Computational Photography - CS 445
Project 3: Gradient-Domain Fusion

by

Deepti Sharma (deeptis2@illinois.edu)

Table of Contents

Deliverables

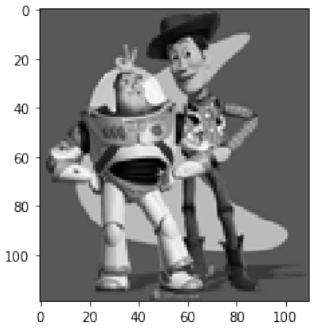
Toy Reconstruction (20 points)

Summary:

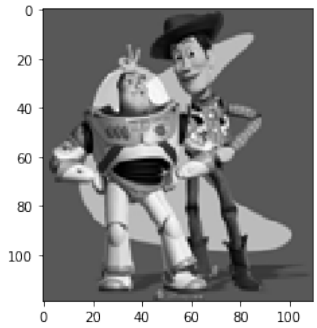
Solve least square problem with these constraints

1. minimize (v(x+1,y)-v(x,y) - (s(x+1,y)-s(x,y)))^2
2. minimize (v(x,y+1)-v(x,y) - (s(x,y+1)-s(x,y)))^2
   Note that these could be solved while adding any constant value to **v**, so we will add one more objective:
3. minimize (v(0,0)-s(0,0))^2

Results



Original Image



Reconstructed Image

Poisson Blending (50 pts)

Summary

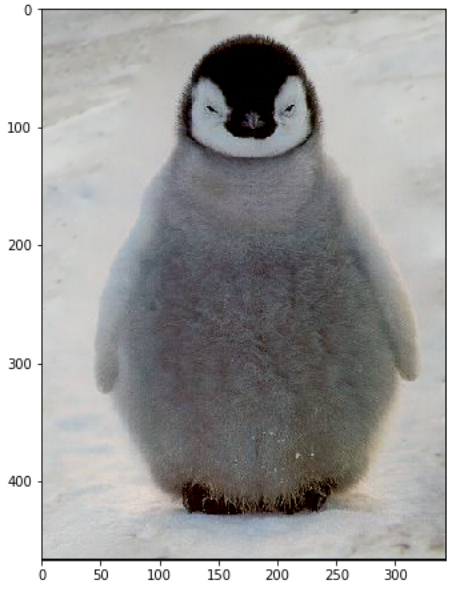
Select Object in source image, and Target location in background image
Create a mask to place the object based on user selection
Take the smallest matrix that covers the masked region and use this smaller image for Gradient Descent Computation
Solve for Least Squares for each Color Plane – RGB with following constraints:

1. minimize (v(x+1,y)-v(x,y) - (s(x+1,y)-s(x,y)))^2
2. minimize (v(x,y+1)-v(x,y) - (s(x,y+1)-s(x,y)))^2
3. minimize (v(x+1,y)-t(x,y) - (s(x+1,y)-s(x,y)))^2
4. minimize (v(x,y+1)-t(x,y) - (s(x,y+1)-s(x,y)))^2

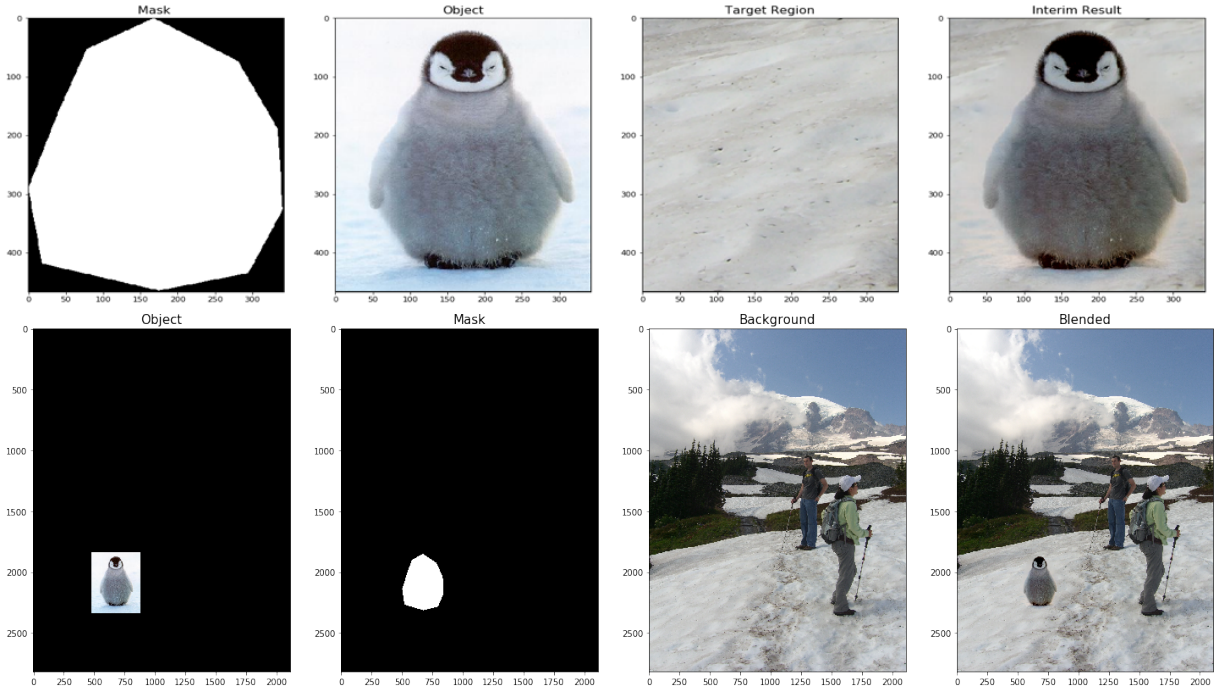
Matrix Size = 2*((im_h)*(im_w-1) + (im_h-1)*(im_w))
Result of Least Squares is the intensity image
Put the smaller image back to the original image in the masked region
Outside the masked region, copy the contents of the background image

Observations

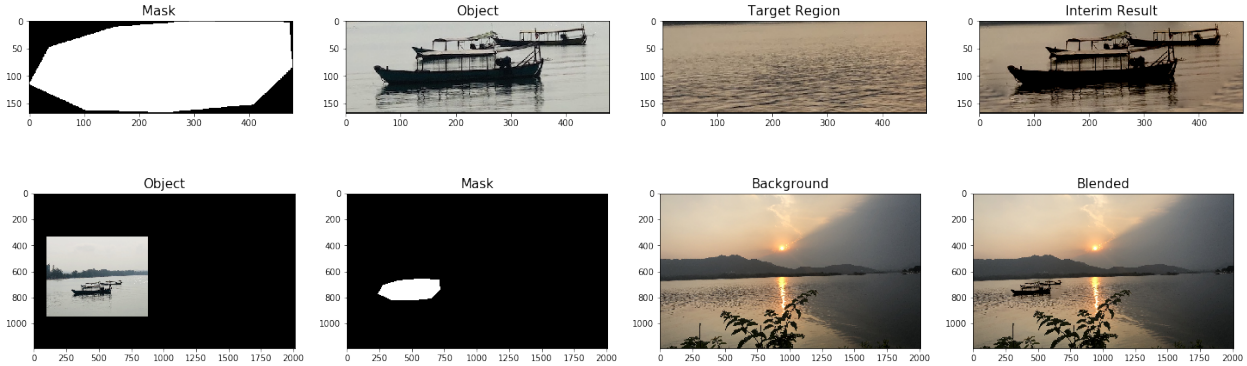
The edges in the result images are smooth but area around the cropped object seems has smoother texture, different from the background image



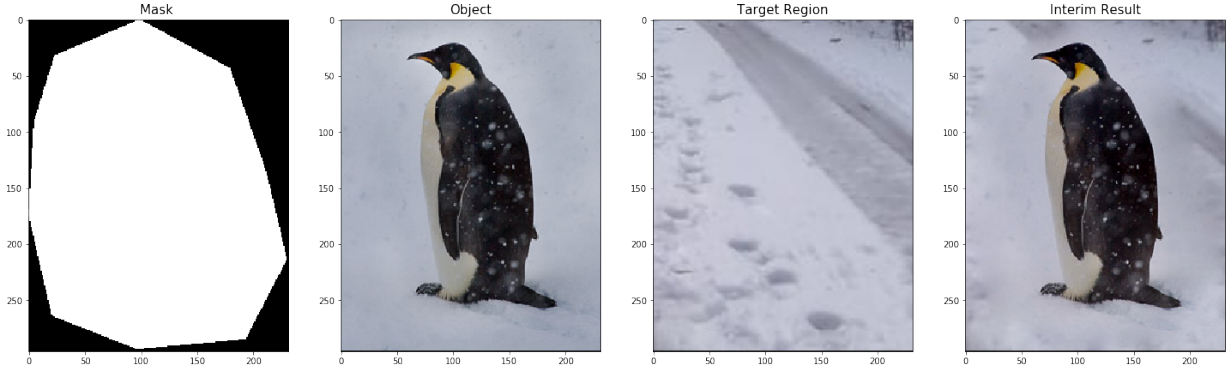
Result 1: Penguin with Hikers

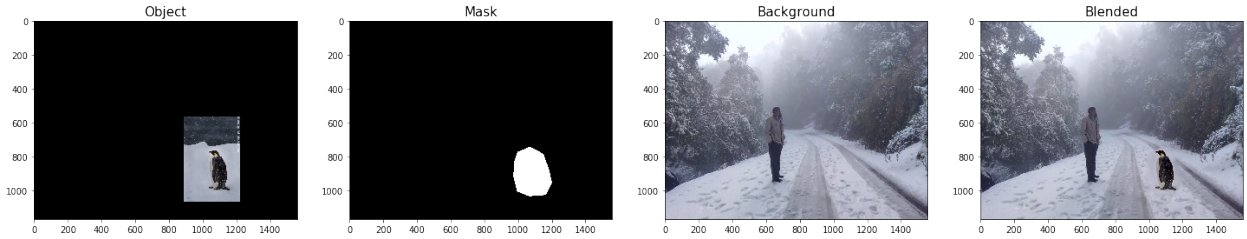


Result 2: Boats in the Lake



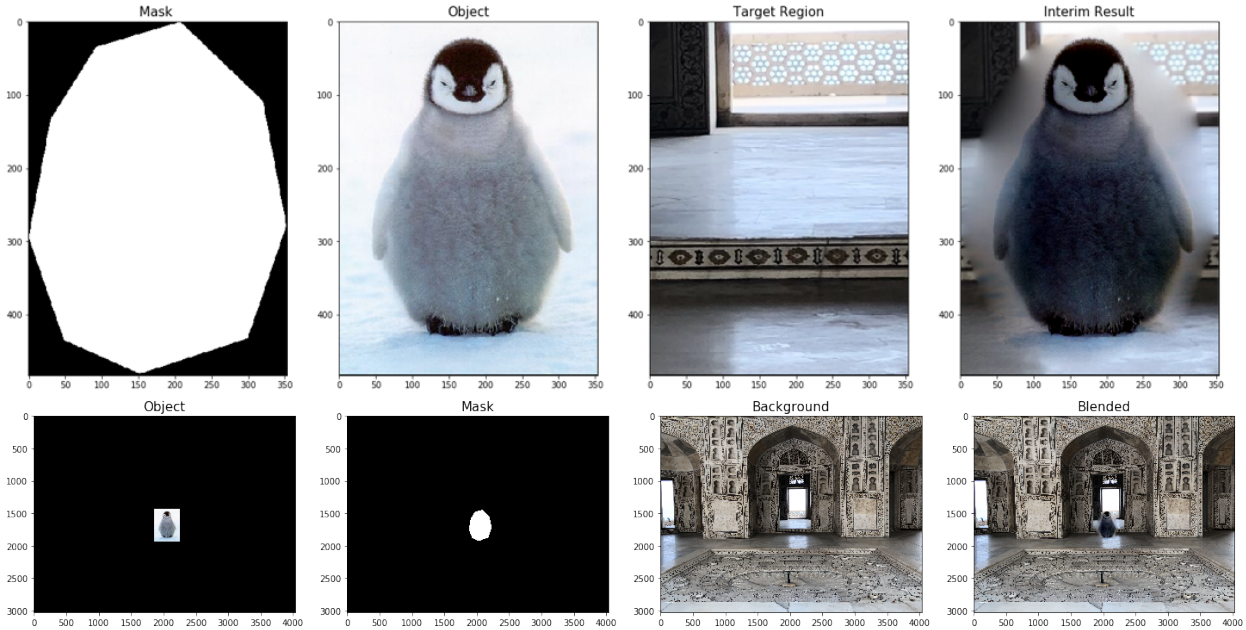
Result 3: Penguin looking at a Man in the Snow



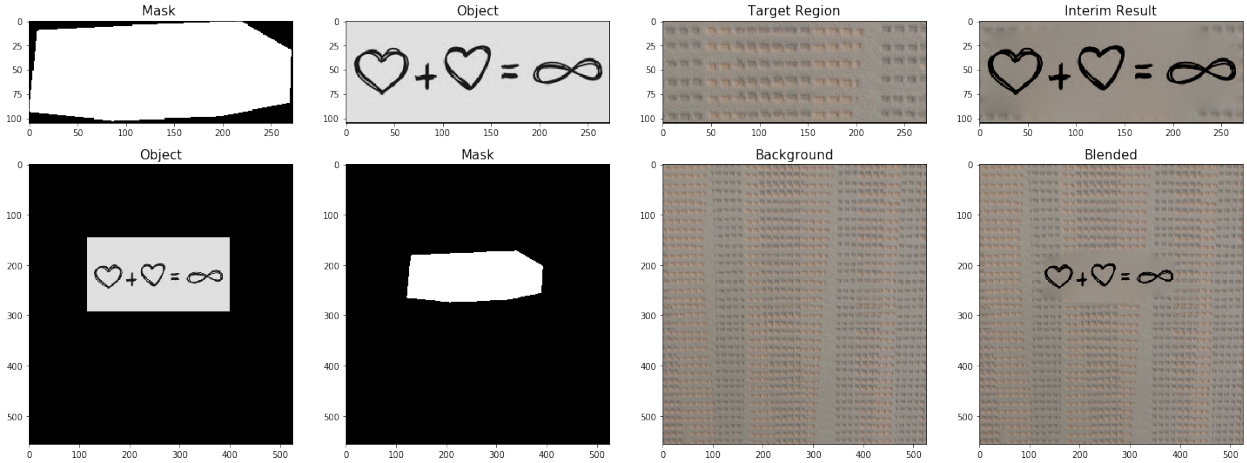


Result 4: Penguin in the Palace

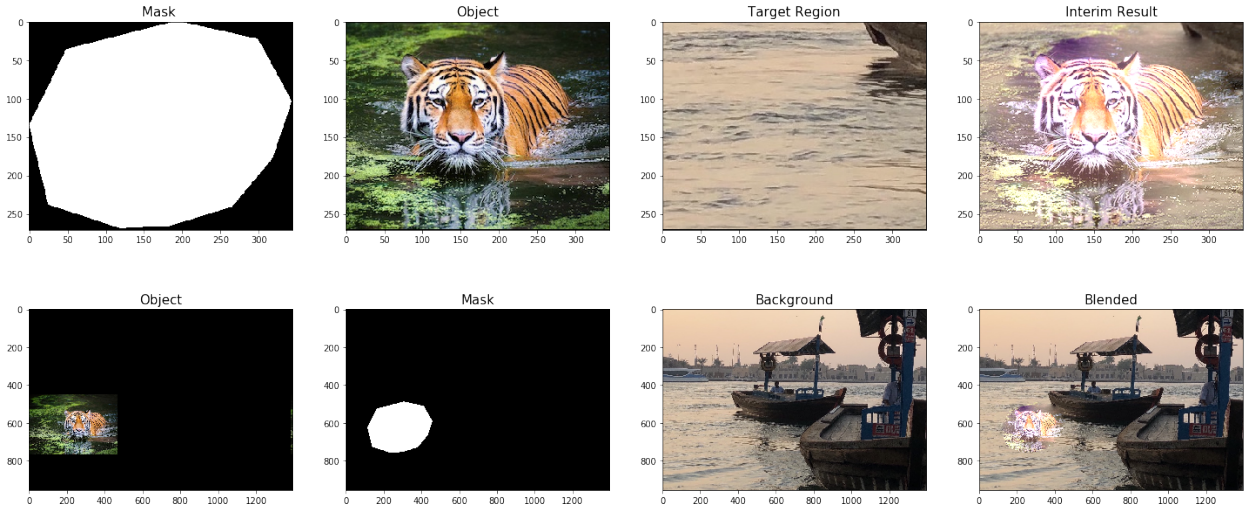
Penguin looks out of place here



Result 5: Pen Writing on a Textured Wall



Result 6: Tiger in Water



Mixed Gradients (20 pts)

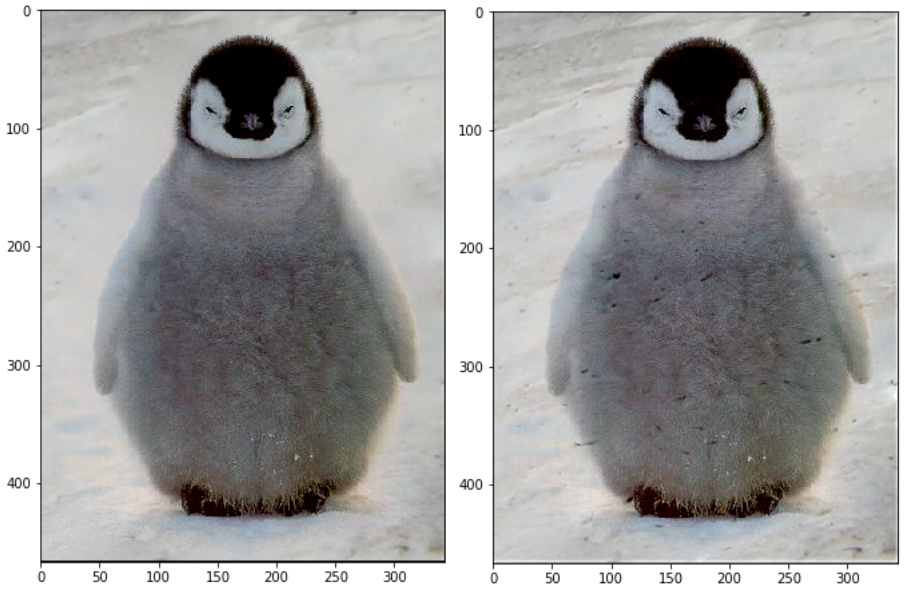
Summary

- Select Object in source image, and Target location in background image
- Create a mask to place the object based on user selection
- Take the smallest matrix that covers the masked region and use this smaller image for Gradient Descent Computation
- Solve for Least Squares for each Color Plane – RGB with following constraints
    1. minimize (v(x+1,y)-v(x,y) - max(abs(s(x+1,y)-s(x,y)),abs(t(x+1,y)-t(x,y))))^2
    2. minimize (v(x,y+1)-v(x,y) - max(abs(s(x,y+1)-s(x,y)),abs(t(x,y+1)-t(x,y))))^2
    3. minimize (v(x+1,y)-t(x,y) - max(abs(s(x+1,y)-s(x,y)),abs(t(x+1,y)-t(x,y))))^2
    4. minimize (v(x,y+1)-t(x,y) - max(abs(s(x,y+1)-s(x,y)),abs(t(x,y+1)-t(x,y))))^2

    - Matrix Size = 2*((im_h)*(im_w-1) + (im_h-1)*(im_w))
- Result of Least Squares is the intensity image
- Put the smaller image back to the original image in the masked region
- Outside the masked region, copy the contents of the background image

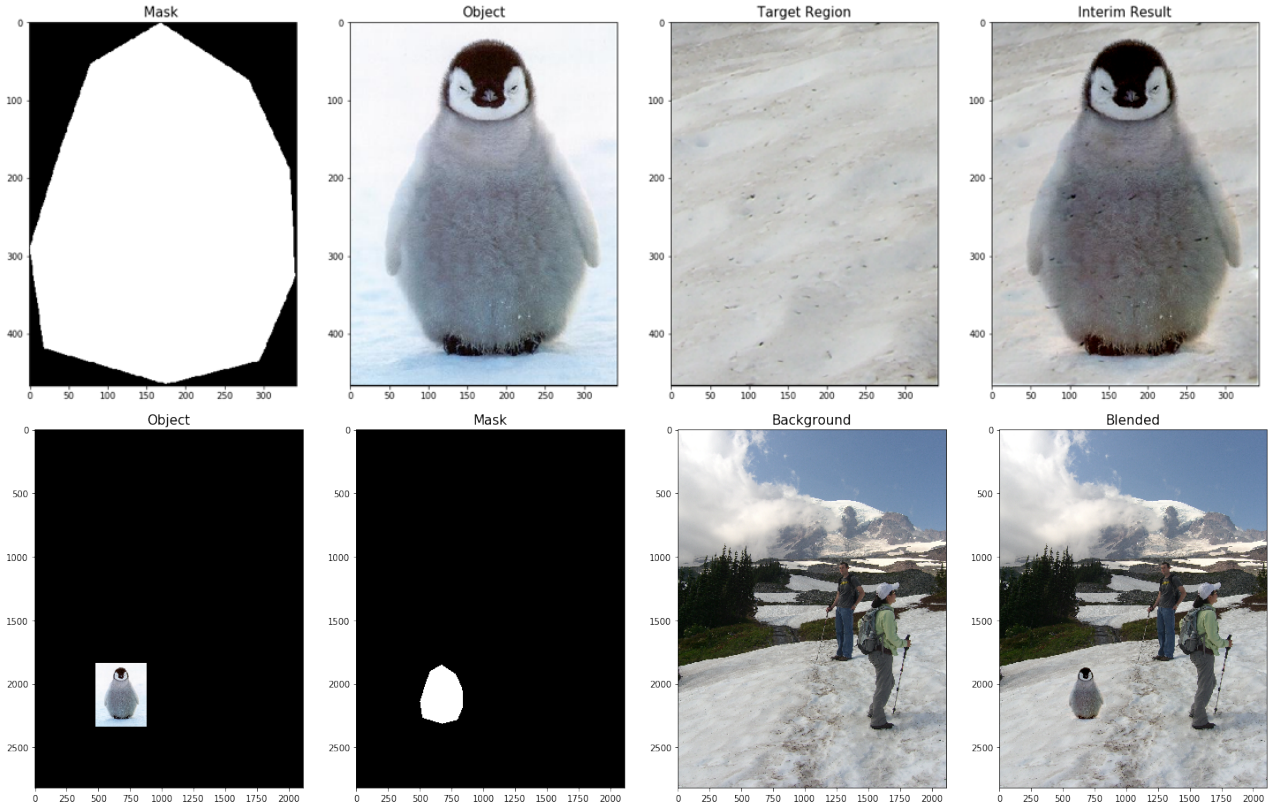
Observations

The edges in the result images are smooth and area around the cropped object has a similar texture as the background image, looks much better compared to the Poisson Blend.

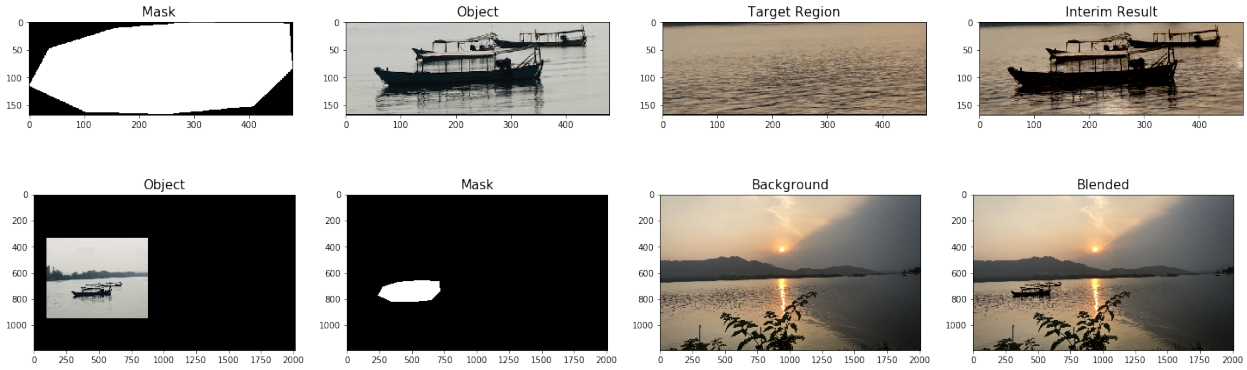


Poisson Blend | Mix Gradients Blend

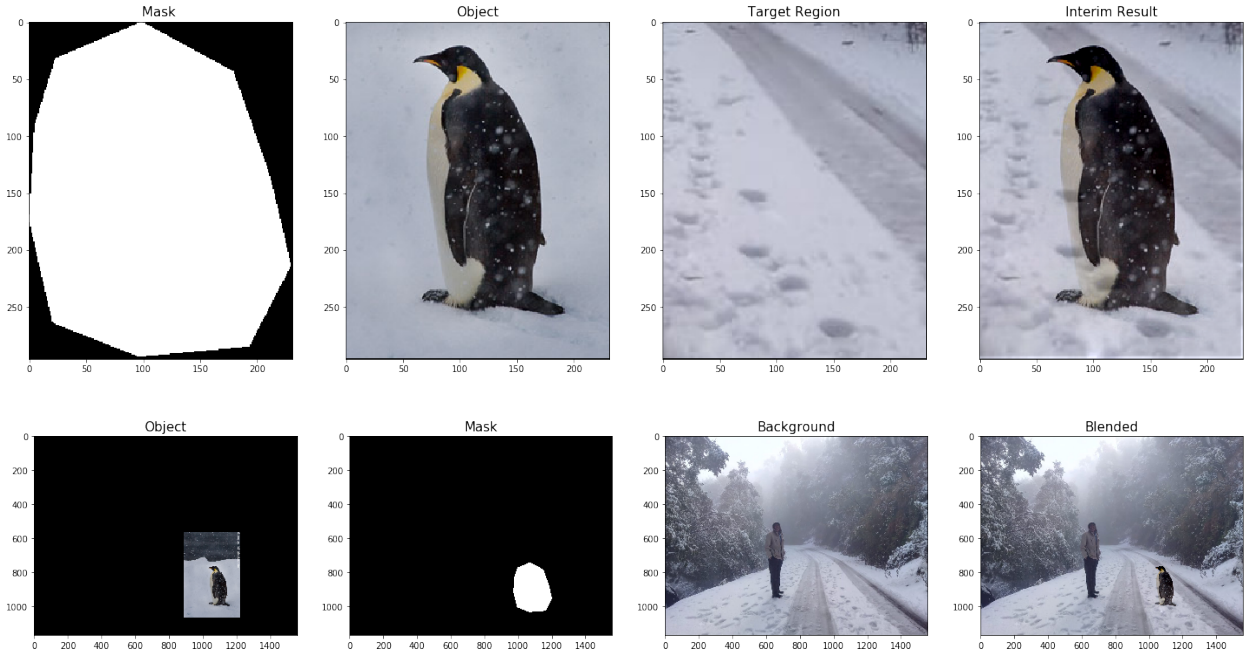
Result 1: Penguin with Hikers



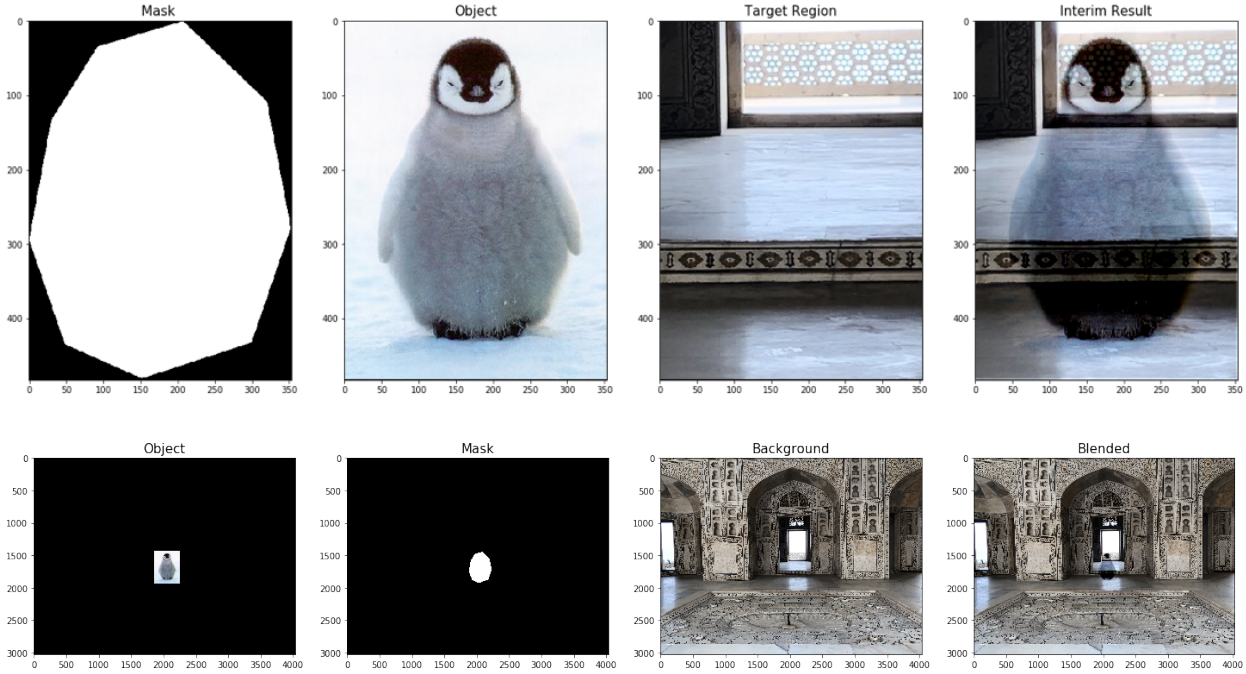
Result 2: Boats in the Lake



Result 3: Penguin looking at a Man in the Snow

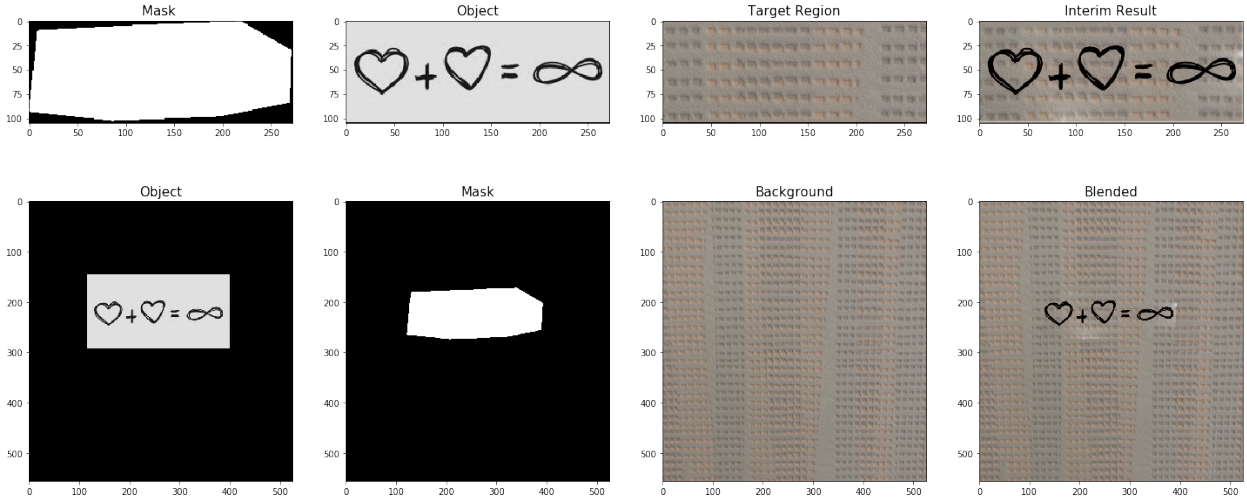


Result 4: Ghost Penguin in the Palace



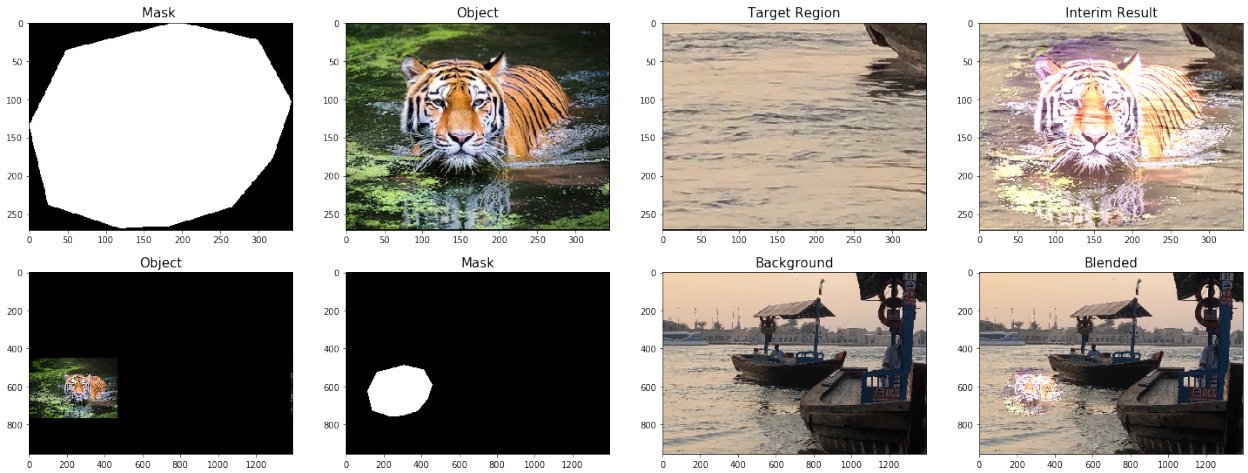
Result 5: Pen Writing on a Textured Wall

This does not blend well with the background



Result 6: Tiger in Water

This does not blend well with the background



Bells & Whistles (Extra Points)

Color2Gray (20 pts)

Summary

Normalize each color channel

C = max(Intensity)/ (1 + max(Intensity)

Normalized Intensity = C * Intensity

Compute Average Intensity

Result 0= Average(Normalized (r), Normalized (g), Normalized (b))

This works for first image colorBlind4 but NOT for colorBlind8

Compute Gradient descent of RGB planes in pairs

Result 1 = gradient(Normalized (r)- Normalized (g))

Result 2 = gradient(Normalized (r)- Normalized (b)))

Result 3 = gradient(Result1, Result2)

Works with both images colorBlind4 and colorBlind8

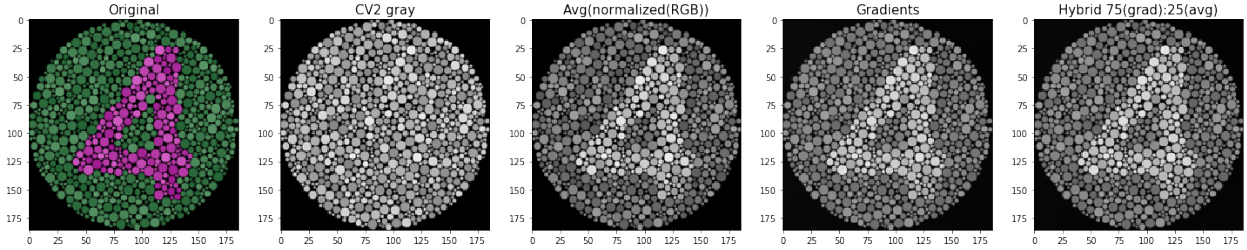
Hybrid

Result = X (Gradient) + (1 – X)(Avg)

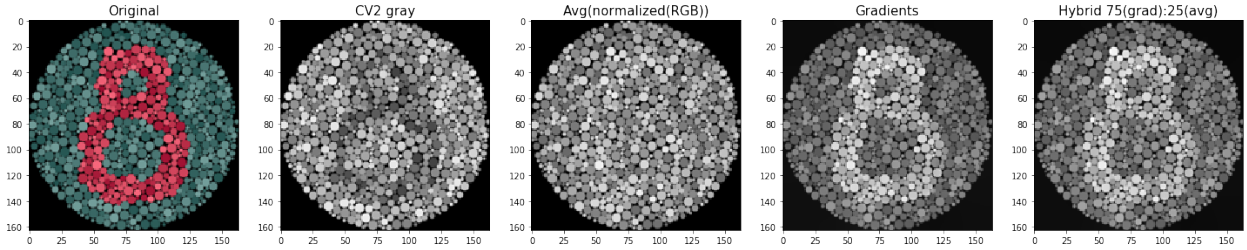
Observations

Gradient descent gives the best contrast against the varying intensities in different color planes

Result 1: Purple 4



Result 2: Red 8



Laplacian pyramid blending (20 pts)

Summary

At L0

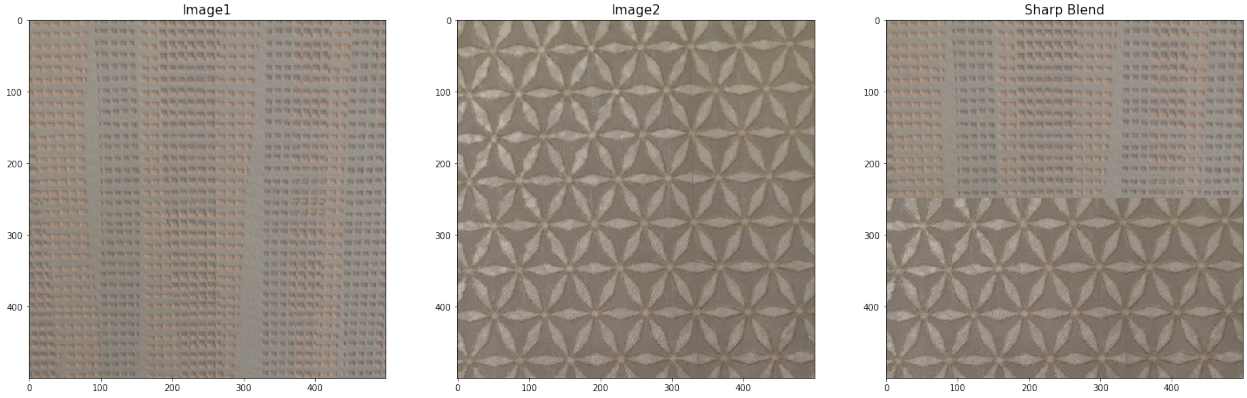
Laplacian Transform of Image 1 (LL0Image1) and Image 2 (LL0Image2) at Level 0
Level 0 Combined = Mask(LL0Image1) + (1-Mask)(LL0Image2)
Apply Gaussian Filter and reduce size of Image1 and Image2

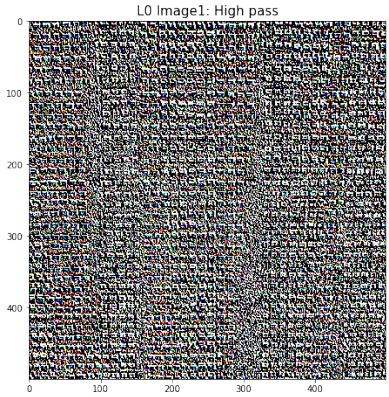
At L1

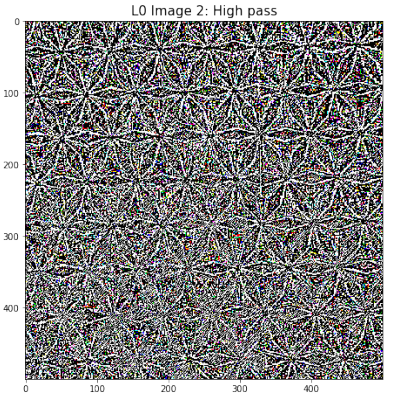
Level 1 Combined = Mask(GL1Image1) + (1-Mask)(GL1Image2)

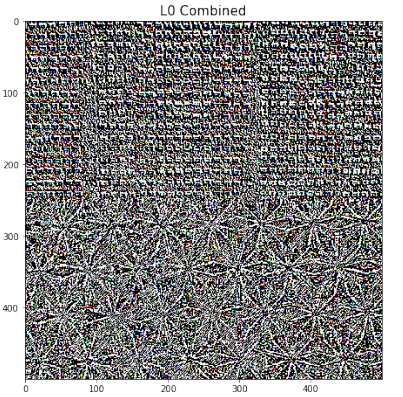
Up-sample Level 1 Combined image and add it to Level 0 Combined to get final results

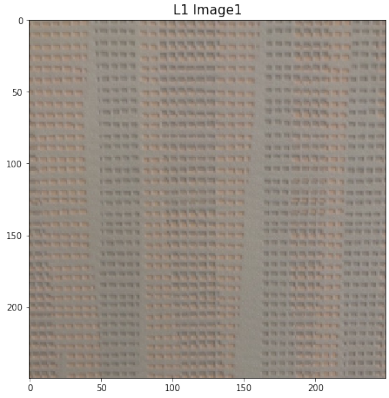
Result 1: Two Textures

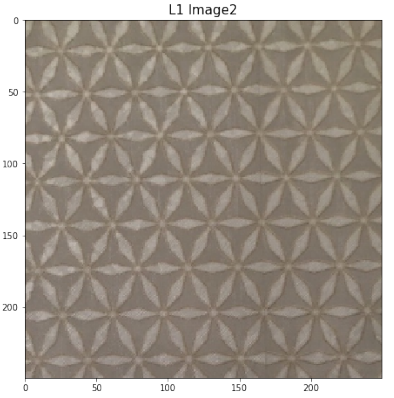


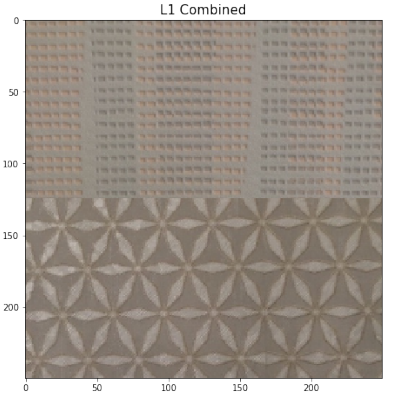


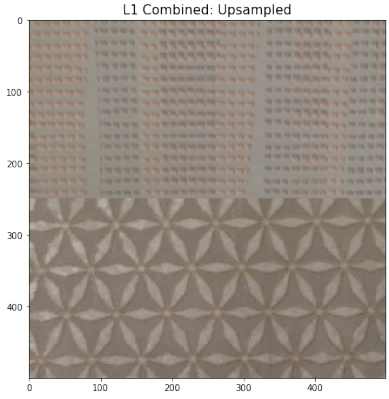


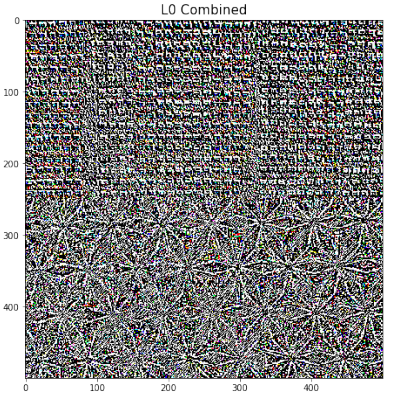


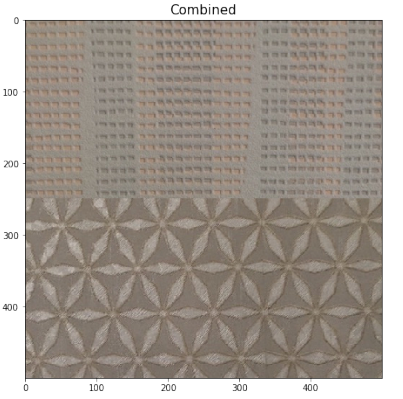




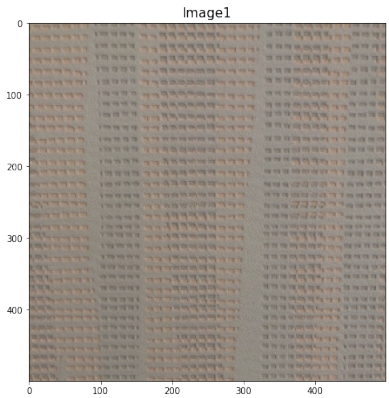


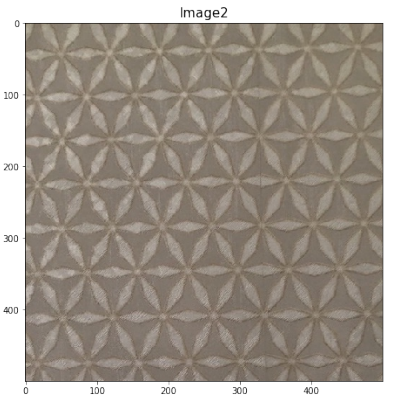


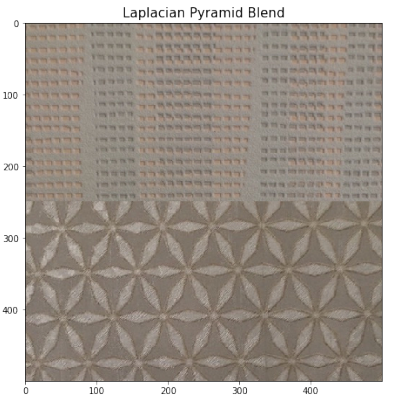




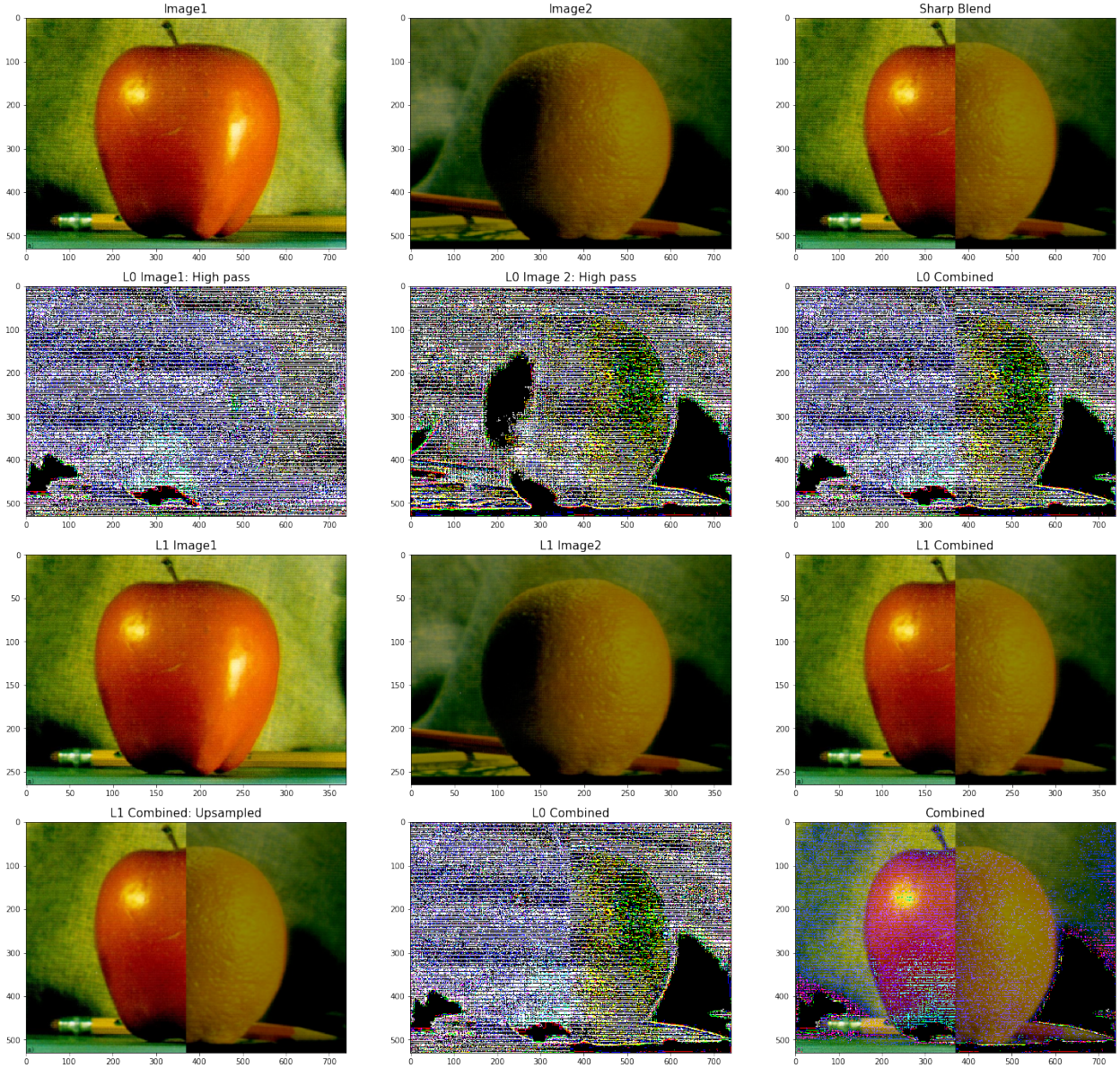
Final Results:







Result 2 : Apple and Orange



Final Results