**1. INTRODUCTION**

**1.1 Over view of the project**

There are people facing problems in reading the local regional languages. So it becomes obvious that they have to suffer by this problem of understanding the language. So our software gives them a platform where they can give an image, containing of those words which they find difficult to read and understand, as an input which gives translated English version of those words as output.

**1.2 Motivation**

One of our teammates faced the regional language problem in China. There was a lot of difficulty even in finding a right person to translate Chinese in English language. When this was known to us, we thought that even for the visitors or people who are new in our state will face same problem in reading Kannada language. Hence, we got motivated and thus decided to solve this problem for the people who are new in the city. This can solve the problems of students from different states, and also to those who are unable to read and understand Kannada language.

**1.3 Objectives of the project**

The objectives of the project are as follows:

* User should not face difficulty in reading Kannada language.
* The translation should happen, when the user uploads an image of the script to be translated.
* The translation should happen irrespective of the type of image the user uploads (colored background, colored script, varied font styles and font sizes)

**1.4 Problem Statement**

To develop Software that can translate Kannada script to English from an image. The people who do not know Kannada (read/write) find it difficult to understand at times. This will not help those people sometimes and it is not handy. Hence this is the problem.

**2. PROPOSED SYSTEM**

**2.1 Description of the proposed system with simple block diagram**

Translated English script

Click On

Translate Button

Choose image

From System

Click on

‘Choose file’

Button

Fig 2.1 block diagram of the Proposed system

The Fig 2.1 explains that when user clicks on ‘choose file’ button, a dialog box of images appears where user chooses an image and then clicking on translate button the output is generated with translated English script.

**2.2 Description of Target users**

Our target users are those who are not familiar with the Kannada language. The visitors, the non-regional students- those who stay in hostels, they find it difficult to read the boards of any hotels, menu cards, shops, importantly the name of the areas, employees new to the city who are unaware about the language.

**2.3 Advantages/applications of proposed system**

1. Improve Communication in Multiple Languages: To remain competitive in today’s fast-paced global marketplace, companies must chart a course to growth fuelled by highly efficient international networks, local market penetration strategies and solutions designed to meet the increasingly sophisticated needs of their customers. This implies worldwide communication with partners, customers and employees in a variety of languages. The software generates publishable quality translations for organizations to bridge the multilingual communications gap both internally and externally.
2. Deliver Information in Multiple Languages: Language is an essential driver of enterprise growth. The ever-expanding volume of information to be translated demands increased efforts to conceive, draft, publish, and update texts in multiple languages. Websites, technical documentation, knowledge bases, strategic documents, and other information enhance their full value if they are translated consistently across the enterprise. The same information must be understood by and communicated to employees, customers and partners worldwide.
3. Translate More for Less: Machine translation eliminates the main constraints of human translation: cost and capacity.A translator usually translates 2,000 words per day for 20 cents per word.As the volume of information grows, it surpasses the capacity of human translators and enterprise budgets. Corporations want to translate more for less and they need to do it fast.
4. Increase Human Translation Productivity: Machine translation can be used by human translation service providers as a way to boost productivity. They also help localization groups translate more content at lower costs in a timely manner, maintain budget control and achieve time-to-market objectives.

**2.4 Scope**

Language translation is very critical if every basic text is not in English. Travelers, tourists find it difficult to translate and understand the regional language script. So, it becomes very handy if a person can upload a picture and translate the text from the picture. The existing apps concentrate on translation only when text is types manually or in picture, the text which has to be translated has to be highlighted. Hence there is scope to develop software that will translate the text from the image, once we chose and upload the image.

4

**3. SOFTWARE REQUIREMENTS SPECIFICATION**

**3.1 Overview of SRS**

In a project, a requirements specification is a document providing an adequate and unambiguous description of the task load, together with a description of the desired results, the essential conditions to which the service must conform and the characteristics or features of each deliverable.

The specific goal of a good translation requirements specification (TRS) is to establish the basis for agreement between the customers and the vendors, to determine if the translation specified meets their needs, and help the vendor select the most appropriate resources and prepare a realistic schedule.

**3.2 Requirement Specification**

Project requirements must be concise and straightforward to be read and followed.

3.2.1 Functional Requirements

Behavioral requirements describing all the cases where the system uses the functional requirements are captured in [use cases](https://en.wikipedia.org/wiki/Use_case).

1) User shall be able to choose image from System

1.1) User shall be able to Choose picture with Kannada script.

2) User shall be able to get the English Script after translation

2.1) User shall be able to get the translated text.

2.2) User shall be able to get the good understandable translation.

3.2.2 Use Case Diagram

Usecase diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors).

1. Users:

1.1 User shall execute the code

1.2 User shall click on ‘choose file’ button to choose the picture

1.3 User shall press the translate button.

2. Image translator:

2.1 System shall scan and convert the image to gray scale the image after it has been chosen.

2.2 System shall retrieve the text.

2.3 System shall translate the Kannada script to English script

2.4 System shall display the text on the text area as translated script.

The Fig 3.1 and 3.2 show the user case diagram of above mentioned actors.

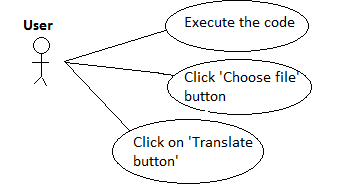


Fig 3.1 Use Case diagram of User

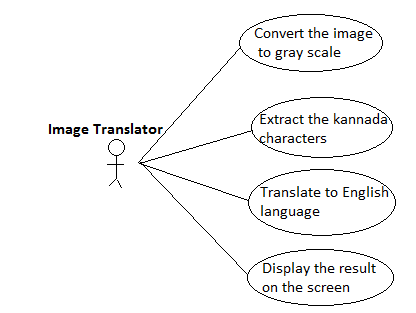


Fig 3.2 Use Case diagram of Image Translator

3.2.3 Use Case descriptions using scenarios

**Use Case 01:** Choose the image from system

* + Primary Actor – Person who is unaware about the kannada script
  + Purpose – To translate the Kannada script to English
  + Pre Conditions –
    1. User has images stored in system.
  + Post Conditions –
    1. User should be able to choose only one picture with Kannada script at a time.
  + Main success scenario –
    1. User opens the software
    2. User chooses a single image at a time
    3. User asks the system to translate the image.
  + Exceptional scenario -

a.1) If the source image is not chosen well, then it fails to translate.

**Use Case 02:** Image Translator

* + Primary Actor – The software to translate the script
  + Purpose – To help the user to understand the kannada script in English
  + Pre Conditions –
    1. Software accepts the image from system.
  + Post Conditions –
    1. Software translates to English.
    2. Software gives the translated script as output.
  + Main success scenario –
    1. Software checks if the script can be translated or not.
    2. Software will be able to translate few colored images.
    3. Software will give the meaningful translated sentence.
  + Exceptional scenario -

b.1) if the source image has hand written script, then it fails to translate

c.1) if the source script is in different language other than Kannada, then it fails to translate.

3.2.4 Nonfunctional Requirements

3.2.4.1 Ease of use

* System should be able to choose images.

3.2.4.2 Reliability:

* System should make sure that it should not crash while translating.

3.2.4.3 Efficiency

* The system should translate the script correctly with logicality.
* The system should translate the script in an organized order and not jumbled.

3.2.4.4 Robustness:

* The system should make sure that the user’s data is not compromised at the time of error or error recovery.
* The system should make sure to get trust to use the system.

**3.3 Software and Hardware requirement specifications**

The software requirements for the language translation system is listed below

1. Ubuntu 16.10
2. Qt Designer for python GUI
3. Lucid chart for diagrams

The hardware requirements for the system are as follows

1. Laptop
2. RAM - 4GB
3. Free space 150 MB

**3.4 GUI of proposed system**

The GUI for the system has only one screen and it is shown in Fig 3.3

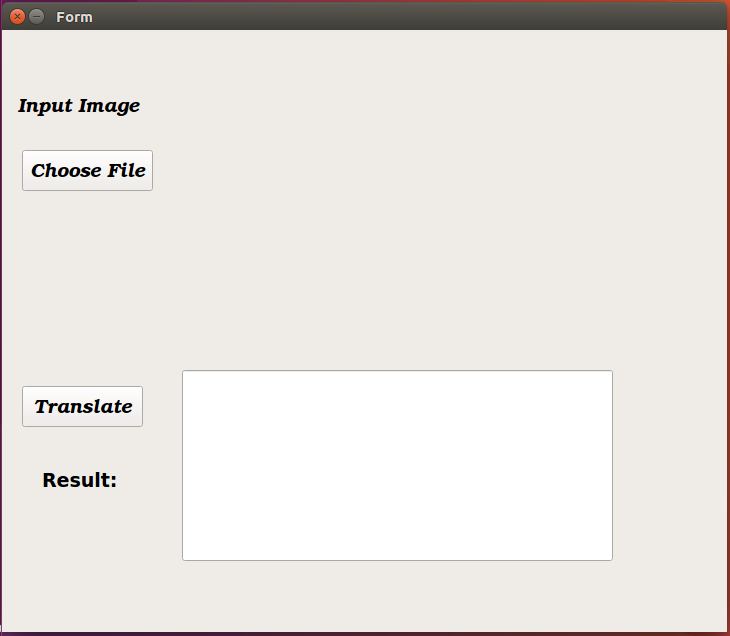


Fig 3.3 GUI of the proposed System

**3.5 Acceptance test plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl. NO | Test case | Input | Expected Output | Actual Output |
| 1 | Character recognition and translation | Colored background image | Characters are recognized and translated correctly. |  |
| 2 | Character recognition and translation | Hand written image | Not recognized properly and translated accordingly |  |
| 3 | Character recognition and translation | Black and white image | Characters are recognized and translated correctly. |  |
| 4 | Character recognition and translation | Designed background image | Characters are recognized and translated correctly. |  |
| 5 | Character recognition and translation | Single character in the image | It is not translated. |  |
| 6 | Character recognition and translation | Colored background with colored script | Characters are recognized and translated but not appropriately. |  |
| 7 | Validation | On click of ‘Translate’ button before on click of ‘Choose’ button |  |  |

Table 3.1 Test Plan for the proposed System

**4. SYSTEM DESIGN**

**4.1 Architecture of the system**

The system to translate the script from an image can be done by using two modules. One to extract the characters from an input image and two to translate those extracted characters.

The Fig 4.1 says the user imports the image, the image having the script is being extracted by the OCR module, and the extracted text is then fed to the translator module. The translator module translates the possible script to English and the same is given as output.

OCR

Extract Characters

Check database

Source image

Translator

Generate

DATABASE

RESULT

Translated output script

Fig 4.1 Architecture of the Proposed System

**4.2 Level 0 DFD**

DFD Level 0 is designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. The figure for DFD level 0 is shown in fig 4.2

1. **External entity:** an outside system that sends or receives data, communicating with the system being diagrammed. *User, database and translator engine* are the external entities, which are communicating with the software/tool.
2. **Process:** any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. Here the *software/tool* is the process, it gives the final output.
3. **Data flow:** the route that data takes between the external entities, processes and data stores.

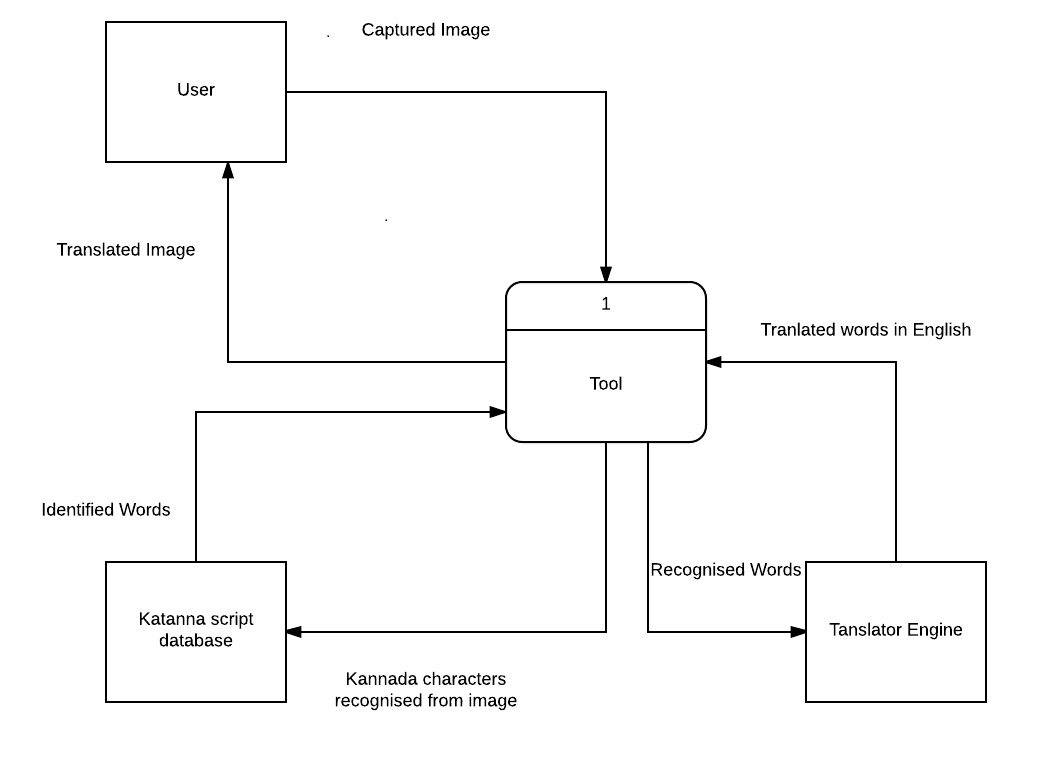


Fig 4.2 DFD level 0

**4.3 Detailed DFD for the proposed system**

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. Here we have an extra entity i.e. data store. Data store is a files or repositories that hold the database containing Kannada ACII values.

User imports the image into the OCR module, the OCR module extracts characters from data store, and then the extracted data is checked from the Kannada script database. The formed words which all are matched are displayed on screen, and these words are sent to the translation module

The figure for detailed DFD for the system is shown in fig 4.3

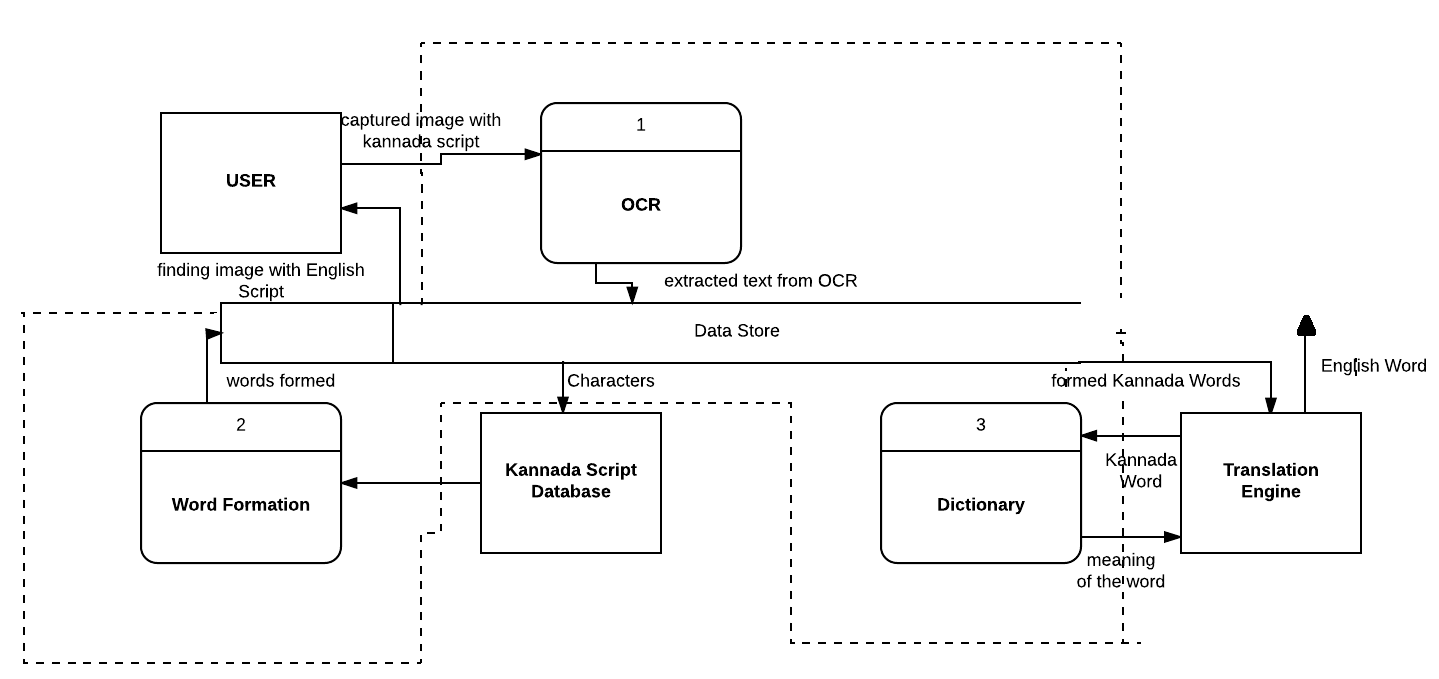


Fig 4.3 DFD Level 1

**4.4 Class Diagram**

The below fig 4.4 shows that in the class ‘SystemImage’, the user imports the image having the Kannada script from the system and then sends the image to the ‘OCR’ class, the characters from the image are extracted by this class and then it is loaded to ‘TranslatorEngine’ module, here the characters or the words are sent to the dictionary and are translated in English language.

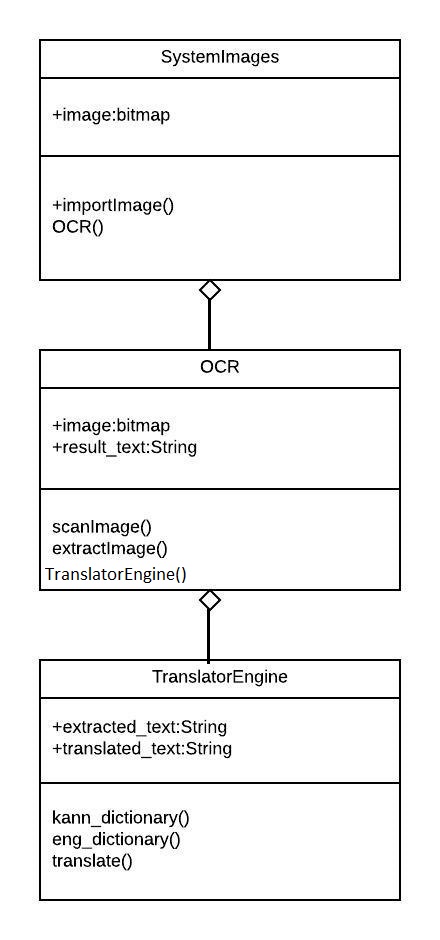


Fig 4.4 Class Diagram for the Proposed System

**4.5 Activity Diagram**

The fig 4.5 shows, from the start point, we choose an image having .the Kannada script and load it, if the image is loaded properly then the process continues further if not then choose the image again from start. The characters from the image are extracted in the OCR module. If the characters recognized then the characters are sent to the translator engine if not then the process stops. The translator engine uses the database having Kannada script, Kannada and English dictionary to translate the Kannada script to English and at the end the output is the meaningful English translated script.

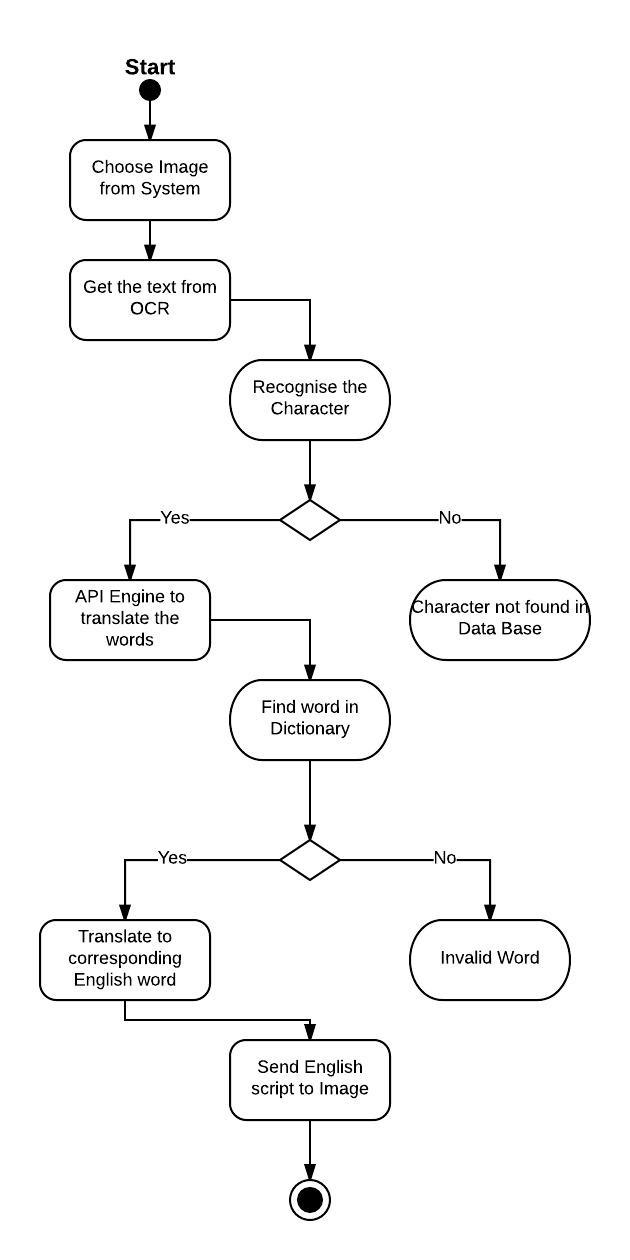


Fig 4.5 Activity Diagram for the Proposed System

**4.6 State Transition Diagram**

State transition diagram The fig 4.6 we have 2 states one is ocr and another translator engine, they describe the behavior of individual objects over the full set of used cases that affect those objects. And it is not showing the collaboration between objects that cause the transition. The behavior of OCR is to scan and generate a gray scaled image to extract the characters of the image using the dataset. And the translator fetches the extracted text from the OCR and the output of translated script is generated.

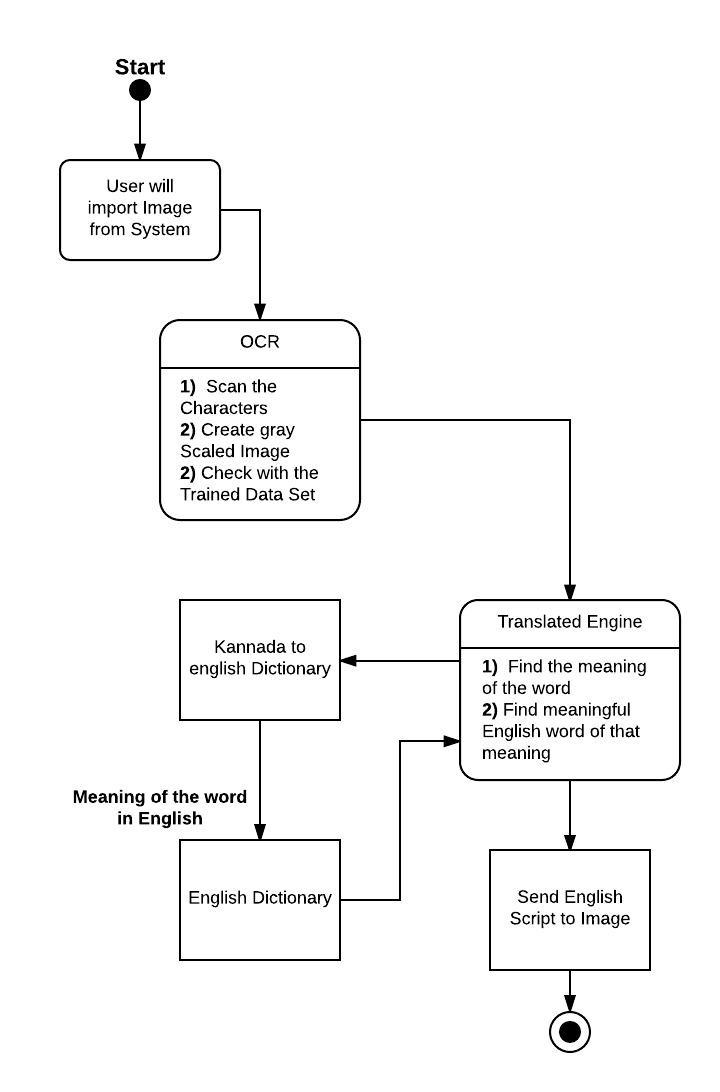


Fig 4.6 State diagram of proposed system

**4.7 Data Structure used**

* This project includes a well-defined data structure that manipulates and stores data in the Database.

**5. IMPEMENTATION**

**5.1 Proposed Methodology**

The proposed system has OCR module and Translation module.

1. The OCR (Optical Character Recognition) algorithm relies on a set of learned characters. It compares the characters in the scanned image file to the characters in this learned set. Generating the learned set is quite simple. Learned set requires an image file with the desired characters in the desired font be created, and a text file representing the characters in this image file.

Four basic algorithms:

• Image labelling: It is the main algorithm  
• Finding boundary and Generating X, Y coordinate pixel array.  
• Forming words.

Image text labelling algorithm:

It uses the Two-pass algorithm, which is relatively simple to implement and understand, the two-pass algorithm iterates through 2-dimensional, binary data. The algorithm makes two passes over the image: one pass to record equivalences and assign temporary text areas and the second to replace each temporary text area by the text area of its equivalence class.

1. Translator uses a model called Neural Machine Translation, where a kind of recurrent neural network based model is trained to generate a sequence of tokens from another sequence of tokens (hence, translation)

**5.2 Modules**

The proposed language translation system consists of two main modules;

* Ocr module
* Translator Engine module

1. OCR module description

The user chooses an image having Kannada script, and then this image is loaded as input to OCR module. In this module all the Kannada characters are extracted. This extraction of characters uses a database. This database consists of Kannada script ASCII values. Comparing with the database values, OCR module extracts the Kannada characters from the image.

Module name: OCR()

Input: Image having Kannada script

Output: Extracted Kannada script

Code or pseudocode:

def imageMoments()

imageFileName = input("enter the image name with absolute path ") image = Image.open(imageFileName)

gray\_image = cv2.cvtColor(enhanced\_image, cv2.COLOR\_BGR2GRAY)  
    filename = "{}.png".format(os.getpid())  
    cv2.imwrite(filename,enhanced\_image )

tessdata\_dir\_config='--tessdata-dir "/home/sahana/mini/tesseract-master/tessdata"'

text=pytesseract.image\_to\_string(Image.open(filename),lang='kan',config = tessdata\_dir\_config)

2) Translator Engine module description

The output from the OCR module acts as a input to the Translator module. Here the extracted Kannada characters are sent to the trained dictionary, where it generates a sequence of characters which are translated in English.

Module name: Translator\_Engine()

Input: Kannada characters

Output: English characters

Code or pseudocode:

translator = Translator()  
    translations=translator.translate(text,src='kn',dest='en')  
    print(translations)

**6. TEST PLAN AND TEST CASES**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. NO | Input Image | Expected Output | Actual Output |
| 1 | Colored background image | Characters are recognized and translated correctly. | Characters are recognized and translated correctly but only for few. |
| 2 | Hand written image | Not recognized properly and translated accordingly | Error – Hand written images are not accepted. |
| 3 | Black and white image | Characters are recognized and translated correctly. | Characters are recognized and translated correctly |
| 4 | Designed background image | Characters are recognized and translated correctly. | Characters are recognized and translated correctly. |
| 5 | Single character in the image | It is not translated. | It is not translated. |
| 6 | Colored background with colored script | Characters are recognized and translated but not appropriately. | Characters are recognized and translated but not appropriately. |
| 7 | Validation | On click of ‘Translate’ button before on click of ‘Choose’ button | Error- Image has to be chosen before translation. |

Table 6.1 for the proposed System

**7. RESULT AND DISCUSSIONS**

The figures 7.1, 7.2, 7. 3, we can see there are two dialog boxes. In the main dialog box-the user chooses an image by clicking the ‘Choose File’ button. The image appears on the text area provided. On clicking ‘Translate’ button, the process takes few seconds and finally an output of translated English script appears on the second text area. The other dialog box displays the converted gray scale image of the input image.

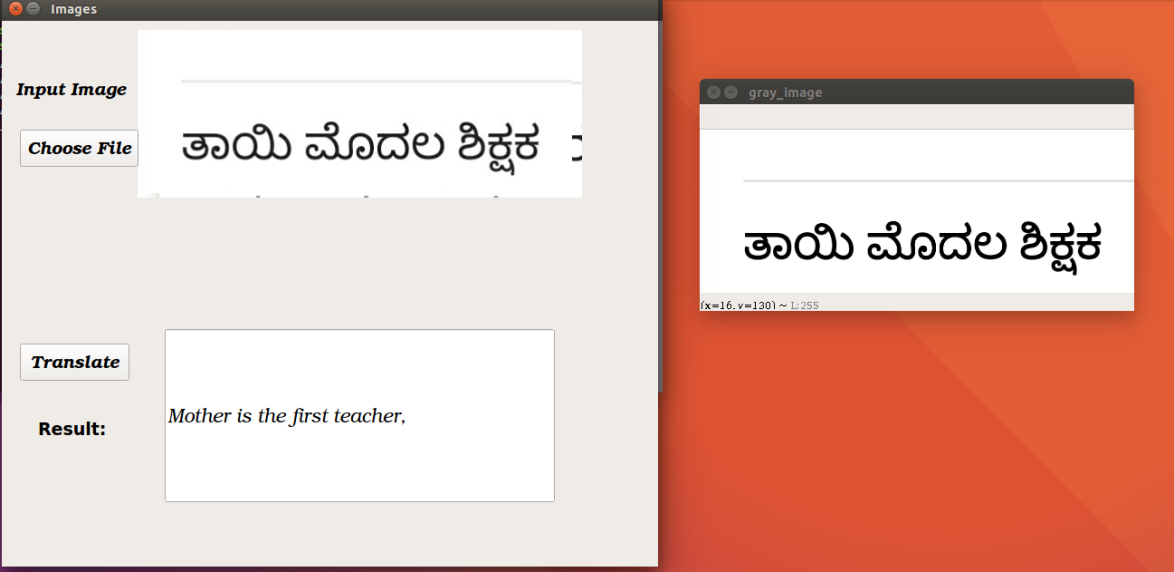


Fig 7.1 Black and white image - valid

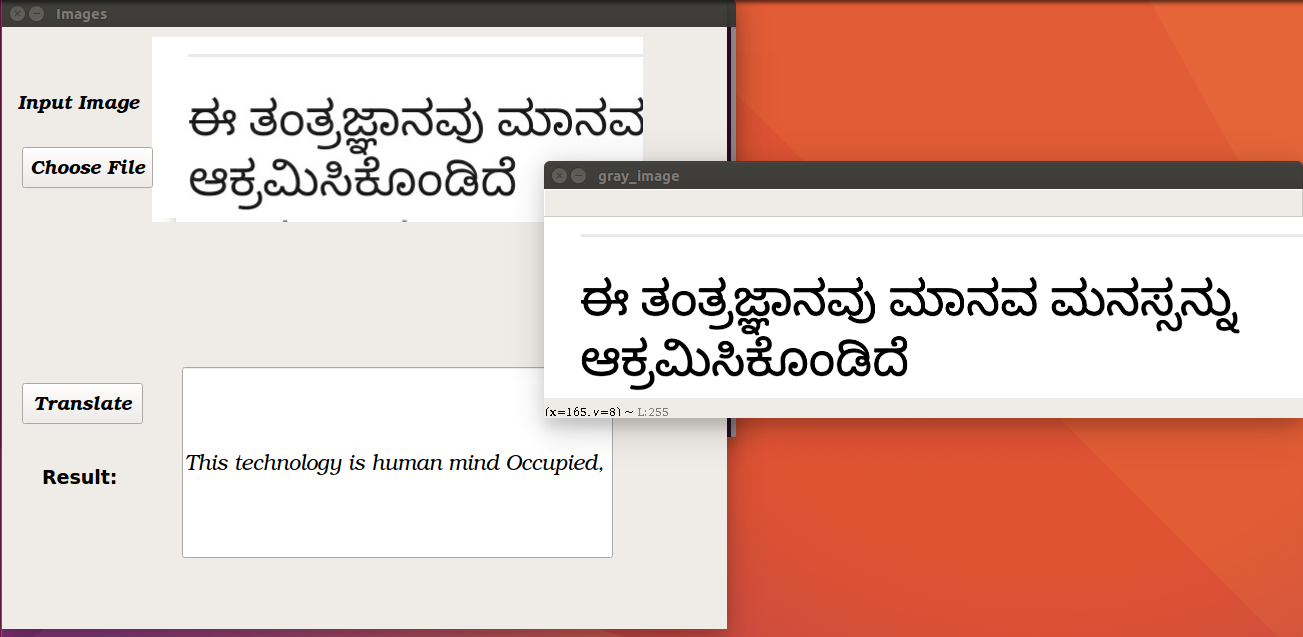


Fig 7.2 Black and white image - valid

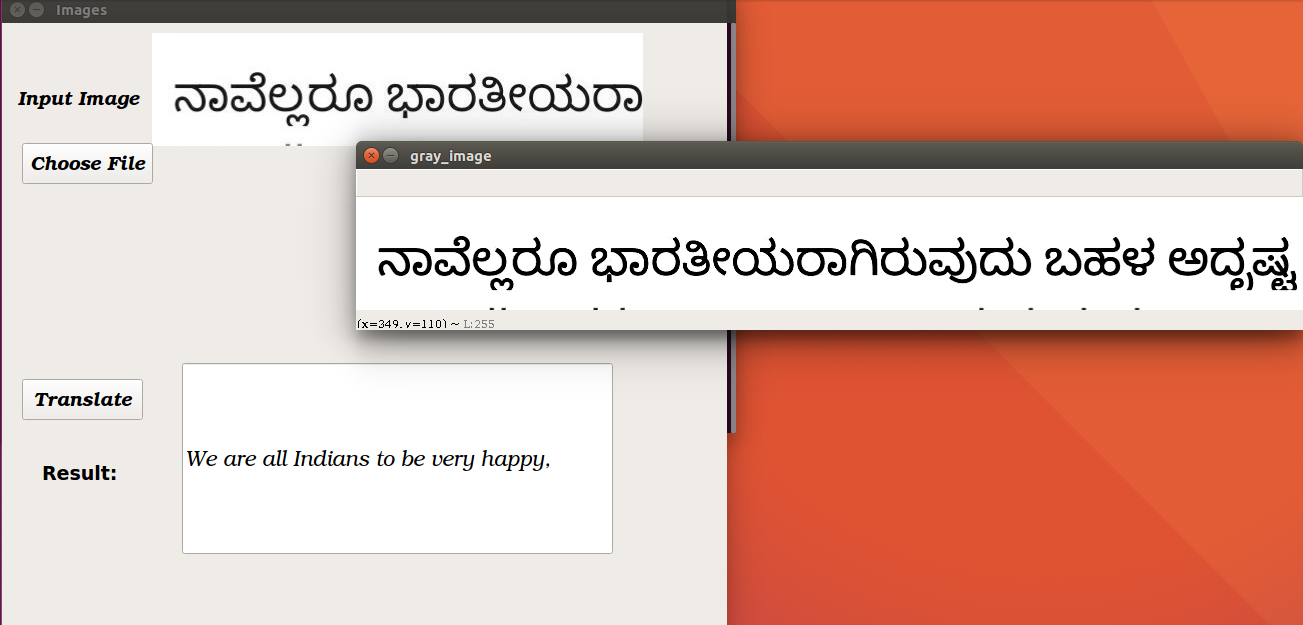


Fig 7.3 Black and white image – valid

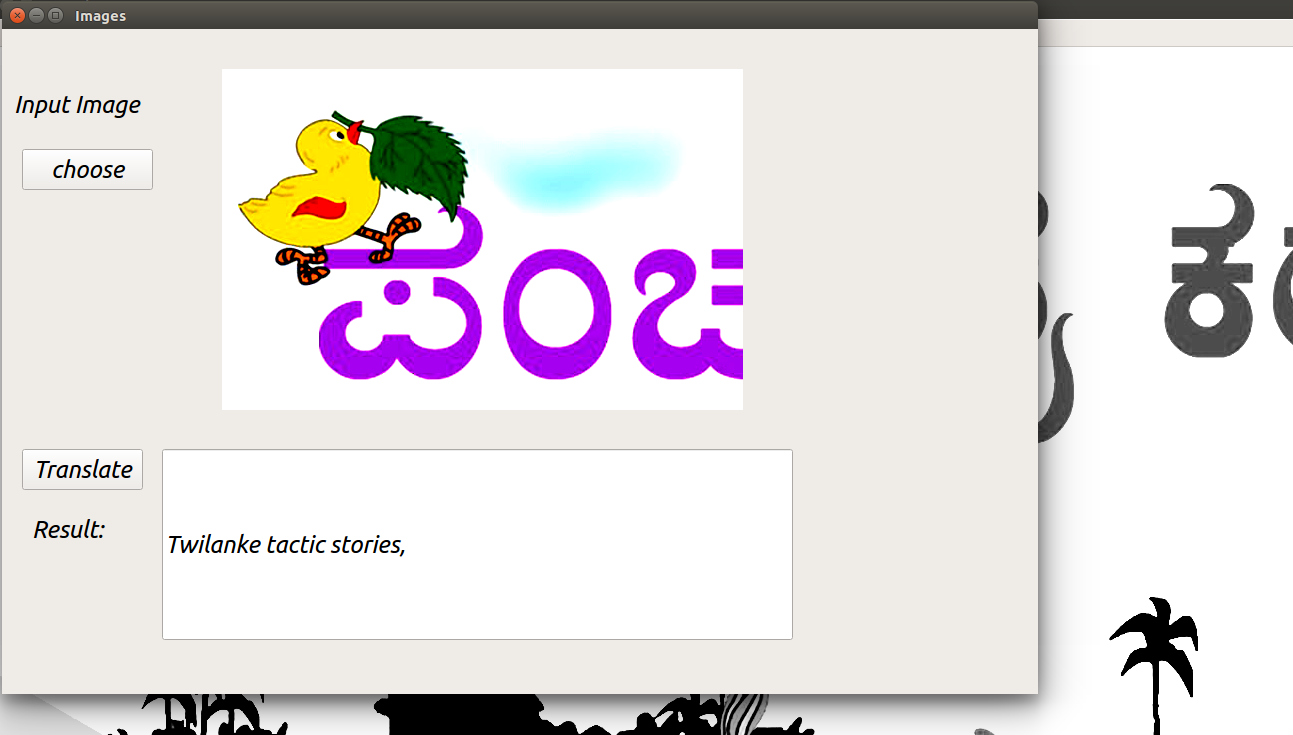


Fig 7.4 Colored image - valid

In fig 7.5, if we click on translate button without importing any image then the error is shown.

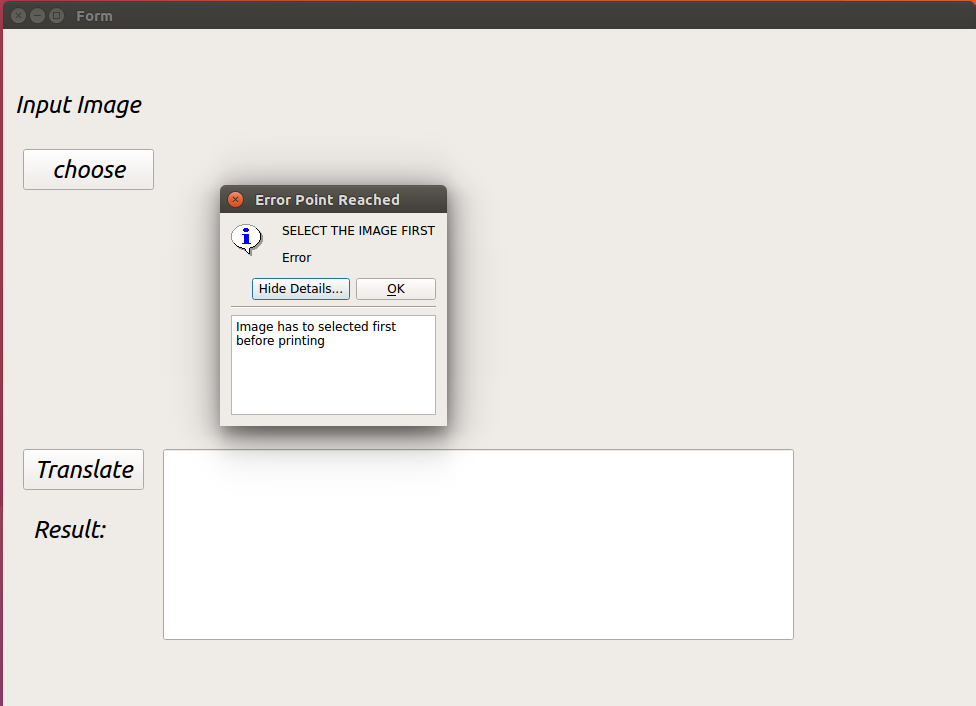


Fig 7.5 Error message

In the fig 7.6 it shows an error when the input image is hand written script.

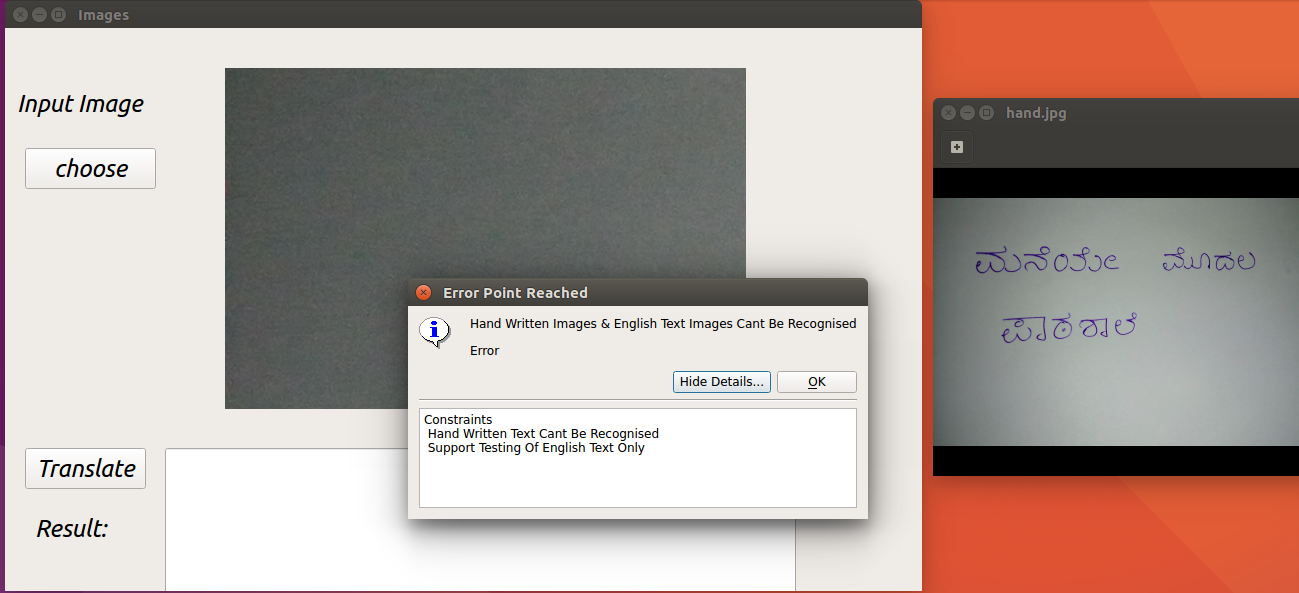


Fig 7.6 Hand written image – invalid

**8. CONCLUTION AND FUTURE SCOPE**

The developed software helps the user to choose the image and translate it at his/her fingertips. Basically any image gets converted to gray-scale first and then the characters are extracted and translated. The software also gives the pixel values and the RGB values of the image chosen.

In the future, we can enhance the proposed system to be implemented as a mobile application. And instead of choosing the image from the system, we can capture the image from the mobile camera itself.

**9. REFERENCE/ BIBLIOGRAPHY**

[1[]https://www.lucidchart.com/documents#docs?folder\_id=home&browser=icon&sort=saved-desc](#_top)

[2] [https://wiki.qt.io/Install\_Qt\_5\_on\_Ubuntu](#_top)

**10. APPENDIX**

A. Description on Technology used

Python:

In terms of software, the technology used in building this system is python. Python is becoming the preferred language to code since it is quite easy to learn as well as it is strong in the usage of dictionaries. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

B. Explanation on Tools:

Qt Designer:

Qt Designer is Qt's tool for designing and building graphical user interfaces (GUIs) from Qt components. It is possible to compose and customize widgets or dialogs in a what-you-see-is-what-you-get (WYSIWYG) manner, and test them using different styles and resolutions. Using its simple drag and drop interface, a GUI interface can be quickly built without having to write the code. It is however, not an IDE such as Visual Studio. Hence, Qt Designer does not have the facility to debug and build the application.

Sublime Text 3:

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.