Introduction:

The goal of this project is to develop an intelligent software agent that can digitize manual, hand-written application forms. The agent will utilize Optical Character Recognition (OCR) technology to read the information on the form, and then update the digital version with the appropriate data. This will streamline the process of collecting and storing application information, reducing the possibility of errors, and making it easier to access and manage the data.

Background:

Currently, manual hand-written application forms are a common method for collecting information from applicants. However, these forms can be difficult to read, prone to errors, and time-consuming to process. Additionally, it can be challenging to access and manage the information once it has been collected. Digitizing the process can help to overcome these challenges by making it easier to read, store, and manage the data.

Methodology:

The intelligent software agent will be developed using OCR technology. The agent will be trained to recognize and extract the relevant information from the hand-written forms, such as the applicant's name, address, and other contact information. The extracted data will then be used to update the digital version of the form. The agent will be trained using a dataset of sample forms, and the performance will be evaluated using metrics such as accuracy and speed.

Implementation:

The software agent will be implemented in two phases. In the first phase, the agent will be trained to recognize the information on the forms. This will involve pre-processing the forms to remove any noise and improve the quality of the images, as well as training the OCR model using a dataset of sample forms. In the second phase, the agent will be integrated with the digital form, and the data will be updated automatically as the forms are processed.

Expected Results:

The expected results of this project include an increase in the speed and accuracy of the form processing, as well as improved accessibility and management of the data. The digitization of the process will also reduce the possibility of errors, as well as make it easier to access and analyse the data.

Conclusion:

The intelligent software agent developed in this project will streamline the process of collecting and storing application information, by digitizing manual hand-written forms and using OCR technology to read and update the form. This will provide significant benefits in terms of speed, accuracy, and data management, making the process more efficient and effective.

To develop the intelligent software agent for digitizing manual, hand-written application forms using Python, several tools may be required:

1. Optical Character Recognition (OCR) Library: Python has several libraries for OCR, such as Tesseract, pytesseract, and pyocr that can be used to recognize and extract the text from images of the forms.
2. Image Processing Libraries: Libraries such as OpenCV, PIL, and scikit-image can be used to pre-process the images of the forms, such as resizing, cropping, and removing noise.
3. Machine Learning Libraries: Libraries such as scikit-learn, TensorFlow, and Keras can be used to train the OCR model to recognize the text on the forms.
4. Natural Language Processing (NLP) Libraries: Libraries such as NLTK and spaCy can be used to extract and analyze the information from the text, such as the applicant's name and address.
5. Data Management Libraries: Libraries such as pandas and NumPy can be used to store and manage the data in a structured format, such as a CSV or database.
6. GUI Libraries: Libraries such as PyQt, Tkinter and wxPython can be used to create a user interface for the software agent, which would allow the user to interact with the agent, and upload the forms, check the results and so on.

Code 1

import cv2

import pytesseract

# Read the image

image = cv2.imread("form.jpg")

# Convert the image to grayscale

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Apply Otsu threshold

thresh = cv2.threshold(gray, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU)[1]

# Pass the image to tesseract

text = pytesseract.image\_to\_string(thresh)

print(text)

—