Chapter 10
Bilingual OCR System

10.1 Introduction
In India, the state of Karnataka has Kannada as its official language, though many
national organizations such as banks use both English and Kannada. The entire
document in the government offices of the Karnataka state also usually appear in two
languages, Kannada and English. This is the major reason for choosing these two
languages for testing with the multi-script, multi-lingual document processing model
discussed in chapter 8. There was no complete example of automation of multi-script
document processing in that chapter. The aim of the automation of document
processing is to convert the scanned paper document to machine readable codes such
as ASCII. In this chapter the same is tested by considering the bilingual documents
printed in English and Kannada languages. Here the document image is finally
converted into ASCII files.

The major function of an OCR system which can read two language scripts,
English and Kannada, is to separate the words in the document script-wise and then to
feed them to the corresponding OCR systems to convert the same to machine codes.
For this purpose, the Kannada OCR system which has been developed in the previous
chapter is used. An English OCR system is implemented in this chapter and both
Kannada and English OCR systems are combined to form the bilingual OCR system,
along with the individual word script identification system used in chapters 5 and 8.

10.2 Related work
Researchers of international community have some contributions in the document
script identification field [16], [17], [100], [101] but have rarely attempted to process
multi-script, multi-lingual documents. This may be due to the fact that such multi-
script, multi-lingual documents appear only in Indian Society. At the national level
the main contribution so far has been from B. B. Chaudhuri and U. Pal. They have
discussed an OCR system to read two Indian language scripts, Bengali and Devnagari (Hindi) [42]. The description of their system is as follows.

The diagrammatic structure of their OCR system is shown in figure 10.1.

The above system performs text digitization, gray-tone to two-tone conversion, line and word detection, and character segmentation followed by actual character recognition. In this system they have adopted a manual method to switch between the Bengali and Devnagari, depending upon the document content. Depending upon the position of the mechanical switch, the OCR system detects the features required for classification and then the characters are converted to suitable machine codes.

10.3 Developed Bilingual OCR system

The following important facts are to be noted in B. B. Chaudhuri's system discussed above. It requires the user to operate a manual switch to perform the OCR function, which indicates that there is no script identification technique embedded in this system. It is just a simple combination of the two character recognizers and hence may not be useful for processing bilingual documents directly. For processing bilingual documents, that is documents printed in two languages, they first recommend that the words be separated, script-wise before feeding them to their OCR
system, using their word script separation technique [80] (This has been discussed in
chapter 5).

In the developed bilingual OCR system presented here the above drawback is
eliminated. In this system the bilingual document can be directly processed and there
is no need to separate the words script-wise beforehand, because the automatic
neural-based script identification system for individual word is embedded. The
developed bilingual OCR system is shown in figure 10.2.

![Developed bilingual OCR system](image)

Figure 10.2: Developed bilingual OCR system.
The diagram of the developed bilingual OCR system presents the complete implementation details. The system has adopted neural based multi-script, multi-lingual document processing model explained in the chapter 8 of this thesis, for two languages, Kannada and English. It includes automatic word script identification system for two language scripts. This subsystem has already been discussed in chapter 5. It includes the Kannada OCR system which has been developed in chapter 9. The English OCR system for single font, has been developed specifically for this purpose. This system has been explained in the next subsection. The bilingual OCR system algorithm (10.1) is as follows.

Algorithm 10.1: Bilingual OCR system algorithm
1. Accept the bilingual document image printed in English and Kannada languages.
2. Call segmentation algorithms 5.1, 5.2 and 5.3, which return individual word document images of the input document in a words cell array and also the total number of words.
3. Initialize two new arrays lang1_words and lang2_words.
4. Make $i = 1$.
5. Select $i^{th}$ word document image from the words cell array.
6. Extract the features from the above word document image (dynamic feature extraction algorithm 5.5).
7. Pass the above extracted features to individual word script identification system (developed in chapter 5). Identify the script of the word.
8. If the script of the word is Kannada (classifier output '1') store the word in lang1_words array, else (classifier output '2') store the word in lang2_words array.
9. Make $i = i + 1$.
10. If $i$ is less than or equal to the total number of words (found in step 2) in the document go to step 5 else go to next step 11.
11. Pass all the words of `lang1_words` array to the Kannada OCR system (Kannada OCR system of chapter 9). The system segments each word into characters and recognizes them by decomposing into subparts. It reconstructs decomposed parts by allotting them suitable ASCII codes of the specific font, Shreelipi 851. These mapped ASCII codes are written into new text file 'Kan1.txt'.

12. Pass all the words of `lang2_words` array to the English OCR system (from step 2 of algorithm 10.2). The system segments each word into isolated characters and extracts the features. These features are fed to the PNN classifier which has been trained on that specific font (Courier New font) with small letters a to z and capital letters A to Z. This classifier recognizes English characters and writes equivalent ASCII codes into the new text file 'Eng1.txt'.

10.3.1 English OCR system

An English OCR system has been developed specifically to include in the bilingual OCR system. Since all the bilingual documents considered for experimentation have single font characters, single font character recognizer serves the purpose. To develop a single font English OCR system, first the document (shown in figure 10.3) is considered. In this document, all the possible English capital letters, the English small letters (of Courier New font) and the full stop, are present. If the developed English OCR system recognizes these characters in the input bilingual document, that may be sufficient to test the bilingual OCR system presently.

```
abcde fghi jklmn opqrst uv wxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

Figure 10.3: Sample English alphabet used in English OCR system.

From the document shown in figure 10.3, all the characters are segmented and normalized to a fixed size. Using these normalized characters, the probabilistic neural network based recognizer is designed. For this purpose a bar mask encoder type of
A feature extractor is used. It extracts 50 features and provides them to the neural network based system. The trained network is now ready for recognizing unknown characters. All the recognized characters are given tags of ASCII codes and are written into an ASCII file. To reconstruct the sentences, the information of the input document is utilized.

![Diagram of English character recognition system](image)

**Figure 10.4:** English character recognition system.
The algorithms used in the construction of above OCR system are as follows.

Algorithm 10.2: English OCR system.
1. Accept the document image (written in specific font Courier New, considered for English OCR system).
2. Segment the above document image into lines, words and characters, by segmentation procedures 5.1 and 5.2. They return words cell array which contains all the segmented words.
3. Make \( i = 1 \).
4. Segment \( i^{th} \) word into characters (algorithm 5.3). Let total number of characters be total_char.
5. Make \( j = 1 \).
6. Present \( j^{th} \) character to the feature extractor. It extracts 50 features.
7. Pass the above features to the trained probabilistic neural classifier.
8. PNN classifier recognizes the character and also associated ASCII code for that character.
9. Write the above ASCII code in the previously opened file 'Eng1.txt'.
10. Make \( j = j + 1 \).
11. If \( j \) is less than or equal to total number of characters (total_char) in the word go to step 6 else go to next step.
12. Write a blank space in the opened file 'Eng1.txt', indicating end of word.
13. Make \( i = i + 1 \).
14. If \( i \) is less than or equal to total number of words go to step 4, else go to step 15.
15. Close the 'Eng1.txt' file.

The above algorithm should be run the first time with the document image shown in figure 10.3. After the PNN classifier has been trained on these alphabets, the trained classifier can be used on unknown English document images.
10.4 Experiments, results and discussions

Two experiments are conducted. The first one is to test the English OCR system. The second one is to test the bilingual OCR system. To test the English OCR system, documents are written in MS Paint using Courier New font of size 11. The English document images (in '.bmp' format) thus created are fed to the English OCR system, which produces the ASCII text file as an output. A sample input document image and the obtained result output ASCII file are shown in figure 10.5.

Figure 10.5: (i) Sample English input document image (ii) Corresponding output file of the English OCR system.
A second experiment is conducted to test bilingual OCR system. To test the system, bilingual document images are created as follows. Separate English and Kannada document images are created using the corresponding fonts in the MS Paint package and saved in '.bmp', image format. Then individual lines from English document image and Kannada document image are manually cut and pasted in the new image file. Thus bilingual documents are obtained. Such bilingual documents are fed to the bilingual OCR system. A sample input bilingual document image and the two ASCII files obtained are shown in figures 10.6a and 10.6b.
The above figures show that the results obtained are very good. However some of the limitations and problems are seen. The first limitation is that the complete system works for only one font of English and one font of Kannada. The second is about the errors in the system. Errors of segmentation and word script identification are very critical. If a single word is labeled wrongly, all the characters in that word will produce erroneous outputs. To avoid such errors, one more subsystem, discussed in chapter 9, character script class confirmation system could be added to the present system. Similarly, if line segmentation produces errors all the words and characters in that line produce errors in the output files. All these things should be handled carefully.

10.5 Conclusions

This chapter presented a English OCR system specifically built to embed into the bilingual OCR system for English and Kannada language scripts. The bilingual OCR system has been built based on the multi-script, multi-lingual document processing model which was proposed in chapter 8. The system includes segmentation modules, feature extraction modules, script identification modules, English OCR system and Kannada OCR system. Such experimentation with English and Kannada scripts is first of its kind to the best of our knowledge. The results obtained are very good and prove that the approach followed is very effective.