INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

CSE578: Computer Vision

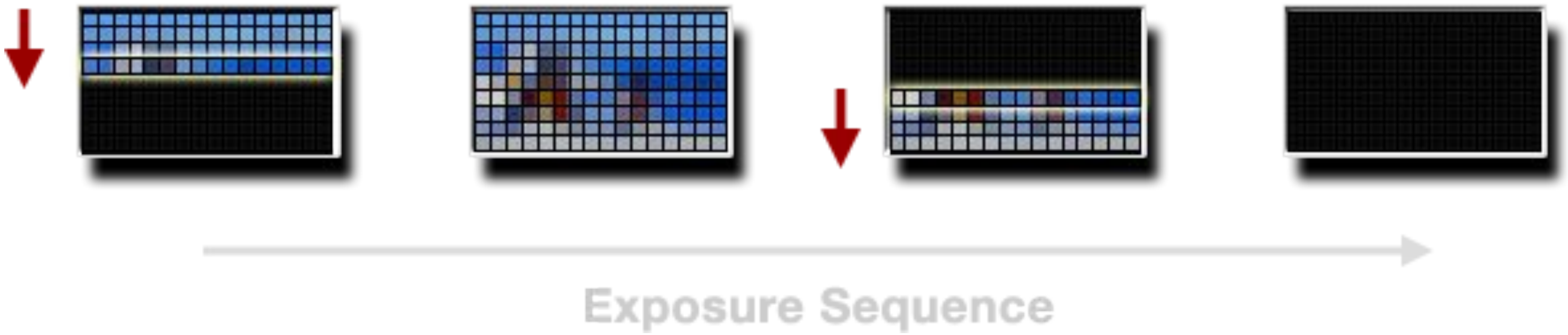
Unrolling the Shutter

CNN to correct motion distortions

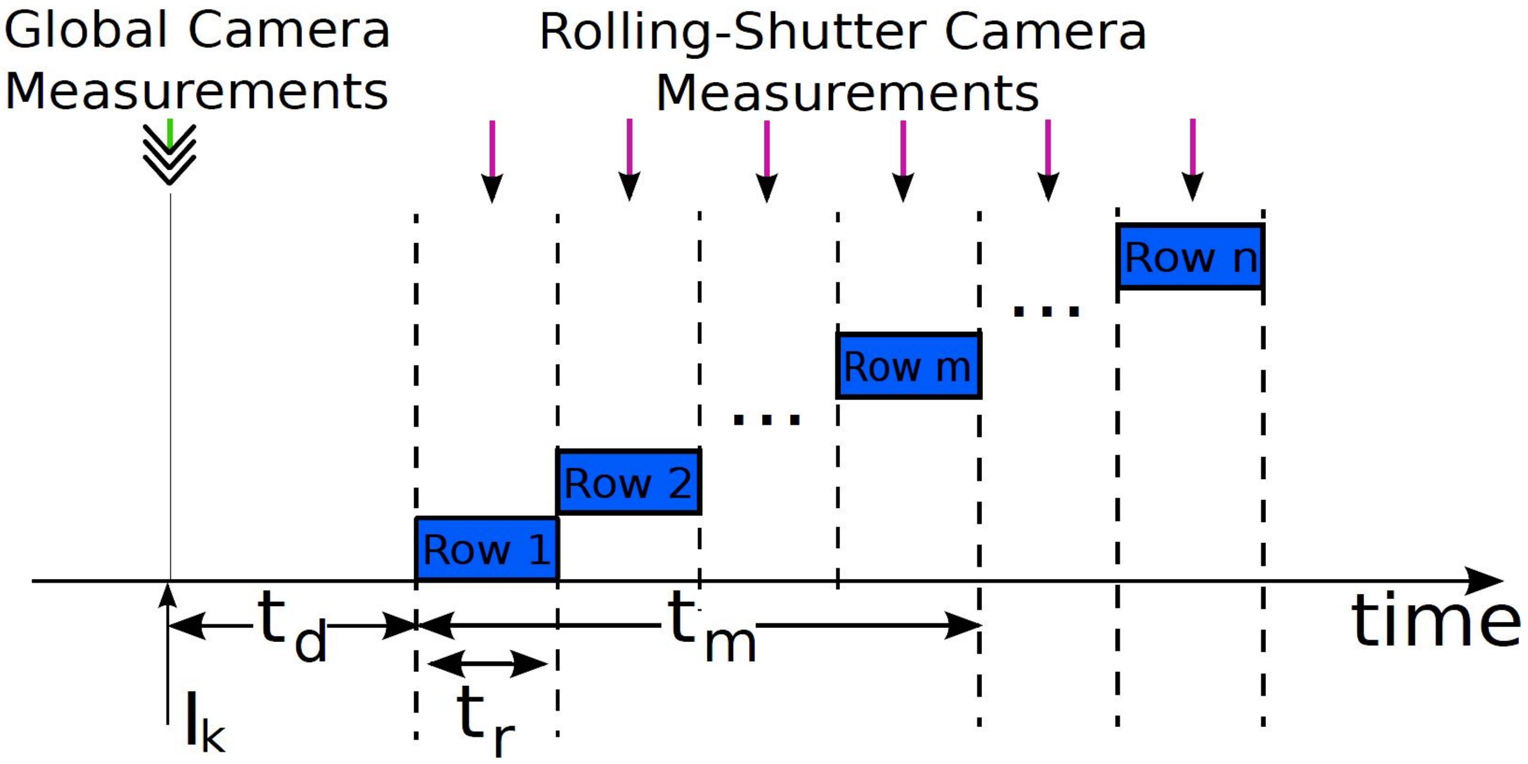
TA: Anjali Shenoy
Faculty Mentors:
Dr. Anoop Namboodri, Dr. Avinash Sharma

Anjan Kumar (201501238)
Jyotish (20161217)

What is Rolling Shutter?

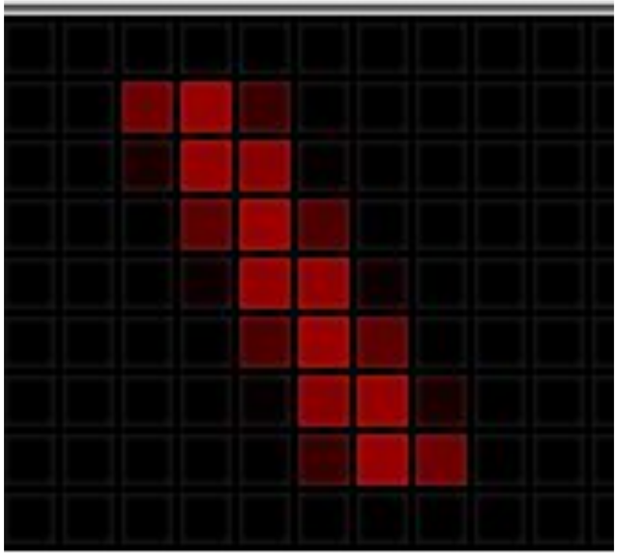


What is Rolling Shutter?



What is Rolling Shutter?

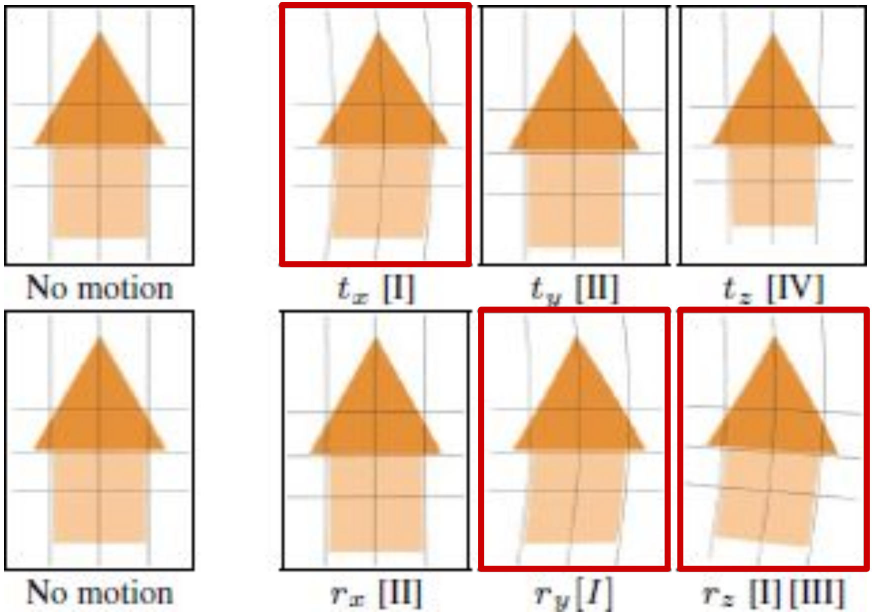
Horizontal Motion
(object moving fast relative to sensor speed)



Recorded Frame
(position shifts as shutter sweeps downward)

What is Rolling Shutter?

What is Rolling Shutter?



I. Vertical Curvature
II. Vertical Stretching or Shrinking
III. Horizontal Curvature
IV. Vertical Scale Change

Global shutter

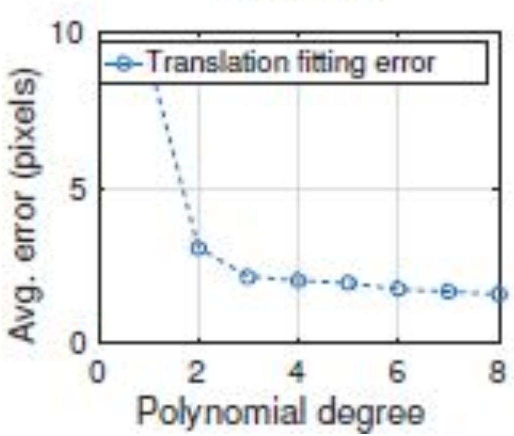
$$\mathtt{I}(\mathtt{p_i}) = \frac{1}{\mathtt{t_e}} \int_0^{\mathtt{t_e}} \mathtt{I}(\mathtt{w}(\mathtt{T}(\mathtt{t}\mathbf{f_v}), \mathtt{o_{i,t}}))\mathtt{dt}$$

Rolling shutter

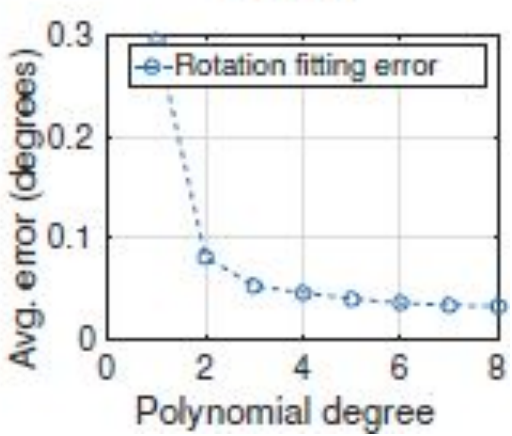
$$\mathtt{I_r}(\mathtt{p_i}) = \frac{1}{\mathtt{t_e}} \int_{\mathtt{r.t_d}}^{\mathtt{r.t_d}+\mathtt{t_e}} \mathtt{I_r}(\mathtt{w}(\mathtt{T}(-\mathtt{t}\mathbf{f_v}), \mathtt{o_{i,t}}))\mathtt{dt}$$

Symbol	**Meaning**
$\mathtt{t}$	Time
$\mathtt{I}(\mathtt{p},\mathtt{t}) : \Omega \times \mathbb{R}^+ \to \mathbb{R}^+$	Color Image Brightness Fn. wrt time
$\mathtt{I}(\mathtt{p}) : \Omega \times \mathbb{R}^+$	Color Image Brightness Fn.
$\mathtt{D}(\mathtt{p},\mathtt{t}) : \Omega \times \mathbb{R}^+ \to \mathbb{R}^+$	Depth Image Brightness Fn. wrt time
$\mathtt{D}(\mathtt{p}) : \Omega \times \mathbb{R}^+$	Depth Image Brightness Fn.
$\mathtt{K}$	Intrinsic Calibration Matrix
$\mathtt{p_i}$	p_i in homogeneous co-ordinates
$\mathtt{o_i} = \mathtt{K}^{-1}\bar{\mathtt{p}}_\mathtt{i}\mathtt{D}(\mathtt{p_i})$	3D pt. corresponding to p_i
$\mathbf{f_v} = (\mathtt{v}, \omega) \in \mathbb{R}^6$	v: Linear velocity, ω: Angular Velocity
$\mathtt{T}$	An $\mathbf{SE}(3)$ transformation
$\mathtt{w}(\mathtt{T},\mathtt{o_i})$	warping function, transforms o_i by $\mathtt{T} \in \mathbf{SE}(3)$ followed by projection
$\mathtt{t_e},\mathtt{t_d}$	Row exposure time, Line delay

Trajectory Fitting using polynomials



Fitting error for t_x



Fitting error for r_z

- ➔ Fit an n^{th} degree polynomial to t_x and r_z motions
- ➔ It is observed that the average fitting error with respect to human hand-held motion almost converges after n=3

Dataset Generation

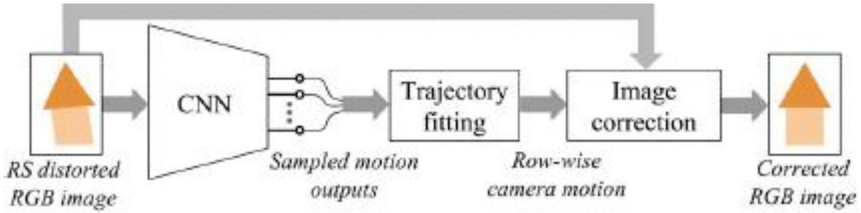
- Choose an n-th degree polynomial
- For every sample to generate
    - Choose the coefficients of the polynomial at random
    - Compute motion vectors for every row
    - Compute inverse homographies for each pixel in the sample to undistorted image
    - Perform bilinear interpolation for required pixels as required pixels may not exist in the undistorted image

Related Works

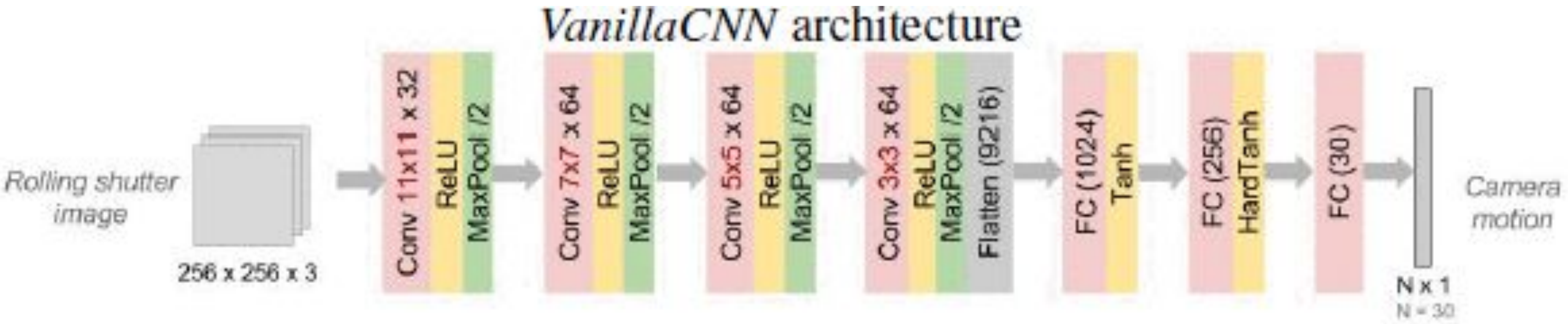
1. Most prior works in RS correction use multiple images to correct RS correction.
2. Very few works concentrate on correcting RS effect in a single image.
   a. "Correcting rolling-shutter distortion of CMOS sensors using facial feature detection"
   b. "From Bows to Arrows: Rolling Shutter Rectification of Urban Scenes"
   c. "Rolling Shutter Motion Deblurring"
3. In the absence of Motion blur, a and b correct faces and urban scenes using respective features. In this work the author tries to correct both using a common representation.
4. c corrects the scene using blind deblurring. It estimates the image trajectory and used Gauss Newton optimisation to solve for the trajectory. One drawback of it being the it can't handle in plane rotations which are very common.

Why a Neural Network?

- ➔ Traditional methods need different features for each type of image
  - ◆ Facial features to un-distort images focusing human faces
  - ◆ Curves to un-distort urban images
- ➔ These methods are tailored for specific image classes and are thus heavily dependent on the extraction of their respective scene-specific features
- ➔ Spatial convolution filters to learn scene-specific features
- ➔ Row and column wise convolution filters to learn camera motion along t_x and r_z

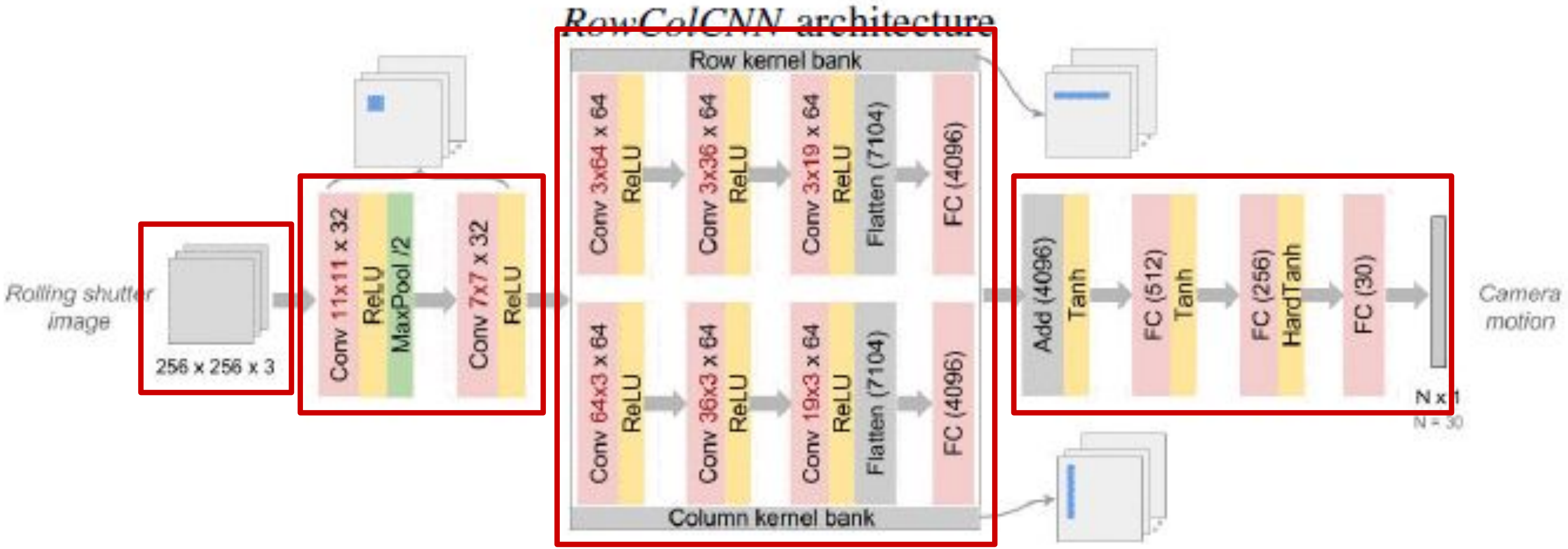


Typical CNN

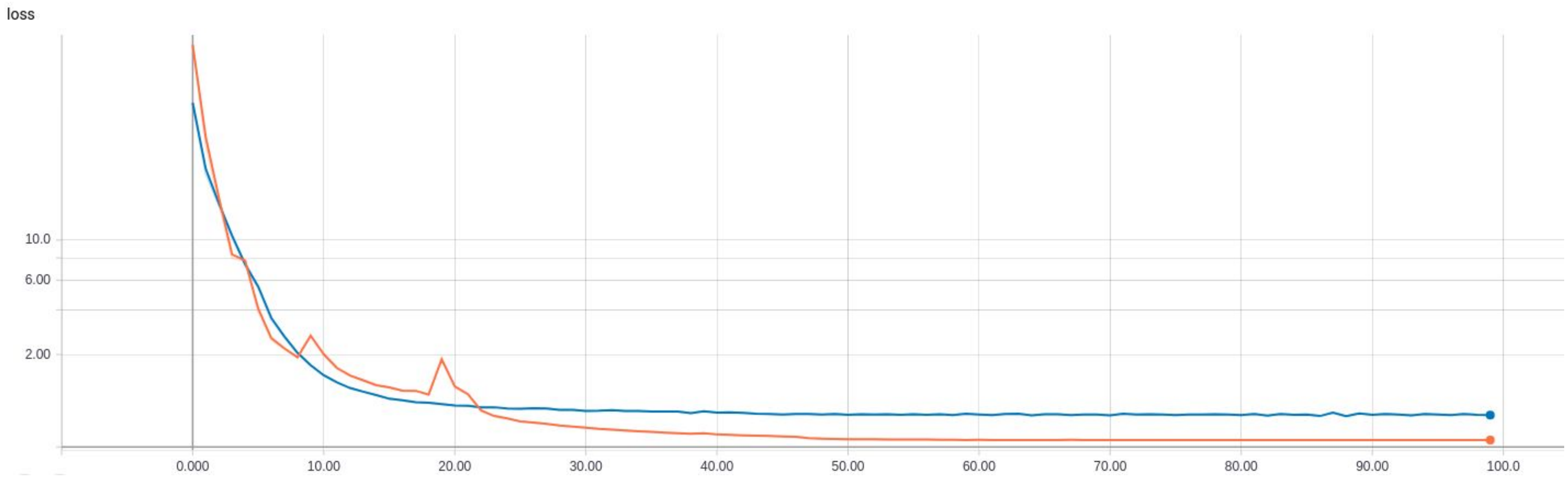


VanillaCNN architecture
Rolling shutter image
256 x 256 x 3
Conv 11x11 x 32
ReLU
MaxPool /2
Conv 7x7 x 64
ReLU
MaxPool /2
Conv 5x5 x 64
ReLU
MaxPool /2
Conv 3x3 x 64
ReLU
MaxPool /2
Flatten (9216)
FC (1024)
Tanh
FC (256)
HardTanh
FC (30)
N x 1
N = 30
Camera motion

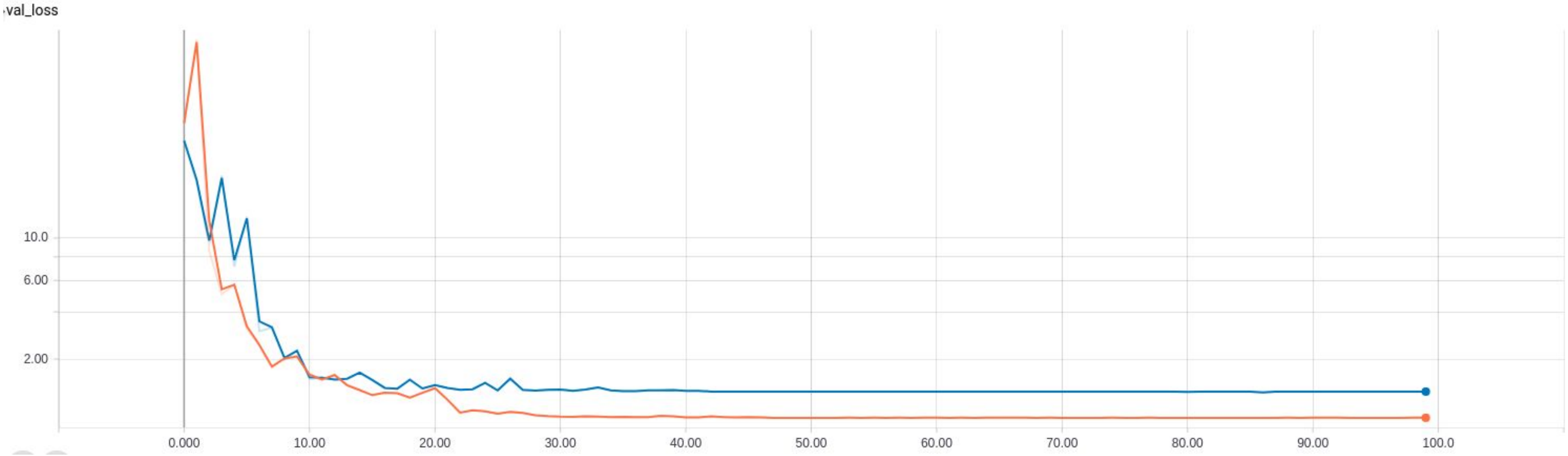
Proposed Network



Training - Loss on training data

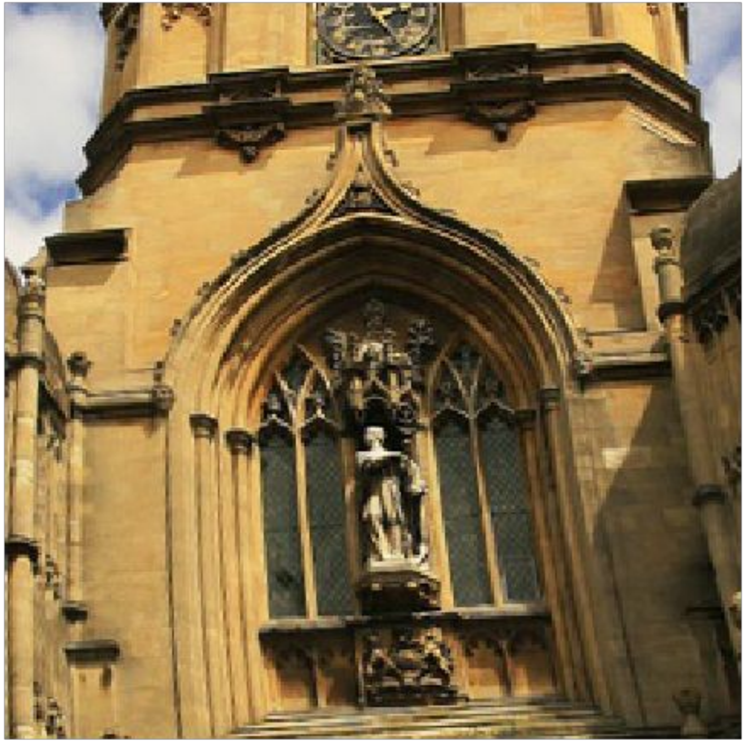


Training - Loss on validation data

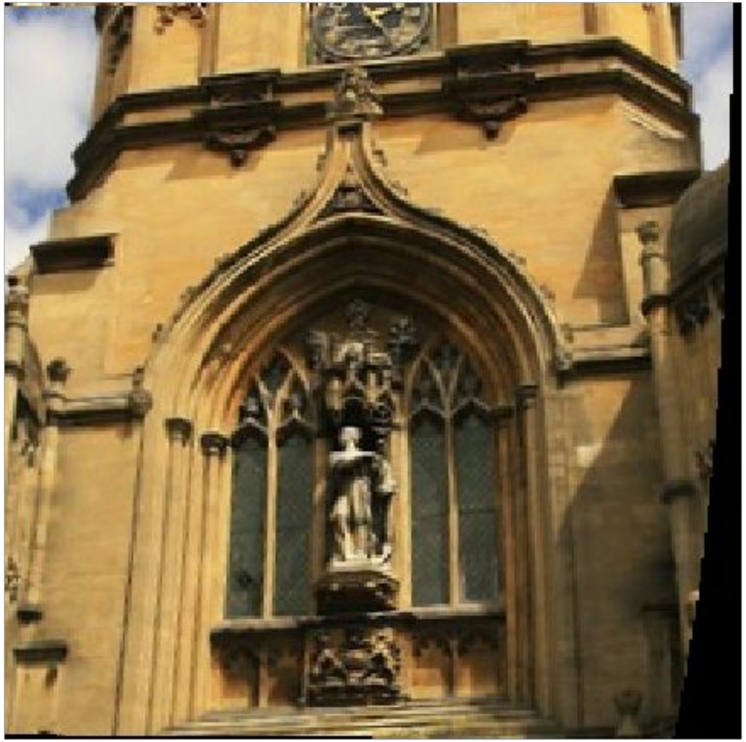


Results - Row Column CNN

Distorted Image

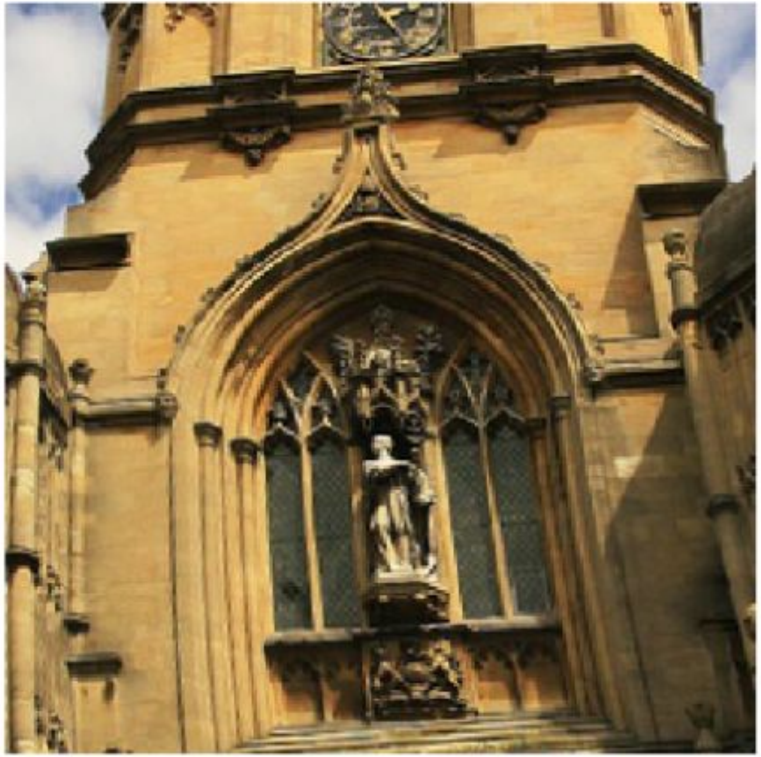


Undistorted Image

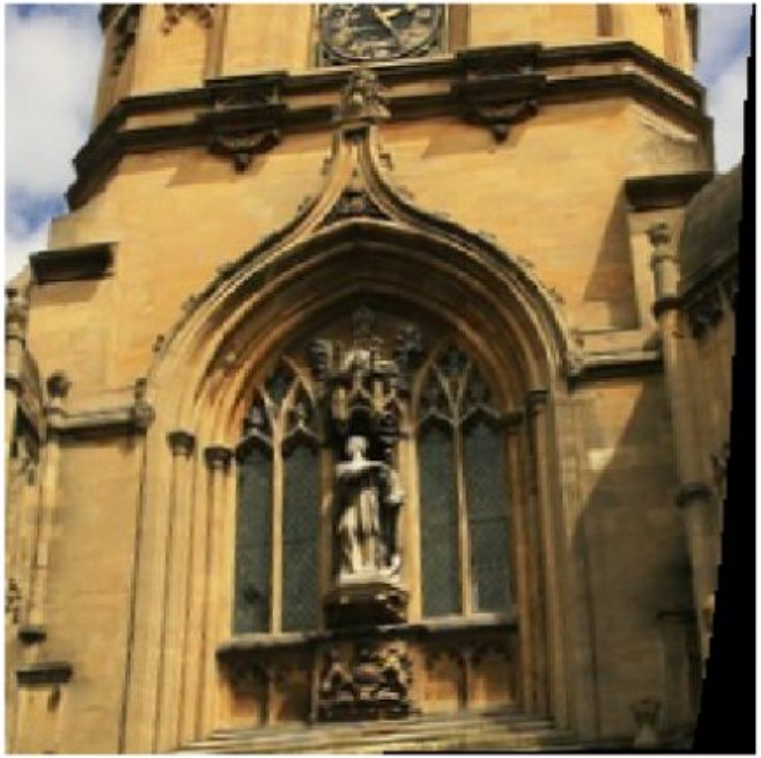


Results - Vanilla CNN

Distorted Image



Undistorted Image



Results - Row Column CNN

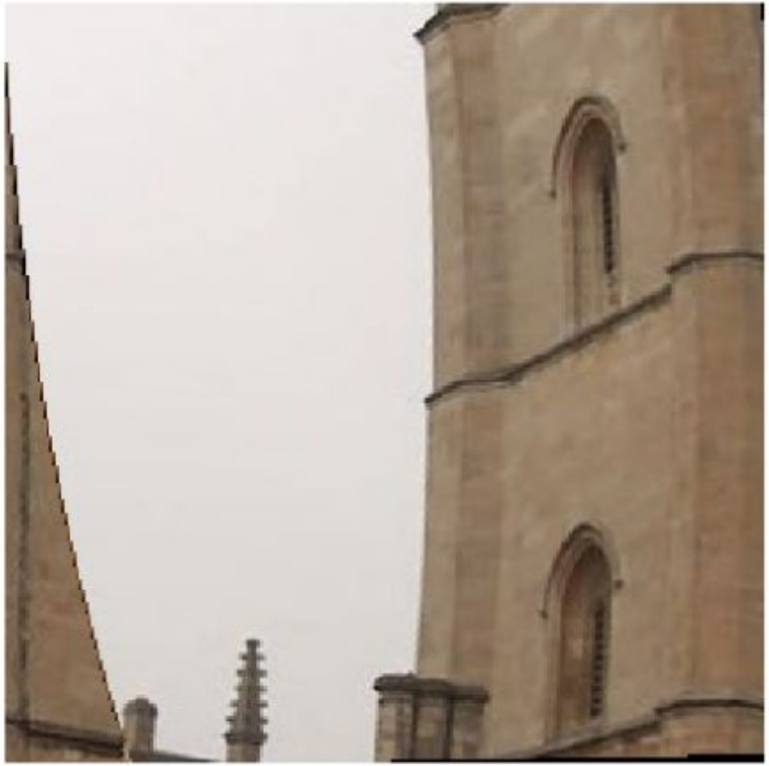
Distorted Image

Undistorted Image

Results - Vanilla CNN

Distorted Image

Undistorted Image



Results - Row Column CNN

Distorted Image

Undistorted Image

Results - Vanilla CNN

Distorted Image

Undistorted Image

Metrics

- ➔ Root Mean Square Error (RSME) → E_1
  - Error between the CNN output and ground truth t_x and r_z motion vectors
- ➔ Peak Signal to Noise Ratio (PSNR) → E_2
  - Error between the undistorted image and ground truth image
- ➔ Degree of rotation → E_3
  -

High E_1 and Low E_2, E_3 is desired

Extension - Appearance Flow Fields

1. This approach is based on the fact that pixels in GS image and RS image are highly correlated.
2. We predict dense appearance Flow fields for an RS image that specifies how to reconstruct a GS image from the said RS image.
3. Specifically for each pixel i in the GS image, the appearance flow vector f(i) specifies the coordinate at the RS image where pixel value is sampled to reconstruct to reconstruct pixel i.

Extension - Appearance Flow Fields

Possible benefits:

1. It alleviates the perceptual blurriness in images generated by CNN trained with L2 loss. By constraining the CNN to only utilize pixels available in the input image, we are able to avoid the undesirable local minimum obtained by predicting the mean colors around texture/edge boundaries that lead to blurriness in the resulting image.
2. The color identity of the instance is preserved by construction since the synthesized view is reconstructed using only pixels from the same instance.
3. The appearance flow field enables intuitive interpretation of the network output since we can visualize exactly how the target view is constructed with the input pixels

Extension - Appearance Flow Fields : Architecture

Three major components :

1. Input RS image encoder – extracts relevant features about the scenes like urban scenes and facial features.
2. Trajectory transformation encoder : maps the specified handshake trajectory to a higher-dimensional hidden representation.
3. Synthesis decoder : Outputs appearance flow field.

Thank You!