Deblurring Face Images Using Exemplars

Digital Image Processing Semester Project

Mentor TA: Anushree Korturti

Github Repo: https://bit.ly/3xPX2DX

Team Newt

Laksh Nanwani (CSE - 2021701002)
Manasvi Vaidyula (CSE - 2019101012)
Mehul Mathur (CSE - 2019101046)
Pooja Desur (CSE - 2019101112)

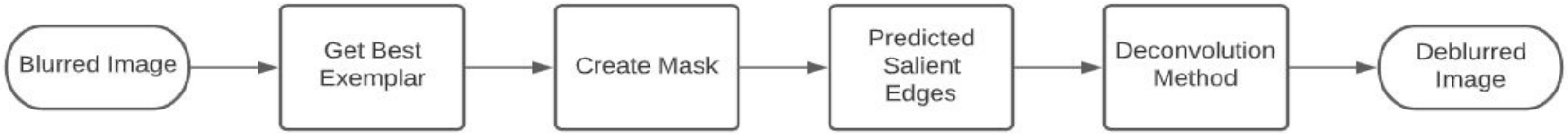
Paper Overview

- Aims to solve the problem of de-blurring of face images by using the more salient features of a human face
- Focused on motion blur images
- Salient features concluded by paper are - eyes , mouth and lower contour.
- Exemplar-based algorithm to extract blur kernel
  - Avoids incorrect or noisy information from extra edges (for example - hair)
  - Works for highly blurred images with fewer textures
- Non-blind deconvolution to recover latent image.
- Better and faster convergence than coarse-to-fine approaches of blur kernel estimation

Datasets

- Used in Paper - http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html
  - CIE Dataset
    - 2435 images
    - Faces with different expressions and orientations
    - Issue: paid
- Datasets Tried
  - Multi PIE - http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html
    - issue : paid
  - CVL - http://lrv.fri.uni-lj.si/facedb.html
    - Issue: paid
  - http://robotics.csie.ncku.edu.tw/Databases/FaceDetect_PoseEstimate.htm
    - Issue: no expressions
    - Issue: Only 90 images
- Dataset Used - https://www.kaggle.com/jonathanoheix/face-expression-recognition-dataset
  - Extracted 2000 images - https://drive.google.com/drive/folders/1acmtZQFaZP2D-l3ESXVikd13sIzCJpfa?usp=sharing(
  - Different expressions - happy, angry, disgust, fear, neutral, sad, surprise
  - Various orientations

Pipeline



Note: Blurred Images were taken by applying horizontal motion blur on images from the dataset

Exemplars

- The reference image in our dataset used to deblur image
- Task : Find best matched exemplar for given blurred image B
- How?
  - Get max over NCC over translated exemplar gradient and blurred image gradient
  - Translated from -15 to 15
  - Max value over all exemplars

$$v_i = \max_t \left\{ \frac{\sum_x \nabla B(x) \nabla T_i(x+t)}{\|\nabla B(x)\|_2 \|\nabla T_i(x+t)\|_2} \right\},$$

Exemplars

Here are some of the examples from exemplars matched through cross-correlation values (when the original image is not present)

Blurred Image

Exemplar

Blurred Image

Exemplar

Blurred Image

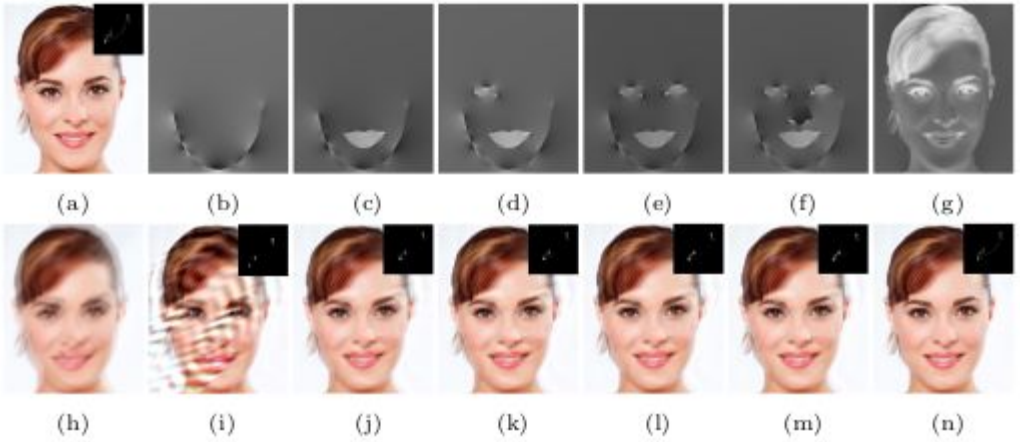
Exemplar

Blurred Image

Exemplar

Mask

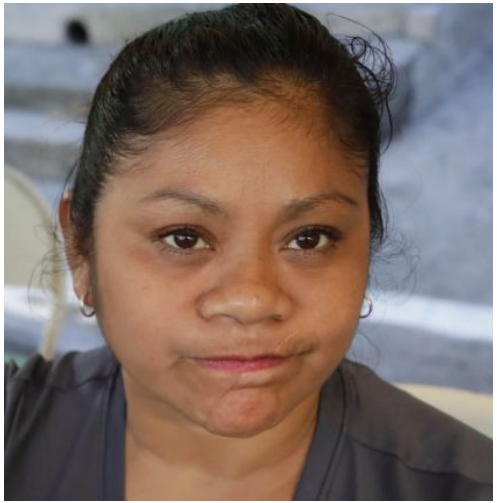
- Using the best exemplar, create a mask
- Best mask as found by paper -
  - Eyes
  - Mouth
  - Lower Contours (Jaw)



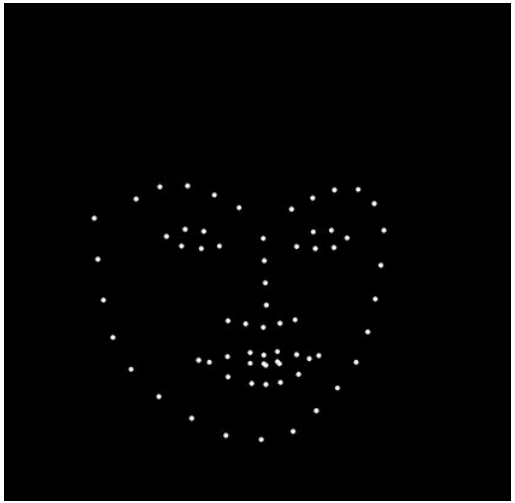
Creating the Mask

- Dlib is used to generated facial landmarks (68 points)
- Nose and eyebrows landmarks are removed
- Isolated Facial Parts are connected using cv2.line, generating a mask

Exemplar Image



Landmarks detected



Mask generated

Salient Edges

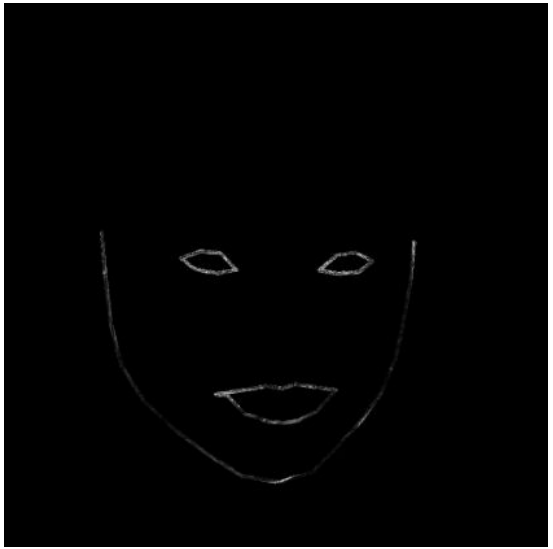
$$\nabla S_{i^*}(x) = \begin{cases} \nabla T_{i^*}(x), & \text{if } x \in \{x | \mathcal{M}_{i^*}(x) = 1\}, \\ 0, & \text{otherwise.} \end{cases}$$

- Edges of the exemplar are taken
- Mask from previous step is applied to these edges, giving Salient edges

Mask

Edges

Salient Edges



Blur Kernel Estimation

Algorithm 1 Solving (6)

Input: Blur image B and estimated kernel k.
$I \leftarrow B$, $\beta \leftarrow 2\lambda$.
repeat
  solve for $\mathbf{w}$ using (11).
  solve for I using (10).
  $\beta \leftarrow 2\beta$.
until $\beta > 1e^5$
Output: Latent image I.

Algorithm 2 Blur kernel estimation algorithm

Input: Blur image B and predicted salient edges ∇S.
for $l = 1 \rightarrow n$ **do**
  solve for k using (7).
  solve for I using Algorithm 1.
  $\nabla S \leftarrow \nabla I$.
end for
Output: Blur kernel k and intermediate latent image I.

Blur Kernel Estimation

Brief steps for the entire algorithm:

1. Obtain a blur kernel estimated through gradient descent (Equation (1)).
2. Obtain a Latent image I using the blur kernel from 1.
   a. Take an initial value of lambda and beta
   b. Get auxiliary variable w using equation (2)
   c. Get the latent image I using the fourier solution of (3) (equation (4))
   d. Repeat from b. Till beta reaches a max value
3. Update the salient edges (del S) with del I.
4. Repeat from 1. For a number of iterations
5. Get final latent image as the output

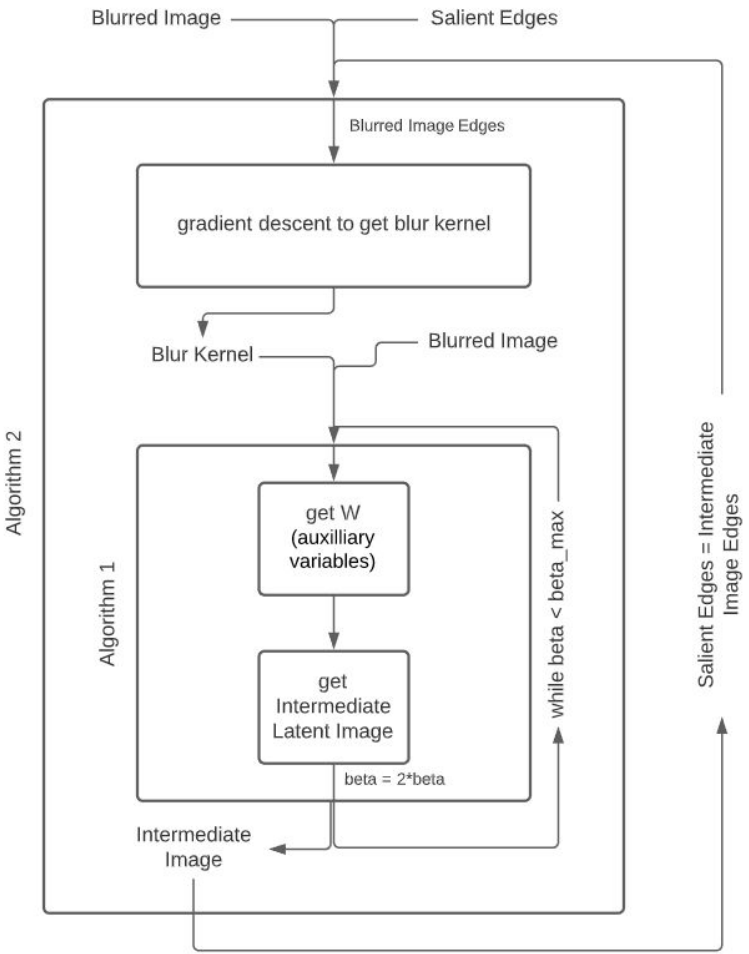
$$\min_k \|\nabla S * k - \nabla B\|_2^2 + \gamma \|k\|_2^2, \quad \text{——— (1)}$$

$$\mathbf{w} = \begin{cases} \nabla I, & |\nabla I|^2 \geqslant \frac{\lambda}{\beta}, \\ 0, & \text{otherwise.} \end{cases} \quad \text{——— (2)}$$

$$\min_I \|I * k - B\|_2^2 + \beta \|\mathbf{w} - \nabla I\|_2^2, \quad \text{——— (3)}$$

$$I = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(k)}\mathcal{F}(B) + \beta(\overline{\mathcal{F}(\partial_x)}\mathcal{F}(w_x) + \overline{\mathcal{F}(\partial_x)}\mathcal{F}(w_y))}{\overline{\mathcal{F}(k)}\mathcal{F}(k) + \beta(\overline{\mathcal{F}(\partial_x)}\mathcal{F}(\partial_x) + \overline{\mathcal{F}(\partial_y)}\mathcal{F}(\partial_y))}\right). \quad \text{——— (4)}$$

Diagramatic Explanation

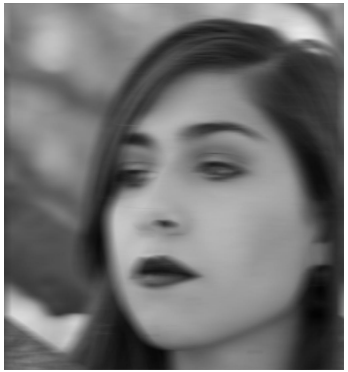


Hyperparameters Used

- Number of Total Iterations: 50
- Gradient Descent for K in Algo 2
  - Epochs: 100
  - Tolerance: 1e-10
  - Learning rate: 1e-10
- Algo 1
  - Initial beta: 1
  - Beta_max: 256

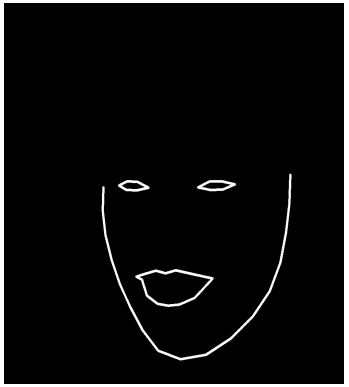
Final Results

Blurred image

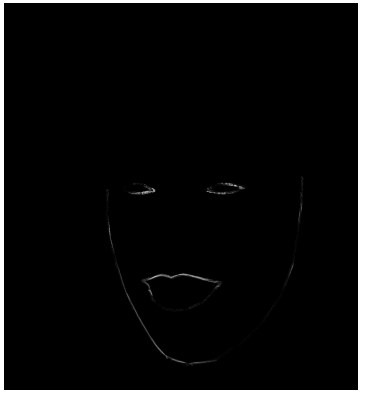


Matched Exemplar

Mask



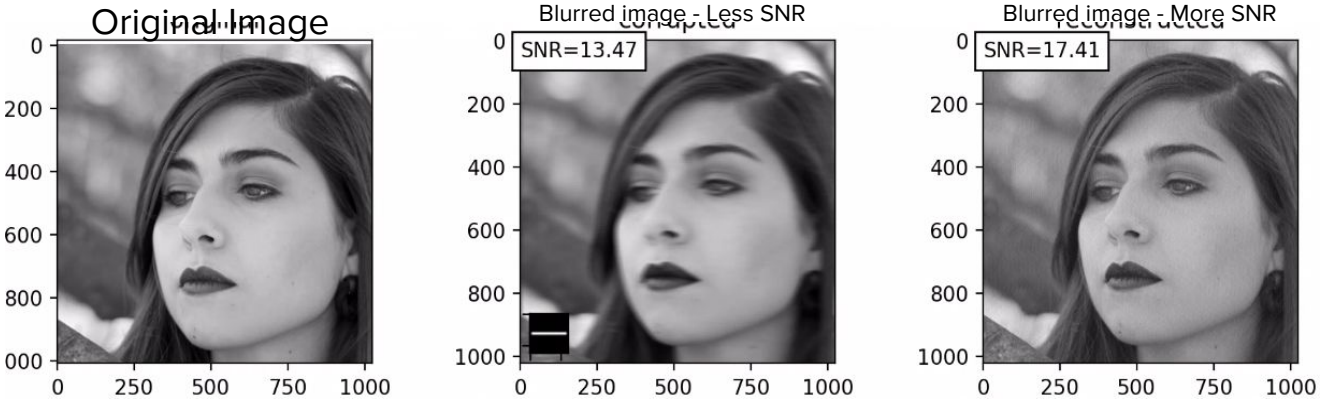
Salient Edges



Epochs

Evaluation Metrics - SNR Ratio

- SNR Ratio
  - Ratio of average signal value to standard deviation of signal
  - Measurement for quality of image
  - To measured the improvement we compared the signal-to-noise ratios of original,blurred and original, de-blurred images
  - Desired : latter to have higher SNR values
- For evaluation - SNR between a true clear image and image x:
  - Variance of clear image (signal) divided by Mean((x - true image)^2) (noise)
  - As seen below , the SNR of the deblurred output is more than that of the blurred image , indicating good retrieval



References

- Paper Implemented: https://faculty.ucmerced.edu/mhyang/papers/eccv14_deblur.pdf
- https://papers.nips.cc/paper/2009/hash/3dd48ab31d016ffcbf3314df2b3cb9ce-Abstract.html
- https://www.murtazahassan.com/courses/opencv-projects/
- https://inst.eecs.berkeley.edu/~cs194-26/fa17/Lectures/ConvEdgesTemplate.pdf
- http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=1F40310FF3E4F778B8253D4FFDCF91CF?doi=10.1.1.295.9311&rep=rep1&type=pdf

Division Of Work

- We took advantage of the VScode LiveShare extension which allows multiple people to edit the same codebase simultaneously
- Laksh - Gradient Descent for kernel estimation , Algorithm 2
- Manasvi - Algorithm 2, motion blur, GUI
- Mehul - Algorithm 1, GUI
- Pooja - best exemplar, mask creation, Algorithm 1