Data-driven Hallucination

Team: **Image Processors**

Manish Sharma	-	2018101073	Btech. CSE
Akshat Goyal	-	2018101075	Btech. CSE
Dixit Kumar Garg	-	2018101077	Btech. CSE
Khadiravana	-	2019121008	Btech. CSD

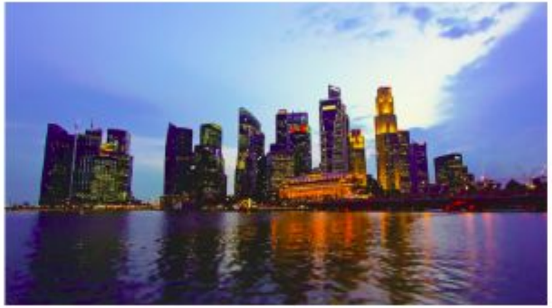
Mentor TA - Prajwal Krishna

Repo Link

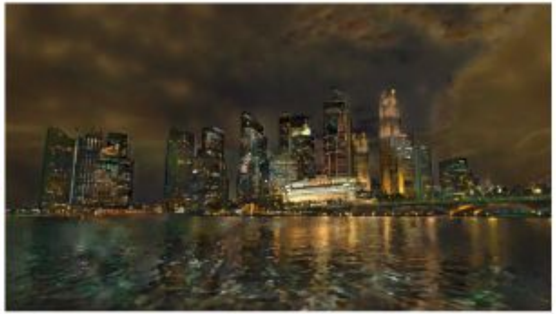
Original Paper link

Problem statement

Given a single image as input, we want to automatically create a plausible-looking photo that appears as though it was taken at a different time of the day. This should be done using a fixed database of time lapse videos, of which the input image may not be a part of. Also we want to make the image look realistic, while preserving the colour schema.



Input image at “blue hour”



Hallucinations at night

Utility

Most photographers cannot be at the right place at the perfect time and end up taking photos in the middle of the day when lighting is harsh. In this project, we implement an automatic technique that takes a single outdoor photo as input and seeks to hallucinate an image of the same scene taken at a different time of day.

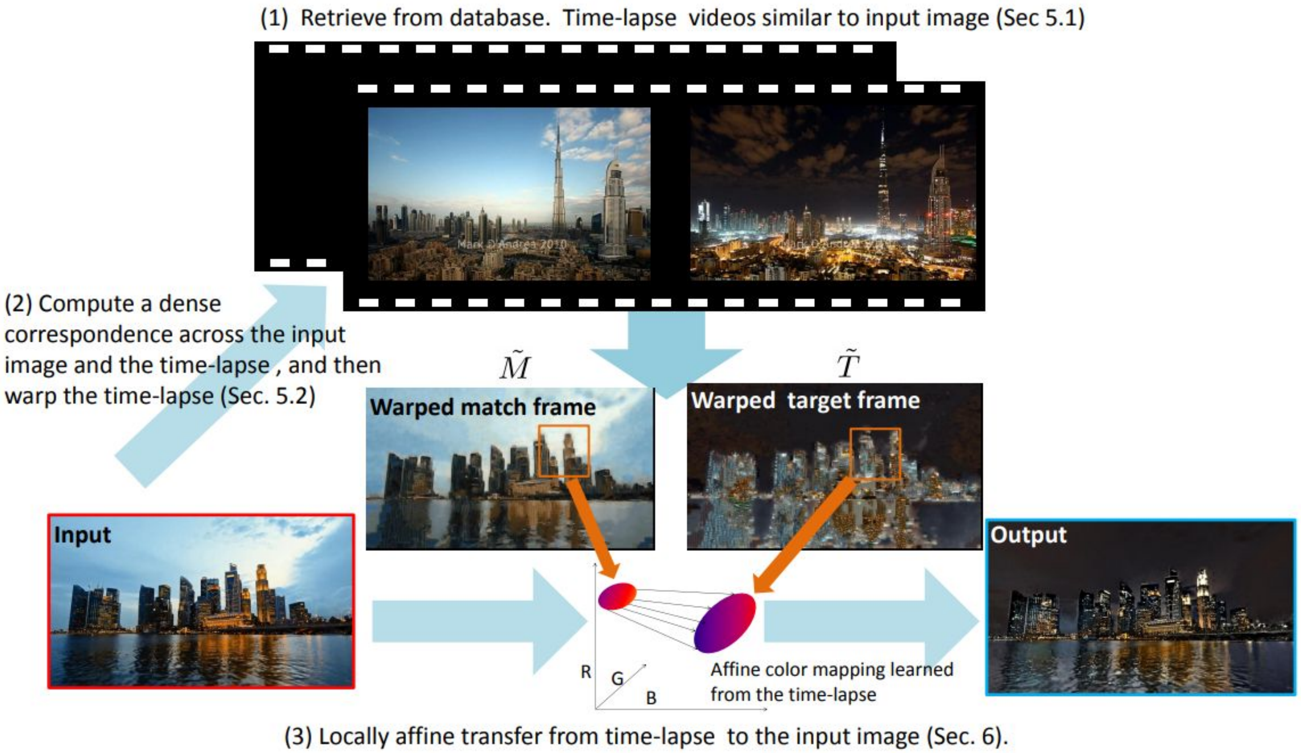
Golden Hour

Blue Hour

Milestones

1. Reading and understanding the paper
2. Global Matching
3. Frame Selection
4. Local Matching
5. Locally Affine color transfer
6. Denoising image

1. Reading and understanding the paper


(1) Retrieve from database. Time-lapse videos similar to input image (Sec 5.1)
Mark D'Andrea 2010
(2) Compute a dense correspondence across the input image and the time-lapse , and then warp the time-lapse (Sec. 5.2)
$\tilde{M}$
Warped match frame
$\tilde{T}$
Warped target frame
Input
R
G
B
Affine color mapping learned from the time-lapse
Output
(3) Locally affine transfer from time-lapse to the input image (Sec. 6).

2. Global Matching

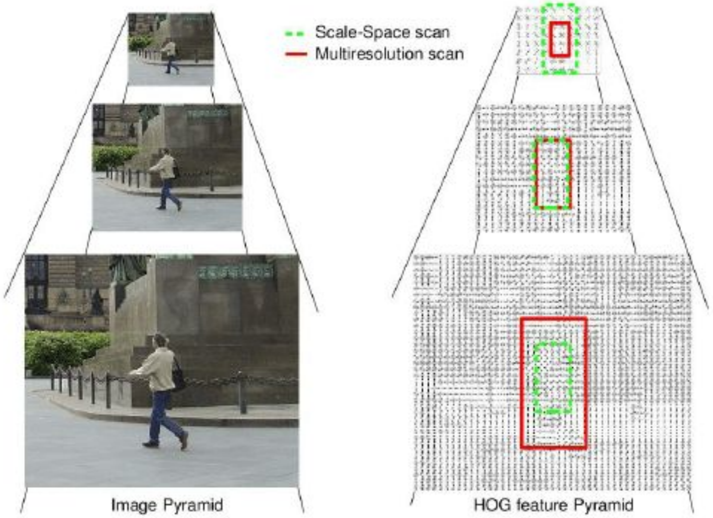
The first step of our algorithm is to identify the videos showing a scene similar to the given input image. We employ a standard scene matching technique in computer vision. We sample 5 regularly spaced frames from each video, and then compare the input to all these sampled frames.To assign a score to each time-lapse video, we use the highest similarity score in feature space of its sampled frames. We used Histograms of Oriented Gradients (HOG) [Dalal and Triggs 2005] for getting features.

(1) Retrieve from database. Time-lapse videos similar to input image

Pyramid-HOG Algorithm

For selecting best videos(global matching) and selecting the best frame , we need to compare images. To reduce computational cost we need a compact representation which is provided by Pyramid Histogram of Oriented Gradient (PHOG).

The objective of the PHOG is to take the spatial property of the local shape into account while representing an image by HOG. The spatial information is represented by tiling the image into regions at multiple resolutions based on spatial pyramid matching .



3. Frame-Selection

Now that we have a set of matching videos, for each of them, we seek to retrieve a frame that matches the time of day of the input image. We use the color histogram and L2 norm to pick the matched frame.

Let H() be the histogram function. I be the image and M be the matched frame and X be the set of frames taken from the matched video at equal intervals then,

$$H(i) = [I_o, I_1, I_2, \ldots, I_{255}]$$

$$cost(x) = \sum_i (I_i - X_i)^2$$

$$M = X \ni (cost(X) = \min_{\forall y \in X} cost(y))$$

4. Local matching

1. We seek to pair each pixel in the input image I with a pixel in the match frame M
2. Methods like SIFT are not used because they are designed to match with a single image and are not designed for videos.
3. For this, we formulate the problem as a Markov random field (MRF) using a data term and pairwise term.
4. MRF formulation allows us to exploit structure found in time-lapse video
5. Scene geometry remains constant over the frames as the camera does not move in time-lapse videos

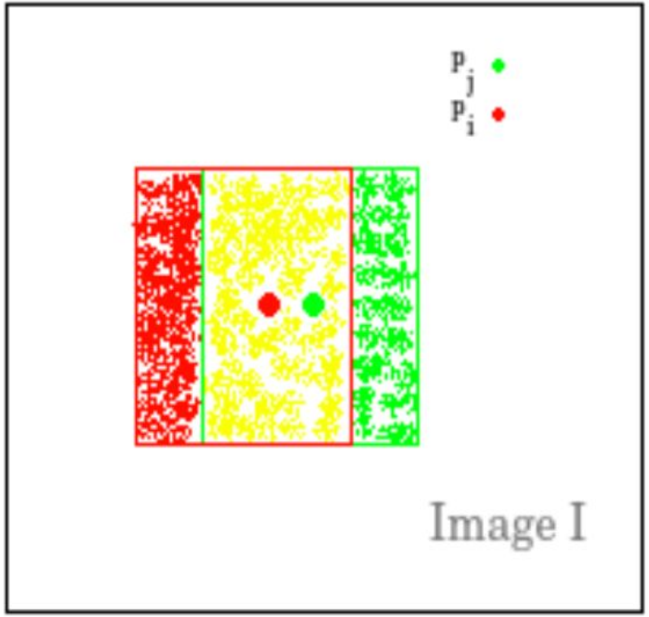
Energy Function (Data Term)

For each patch in I(Input images), we seek a patch in M(best matched frame) that looks similar to it. We use the L2 norm over square patches of side length 2r + 1. For pixel p ∈ I and the corresponding pixel q ∈ M , we calculate pixel wise difference in intensities for the patch centered around the pixel.

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

Energy function (Pairwise term)

1. A pairwise MRF term was introduced to gain additional knowledge from the video, that the matches should remain consistent throughout the video.
2. For two adjacent pixels pi and pj in I, we name Ω the set of the overlapping pixels between the two patches centered at pi and pj.
3. For two adjacent pixels pi and pj in I, we name Ω the set of the overlapping pixels between the two patches centered at pi and pj.
4. Yellow region represents Ω, the set of overlapping pixels



Energy function (Pairwise term) (Cont.)

1. For each pixel o∈Ω, we define the offsets δi=o−pi and δj=o−pj.
2. For the energy we use L2 norm within each frame
3. Then we take the L∞ across the frames of the video

$$E_2(q_i, q_j) = \max_t \sum_{o \in \Omega} \left\| V_t(q_i + \delta_i) - V_t(q_j + \delta_j) \right\|^2$$

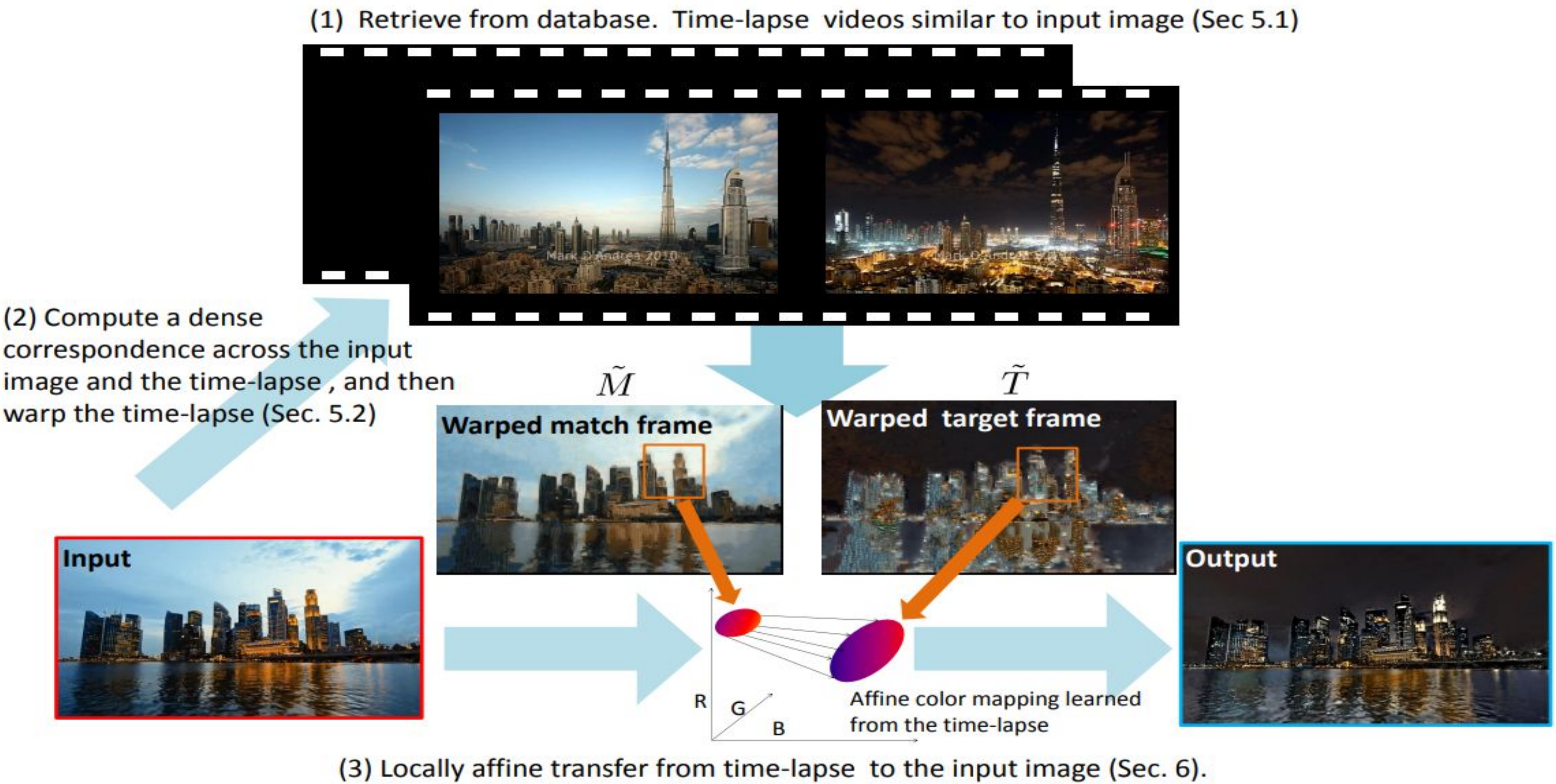
MRF Graph (cont.)

The MRF graph made is of dimensions of the input image, with the preliminary terms calculated using only input image and matched frame, whereas the edge weights are calculated using all the frames of the matched video.

5. Locally Affine Color Transfer

The transfer from the input image I, the warped match frame M˜ , the warped target frame T˜, and output the hallucinated image O involves computing affine model A_k for each of the k patches.

A naive solution would be to compute each affine model Ak as a regression between the k th patch of M˜ and its counterpart in T˜, and then independently apply Ak to the k th patch of I for each k. However, the boundary between any two patches of O would not be locally affine with respect to I, and would make O have a different structure from I, e.g., allows for spurious discontinuities to appear at patch boundaries.



(1) Retrieve from database. Time-lapse videos similar to input image (Sec 5.1)

(2) Compute a dense correspondence across the input image and the time-lapse , and then warp the time-lapse (Sec. 5.2)

Affine color mapping learned from the time-lapse

(3) Locally affine transfer from time-lapse to the input image (Sec. 6).

Locally Affine Color Transfer (cont.)

We want to explain the color variations observed in the timelapse video. We seek a series of affine models that locally describe the color variations between T˜ and M˜ . For this we minimize the following term .

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

Locally Affine Color Transfer (cont.)

We want a result that has the same structure as the input and that exhibits the same color change as seen in the time-lapse video. We seek an output O that is locally affine to I, and explained by the same affine models .

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

Locally Affine Color Transfer (cont.)

Finally, we add a regularization term on the A_k matrices to make sure they are not wildly different.. For this we regularize A_k using a global affine model G, the regression by the entire picture of $M^{\sim}$ and $T^{\sim}$ with the Frobenius norm.

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

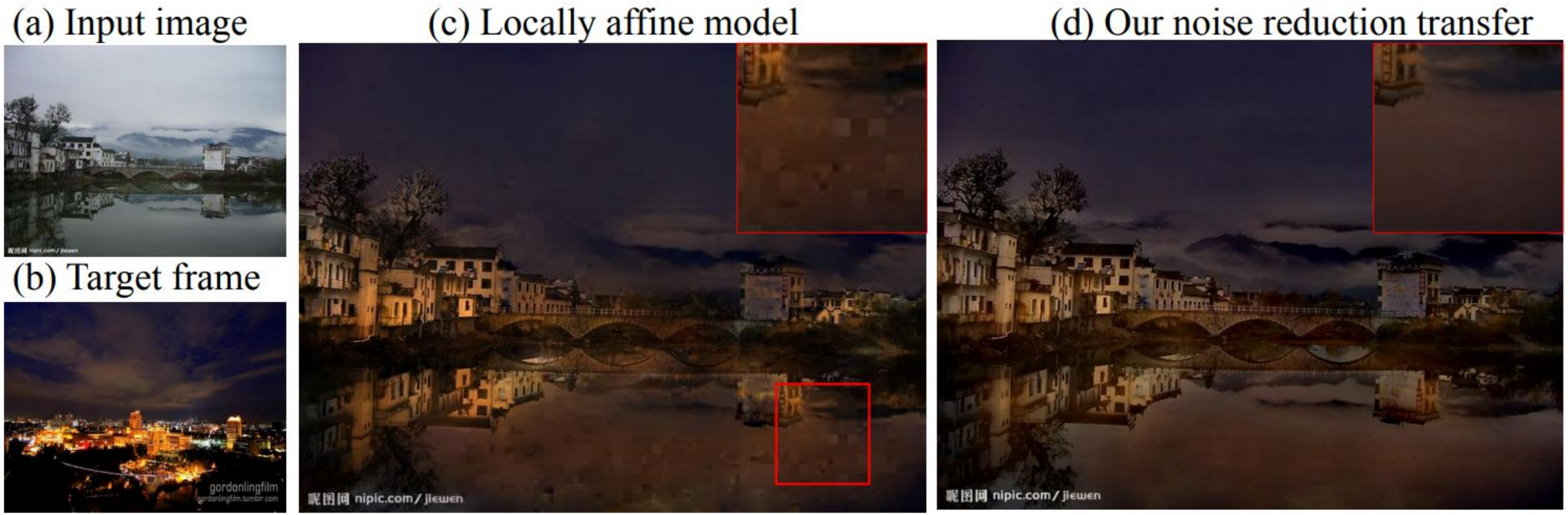
6. Denoising image

The affine mapping has a side effect that it may magnify the noise existing at the input image, such as sensor noise or quantization noise.

We first use bilateral filtering to decompose the input image into a detail layer and a base layers, the latter being mostly noise-free. We then apply our locally affine transfer to the base layer instead of the input image.

Finally, we obtain the final result by adding the detail layer back to the transferred base layer. Since the base layer is clean, the noise is not magnified. Compared to directly taking the input image, we significantly reduce the noise.

Denoising image (cont.)



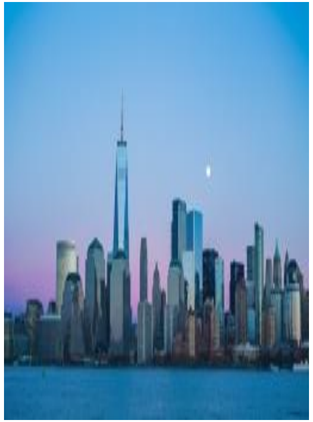
(a) Input image

(b) Target frame

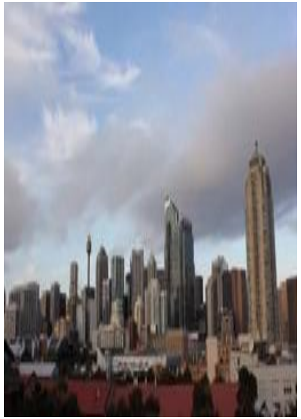
(c) Locally affine model

(d) Our noise reduction transfer

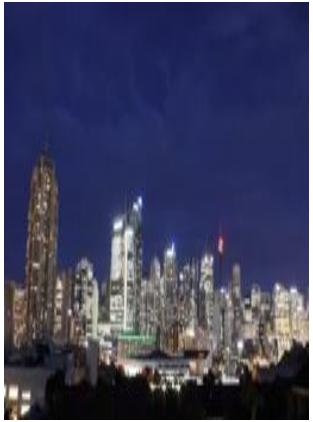
Results:



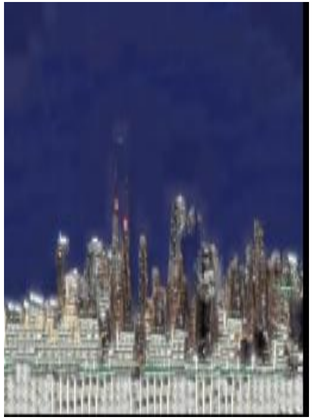
Input Image



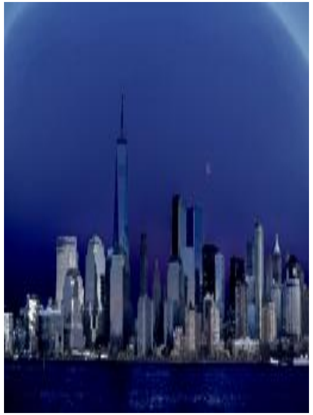
Best Frame



Reference Image

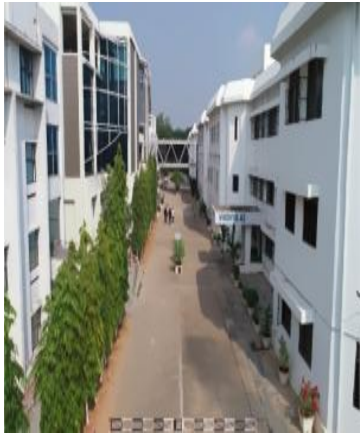


Wrap Image

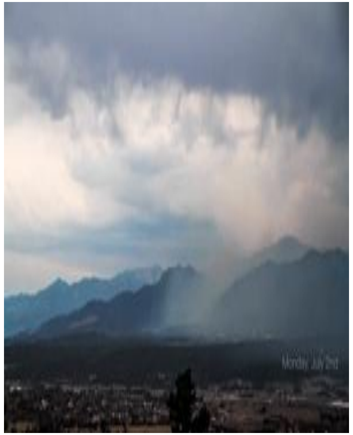


Output Image

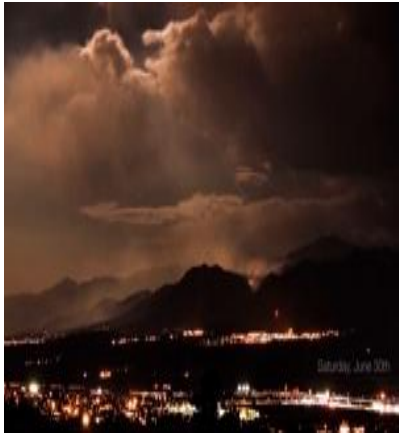
Results:



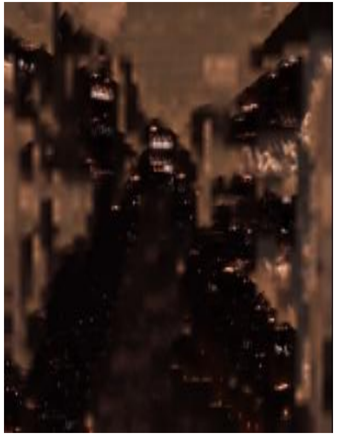
Input Image



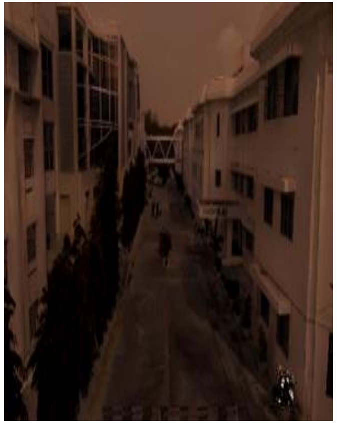
Best Frame



Reference Image

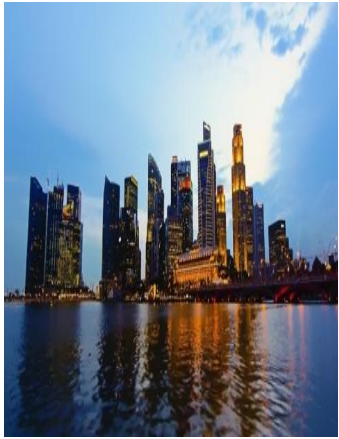


Wrap Image

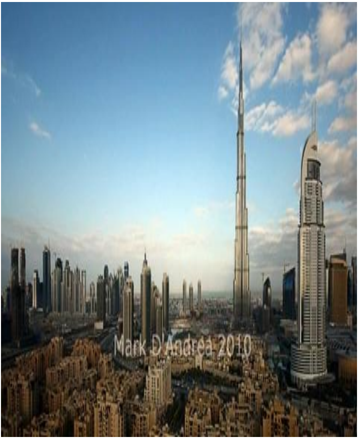


Output Image

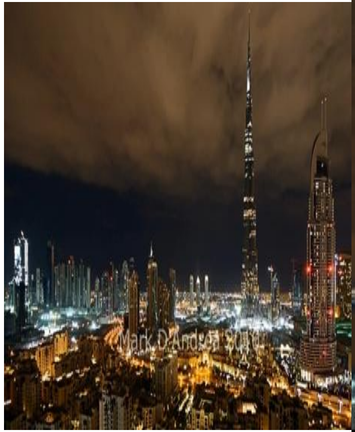
Results:



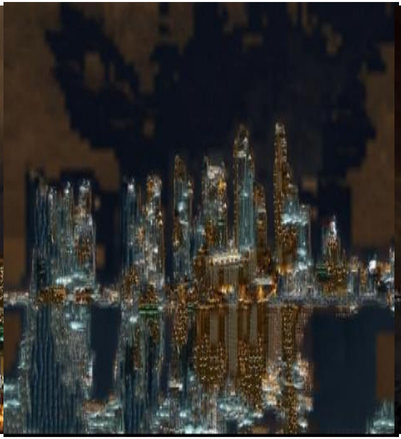
Input Image



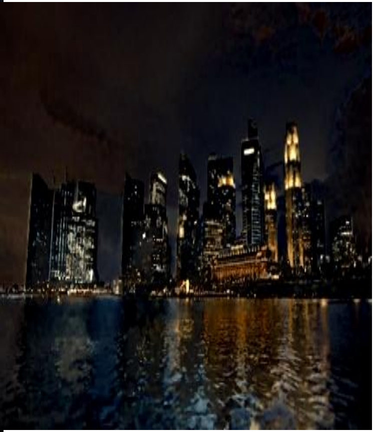
Best frame



Reference Image

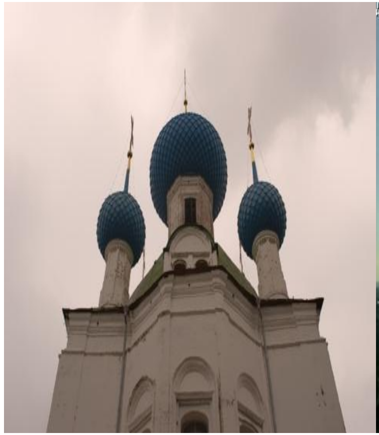


Warped Image

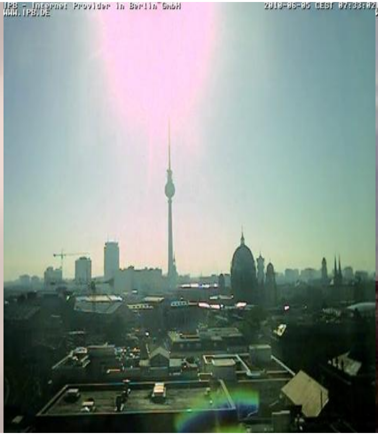


Output Image

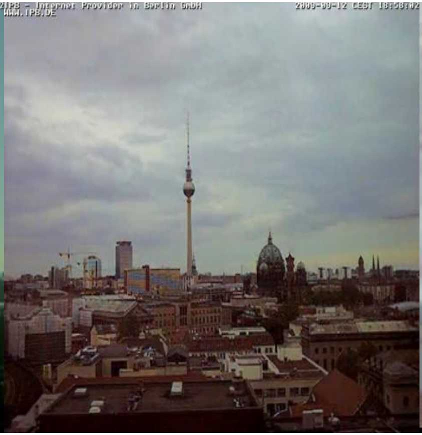
Results:



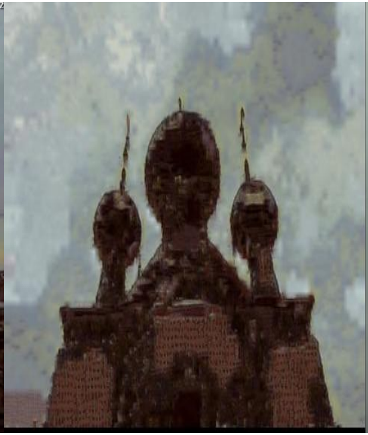
Input Image



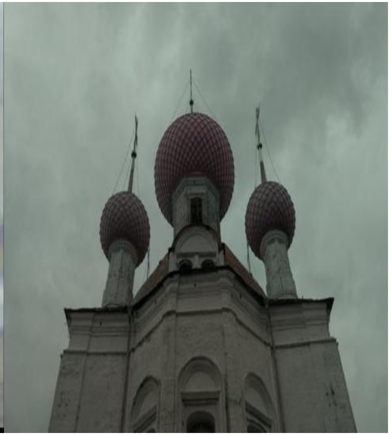
Best Frame



Reference Image

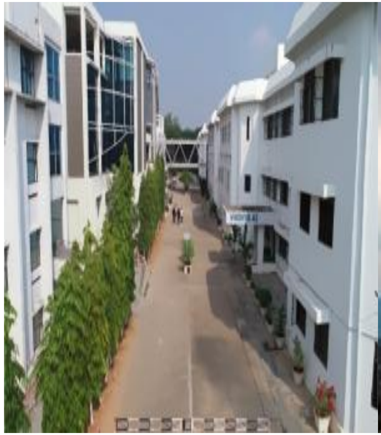


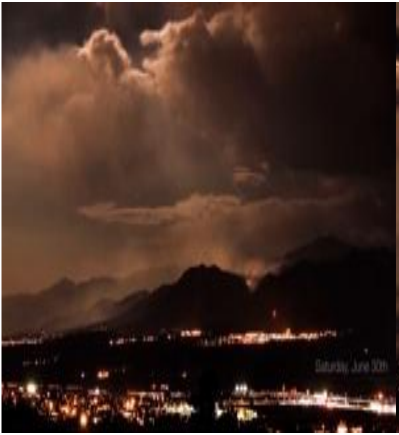
Wrap Image

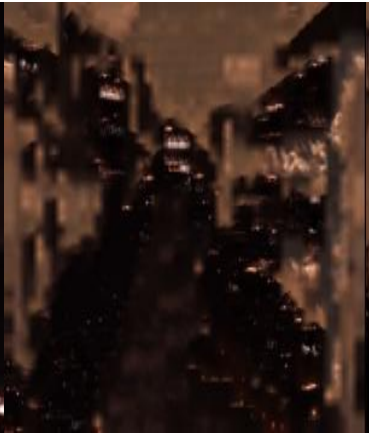


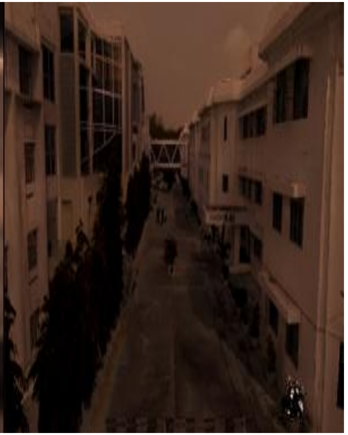
Output Image

Results:









Input Image

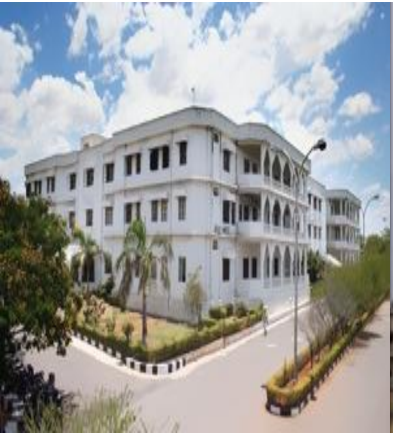
Best Frame

Reference Image

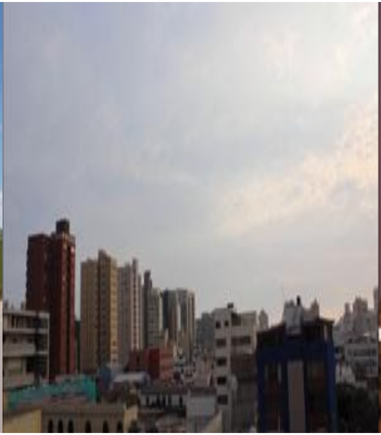
Wrap Image

Output Image

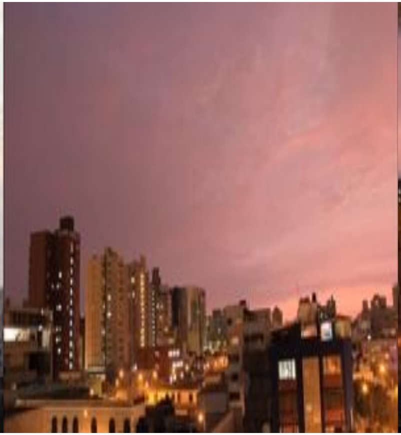
Results:



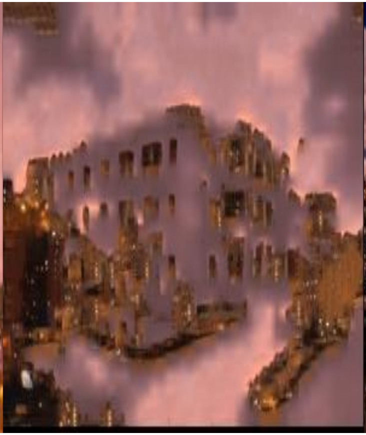
Input Image



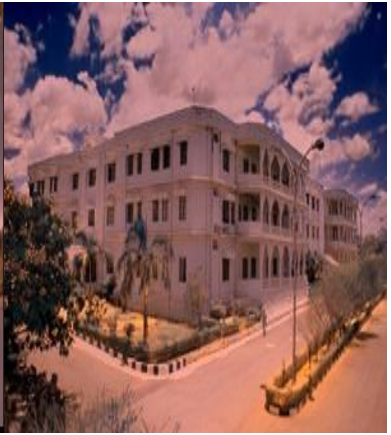
Best Frame



Reference Image

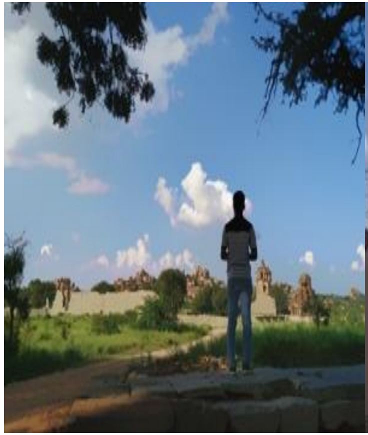


Wrap Image



Output Image

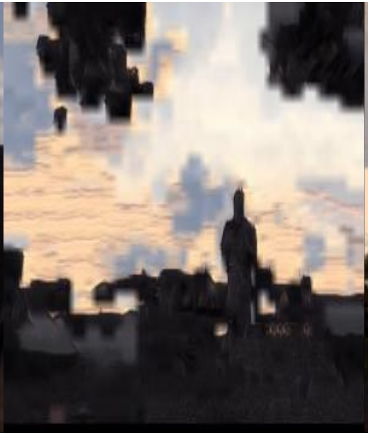
Results:



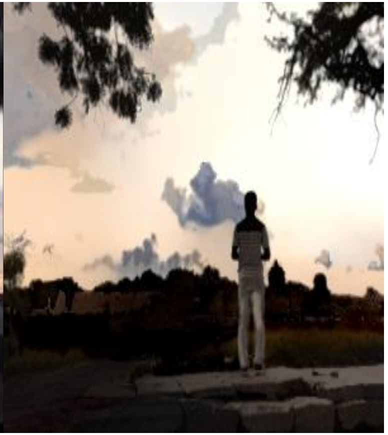
Input Image

Best Frame

Reference Image



Wrap Image



Output Image

Division of work

- We did pair programming (mostly)
- Dixit Kumar Garg: Global Matching, Local matching
- Akshat Goyal : Global matching, Local affine color transfer
- Manish Sharma : Frame Selection, Local affine color transfer
- Khadiravana : Local matching, Denoising

Thankyou!!