Summarizing Visual Data Using Bidirectional Similarity

Team name - LMTG
Project-id - 12
Project link -
https://github.com/Computer-Vision-IIITH-2021/project-lmtg

Mohit pavan kumar - 2018102016
Tejaswini Anuhya Suma - 2018102018
Gowthami Gongati - 2018102048
Lakshmi Madhuri - 2018101116

Overview of presentation

- Brief summary of Algorithm
- Experiments done on various elements of algorithm(After Mid Progress)
  - Importance based cropping technique vs bidirectional similarity technique.(done)
  - Naive Bidirectional similarity vs Optimized Bidirectional similarity(d)
  - Experiment on the function used to check bidirectional similarity
    - Sum of squared distances vs correlation function(done)
    - Both of the above process performed with different weights
      - Uniformed vs non-Uniformed weights.
  - Gradual resizing by cv2.resize vs Gradual resizing using gaussian pyramid.
- Conclusions

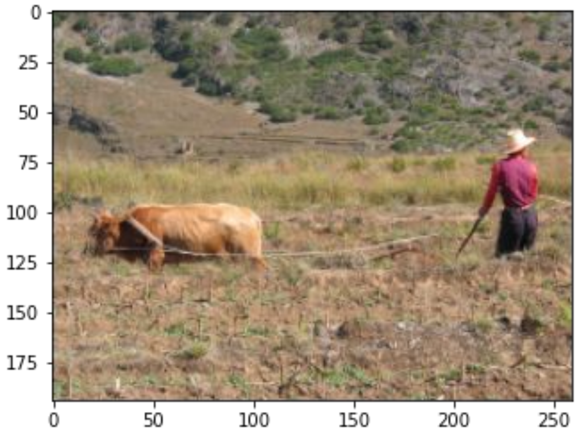
Summary of the algorithm

- The intention of the paper is to perform image summarization by attempting to reduce redundant data in the image. It uses bidirectional similarity to calculate the distance measure and try to minimize it to remove the redundant information.
- There are 4 main steps of this algorithm which happens in every block in the whole process
    - To take target image as a resized version of source image
    - Find the patches which are bidirectionally similar
    - Update the pixels in the target image with the use of formula given
    - Downscale this image by fixed value for every block until we get the final output.
- The target image according to the requirements of the algorithm should a visually similar image so that when we attempt to find the patches which are bidirectionally similar we get the patches corresponding to patches in source which are not completely different from source image.
- The pixels of the target image are updated based on the formula for each pixel which is obtained after differentiating the distance measure thereby minimizing it which is also the idea of image summary.

Experiment-1:- Importance based cropping

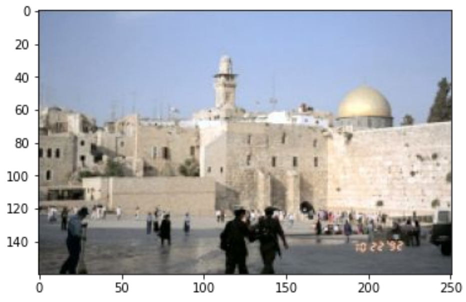
- Importance based cropping is one of the technique which provides better output if the important information is concentrated in one region-spatial or temporal.
- This makes the image free from redundant data and makes the image more aesthetic. It can be done in many ways-neural networks etc
- We chose to use an interactive gui which can be used to identify the important regions and assign weights accordingly.
- After Assigning weights, the regions with less weight/priority are cropped more than regions with more weight/priority

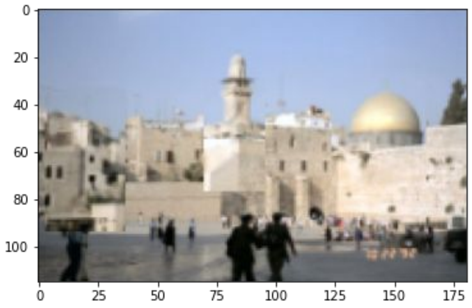
Results of Importance based cropping-(1/2)

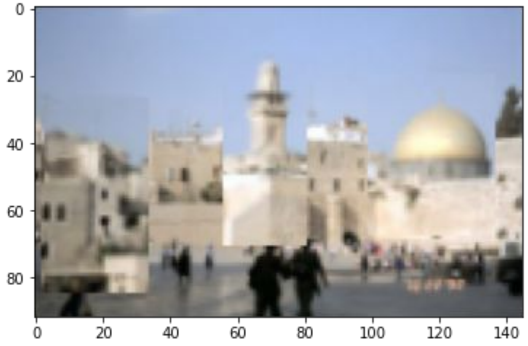


Importance was given to man and the animal through interactive gui. These images represent some of the stages in the whole process.

Results of Importance based cropping-(2/2)







Importance was given to tombs and towers through interactive gui.
These images represent some of the stages in the whole process.

Experiment-2- Naive Bidirectional similarity vs optimized Bidirectional similarity

- Bidirectional similarity has 2 aspects which needs to be taken care of-completeness and coherence.
- Both completeness and coherence requires comparison between every patch in both source and target.
- If we have to compare every patch in S with every other patch in T and vice versa, It would take 4 nested loops for both patches(considering they are 2d)
- In optimized bidirectional similarity, we store the patches of the signal/image which is to be compared with the other once and then use numpy array operations to multiply the matrices directly which gives the distance/correlation measure considerably faster.

Experiment 3:- Distance vs correlation

- In this paper, we attempt to achieve a bidirectionally similar image by trying to minimize the measure of dissimilarity or distance between source -target and vice versa.
- The distance measure used in the paper, is sum of squared distances. 2 kinds of distances are calculated-one each for completeness and coherence and then multiplied by weights after normalizing to get the distance measure for a source and a target.
- However it can also be seen that instead of minimizing the dissimilarity measure, it’s also feasible to maximize similarity which is in turn given by correlation.
- So to find respective bidirectionally similar patches, minimizing distance or maximizing correlation work.

Sum of Squared distances/Euclidean distance

- Sum of squared distance measure can be calculated by subtracting the mean and multiplying one vector/list with transpose of other.
- According to the description of results in the paper, d(S,T) decreases before abruptly increasing because we are losing information beyond that point.
- This point is also the convergence point where on abrupt increase, the output image of the iteration before the increase will be considered as the final output. This is why SSD works differently for different sized images.
- However, according to the paper, the dimensions of final summarized image are given as input and the iterations stop only when we get a summarized image of these dimensions which is also implemented by us.

Result(1) -

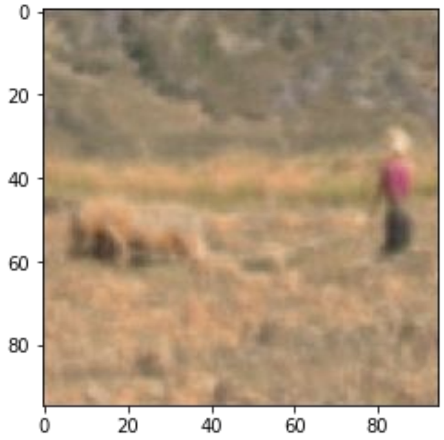


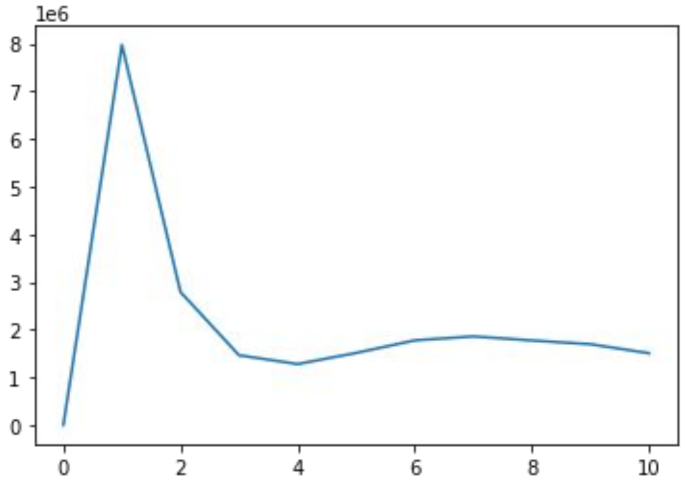
Fig a is the input given which is resized to 250X250 resolution while observing for big sized images and 100X100 resolution while observing for small sized images

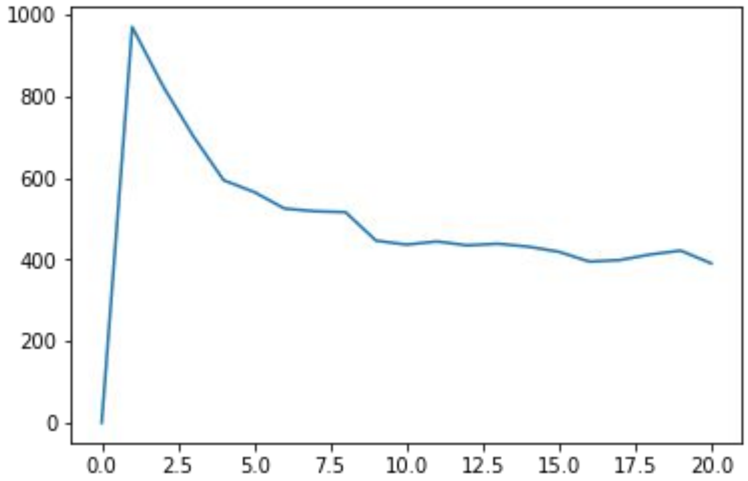
Fig b is the output given while small image is taken

Fig c is the output given while large image is taken

Result(2) -

Distance graphs for the above slide. Fig a for small sized images and Fig b for large sized images





Result(3) -

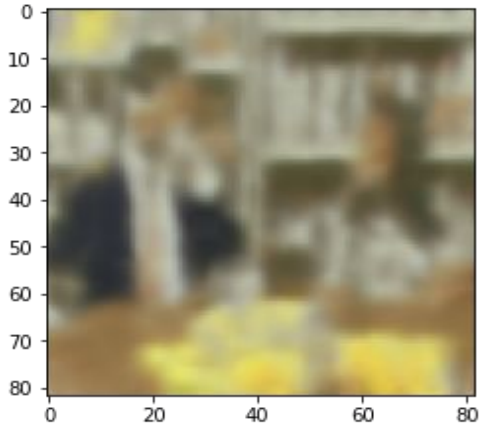


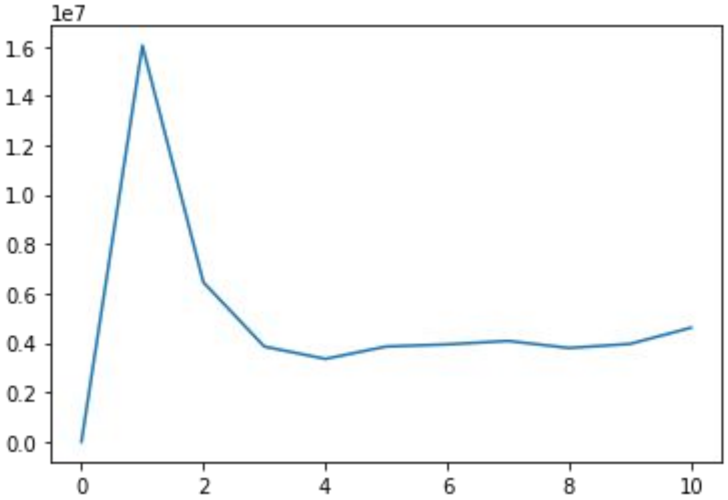
Fig a is the input given which is resized to 200X200 resolution while observing for big sized images and 100X100 resolution while observing for small sized images

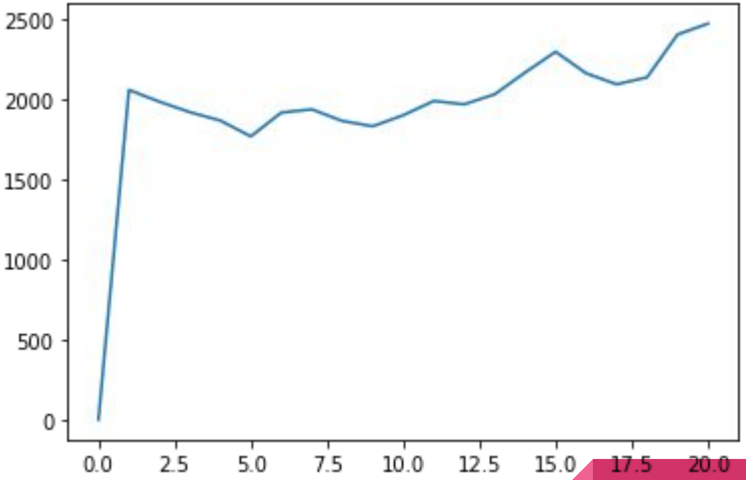
Fig b is the output given while small image is taken

Fig c is the output given while large image is taken

Result(4) -

Distance graphs for the above slide. Fig a for small sized images and Fig b for large sized images

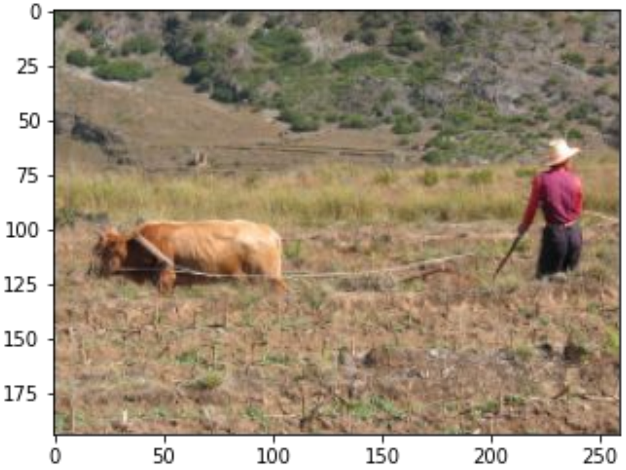




Correlation

- Correlation function is used to obtain a patch which is most similar or least dissimilar, i.e a measure used while matching a part of image/template
- Similar to distance function, correlation function is also used twice for ensuring completeness and coherence while storing the patches which correspond to most similar/least dissimilar to each other.
- These patches are there by used to transform the image and the iterations go on until the size of summarized image matches with dimensions given as input.
- Correlation has some advantages over distance measure which is why correlation is favoured in upcoming experiments.

Result(1) -



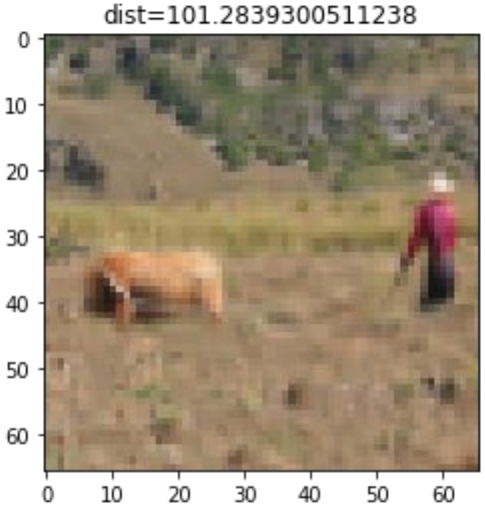


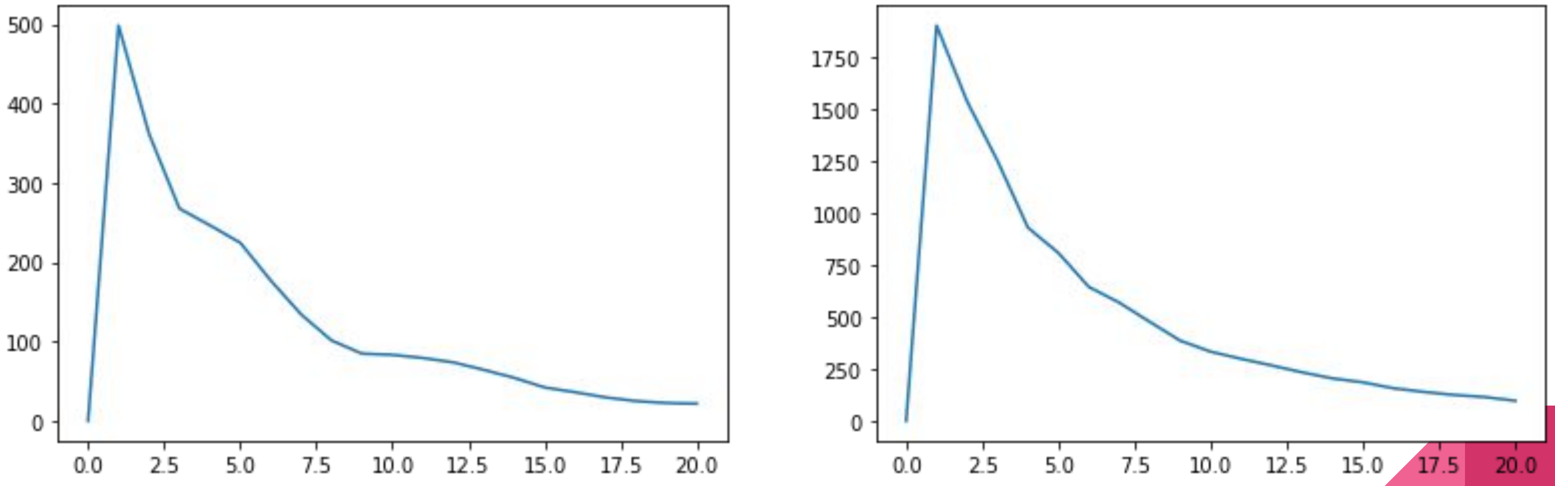
Fig a is the input given which is resized to 200X200 resolution while observing for big sized images and 100X100 resolution while observing for small sized images

Fig b is the output given while small image is taken

Fig c is the output given while large image is taken

Result(2) -

Distance graphs for the above slide. Fig a for small sized images and Fig b for large sized images



Result(3) -

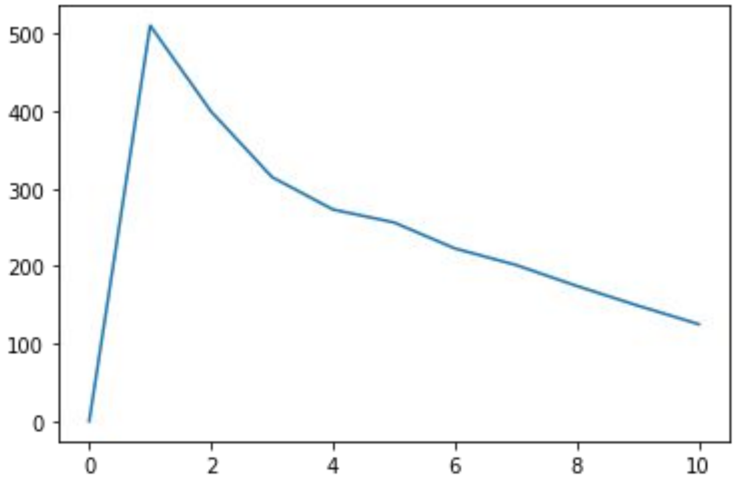
Fig a is the input given which is resized to 200X200 resolution while observing for big sized images and 100X100 resolution while observing for small sized images

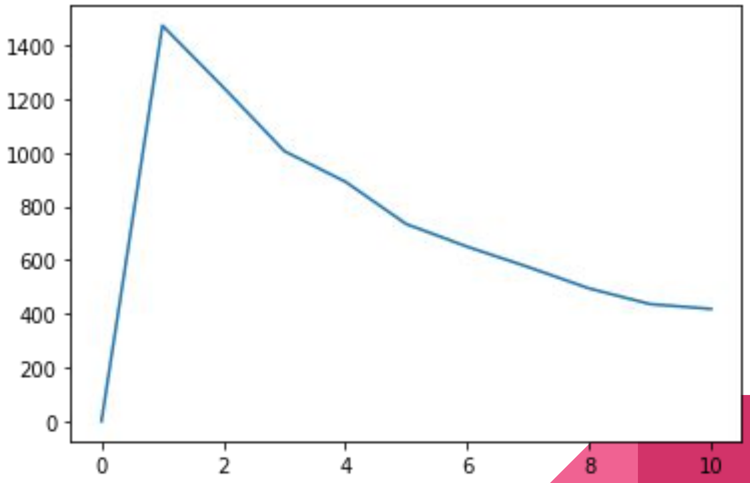
Fig b is the output given while small image is taken

Fig c is the output given while large image is taken

Result(4):-

Distance graphs for the above slide. Fig a for small sized images and Fig b for large sized images





Experiment 4:- Uniform vs Non Uniform weights

- While calculating distance or correlation measure, both the terms completeness and coherence are multiplied with appropriate weights because weight indicate importance of the patch/pixel.
- In the paper, initially weights are taken to be uniform that is 1/Ns and 1/Nt (where Ns and Nt are number of pixels in source and target image) and 0.5 when distances are normalized. These weights are same through out the image. All above experiments are done with uniform weights.
- For Non uniform weights, an interactive gui is taken similar to importance based cropping and important parts are given more weight in the process of patch match and image transform. Here, unlike cropping, the weights are multiplied while image transform which helps in removing redundant/less priority
data in a more refined way

Result(1) -

Fig a is the input given to correlation function to determine most similar patches.

Fig b is the output given when uniform weights are taken

Result(2) -

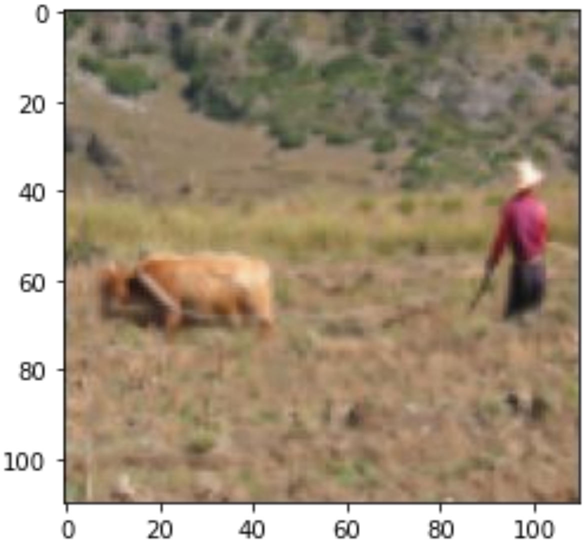


Fig a is the input given to correlation function to determine most similar patches.

Fig b is the output given when non uniform weights are taken

Experiment 5:- Using Gaussian pyramid for resizing

- Image pyramid refers to the way of representing an image at multiple resolutions. The idea behind this is that features that may go undetected at one resolution can be easily detected at some other resolution. For instance, if the region of interest is large in size, a low-resolution image or coarse view is sufficient. While for small objects, it’s beneficial to examine them at high resolution.
- In all the experiments above, we have scaled down the images gradually by fixing the step by which they can scale down. However such resizing can also be achieved by using gaussian pyramid.
- Instead of directly down scaling by fixing the step, the output of one block is given as input to cv2.pyrDown which gives the lower resolution image for a given higher resolution image. This output is passed an input to next block.

Result(1)(non gaussian with uniform weights) -

Fig a is the input given to the correlation function to find similar patches.

Fig b is the output given while uniform weights are taken and resizing is done gradually by fixing step.

Result(2)(gaussian without weights):-

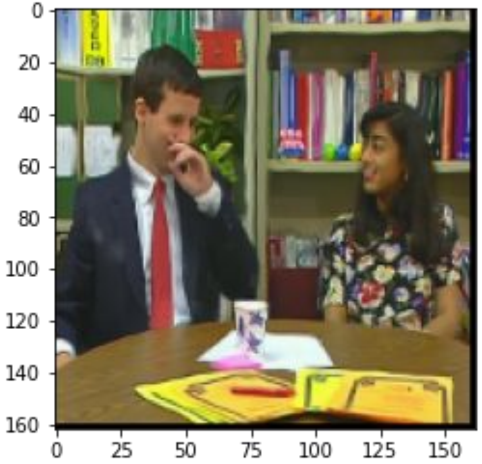


Fig a is the input given to the correlation function to find similar patches.

Fig b is the output given while uniform weights are taken and resizing is done gradually by using cv2.pyrDown library to make a gaussian pyramid

Result(3) -

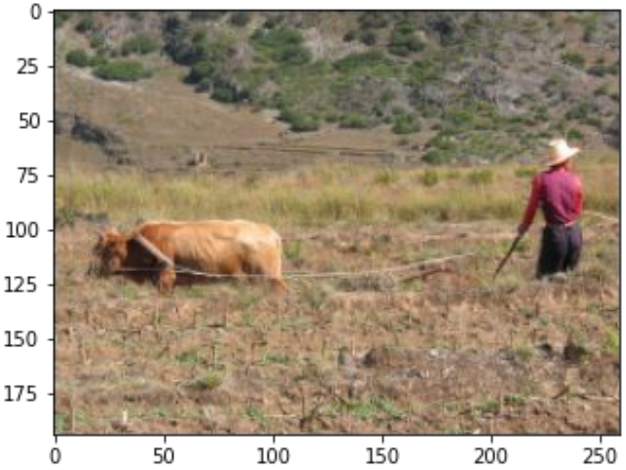
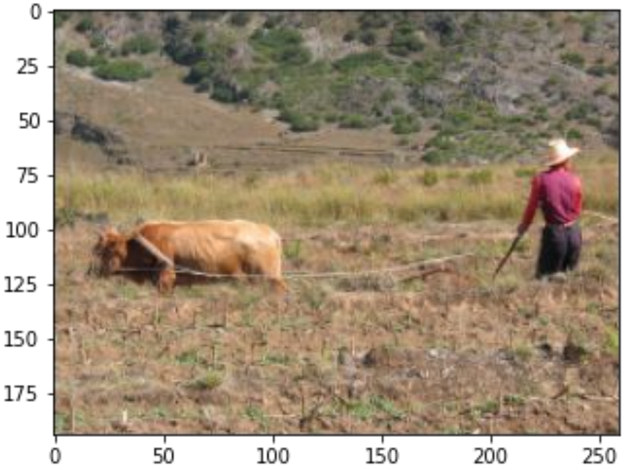


Fig a is the input given to correlation function to determine most similar patches.

Fig b is the output given when non-uniform weights are taken and resizing is done by fixing a step manually for gradual resizing

Result(4):-



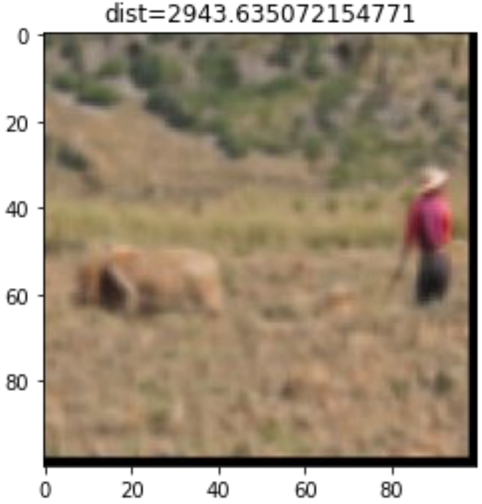


Fig a is the input given to the correlation function to find similar patches.

Fig b is the output given while non-uniform weights are taken and resizing is done

gradually by using cv2.pyrDown library to make a gaussian pyramid

Observations:-

- Based on first experiment, it can be said that although cropping technique looks good for one example, it fails for the example where important data isnt concentrated and redundant data is mixed with it. So the approach given in the paper is better than importance based cropping.
- Based on second experiment, it can be said that the speed of the algorithm has increased after removing the loops and can be used to analyze bigger sized images than before.
- Based on third experiment it can be said that correlation function is better suited to determine most similar patches because if distance graphs are observed for both correlation and ssd, ssd is more sensitive to the size of image as compared to correlation function. The output of correlation function is also better in some cases because of which for later experiments, correlation function is used.
- Based on fourth experiment it can be said that giving non uniform weights gives better results than uniform weights because the data is higher priority is retained and redundant data is being removed.
- Based on fifth experiment, it can be said that gaussian resizing doesn't yield better outputs because it resizes the images by half in one iteration and thereby loses important data. So gradual resizing by fixing a step is more preferred through out the experiments.

Conclusions

- ➢ Five different experiments done includes importance based cropping technique, bidirectional similarity technique, naive bidirectional similarity, optimized bidirectional similarity, gradual resizing by cv2 and gaussian pyramid.
- ➢ The summarized image in any case, is not exactly similar to the output given in the paper but the loss of redundant can be seen.
- ➢ Described a principled approach to retargeting and summarization of visual data by optimizing this bidirectional similarity measure.
- ➢ And the applications of this approach to summarization, data completion and removal, image synthesis, image collage, photo reshuffling and auto-cropping.

Contributions

Mohit Pavan kumar:-

Experiment-4, debugging, Importance based cropping

Tejaswini Anuhya Suma:-

Experiment-5,analysis of experiments and graphs, Importance based cropping

Gowthami Gongati:-

Experiment-3, Naive implementation of bidirectional similarity

Lakshmi Madhuri Yarava:-

Experiment-1,2, optimization of bidirectional similarity

Thank You