Time Hallucination

- **Team :** *POTATO HEAD*

- **Members :**
- - Srividya Srigiri
- - Poojitha Mittapalli
- - Goutham C M

Problem being solved

Given a single input image we try to hallucinate the same scene at a different time of the day.

For this we use a database of timelapse video to get the information for hallucinating a different time of the day.

Introduction

Time hallucination is the synthesis of an image at a different time of day from an input image.

Time of day and lighting conditions for photographers.

Most photographers cannot be at right place at the right time and end up taking photos in some harsh lightning.

We implement an automatic technique that taken a single outdoor image as input and seeks to hallucinate the image getting the same scene at a different time of the day.

Overview of the method

We try to match the input image with frames from a time lapse database.

We match the input image globally to time-lapse videos of similar scenes, and find a dense correspondence based on a Markov random field

After getting a pair of time-lapse frames (frame at target time and at the time of the given image) we transfer the appearance variation between the time-lapse frames to the input image.

Algorithm :

Global Matching

Local Matching

Locally affine color transfer

Global Matching

- Firstly we find the video showing the scene similar to the input image.
- We sample each video in to 5 equally spaced frames and compare these frames to the input image.
- To attain a quantitative measure of similarity we assign a similarity score to each frame and this will be done by Histograms of Oriented Gradients(HOG) method.
- The frame that get highest similarity score or the frame that matches with the time of the input image is the *Matched frame(M)*

Local Matching

- We try to pair each pixel of input image(I) to the matched frame(M).
- The similarities between the matched and the time-lapse video can be modeled using the Markov random field and the data term is defined as:

$$E_1 = \sum_{i=-r}^{+r} \sum_{j=-r}^{+r} \left\| I(x_p + i, y_p + j) - M(x_q + i, y_q + j) \right\|^2$$

- The pair-wise term is defined by: $E_2(q_i, q_j) = \max_t \sum_{o \in \Omega} \left\| V_t(q_i + \delta_i) - V_t(q_j + \delta_j) \right\|^2$

Which assign the score depending on the similarity between the successive time frames in the image.

- We try to minimize the value of the energy term to get the best match between video and image. $\sum_{i \in I} E_1(p_i, q_i) \quad + \quad \lambda \sum_{i \in I, j \in N_i} E_2(q_i, q_j)$

Locally affine color transfer

- In this step we apply the color correspondence from the matched frame to the target frame onto the input frame.
- Using affine models that define the transfer between the wrapped images, we can the output image at the required time.
- Naively, Affine model can be defined as the regression between kth patch of the matched frame and patch of the target image.

L2-optimal locally affine model

- Let $v_k()$ be the 3xN matrix with each column denoting the color of a pixel in the kth patch of an image.
- Let $v_{k'}()$ be the 4xN matrix with augmented column of ones to $v_k()$.
- A_k is a 3x4 matrix which represents the local affine functions that transforms M' to T'.

$$\sum_k \left\| \mathbf{v}_k(\tilde{T}) - \mathbf{A}_k \, \bar{\mathbf{v}}_k(\tilde{M}) \right\|_{\mathsf{F}}^2$$

- We also apply least square formulation over output image and input image.

$$\sum_k \left\| \mathbf{v}_k(O) - \mathbf{A}_k \, \bar{\mathbf{v}}_k(I) \right\|_{\mathsf{F}}^2$$

- Where O is output image.

$$O = \arg\min_{O,\{\mathbf{A}_k\}} \sum_k \left\| \mathbf{v}_k(O) - \mathbf{A}_k \, \bar{\mathbf{v}}_k(I) \right\|^2 + \epsilon \sum_k \left\| \mathbf{v}_k(\tilde{T}) - \mathbf{A}_k \, \bar{\mathbf{v}}_k(\tilde{M}) \right\|^2 + \gamma \sum_k \left\| \mathbf{A}_k - \mathbf{G} \right\|_{\mathsf{F}}^2 \quad (7)$$

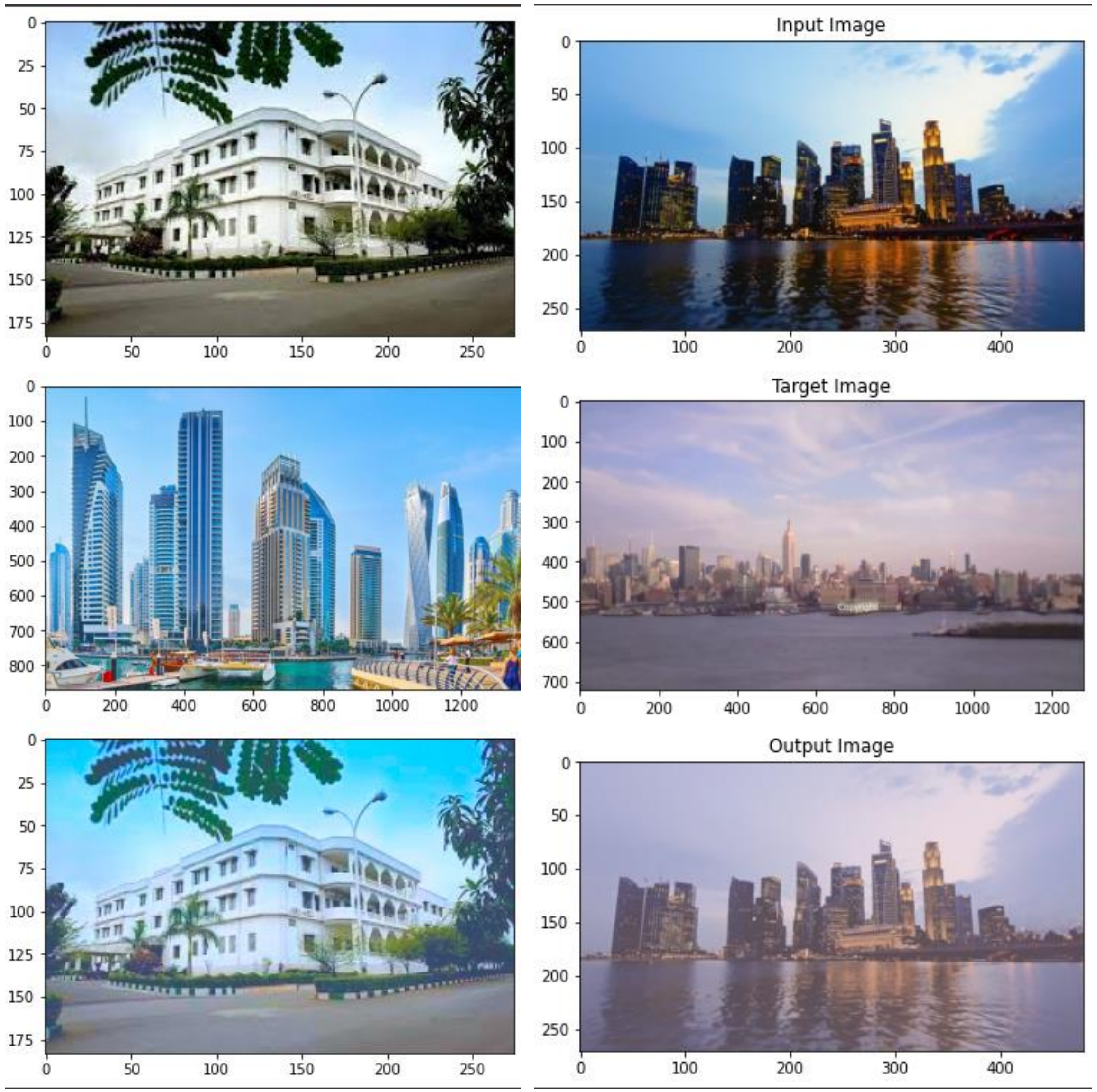
Dealing with Noisy input

- To decrease the effect of noise on output image, we deal the input image layer by layer.
- We first decompose the input image into base layer and a detail layer bilateral filtering, then apply the affine model to base layer before adding the detail layer.
- Compared to directly processing the input image we prevent the problem of increasing noise

Color Transfer

- Initially, the image is converted from RGB color space to LAB color space.
- Now for each axis, mean and standard deviations are computed for both the target and source image.
- First, the mean is subtracted from the source image and scaled with the standard deviation of the source image and target image.
- Now the current color space of the source image conform to the target image and finally their color space is converted back to the RGB color space.

Results



Challenges

- Computationally expensive, since it takes around 12-15 seconds to find the descriptors for every video and there are 450 videos in total making the time to compute descriptors to around 7-8 minutes.
- All the videos are of unequal durations making it difficult to pinpoint a particular time frame for all of them

Work split-up

Team Member	Work Contribution
Srividya	• Local matching, Locally affine color transfer, L2 optimal locally affine model. • Code : GettingTargetFrame, referenceFrame, testCode. • Presentation : Local matching, Affine models and L2 optimal affine model.
Poojitha	• Global matching, L2 optimal locally affine model, Dealing with noise inputs. • Parts of Color transfer, Dealing with testing the functions and outputs. • Presentation : Introduction, Algorithm overview and Global matching.
Goutham	• A great detailed info of Color transfer (actual implementation of code). • Global matching, HOG descriptors, Parts of Color transfer. • Presentation : Dealing with noisy input, Color transfer, Results and challenges.

Future Scopes

INCORPORATING NEURAL NETWORK MODELS TO FIND THE BEST MATCHING FRAMES/VIDEOS TO ENHANCE THE RESULTS.

DEALING WITH NOISY INPUT.

THANK YOU