Create FCGR images for CCAT-10 text

In[1507]:=
```
letterRules1 =
  Dispatch[{"p" | "d" → "b", "k" | "q" | "x" | "z" → "c", "h" | "j" → "g", "f" | "v" → "w",
    "y" → "i", "l" → "r", "\t" | "\n" | ";" | "," | "?" | "!" | ":" | "." | "(" |
      ")" | "-" | "+" | "*" | "&" | "_" | "=" | "\"" | "/" | "$" | "'" | "[" |
      "]" | "(" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" → "0"}];
letterRules2 = Dispatch[{"g" → {0, 0}, "i" → {0, 1}, "t" → {0, 2}, "m" → {0, 3},
    "r" → {1, 0}, "a" → {1, 1}, " " → {1, 2}, "s" → {1, 3}, "e" → {2, 0}, "n" → {2, 1},
    "0" → {2, 2}, "u" → {2, 3}, "b" → {3, 0}, "w" → {3, 1}, "c" → {3, 2}, "o" → {3, 3}}];
```

In[1509]:=
```
makePositionsC = Compile[{{shifts, _Integer, 2}, {k, _Integer}},
   Module[{posns},
    posns = FoldList[IntegerPart[(#1 + #2) / 2.] &, {2 ^ k / 2, 2 ^ k / 2}, (2. ^ k) * shifts];
    Rest[1 + Round @ posns]
   ], RuntimeOptions → "Speed", CompilationTarget → "C"];

FCGR[chars_, k_] := Module[
  {posns, newposns},
  newposns = makePositionsC[chars, k];
  Normal[SparseArray[Apply[Rule, Tally[newposns], {1}], {2 ^ k, 2 ^ k}]]
 ]
```

In[1511]:=
```
c10Train =
  FileNames[FileNameJoin[{$HomeDirectory, "Documents", "c10", "C10train", "*"}]];
c10Test = FileNames[FileNameJoin[
    {$HomeDirectory, "Documents", "c10", "C10test", "*"}]];
allC10TrainFiles = FileNames[FileNameJoin[
    {$HomeDirectory, "Documents", "c10", "C10train", "*", "*"}]];
allC10TestFiles = FileNames[FileNameJoin[
    {$HomeDirectory, "Documents", "c10", "C10test", "*", "*"}]];
```

In[1515]:=
```
allTrainWritings = Map[Import[#, "Text"] &, allC10TrainFiles];
allTestWritings = Map[Import[#, "Text"] &, allC10TestFiles];
allTrainWritingsPartitioned = Partition[allTrainWritings, 50];
allTestWritingsPartitioned = Partition[allTestWritings, 50];

trainauthors = Map[FileNameTake, c10Train];
ltlen = Length[trainauthors];
testauthors = Map[FileNameTake, c10Test];
```

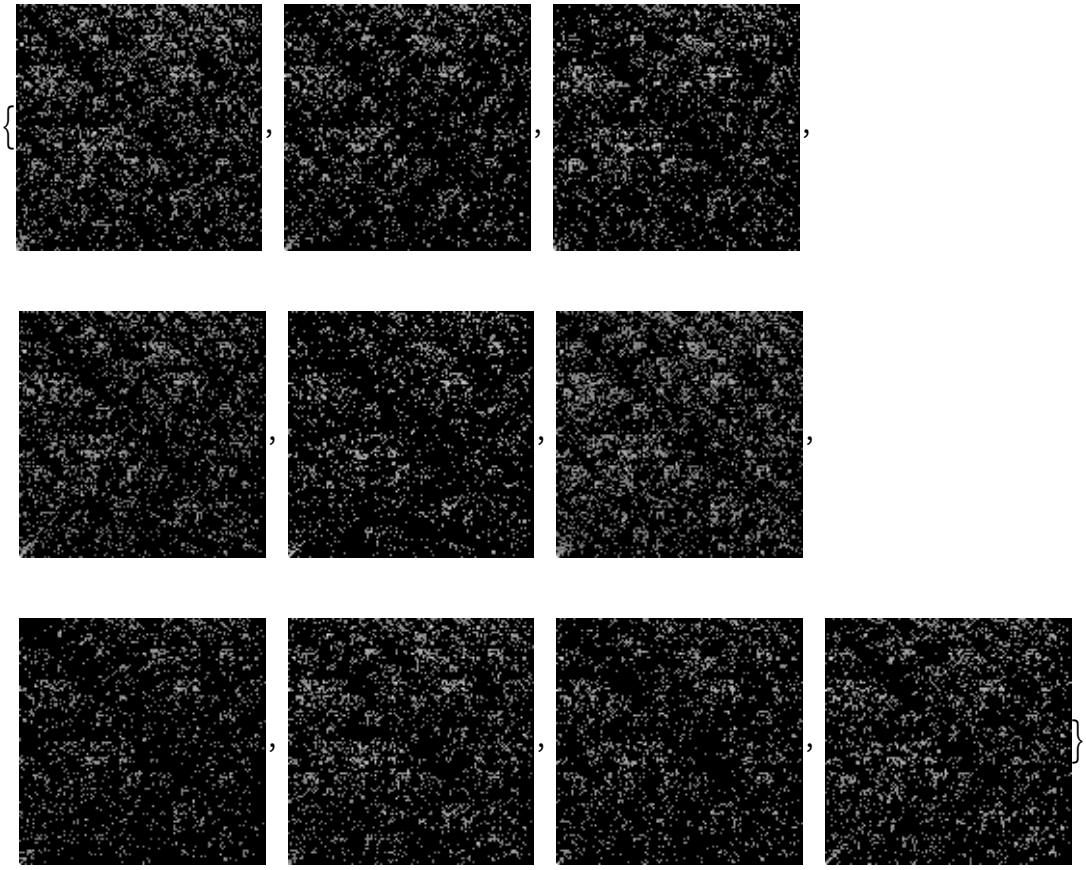
In[1522]:=
```
traintextLetters = Map[
    Characters[ToLowerCase[RemoveDiacritics[#]]] &, allTrainWritingsPartitioned , {2}];
traindigitseqs = Map[Developer`ToPackedArray[
      (IntegerDigits[Flatten[# /. letterRules1 /. letterRules2], 2, 2] /.
        IntegerDigits[__] :> Nothing)] &, traintextLetters , {2}];
testtextLetters = Map[Characters[ToLowerCase[RemoveDiacritics[#]]] &,
    allTestWritingsPartitioned , {2}];
testdigitseqs = Map[Developer`ToPackedArray[
      (IntegerDigits[Flatten[# /. letterRules1 /. letterRules2], 2, 2] /.
        IntegerDigits[__] :> Nothing)] &, testtextLetters , {2}];
```

In[1538]:=
```
pixLevel = 7;
trainimages1a = Table[FCGR[traindigitseqs[[j, k]], pixLevel],
    {j, ltlen}, {k, Length[traindigitseqs[[j]]]}];
testimages1a = Table[FCGR[testdigitseqs[[j, k]], pixLevel],
    {j, ltlen}, {k, Length[testdigitseqs[[j]]]}];
```

In[1541]:=
```
expon = 20 / 100;
trainimages1 = Map[(# / N[Max[#]])^expon &, trainimages1a , {2}];
testimages1 = Map[(# / N[Max[#]])^expon &, testimages1a , {2}];
trainSetLabels =
  Flatten[Table[ConstantArray[trainauthors[[j]], Length[trainimages1[[j]]]], {j, ltlen}]];
trainImages = Apply[Join, trainimages1];
testSetLabels =
  Flatten[Table[ConstantArray[testauthors[[j]], Length[testimages1[[j]]]], {j, ltlen}]];
testImages = Apply[Join, testimages1];
```

In[1548]:=
```
firstTrainImages = Map[Image, trainImages[[1 ;; -1 ;; 50]]]
```

Out[1548]=



In[1549]:=
```
lastTrainImages = Map[Image, trainImages[[50 ;; -1 ;; 50]]]
```

Out[1549]=