Algorithm 1 Building the Coordinate Matrix

Input: Template set T= {T1,T2, …,TN} where ti is a sample for the ith template (one sample per template) , N is the number of the templates

Key set K={K1,K2,…,KN} where Ki ={ki1, ki2, kni }is a set that includes keys in the ith template, kij is the jth key from the ith template, ni is the number of the keys in the ith template.

Output: CM: Coordinate Matrix that has the coordinates of all labels, (xij, yij) for every 1≤i≤N , 1≤j≤ni

CM=ᵩ

For i € {1,.., N}

For j € {1,.., N}

//OCR function returns the coordinates of the given key from the document

(xij, yij) = OCR (Ki, kij)

CM.append(Ki, kij, (xij, yij))

End for

End for

Return CM

Algorithm 2 Classifying New Documents

Input: Coordinate Matrix CM,

Test Document Doc, the document to be classified

Maximum\_penalty,

Output: Doc\_Class, the detected class

For i € {1,.., N}

For j € {1,.., N}

//Dij represents the distance from kij to its counterpart in DOC

If OCR (Doc, kij) == NULL // the key is not found

Dij= Maximum\_penalty

Else

(txij, tyij) = OCR (Doc, kij)

Dij= | xij - txij |+| yij - tyij |

If Dij > Maximum\_penalty

Dij = Maximum\_penalty

End for

End for

//identifying the class

Doc\_Class=NULL

Doc\_Distance= Maximum\_penalty

For i € {1,.., N}

If ()< Doc\_Distance

Doc\_Class = i

Doc\_Distance =

End for

Return Doc\_Class, Doc\_Distance