**A**

**PROJECT REPORT ON**

# AUTOMATING E-GOVERNMENT SERVICE WITH ARTIFICIAL INTELLIGENCE



SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF

**DIPLOMA IN COMPUTER ENGINEERING**

## SUBMITTED BY

**V.RAMYA (20090-CM-080)**

**Under the valuable guidance of**

**B.KAVYA SREE, B. Tech**

**DEPARTMENT OF COMPUTER ENGINEERING**

# VEMUGANTI MANOHAR RAO POLYTECHNIC

**Rampur, Warangal (TS)-506151**

**(2020-2023)**

## VEMUGANTI MANOHAR RAO POLYTECHNIC

**(Sponsored by Manohara Educational Society)**

**Rampur, Warangal (T.S.)-506 151**



# CERTIFICATE

This is to certify that the project entitled

“**AUTOMATING E-GOVERNMENT SERVICES WITH ARTIFICIAL INTELLIGENCE**”

Is carried out by **V.RAMYA(20090-CM-080)**, in partial fulfilment of the

requirement for **DIPLOMA IN COMPUTER ENGINEERING**, by the **State Board of Technical Education & Training (TS), Hyderabad**, is an award of bonafide work carried out by them under our guidance and supervision. The results embodied in this project report have not been submitted to any other institutions for the award of Diploma.

**PROJECT GUIDE HEAD OF DEPARTMANT**

**Miss. B.KAVYA SREE Mr. B. VENKANNA**

**B. tech M. tech**

**EXTERNAL EXAMINER PRINCIPAL**

**Dr . P. YELLAIAH**

**M. tech , Ph .D**

## DECLARATION

I declare that the work reported in the present entitled

**“AUTOMATING E-GOVERNANCE SERVICES WITH ARTIFICIAL**

**INTELLIGENCE”** is record of work done in the Department of Computer Engineering, VMR Polytechnic, affiliated to **SBTET,** Hyderabad and **AICTE,** New Delhi.

The result of this work has not been submitted to any university or any other institution for the award of any degree.

**SUBMITTED BY**

V.RAMYA (20090-CM-080)

# 

# ACKNOWLEDGMENT

We wish to express our heartfelt gratitude to **Sri. Dr .P . Yellaiah ,** Principal of

**VEMUGANTI MANOHAR RAO POLYTECHNIC,** Rampur, Warangal for

providing all the facilities required for the seminar presentation.

We wish to take opportunity to express our sincere thanks to **Sri .B .Venkanna,**

**H.O.D.** of Computer Engineering Department for the interest technical support and suggestions during the seminar leading to our success.

We are obliged and grateful to our guide **Miss. B. KAVYA SREE,** Lecturer in Computer Engineering Department for her valuable suggestion and sagacious guidance in all aspects during the technical of our training.

We hereby express our sincere thanks regards to all teaching staff and nonteaching staff, parents and friends who extended their helping hand in the accomplishment of the project.

# ABSTRACT

Artificial Intelligence (AI) has recently advanced the state-of-art results in an ever-growing number of domains. However, it still faces several challenges that hinder its deployment in the e-government applications-both for improving the egovernment systems and the e-government-citizens interactions. In this paper, we address the challenges of e-government systems and propose a framework that utilizes AI technologies to automate and facilitate e-government services. Specifically, we first outline a framework for the management of e-government information resources. Second, we develop a set of deep learning models that aim to automate several e-government services.

Third, we propose a smart e-government platform architecture that supports the development and implementation of AI applications of e-government. Our overarching goal is to utilize trustworthy AI techniques in advancing the current state of e-government services in order to minimize processing times, reduce costs, and improve citizens' satisfaction.

# INDEX

**TABLE OF CONCEPTS**

1. **INTRODUCTION** 
   1. **Introduction to AI based on Machine learning…………………….1**
2. **LITERATURE SURVEY** 
   1. **Deep residual learning for E-Government………………….……….4**
3. **SYSTEM ANALYSIS** 
   1. **Existing System………………………………………………………...6**
   2. **Proposed System………..………………………………………..…….6**
   3. **System Requirements……………………………………..…………...7**
4. **SYSTEM DESIGN** 
   1. **System Architecture………………………………………………….8**
   2. **Data Flow Diagrams…………………………………………………...9**
   3. **Implementations……………………………………………...……….13**
   4. **Algorithm……………………………………………………..……….14**

1. **SOFTWARE ENVIRONMENT………………………………..……....16**
2. **SYSTEM TEST……………………...…………………………………..22**
3. **CODE…………………………………………………………………….25**
4. **IMPLEMENTATION 0F PROCESS USING IMAGES……………..33**
5. **FUTURE ENHANCEMENT……………………………….……...…..40**
6. **CONCLUSION……….………………………………….………….….41**
7. **BIBILIOGRAPHY……………………………………….……………42**

## 1.INTRODUCTION

Artificial Intelligence (AI) has been around for some decades in several theoretical forms and complicated systems; however, only recent advances in computational powers and big data have enabled AI to achieve outstanding results in an ever-growing number of domains. For example, medical applications natural language and several other domains. AI can be defined as the ability of a computer to imitate the intelligence of human behaviour while improving its own performance. AI is not only robotics, rather an intelligent behaviour of an autonomous machine that describes the brain of the machine and not its body; it can drive a car, play a game, and perform diverse sophisticated jobs. AI is a field that falls at the intersections of several other domains, including Deep Learning,, Natural Languages Processing Context Awareness and Data Security and Privacy. Machine Learning (ML) is the ability of an algorithm to learn from prior data in order to produce a smart behaviour and make correct decisions in various situations that it has never faced before. ML algorithms are enabled by training a computational model, which is the process of exposing an algorithm to a large dataset (e.g., citizens’ demographics) in order to predict future behaviours (e.g., employment rates). The process of learning from prior datasets is known as a supervised learning. Unlike traditional ML algorithms, Deep Learning, a subfield of ML, has emerged to outcome the limitations of prior ML algorithms. Deep learning can be defined as a mapping function that maps raw input data (e.g., a medical image) to the desired output (e.g., diagnosis) by minimizing a loss function using some optimization approach, such as stochastic gradient descent.

Deep learning algorithms, inspired by the neural networks in the human brain, are built with a large number of hierarchical artificial neural networks that map the raw input data (inserted at the input layer) to the desired output (produced at the output layer) through a large number of layers (known as hidden layers), and thus the name deep learning. The hidden layers are responsible for the actual mapping process, which is a series of simple but nonlinear mathematical operations (i.e., a dot product followed by a nonlinear process). The main advantage of deep learning is that it does not require feature engineering. Despite the fact that deep learning has improved the state-of-art results in several domains, it is still evident that egovernment applications face several challenges regarding adapting deep learning . First, given the recent and rapid advances in the deep learning domain, it is becoming more difficult to find experts of this technology who are capable of developing efficient and reliable AI applications, especially in third world countries. Second, the development lifecycle of AI projects, specially deep learning, has introduced a new set of development challenges. In particular, traditional software development focuses on meeting a set of required functional and non-functional requirements; in contract, deep learning development focuses on optimizing a specific metric based on a large set of parameters, which is done in a unsystematic search approach. Third, integrating AI and deep learning applications in e-government services requires strong policies and measures on data security and privacy.

However, there are still challenges that hinder the creation of concrete standards for data security and privacy, including citizen-government trust, transparency, and other technical difficulties related to developing and implementing secure systems. E-government is the application of employing advanced electronic techniques–and web services–to present, exchange, and advance the government’s services for citizens and businesses with a goal of improving the productivity while reducing the cost. E-government plays a critical role in advancing the economy of the government, citizens, and industry, especially for developing countries. It facilitates the business-to-business transactions and tasks , brings customers closer to businesses , allow productive interactions between the government and citizens, government and enterprises and inter-agency and relationships in more convenient, transparent and economic ways. The ultimate goal of the e-government is to enhance the quality and efficiency of the government services while reducing cost. Moreover, implementing e-government

applications can foster several other advantages including, but not limited to, the following: • Transparency

* Trust
* Citizen participation
* Environment support

In contrast, implementing e-government applications still faces several challenges, including the following:

* Trust
* Lack of experts
* Inaccessibility
* Security

While there are several studies conducted for enhancing e-government services, only a few of them address utilizing recent advances in AI and deep learning in the automation of egovernment services . Therefore, there is still an urgent need to utilize state-of-the-art AI techniques and algorithms to address e-government challenges and needs. In this paper, we propose a novel framework that utilizes recent advances in AI to improve the e-government systems and their interactions with the citizens. First, we propose a framework to automate and facilitate the management of e-government systems using AI techniques. Second, we develop and present several deep learning models that aim at automating e-government services for Arabic speaking countries including automatic recognition of hand-written digits and letters and sentiment analysis. Third, we propose an platform for smart e-government services development and implementation. The rest of this paper is organized as follows: Section two presents the current state of the national and international e-government performance indices. Section three proposes an advanced management framework for e-government information resources. Section four presents our deep learning models. Section five suggests a platform for smart e-government services. The conclusion comes in the sixth Section.

## 2.LITERATURE SURVEY

**2.1** **Deep residual learning for E-Government :**

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers - 8× deeper than VGG nets [40] but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR10 with 100 and 1000 layers. The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions1, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

In order to detect the cerebral microbleed (CMB) voxels within brain, we used susceptibility weighted imaging to scan the subjects. Then, we used under sampling to solve the accuracy paradox caused from the imbalanced data between CMB voxels and non-CMB voxels. we developed a seven-layer deep neural network (DNN), which includes one input layer, four sparse autoencoder layers, one softmax layer, and one output layer. Our simulation showed this method achieved a sensitivity of 95.13%, a specificity of 93.33%, and an accuracy of 94.23%.

The result is better than three state-of-the-art approaches.

Solving the visual symbol grounding problem has long been a goal of artificial intelligence. The field appears to be advancing closer to this goal with recent breakthroughs in deep learning for natural language grounding in static images. In this paper, we propose to translate videos directly to sentences using a unified deep neural network with both convolutional and recurrent structure. Described video datasets are scarce, and most existing methods have been applied to toy domains with a small vocabulary of possible words. By transferring knowledge from 1.2M+ images with category labels and 100,000+ images with captions, our method is able to create sentence descriptions of open-domain videos with large vocabularies. We compare our approach with recent work using language generation metrics, subject, verb, and object prediction accuracy, and a human evaluation.

## 3.SYSTEM ANALYSIS

**3.1 EXISTING SYSTEM:**

Recently, many countries have adopted e-government services in various departments and many autonomous applications . While there are several studies conducted for enhancing egovernment services, only a few of them address utilizing recent advances in AI and deep learning in the automation of e-government services. Therefore, there is still an urgent need to utilize state-of-the-art AI techniques and algorithms to address e-government challenges and needs.

In contrast, implementing e-government applications still faces several challenges, including the following:

* Trust
* Lack of experts
* Inaccessibility
* Security

* 1. **PROPOSED SYSTEM:**

In this paper we are describing concept to automate government services with Artificial Intelligence technology such as Deep Learning algorithm called Convolution Neural Networks (CNN). Government can introduce new schemes on internet and peoples can read news and notifications of such schemes and then peoples can write opinion about such schemes and this opinions can help government in taking better decisions. To detect public opinions about schemes automatically we need to have software like human brains which can easily understand the opinion which peoples are writing is in favour of positive or negative.

To build such automated opinion detection author is suggesting to build CNN model which can work like human brains. This CNN model can be generated for any services and we can make it to work like automated decision making without any human interactions. To suggest this technique author already describing concept to implement multiple models in which one model can detect or recognize human hand written digits and second model can detect sentiment from text sentences which can be given by human about government schemes. In our extension model we added another model which can detect sentiment from person face image. Person face expressions can describe sentiments better than words or sentences. So our extension work can predict sentiments from person face images.

* 1. **SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

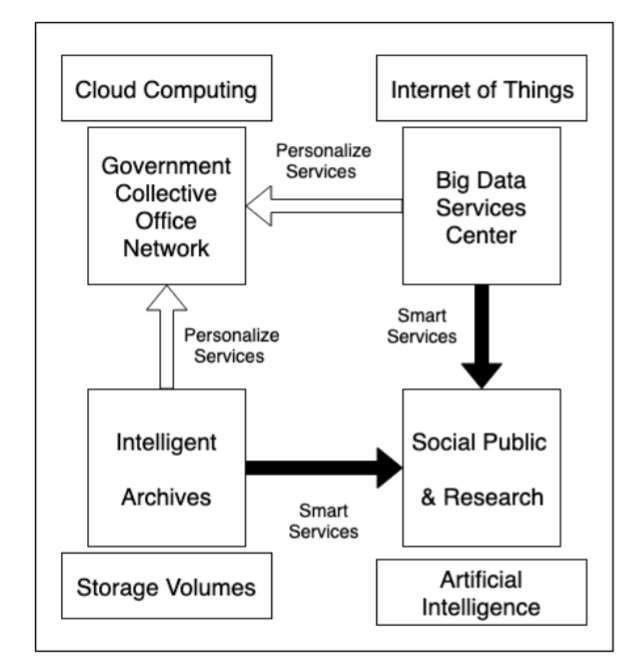
* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 512 Mb.

**SOFTWARE REQUIREMENTS:**

* **Operating System:** Windows • **Coding Language**: Python

## 4.SYSTEM DESIGN

**4.1 SYSTEM ARCHITECTURE:**



**4.2 DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modelling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is
4. modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
5. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

**NO**

**User**

**Check**

**Unauthorize**

**d user**

Generate Text & Image

Based Sentiment Detection

Upload

Test

Image

&Recognize Digit

Write

Your Opinion About

Government Policies

**End**

**process**

Upload

Your

Face

Photo

Expression

About

View

People

Sentiments

From Opinions

Detect

From

Sentiments

Face Expression Photo

### YES

Generate hand Written Digits

Recognition Deep Learning

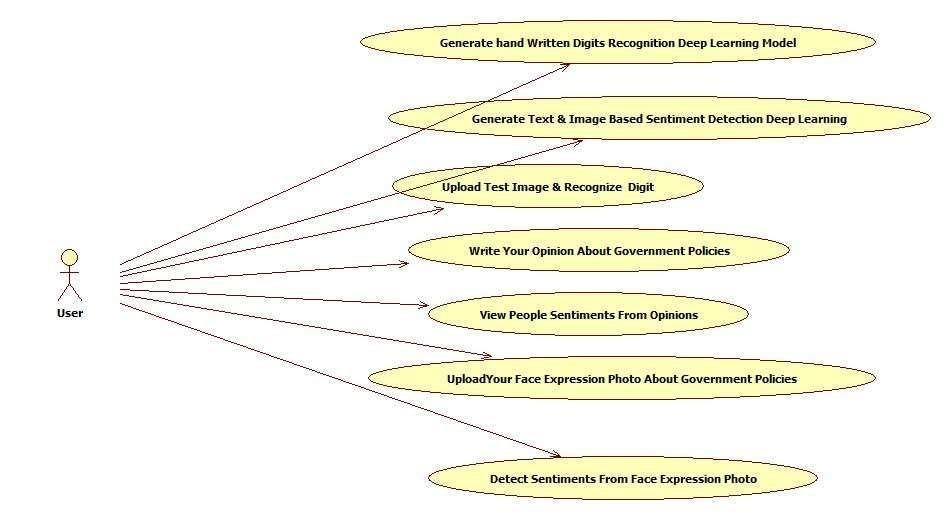
**UML DIAGRAMS:**

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

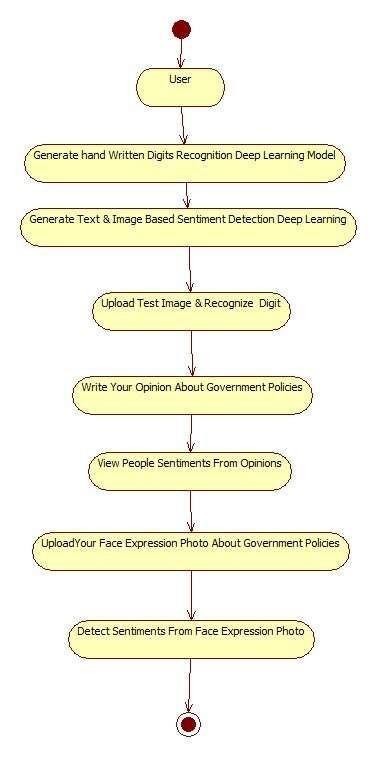
**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can depicted.



**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**4.3 IMPLEMENTATION**:

**MODULES:**

1. Generate Hand Written Digits Recognition Deep Learning Model: using this model we are building CNN based hand written model which take digit image as input and then predict the name of digit. CNN model can be generated by taking two types of images called train (train images contain all possible shapes of digits human can write in all possible ways) and test (Using test images train model will be tested whether its giving better prediction accuracy). Using all train images CNN will build the training model. While building model we will extract features from train images and then build a model. While testing also we will extract features from test image and then apply train model on that test image to classify it.
2. Generate Text & Image Based Sentiment Detection Deep Learning Model: using this module we will generate text and image based sentiment detection model. All possible positive and negative words will be used to generate text based sentiment model. All different types of facial expression images will be used to generate image based sentiment model. Whenever we input text or image then train model will be applied on that input to predict its sentiments.
3. Upload Test Image & Recognize Digit: By using this module we will upload text image and apply train model to recognize digit.
4. Write Your Opinion About Government Policies: using this module we will accept user’s opinion and then save that opinion inside application to detect sentiment from opinion.
5. View Peoples Sentiments From Opinions: using this module user can see all users opinion and their sentiments detected through CNN model.
6. Upload Your Face Expression Photo About Government Policies: using this module user will upload his image with facial expression which indicates whether user is satisfy with this scheme or not.

Detect Sentiments From Face Expression Photo: using this module different users can see the facial expression image and detected sentiment which is uploaded by past users

**4.4 ALGORITHMS***:*

**CNN:**

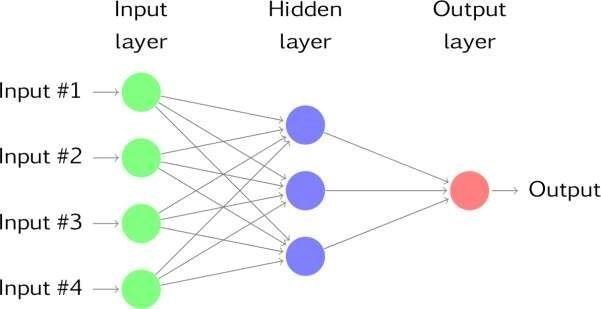
In this paper author describing concept to automate government services with Artificial Intelligence technology such as Deep Learning algorithm called Convolution Neural Networks (CNN). Government can introduce new schemes on internet and peoples can read news and notifications of such schemes and then peoples can write opinion about such schemes and this opinions can help government in taking better decisions. To detect public opinions about schemes automatically we need to have software like human brains which can easily understand the opinion which peoples are writing is in favour of positive or negative.

To build such automated opinion detection author is suggesting to build CNN model which can work like human brains. This CNN model can be generated for any services and we can make it to work like automated decision making without any human interactions. To suggest this technique author already describing concept to implement multiple models in which one model can detect or recognize human hand written digits and second model can detect sentiment from text sentences which can be given by human about government schemes. In our extension model we added another model which can detect sentiment from person face image. Person face expressions can describe sentiments better than words or sentences. So our extension work can predict sentiments from person face images.

To demonstrate how to build a convolutional neural network based image classifier, we shall build a 6 layer neural network that will identify and separate one image from other. This network that we shall build is a very small network that we can run on a CPU as well. Traditional neural networks that are very good at doing image classification have many more parameters and take a lot of time if trained on normal CPU. However, our objective is to show how to build a real-world convolutional neural network using TENSORFLOW.

Neural Networks are essentially mathematical models to solve an optimization problem. They are made of neurons, the basic computation unit of neural networks. A neuron takes an input (say x), do some computation on it (say: multiply it with a variable w and adds another variable b) to produce a value (say; z= wx+b). This value is passed to a non-linear function called activation function (f) to produce the final output(activation) of a neuron. There are many kinds of activation functions. One of the popular activation function is Sigmoid. The neuron which uses sigmoid function as an activation function will be called sigmoid neuron. Depending on the activation functions, neurons are named and there are many kinds of them like RELU, TanH.

If you stack neurons in a single line, it’s called a layer; which is the next building block of neural networks. See below image with layers



To predict image class multiple layers operate on each other to get best match layer and this process continues till no more improvement left.

## 5.SOFTWARE ENVIRONMENT

**Machine Learning : -**

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data*.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

**Need for Machine Learning**

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be

deployed in real life.

**Applications of Machines Learning :-**

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting

**Advantages of Machine learning :-**

* Easily identifies trends and patterns -
* No human intervention needed (automation)
* Continuous Improvement
* Handling multi-dimensional and multi-variety data
* Wide Applications

**Disadvantages of Machine Learning :-**

* Data Acquisition
* Time and Resources
* Interpretation of Results \* High error-susceptibility

**Python :**

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
* GUI Applications (like Kivy, Tkinter, PyQtetc. )
* Web frameworks like Django (used by YouTube, Instagram, Dropbox)
* Image processing (like Opencv, Pillow)
* Web scraping (like Scrapy, Beautiful Soup, Selenium)
* Test frameworks
* Multimedia

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviours. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Advantages of Python :-**

* + Extensive Libraries
  + Extensible
  + Embeddable
  + Improved Productivity
  + IOT Opportunities

**Advantages of Python Over Other Languages**

* + Less Coding
  + Affordable
  + Python is for Everyone

**Disadvantages of Python**

* + Speed Limitations
  + Weak in Mobile Computing and Browsers
  + Design Restrictions
  + Underdeveloped Database Access Layers
  + Simple

**Modules Used in Project :-**

**Tensorflow :**

TensorFlow is a [free an](https://en.wikipedia.org/wiki/Free_software)[d open-sour](https://en.wikipedia.org/wiki/Open-source_software)[ce software library for dataflow and differentiable programming acr](https://en.wikipedia.org/wiki/Library_(computing))oss a range of tasks. It is a symbolic math library, and is also used for [machine learning ap](https://en.wikipedia.org/wiki/Machine_learning)plications such as [neural networks. I](https://en.wikipedia.org/wiki/Neural_networks)t is used for both research and production a[t Google.](https://en.wikipedia.org/wiki/Google)

**Numpy :**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features too.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures.. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyse. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics,

Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython sh](http://ipython.org/)ells, the [Jupyter No](http://jupyter.org/)tebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots an](https://matplotlib.org/tutorials/introductory/sample_plots.html)[d thumbnail gallery.](https://matplotlib.org/gallery/index.html)

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

## 6. SYSTEM TEST

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**TYPES OF TESTS USED IN PROJECT :**

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

### Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

### Test objectives

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

### Features to be tested

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

*Functional testing is centered on the following items:*

**Valid Input** : identified classes of valid input must be accepted.

**Invalid Input** : identified classes of invalid input must be rejected.

**Functions** : identified functions must be exercised.

**Output** : identified classes of application outputs must be exercised.

**Systems/Procedures** : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

## 7.CODE USED IN PROGRAM

from tkinter import messagebox from tkinter import \* from tkinter import simpledialog import tkinter from tkinter import filedialog from tkinter.filedialog import askopenfilename import matplotlib.pyplot as plt import numpy as np import joblib

from keras.models import load\_model from

keras.preprocessing.image import img\_to\_array import cv2

from keras.models import model\_from\_json from keras.preprocessing import image from keras.optimizers import Adam from keras.utils import np\_utils from keras.preprocessing import image import os from numpy import dot from numpy.linalg import norm from keras.models import Sequential from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D

import imutils import nltk

main = tkinter.Tk() main.title("Automating E-Government") main.geometry("1300x1200")

global filename global text\_sentiment\_model

EMOTIONS = ["angry","disgust","scared", "happy", "sad", "surprised","neutral"] global face\_detection global image\_sentiment\_model global digits\_cnn\_model

def digitModel():

global digits\_cnn\_model with

open('models/digits\_cnn\_model.json', "r") as json\_file:

loaded\_model\_json = json\_file.read()

digits\_cnn\_model = model\_from\_json(loaded\_model\_json)

digits\_cnn\_model.load\_weights("models/digits\_cnn\_weights.h5")

digits\_cnn\_model.\_make\_predict\_function() print(digits\_cnn\_model.summary()) text.insert(END,'Digits based Deep Learning CNN Model generated\n')

def sentimentModel():

global text\_sentiment\_model global image\_sentiment\_model global face\_detection

text\_sentiment\_model = joblib.load('models/sentimentModel.pkl') text.insert(END,'Text based

sentiment Deep Learning CNN Model generated\n')

face\_detection = cv2.CascadeClassifier('models/haarcascade\_frontalface\_default.xml')

image\_sentiment\_model = load\_model('models/\_mini\_XCEPTION.106-0.65.hdf5', compile=False)

text.insert(END,'Image based sentiment Deep Learning CNN Model generated\n')

print(image\_sentiment\_model.summary())

def digitRecognize():

global filename

filename = filedialog.askopenfilename(initialdir="testImages")

pathlabel.config(text=filename) text.delete('1.0', END) text.insert(END,filename+" loaded\n");

imagetest = image.load\_img(filename, target\_size = (28,28), grayscale=True)

imagetest = image.img\_to\_array(imagetest) imagetest = np.expand\_dims(imagetest, axis = 0)

pred = digits\_cnn\_model.predict(imagetest.reshape(1, 28, 28, 1))

predicted = str(pred.argmax()) imagedisplay = cv2.imread(filename) orig = imagedisplay.copy() output = imutils.resize(orig, width=400)

cv2.putText(output, "Digits Predicted As : "+predicted, (10, 25),

cv2.FONT\_HERSHEY\_SIMPLEX,0.7, (0, 255, 0), 2) cv2.imshow("Predicted Image Result", output)

cv2.waitKey(0)

def opinion():

user = simpledialog.askstring("Please enter your name", "Username")

opinion = simpledialog.askstring("Government Service Opinion", "Please write your Opinion about government services & policies") f = open("Peoples\_Opinion/opinion.txt",

"a+")

f.write(user+"#"+opinion+"\n")

f.close()

messagebox.showinfo("Thank you for your opinion", "Your opinion saved for reviews")

def stem(textmsg): stemmer =

nltk.stem.PorterStemmer() textmsg\_stem = '' textmsg = textmsg.strip("\n") words = textmsg.split(" ")

words = [stemmer.stem(w) for w in words]

textmsg\_stem = ' '.join(words) return textmsg\_stem

def viewSentiment(): text.delete('1.0', END) with open("Peoples\_Opinion/opinion.txt", "r") as file:

for line in file: line = line.strip('\n')

line = line.strip() arr = line.split("#") text\_processed

= stem(arr[1]) X = [text\_processed] sentiment =

text\_sentiment\_model.predic

t(X) predicts = 'None' if sentiment[0] == 0: predicts = "Negative" if sentiment[0] == 1:

predicts = "Positive"

text.insert(END,"Username : "+arr[0]+"\n");

text.insert(END,"Opinion : "+arr[1]+" : Sentiment Detected As : "+predicts+"\n\n") def uploadPhoto():

filename = filedialog.askopenfilename(initialdir="expression\_images\_to\_upload") user = simpledialog.askstring("Please enter your name", "Username")

policy = simpledialog.askstring("Please enter Government Policy name related to Facial Expression", "Please enter Government Policy name related to Facial Expression") img = cv2.imread(filename)

cv2.imwrite("sentimentImages/"+user+"-"+policy+".jpg",img);

messagebox.showinfo("Your facial expression image accepted for reviews", "Your facial expression image accepted for reviews")

def photoSentiment():

filename = 'sentimentImages' for root, dirs, files in os.walk(filename): for fdata in files: frame = cv2.imread(root+"/"+fdata)

faces =

face\_detection.detectMultiScale(frame,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),fla gs=cv2.CASCADE\_SCALE\_IMAGE)

msg = '' if len(faces) > 0:

faces = sorted(faces, reverse=True,key=lambda x: (x[2] - x[0]) \* (x[3] - x[1]))[0]

(x, y, w, h) = faces cv2.rectangle(frame, (x,y), (x+w,y+h), (0,0,255), 2)

temp = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY) roi =

temp[y:y + h, x:x + w] roi = cv2.resize(roi, (48, 48)) roi = roi.astype("float") / 255.0 roi = img\_to\_array(roi) roi = np.expand\_dims(roi, axis=0)

preds = image\_sentiment\_model.predict(roi)[0]

emotion\_probability = np.max(preds) label =

EMOTIONS[preds.argmax()] msg =

"Sentiment detected as : "+label img\_height, img\_width

= frame.shape[:2]

cv2.putText(frame, msg, (50,40), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5,(0,0,255),

2)

cv2.imshow(fdata,frame)

messagebox.showinfo(fdata, "Sentiment predicted from Facial expression as : "+label)

if cv2.waitKey(10) & 0xFF == ord('q'):

break

cv2.waitKey(0)

cv2.destroyAllWindows()

font = ('times', 16, 'bold')

title = Label(main, text='Automating E-Government Services With Artificial Intelligence',anchor=W, justify=CENTER) title.config(bg='yellow4', fg='white') title.config(font=font) title.config(height=3, width=120) title.place(x=0,y=5)

font1 = ('times', 14, 'bold')

digitButton = Button(main, text="Generate Hand Written Digits Recognition Deep Learning

Model", command=digitModel) digitButton.place(x=50,y=100) digitButton.config(font=font1)

pathlabel = Label(main)

pathlabel.config(bg='green', fg='white') pathlabel.config(font=font1) pathlabel.place(x=50,y=150)

sentimentButton = Button(main, text="Generate Text & Image Based Sentiment Detection Deep Learning Model", command=sentimentModel) sentimentButton.place(x=50,y=200) sentimentButton.config(font=font1)

recognizeButton = Button(main, text="Upload Test Image & Recognize Digit", command=digitRecognize)

recognizeButton.place(x=50,y=250) recognizeButton.config(font=font1)

opinionButton = Button(main, text="Write Your Opinion About Government Policies", command=opinion)

opinionButton.place(x=50,y=300) opinionButton.config(font=font1)

viewButton = Button(main, text="View Peoples Sentiments From Opinions",

command=viewSentiment)

viewButton.place(x=50,y=350) viewButton.config(font=font1)

photoButton = Button(main, text="Upload Your Face Expression Photo About Government

Policies", command=uploadPhoto) photoButton.place(x=50,y=400) photoButton.config(font=font1)

photosentimentButton = Button(main, text="Detect Sentiments From Face Expression Photo", command=photoSentiment)

photosentimentButton.place(x=50,y=450) photosentimentButton.config(font=font1)

font1 = ('times', 12, 'bold')

text=Text(main,height=15,width=80) scroll=Scrollbar(text)

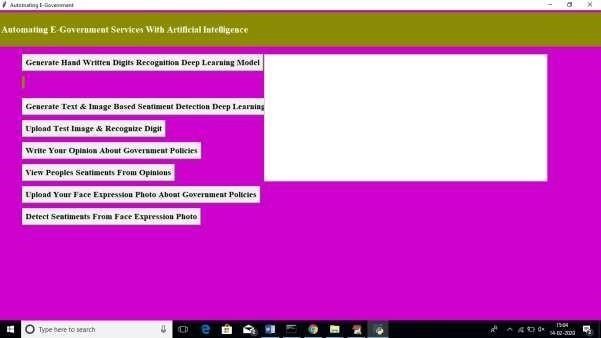
text.configure(yscrollcommand=scroll.set)

text.place(x=600,y=100)

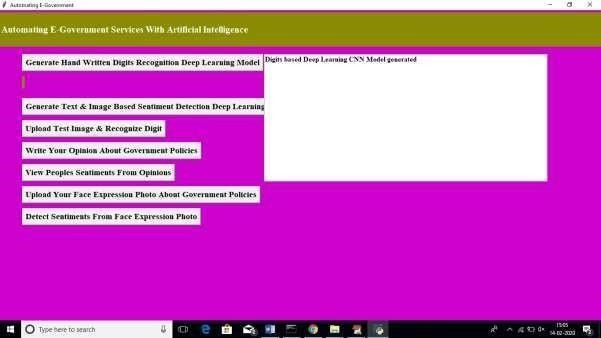
text.config(font=font1) main.config(bg='red') main.mainloo

## 8. WORKING IMAGES

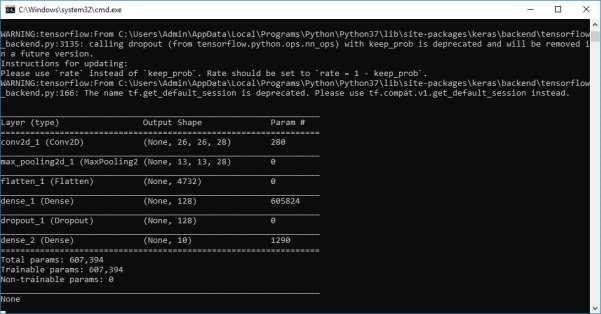
Home Page:



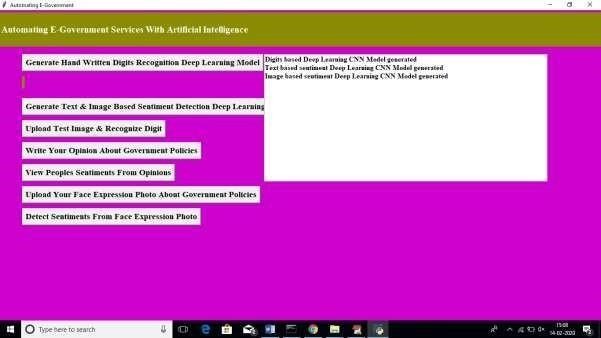
In above screen click on ‘Generate Hand Written Digits Recognition Deep Learning Model’ button to generate CNN digits recognition model



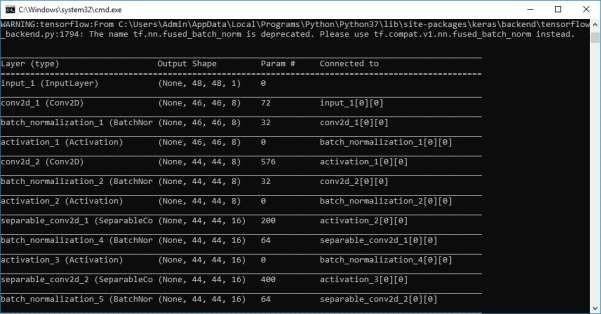
In above screen we can see digits model generated and CNN layer details you can see black console



In above screen we can see Conv2d means convolution or CNN generate image features layer from different size as first layer generate with image size 26, 26 and second generated with 13 and 13 and goes on. Now click on ‘Generate Text & Image Based Sentiment Detection Deep Learning Model’ button to generate CNN for text and image based sentiment detection model.

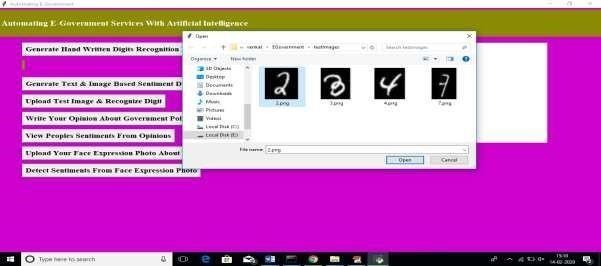


In above screen we can see text and image based CNN model generated. See black screen for more details



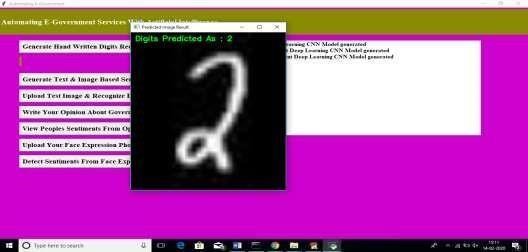
Now click on ‘Upload Test Image & Recognize Digit’ button to upload digit images and to get name of that digit. All digit images saved inside test Images folder

**CONCEPT 1 :**



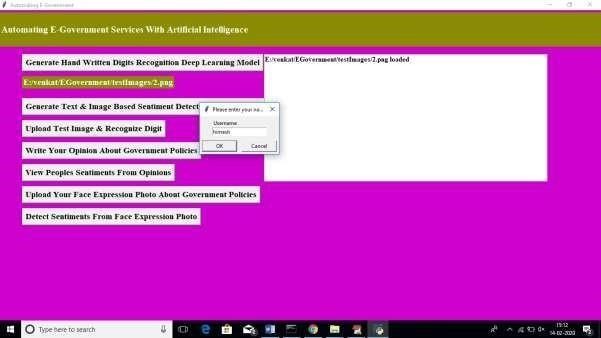
In above screen I am uploading image which contain digit 2 and below is the output of detection

**OUTPUT :**

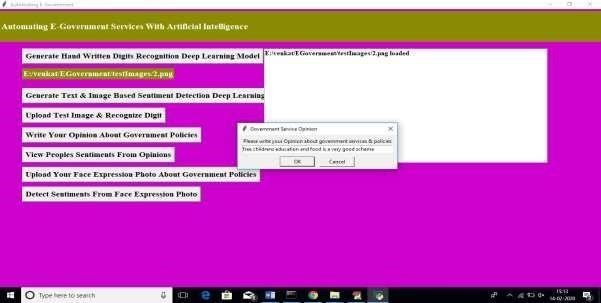


In above screen we can see Digits Predicted as: 2. Now click on ‘Write Your Opinion About Government Policies’ button to write some comments on government policy

**CONCEPT 2:**

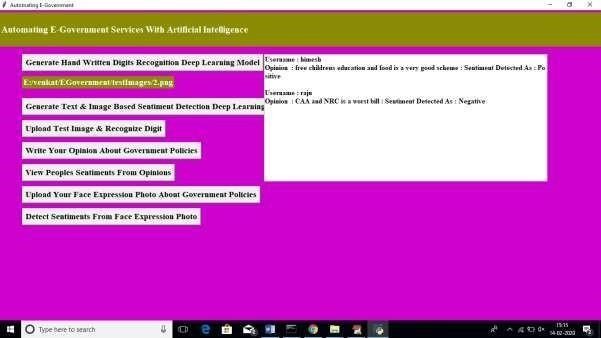


In above screen before writing opinions we need to write username after writing username click ok button to get below screen



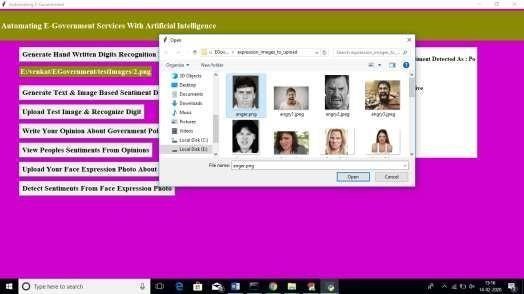
In above screen I wrote some comment on some scheme and application detect sentiment from it as positive or negative. Now click on ‘View Peoples Sentiments From Opinions’ button to view all opinions from past users.

**OUTPUT :**



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

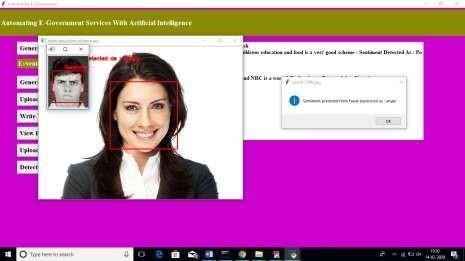
**CONCEPT 3 :**



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

**OUTPUT :**



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

## 9. FUTURE ENHANCEMENT

Future Enhancement is being planned to further analyse and enhance the protocol on reforming policies rather than reforming the process. This approach has been implemented and articulated around the world by different governments to facilitate the governing approach which will increase the public trust and establish a more reliable and transparent system that promotes democracy and provides the more efficient government.

## 10. CONCLUSION

With the recent advances in AI and deep learning technologies, more government agencies are starting to use such technologies to improve their systems and services. However, a large set of challenges hinder the adoption of such technologies, including the lack of experts, computational resources, trust, and AI interpretability. we introduced the definitions of artificial intelligence and e-government, briefly discussed the current state of e-government indices around the world, and then proposed our solutions to advance the current state of egovernment, considering the Gulf Countries as a case study. We proposed a framework for management of government information resources that help manage the e-government lifecycle end-to-end. Then, we proposed a set of deep learning techniques that can help facilitate and automate several e-government services. After that, we proposed a smart platform for AI development and implementation in e-government.

The overarching goal of this paper is to introduce new frameworks and platform to integrate recent advances in AI techniques in the e-government systems and services to improve the overall trust, transparency, and efficiency of e-government.

## 11. BIBILIOGRAPHY

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 neural network 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, . 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>

1. 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.

1. C. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.
2. Y. LeCun, Y. Bengio, and G. Hinton, ‘‘Deep learning,’’ Nature, vol. 521, no. 7553, pp. 436–444, 2015.
3. 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.
4. C. Dwork, ‘‘Differential privacy,’’ in Encyclopedia of Cryptography and Security, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA, USA: Springer, 2011.
5. L. Bottou, ‘‘Large-scale machine learning with stochastic gradient descent,’’ in Proc. COMPSTAT, 2010, pp. 177–186.
6. 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.