# Applying Image Inpainting to Video Post Production

Kino Roy, Joshil Patel, Hyukho Kwon, Bryce Haley, Di Wang

SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

## Problem & Objective

### Problem Description:

Individual shots in a movie require a great deal of visual effects (VFX) work before the final film is released on the big screen. One tedious and time consuming task involves artists removing or replacing objects in a scene, which is known in the industry as painting. The majority of shots in a film require paint work, resulting in a bottleneck in the VFX pipeline and therefore increasing the risk of late delivery.

### Painting Process:

- create a mask of the object
- stencil out the image using that mask
- paint in the hole using various VFX techniques.

### Objective:

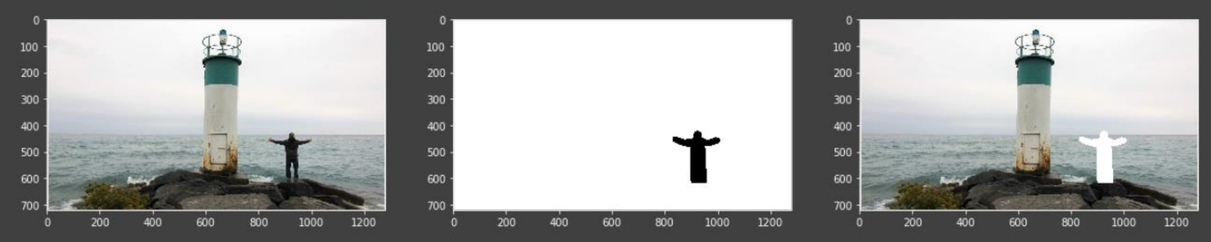
To automate the painting process using machine learning, thus removing or reducing the bottleneck in the VFX pipeline.

## Approach

### Convolutional Layer Input:

Let $\boldsymbol{W}^T$ be the weights for the convolution filter and $b$ be the corresponding bias. $\boldsymbol{X}$ are the pixels values for the current convolution window and $\boldsymbol{M}$ is the corresponding binary mask.

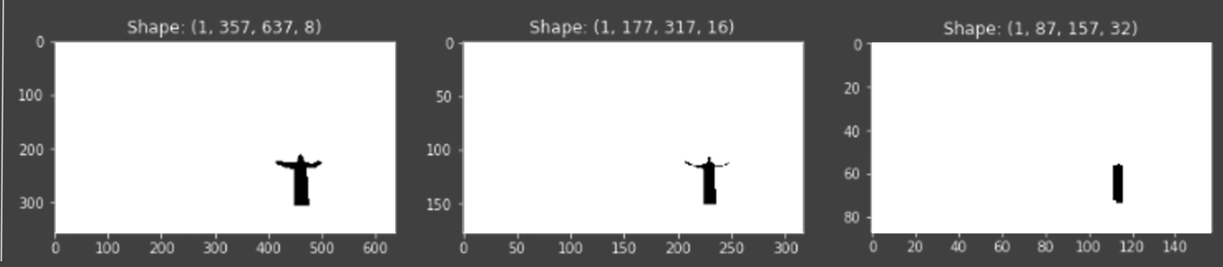
$$x' = \begin{cases} \boldsymbol{W}^T(\boldsymbol{X} \odot \boldsymbol{M}) \frac{\mathbf{1}}{\mathrm{sum}(\boldsymbol{M})} + \boldsymbol{b}, & \text{if } \mathrm{sum}(\boldsymbol{M}) > 0 \\ 0, & \text{otherwise} \end{cases}$$



### Mask Updating:

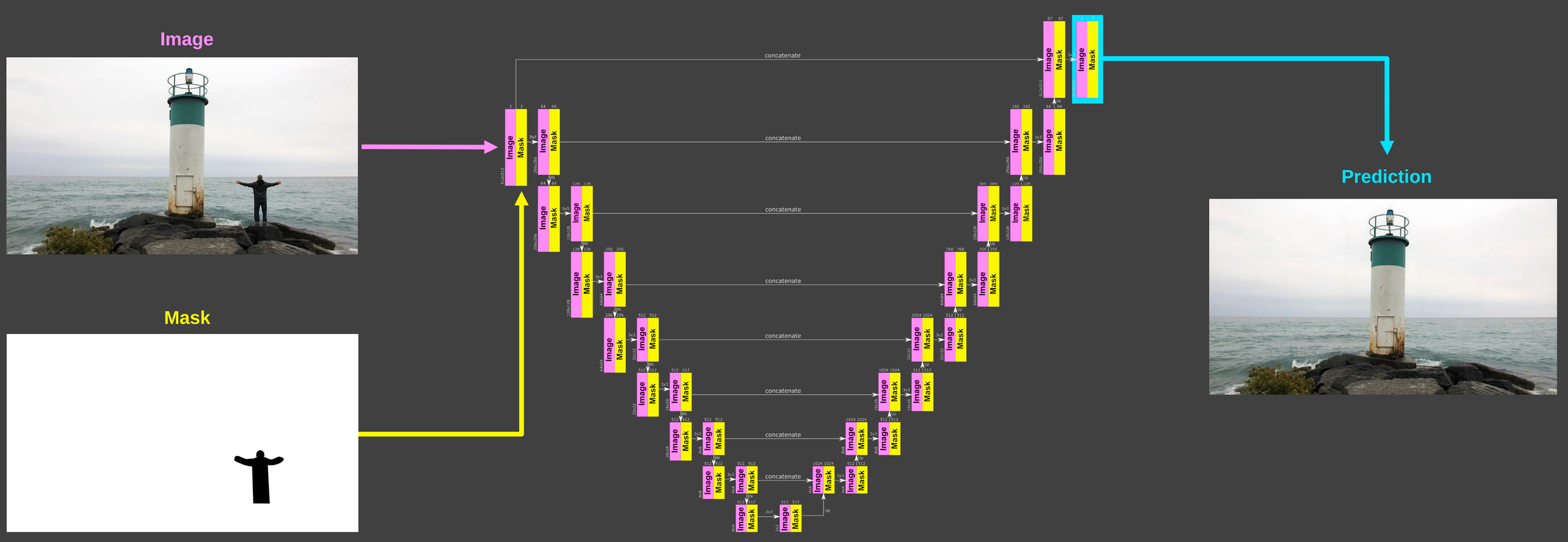
The mask is updated after each partial convolution. If the convolution was able to condition its output on at least one valid input value, then remove the mask for that location.

$$m' = \begin{cases} 1, & \text{if } \mathrm{sum}(\boldsymbol{M}) > 0 \\ 0, & \text{otherwise} \end{cases}$$




**Loss Function:** $\mathcal{L}_{total} = \mathcal{L}_{valid} + 6\mathcal{L}_{hole} + 0.05\mathcal{L}_{perceptual} + 120\left(\mathcal{L}_{style_{out}} + \mathcal{L}_{style_{comp}}\right) + 0.1\mathcal{L}_{tv}$

## U-Net Architecture




[1] Mathias Gruber, PConv-Keras, (2018), GitHub repository, https://github.com/MathiasGruber/PConv-Keras

## Suggested Improvements

### More Data:

Our network was trained on ImageNet data, whereas the original authors trained on multiple datasets for best results. Therefore more data and training would help. In terms of VFX production, adding frames manually painted by artists to the end of the training set could provide added assistance in specializing the net for use on the shot.

### Modifying the Network:

This CNN only looks at each image in isolation. Combining this network with an LSTM may remove the “flickering” present when the images are shown in sequence. Furthermore, this network only accepts low resolution images. Modifying the network to accept higher resolutions may result in better predictions, though will also require longer training time.