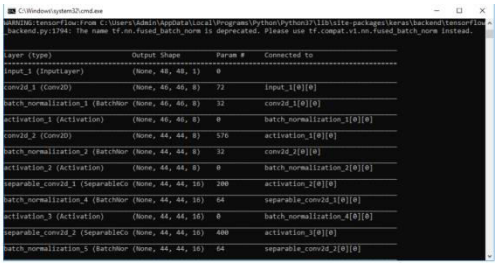


In the interface shown above, the term Conv2D refers to the convolutional layers used in a Convolutional Neural Network (CNN). These layers are responsible for automatically extracting features from input images. As the data moves through successive convolutional layers, the image dimensions reduce, and the feature depth

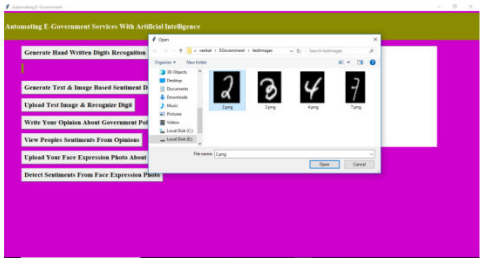
increases. For example, the first Conv2D layer processes the image at a size of 26x26, while the next layer compresses it further to 13x13, and so on—enabling the network to learn complex patterns at different scales.

To begin building the model, click on the 'Generate Text & Image Based Sentiment Detection Deep Learning Model' button. This will trigger the creation of a CNN capable of analyzing both textual data and facial expression images, allowing the system to perform multi-modal sentiment detection.

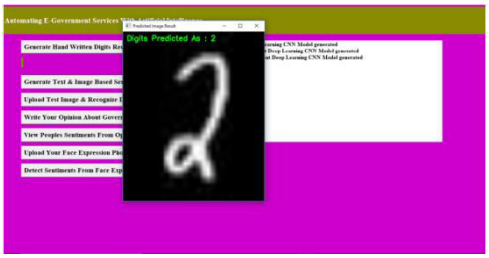
In above screen we can see text and image based CNN model generated. See black screen for more details



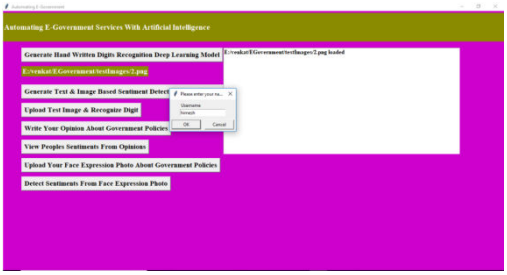
Now click on 'Upload Test Image & Recognize Digit' button to upload digit images and to get name of that digit. All digit images saved inside test Images folder



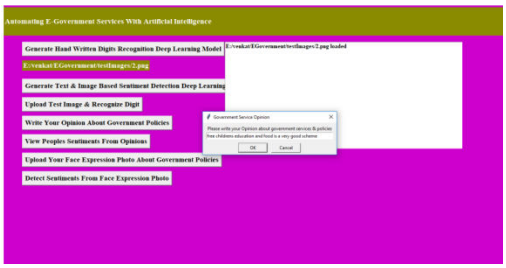
In above screen I am uploading image which contain digit 2 and below is the output of detection



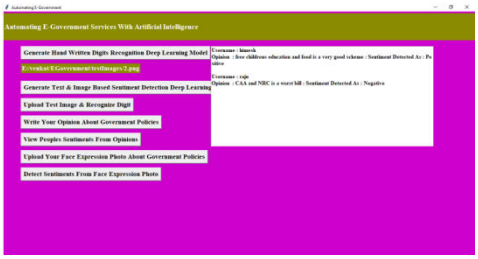
In above screen we can see Digits Predicted as: 2. Now click on 'Write Your Opinion About Government Policies' button to write some comments on government policy



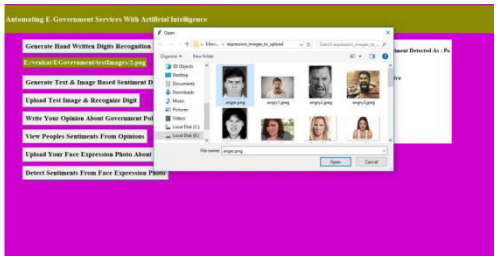
In above screen before writing opinions we need to write username after writing username click ok button to get below screen



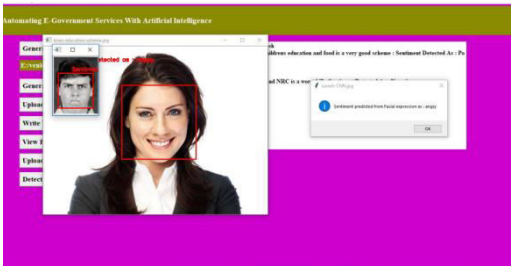
In above screen I wrote some comment on some scheme and application detect sentiment from it as positive or negative. Now click on 'View Peoples Sentiments From Opinions' button to view all opinions from past users.



In above screen text area we can see opinions from all users and in first opinion we got sentiment detected as positive which means user is satisfy with that scheme and for second opinion we got sentiment as negative which means user not happy. Similarly user can upload their image with facial expression which describe whether user is happy or angry.



In above screen I am uploading one anger face image and then application ask to write username and referring scheme name. similarly any number of users can upload their images. Now click on 'Detect Sentiments From Face Expression Photo' button to get all images and its detected sentiments In above screen I am uploading one anger face image and then application ask to write username and referring scheme name. similarly any number of users can upload their images. Now click on 'Detect Sentiments From Face Expression Photo' button to get all images and its detected sentiments.



In above screen we can see all images with facial expression are identified with their sentiments. In dialog box also we can see sentiment result. Similarly you can enter any number of comments or facial images to detect their sentiments.

REQUIREMENT ANALYSIS

The objective of this project was to evaluate and enhance the usability and interface design of several applications, with a focus on making them more user-friendly and intuitive. A key part of this process was ensuring that navigation between screens was smooth and logically structured, allowing users to move through the application effortlessly.

Additionally, efforts were made to minimize user input, reducing the amount of manual typing required to complete tasks. This was crucial in improving overall user experience and efficiency. To further boost accessibility, a browser-compatible version of the application was selected, ensuring it functions seamlessly across a wide range of commonly used web browsers.

INPUT AND OUTPUT DESIGN

OBJECTIVES

INPUT DESIGN: ENHANCING DATA ACCURACY AND USER EXPERIENCE

Input Design refers to the process of transforming user-defined input requirements into a structured, computer-friendly format. Its primary purpose is to ensure data is entered accurately and efficiently into a digital system. By minimizing input errors, it enables the management to obtain reliable information and supports sound decision-making through computerized operations.

To handle high volumes of data seamlessly, input design focuses on creating intuitive and user-centric data entry screens. These interfaces are developed to simplify the input process, reduce mistakes, and support various operations such as data entry, validation, and record browsing. Well-designed input screens enhance user interaction and ensure all necessary functions are accessible within a single interface.

During data entry, the system validates each input for accuracy and relevance. Inputs are collected via interactive screens, and real-time prompts or guidance messages are provided to help users avoid confusion. These feedback mechanisms ensure users understand what is expected, helping to prevent errors. Ultimately, the aim of input design is to produce a clear, logical, and user-friendly input structure that promotes accuracy, speed, and ease of use.

OUTPUT DESIGN:

A high-quality output is one that effectively meets the expectations of its users by presenting processed data in a clear, concise, and actionable format. Outputs serve as the bridge between a system's internal processes and its end users—delivering results, insights, and documentation necessary for informed decision-making.

In the context of system design, **output design** involves planning how information will be displayed on screens for immediate use and how it will be presented in printed or digital documents for records and reports. Because outputs are often the most visible component of a system to users, their design plays a crucial role in user satisfaction and operational efficiency.

KEY PRINCIPLES OF OUTPUT DESIGN:

1. **User-Centric Output Development**: Designing outputs must follow a structured approach to ensure the results align with user needs. The output should be intuitive, readable, and formatted in a way that enhances the user's ability to interpret and act on the information provided.
2. **Effective Presentation Methods**: Output designers must select appropriate formats to present information—be it dashboards, reports, notifications, charts, or printed documents. The choice depends on the urgency, complexity, and type of data being communicated.
3. **Format Creation and Customization**: Outputs may take the form of digital documents, printed reports, interactive charts, or summaries. These should be tailored to reflect the specific information requirements of each user or department, ensuring relevance and clarity.

INTRODUCTION OF PYTHON

Python is a powerful, high-level, interpreted programming language known for its simplicity, readability, and versatility. As an open-source language, it supports a wide range of programming paradigms, including object-oriented, procedural, functional, and imperative programming.

One of Python's most distinctive features is its emphasis on clean and readable code. Unlike many traditional languages that use braces or explicit delimiters, Python relies on **indentation** to define code blocks. This unique approach not only improves readability but also encourages well-structured programming practices.

Python enables developers to write fewer lines of code to accomplish tasks that would typically require more complex syntax in languages like C++ or Java. Its streamlined syntax makes it ideal for both beginners and experienced programmers working on anything from small scripts to large-scale applications.

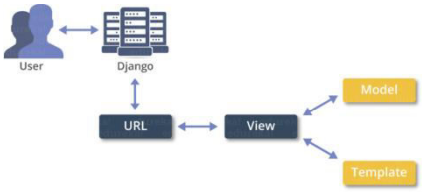
The reference implementation of Python, known as **CPython**, is freely available and developed under an open-source license. It is maintained by the **Python Software Foundation (PSF)** and is supported by a vast global community of contributors.

DJANGO

Django is a robust, high-level web framework crafted in Python, designed to simplify the development of dynamic, data-driven websites. It promotes fast development cycles, modular design, and clean coding practices, allowing developers to build powerful web applications efficiently.

One of Django's biggest strengths lies in its “don’t repeat yourself” (DRY) philosophy, which encourages code reuse and minimizes redundancy. The framework comes pre-packaged with features like an admin panel, authentication system, and ORM (Object-Relational Mapping)—reducing the need for writing boilerplate code.

This feature is a major productivity booster, especially during development or when building content management systems, dashboards, or internal tools.



SYSTEM SPECIFICATION:

HARDWARE REQUIREMENTS:

System : Pentium IV 2.4 GHz.

Hard Disk : 40 GB.

Floppy Drive : 1.44 Mb.

Monitor : 14’ Colour Monitor.

Mouse : Optical Mouse.

Ram : 512 Mb.

SOFTWARE REQUIREMENTS:

Operating system : Windows 7 Ultimate.

Coding Language : Python.

Front-End : Python.

Designing : Html,css,javascript.

Data Base : MySQL.

SYSTEM TEST

The main objective of software testing is to identify defects and ensure quality. Testing is a critical phase in the software development lifecycle where the system is examined under controlled conditions to uncover errors, gaps, or unmet requirements. It helps verify that the software behaves as expected and meets both functional and non-functional requirements.

Testing is more than just error detection—it’s a process to ensure:

- Reliability of the system
- Accuracy in performance
- User satisfaction with the product
- Stability under different usage scenarios

By simulating real-world usage, testing helps in uncovering hidden issues and confirming that the system can handle edge cases and unexpected inputs without failure.

Different types of testing such as unit testing, integration testing, system testing, and user acceptance testing are performed to validate individual components and the system as a whole. Each type is designed to evaluate specific aspects of the software, ensuring thorough coverage and risk mitigation before deployment.

TYPES OF TESTS

UNIT TESTING

Unit testing focuses on verifying the correctness of individual components or modules within a software application. It is the first level of testing, performed soon after the development of a particular function or unit of code.

This type of testing aims to confirm that the internal logic of a unit works as intended, producing the correct output for a given set of inputs. Each conditional path, loop, and logic branch is thoroughly tested to validate functionality.

Unit testing is typically white-box testing, meaning it is done with full knowledge of the code's internal structure. It is a code-level validation, making it highly precise and effective in catching logic errors, boundary conditions, and unexpected behavior early in the development cycle.

Developers usually write unit tests to:

- Validate isolated units of business logic.
- Confirm specific configurations and outcomes.
- Ensure each piece of code meets its functional specification.

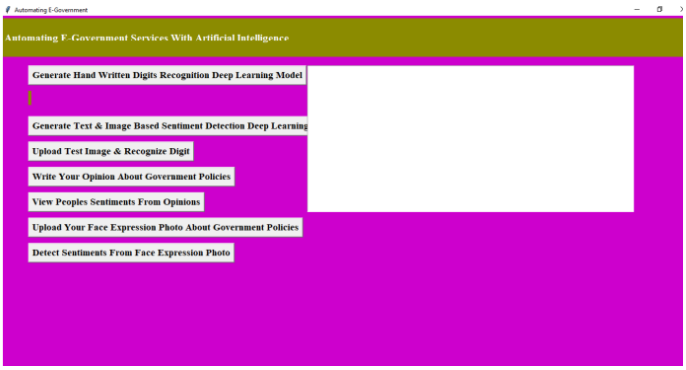
By performing unit testing, developers can detect issues early, simplify debugging, and reduce the risk of errors propagating to higher stages of integration and system testing.

TEST STRATEGY AND APPROACH

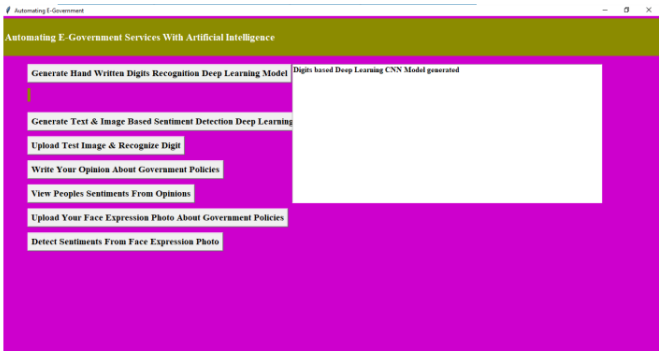
MANUAL FIELD TESTING OVERVIEW

Manual field testing involves executing the application in a real or simulated environment to verify its functionality and responsiveness. Detailed functional test cases are prepared to ensure each component of the user interface operates correctly and efficiently.

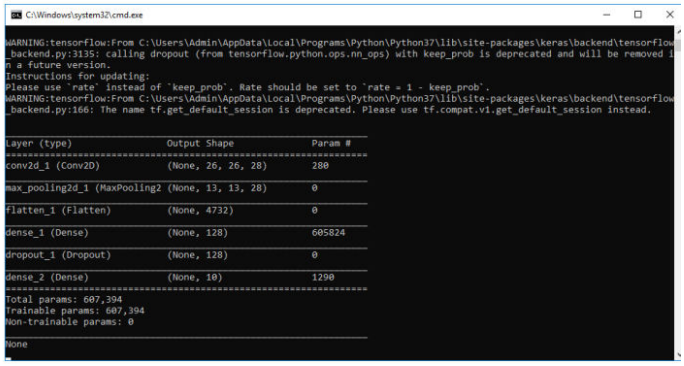
SCREEN SHOTS



In above screen click on 'Generate Hand Written Digits Recognition Deep Learning Model' button to generate CNN digits recognition model



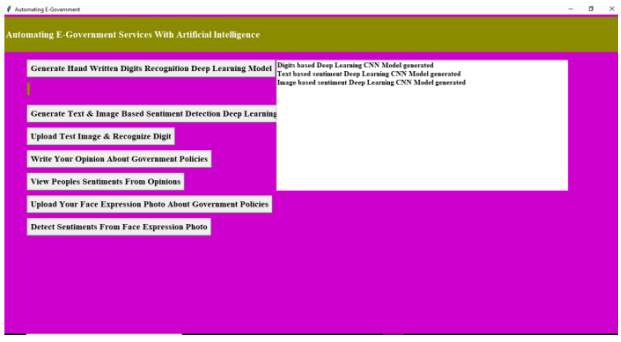
In above screen we can see digits model generated and CNN layer details you can see black console



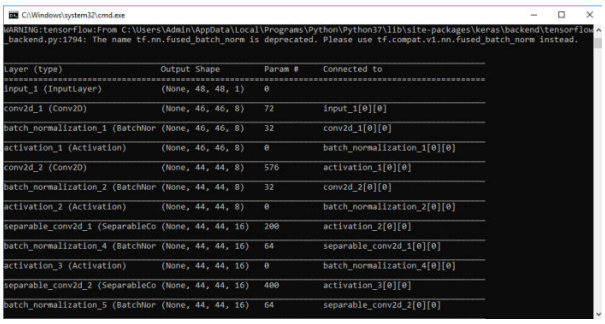
CNN Layer Visualization and Model Generation - Unique Content

In the screen above, we observe the Conv2D layers, which represent convolutional layers within a CNN (Convolutional Neural Network). These layers are responsible for extracting visual features from the input image. The initial convolution layer processes the image at a size of **26x26**, while the subsequent layer reduces it further to **13x13**, continuing this pattern as the depth of the network increases.

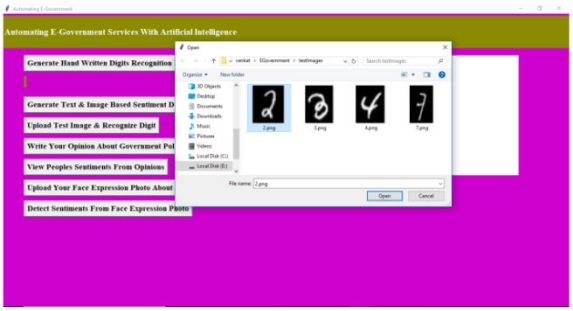
To proceed with creating the sentiment analysis model, simply click the **"Generate Text & Image Based Sentiment Detection Deep Learning Model"** button. This action will trigger the creation of a CNN-based architecture that combines both textual and visual data for a more comprehensive sentiment detection system.



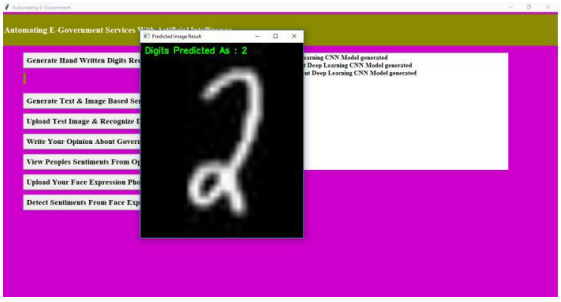
In above screen we can see text and image based CNN model generated. See black screen for more details



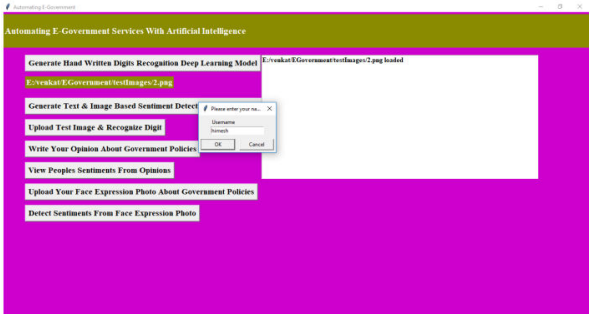
Now click on 'Upload Test Image & Recognize Digit' button to upload digit images and to get name of that digit. All digit images saved inside test Images folder.



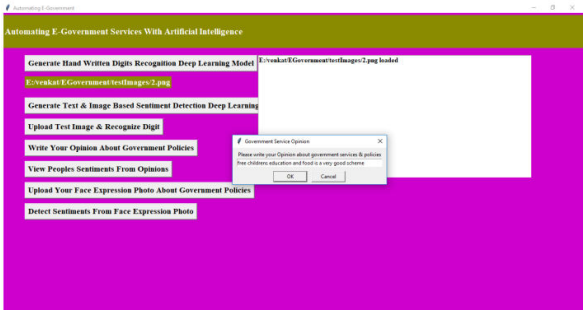
In above screen I am uploading image which contain digit 2 and below is the output of detection



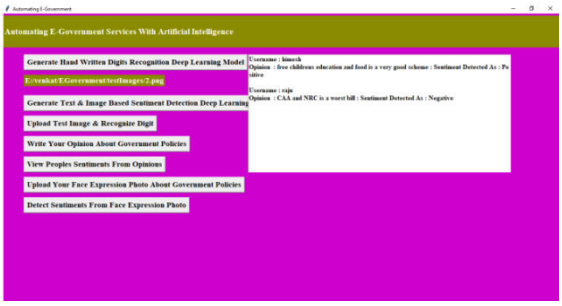
In above screen we can see Digits Predicted as: 2. Now click on 'Write Your Opinion About Government Policies' button to write some comments on government policy



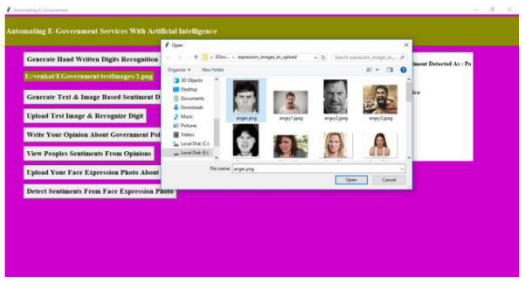
In above screen before writing opinions we need to write username after writing username click ok button to get below screen



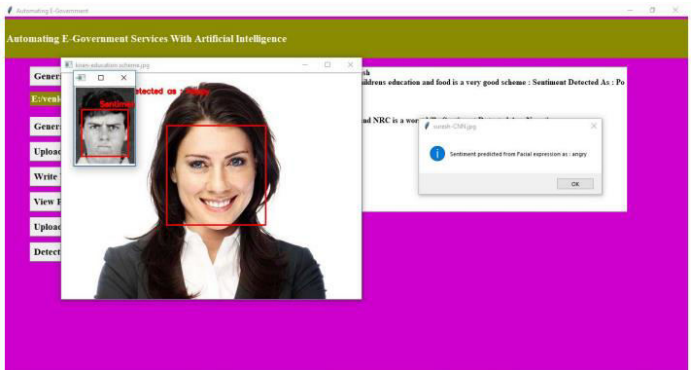
In above screen I wrote some comment on some scheme and application detect sentiment from it as positive or negative. Now click on 'View Peoples Sentiments From Opinions' button to view all opinions from past users.



In above screen text area we can see opinions from all users and in first opinion we got sentiment detected as positive which means user is satisfy with that scheme and for second opinion we got sentiment as negative which means user not happy. Similarly user can upload their image with facial expression which describe whether user is happy or angry



In above screen I am uploading one anger face image and then application ask to write username and referring scheme name. similarly any number of users can upload their images. Now click on 'Detect Sentiments From Face Expression Photo' button to get all images and its detected sentiments



In above screen we can see all images with facial expression are identified with their sentiments. In dialog box also we can see sentiment result. Similarly you can enter any number of comments or facial images to detect their sentiments.

CONCLUSION

Integrating AI into E-Government Systems: A Framework for Advancement With rapid developments in artificial intelligence (AI) and deep learning, government bodies worldwide are exploring these technologies to enhance public services and administrative efficiency. Despite the potential, several obstacles hinder their widespread adoption—ranging from a shortage of skilled professionals and limited computational infrastructure to issues surrounding trust, transparency, and the interpretability of AI systems.This study begins by clearly defining the concepts of AI and e-government, followed by an overview of global e-government development indices. Focusing on the Gulf Cooperation Council (GCC) countries as a case study, we introduce a strategic framework aimed at managing government information systems across their entire lifecycle.We further outline how deep learning methods can streamline and automate various government services, reducing human effort and error. Additionally, we propose a robust, intelligent platform tailored for the development and deployment of AI-driven applications within the public sector.The central objective of this work is to present innovative models and platforms that incorporate modern AI capabilities into government operations—enhancing service delivery through increased efficiency, transparency, and citizen trust.

REFERENCES:

[1] K. He, X. Zhang, S. Ren, and J. Sun, ``Deep residual learning for image recognition," in Proc.IEEE Conf. Comput. Vis. Pattern Recognit.,Jun. 2016, pp. 770_778.

[2] Y.-D. Zhang, Y. Zhang, X.-X. Hou, H. Chen,and S.-H. Wang, ``Sevenlayer deep neuralnetwork based on sparse autoencoder for voxelwise detection of cerebral microbleed," Multimedia Tools Appl., vol. 77, no. 9, pp.10521_10538, May 2018.

[3] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, ``Translating videos to natural language using deep recurrent neural networks," 2014,arXiv:1412.4729. [Online]. Available: https://arxiv.org/abs/1412.4729

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I.Antonoglou, V.Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N.Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, ``Mastering the game of Go with deep neural networks and tree search," Nature, vol. 529, no. 7587, pp. 484_489, 2016.

[5] C. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.

[6] Y. LeCun, Y. Bengio, and G. Hinton, ``Deeplearning," Nature, vol. 521, no. 7553, pp. 436_444, 2015.

[7] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, ``Towards a better understanding of context and context- awareness," in Proc. Int. Symp. Handheld Ubiquitous Comput. Berlin, Germany: Springer, 1999, pp. 304_307.

[8] C. Dwork, ``Differential privacy," in Encyclopedia of Cryptography and Security, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA, USA: Springer, 2011.

[9] L. Bottou, ``Large-scale machine learningwith stochastic gradient descent," in Proc. COMPSTAT, 2010, pp. 177_186.

[10] A. Kankanhalli, Y. Charalabidis, and S. Mellouli, ``IoT and AI for smart government: A research agenda," Government Inf. Quart., vol. 36, no. 2,pp. 304_309, 2019.

[11] J. B. Lee and G. A. Porumbescu,``Engendering inclusive e-government use through citizen IT training programs," Government Inf. Quart., vol. 36, no. 1, pp. 69_76, 2019.

[12] R. Santa, J. B. MacDonald, and M. Ferrer,``The role of trust in e-Government effectiveness, operational effectiveness and user satisfaction: Lessons from Saudi Arabia in e-G2B," Government Inf. Quart., vol. 36, no. 1, pp. 39_50, 2019.

[13] J. D. Twizeyimana and A. Andersson, ``The public value of E-Government_A literature review," Government Inf. Quart., vol. 36, no. 2, pp. 167_178, 2019.

[14] M. Güler, E. Mukul, and G. Büyüközkan,``Analysis of e-government strategies with hesitant fuzzy linguistic multi-criteria decision making techniques," in Proc. Int. Conf. Intell. Fuzzy Syst. Cham, Switzerland:Springer, 2019, pp. 1068_1075.

[15] P. G. Nixon, V. N. Koutrakou, and R. Rawal, eds. Understanding E- Government inEurope: Issues and Challenges. Evanston, IL,USA: Routledge, 2010.

[16] D. A. D. Putra, K. A. Jasmi, B. Basiron, M. Huda, A. Maseleno, K. Shankar, and N. Aminudin, ``Tactical steps for e-government development," Int. J. Pure Appl. Math., vol. 119,no. 15, pp. 2251_2258, 2018.

[17] A. T.-K. Ho, ``Reinventing localgovernments and the e-government initiative," Public Admin. Rev., vol. 62, no. 4, pp. 434_444,2002.

[18] M. Yildiz, ``E-government research: Reviewing the literature, limitations, and ways forward," Government Inf. Quart., vol. 24, no. 3,pp. 646_665, 2007.

[19] H. J. Scholl, E-Government: Information, Technology, and Transforma-tion. Evanston, IL, USA: Routledge, 2015. [20] United Nation E-Government Survey. Accessed: Jul. 2019. [Online]. Available: https://publicadministration.un.org/egovkb/en-us/Reports/UNE