Report of Assignment 1

(Group members: Vidhixa Joshi and Divya Lalwani)

Code Organization:

A file named a1.cpp has main method.

We have included 4 header files as follows:

CommonFunctions.h: It consists of Common operations like creating a binary image and finding max distance from the scores generated in step 4 and 5.

Convolution.h: It consists of functions named convolve_general and convolve_separable which perform convolution as their names suggest.

HammingDistance.h : It consists of a function to calculate hamming distance between two binary images viz. image and template.

DFunction.h: It consists of function to perform Sobel edge detection on image which has been convolved with X and Y gradient respectively. Then we have a function which performs the scoring operation given in step 5.

Description of operations performed to execute a step:

(Note: In all steps, we have used image music1.png as a test image)

Step 2:

We are using a function named convolve_general to perform convolution. It uses a predefined box mean filter. The kernel can be flipped if we use *filter[krows-n][kcols-m]* while performing multiplication. The kernel on convolution smoothens the an image. Since there is no relevant information present in the boundary region for a piece of musical data, we let it be 0. Result of convolution can be seen in scores4.png.

Step 3:

We are using a function named convolve_separable to perform separable convolution. It uses horizontal and vertical 1-D matrix of the mean kernel used in above step.

Step 4:

Firstly, we are converting all grayscale images and templates to binary images. We assume a particular threshold of 70 to divide pixels into either black or white. We then calculate the hamming distance of input image and template (by passing templates one at a time). From hamming function, we are able to generate matching score of input image and template. In our case, since we have three templates, we are getting three scores. Now, from these scores, we are trying to extract the maximum score in max variable.

And based on these values, we are trying to generate symbols. We have set 88, 97, 97 as thresholds for the three templates.

We are generating symbols for all three templates and the notes detected in this step can be seen in output file detected4.png.

Step 5:

We have used a predefined kernel to perform Sobel operation on our image.

The X gradient is:

-1	0	1
-2	0	2
-1	0	1

We perform general convolution between image and this X gradient and let's call it Gx.

The Y gradient is:

1	2	1
0	0	0
1	-2	-1

We perform general convolution between image and this Y gradient and let's call it Gy.

We then perform an operation sqrt(Gx square + Gy square) on these results. And convert the results to a binary scale (0 and 255). The resultant image detects edges of input image as displayed in edges.png.

We apply this sobel operator to both image and template. We apply Distance function to the output of this image and template as per the formula for dynamic programming given in step 5. And then we repeat the operations performed in step 4, like finding the maximum and applying threshold to generate symbols.

As the distance function is quiet time consuming, we have applied this only to template1.png. All the operations can be applied to template2.png and template3.png similarly. For input image music1.png, entire process takes approximately 10 minutes, as we have used the naïve approach.

Step 6:

We have used the edges detected by Sobel in previous steps and used it as base to detect the staves.

In our approach, we start with second row and end on the second to last row (avoiding the boundary) and count the number of edges that are in that particular row. The staff lines are the ones which have most of the continuous pixels. Here we assume that the lines are horizontal and the assumption gives us reasonable results.

We then detect the presence of continuous 10 pixels to be an indication that the row is part of staff. We then check for end of each staff line and store the row number in an vector array. The distance between any two consecutive array element is equal to the node size in that particular image. This can be used to recognize the note type.

The image staves.png shows the detected staves.

Step 7:

We have scaled music4.png to 1000*218 pixels. This gives us good results for all operations that we perform in the system.

The notes A-G can be recognized by taking the difference of values at indices of array mentioned in step 6. The recognition of A-G notes has been accomplished by the group.

List of references:

1. Convolution has been understood from wikipedia.

2. Sobel matrix is assumed to be the standard Sobel matrix found in textbooks and online material.

3. Discussed the method to detect symbols with classmate Harsh Seth.