Computer Vision : Adaptive Pyramid

Assignment 3 Report

Mohamed Mostafa El Hamamsy | 28-9771
Ahmed Saleh Abdelwahab | 28-13165

Methods Used:

- This is a brief description of the functionality of each method:

(utils.py) :

- **has_more_candidates :** this method takes a graph as an input checks on all the cells in the graph to see if there is still a candidate cell (q = 1) if so it returns true otherwise it returns false.

- **min_mean :** this method takes a list of cells as an input and return the value of the mean of the cell with the minimum mean

- **get_result_image :** this method takes as an input the graph representing the image after applying the adaptive pyramid on it at level h and returns an image containing the new values of each pixel, to set the correct value the method finds the parent of the original pixel in the adaptive pyramid and set the value of that pixel to the value of its parent.

(cell.py):

- **__init__:** the constructor of the cell object, takes as an argument the index, row, col and the value of the corresponding pixel then it initializes the values of the cell.

- **__repr__:** toString Method

- **initVals:** initializes the mean/variance of the cell by calculating them using the support vector (a vector that contains all the neighbours to the cell in the current level), sets the correct values of p and q, and sets the size of the cell neighbours to the size of the support vector.

The Algorithm:

(main.py)

The algorithm follows the same steps listed in the paper:

Step 1: The algorithm starts by taking a path entered by the user then it reads the input image then it creates a 2D array of cells where each cell[i][j] matches a corresponding pixel[i][j] in the image

Step 2: Then for each each cell in that 2D array it calculates its support vector (a list of the neighbouring cells) along with the variance and mean and sets all the values to their initial values by calling initVals() method described before, and finally creates a graph which is a list of cells where each cell has its neighbours list.

Step 3: Then the algorithm starts a loop to find the local/sub minima in the graph and marks all the cells that are local/sub minima as surviving (p=1), The algorithm keeps repeating this step while there is still a candidates by using the method has_more_candidates described before to check.

Step 4: After finding all the surviving cells in the current graph the algorithm loops on all the cells in the graph again and for each non surviving cell (p=0) it finds the nearest surviving cell to that cell by calculating the mean difference between all the surviving cells and taking the one with the minimum difference.

Step 5: After finding the closest surviving cell to a non surviving one the root check described in the paper is made by calculating the min_contrast for the non surviving cell and comparing it with the mean difference if it was smaller than the cell is a root (p = 2) otherwise the cell is a non surviving cell and is linked to the closest surviving by setting it as its parent.

Step 6: In this step the algorithm connects the surviving cells with each other, this happens by looping on all the cells and if the cell is non surviving (p = 0) then all its surviving neighbors are connected/linked to its parent (which is a surviving cell) , and if that cell was a root then its connected/linked to all its surviving neighbours.

Step 7: After connecting the surviving cells all together including the roots the new graph is built by looping on the current graph and adding to the new graph only the surviving cells also the method initVals is called again on each surviving cell to recalculate the new mean/variance for that cell and initialize all the parameters again (p= 0, q = 1).

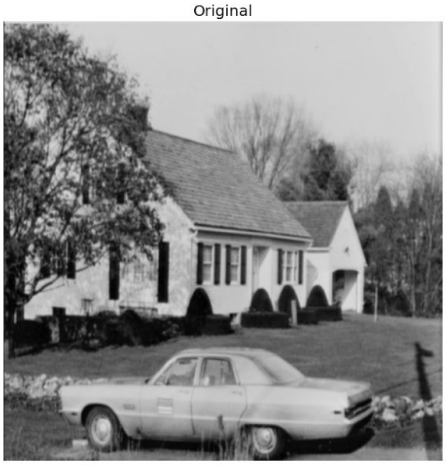
Step 8: Finally the resulting image is constructed by calling get_result_image method described before and the resulting image is added to the result_images array which contains the result image at each level including the original image which is the base of the pyramid (h = 0) and finally the level h is inceremented by 1

Step 9: The algorithm keeps repeating steps 3-8 until the level h reaches the MAX_LEVEL which is entered by the user, and once it terminates the results are displayed on a figure as shown below.

Original

Result at level 1

Original



Result at level 2

Another image: