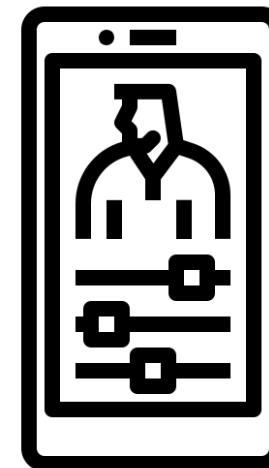


# Neural Style Transfer of Paintings to Portraits and Selfies

*CS 464 – Final Presentation  
Group 7*

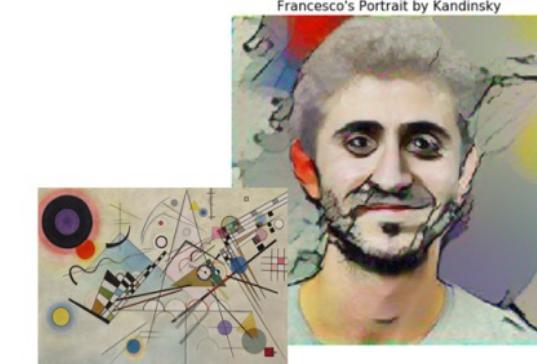
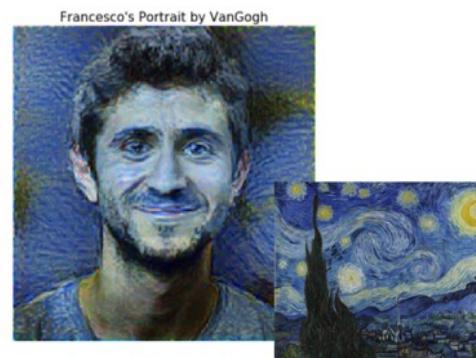
## MOTIVATION

- Selfie editing applications are quite popular.
- Latest applications utilize AI-based technologies.



# PARTICULAR FOCUS

- NST of classical paintings to portraits and selfies



# NEURAL STYLE TRANSFER

- Blending the style and content of two images



Figure 1: Content Image  
(Hayley Williams)



Figure 2: Style Image  
(Angle of Love)

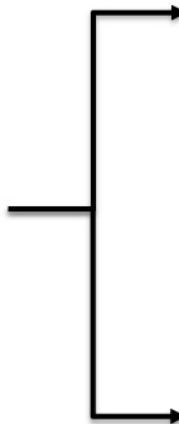


Figure 3: Target Image  
with 1500 iterations



Figure 4: Target Image  
with 2200 iterations

## BACKGROUND RESEARCH

- The optimizer optimizes the weighted sum of style and content losses.

$$L_{total} (i,j,k) = \alpha L_{content} (i,k) + \beta L_{style} (j,k)$$

The diagram illustrates the components of the total loss function. It starts with the equation  $L_{total} (i,j,k) = \alpha L_{content} (i,k) + \beta L_{style} (j,k)$ . Two arrows point from the terms  $\alpha L_{content} (i,k)$  and  $\beta L_{style} (j,k)$  to the labels "Content Weight" and "Style Weight" respectively. From each of these two labels, a blue arrow points to the word "Hyperparameters".

## BACKGROUND RESEARCH

- The content loss function is defined as the squared difference of the content and target images.

$$L_{content} = \sum_l \sum_{i,j} (\alpha C_{i,j}^l - \alpha P_{i,j}^l)^2$$

Content Loss

Content Weight

CONTENT IMAGE      TARGET IMAGE

The diagram illustrates the components of the content loss function. The formula is  $L_{content} = \sum_l \sum_{i,j} (\alpha C_{i,j}^l - \alpha P_{i,j}^l)^2$ . A blue arrow labeled "Content Loss" points to the summation symbol. Two orange arrows labeled "Content Weight" point to the terms  $\alpha C_{i,j}^l$  and  $\alpha P_{i,j}^l$ . Below the summation symbols, blue arrows labeled "CONTENT IMAGE" and "TARGET IMAGE" point to the subscripts  $i$  and  $j$  respectively.

## BACKGROUND RESEARCH

- The style loss function is the squared difference between the gram matrices of the style and target images.
- Gram matrices are computed multiplying style and target tensors by their transposes.

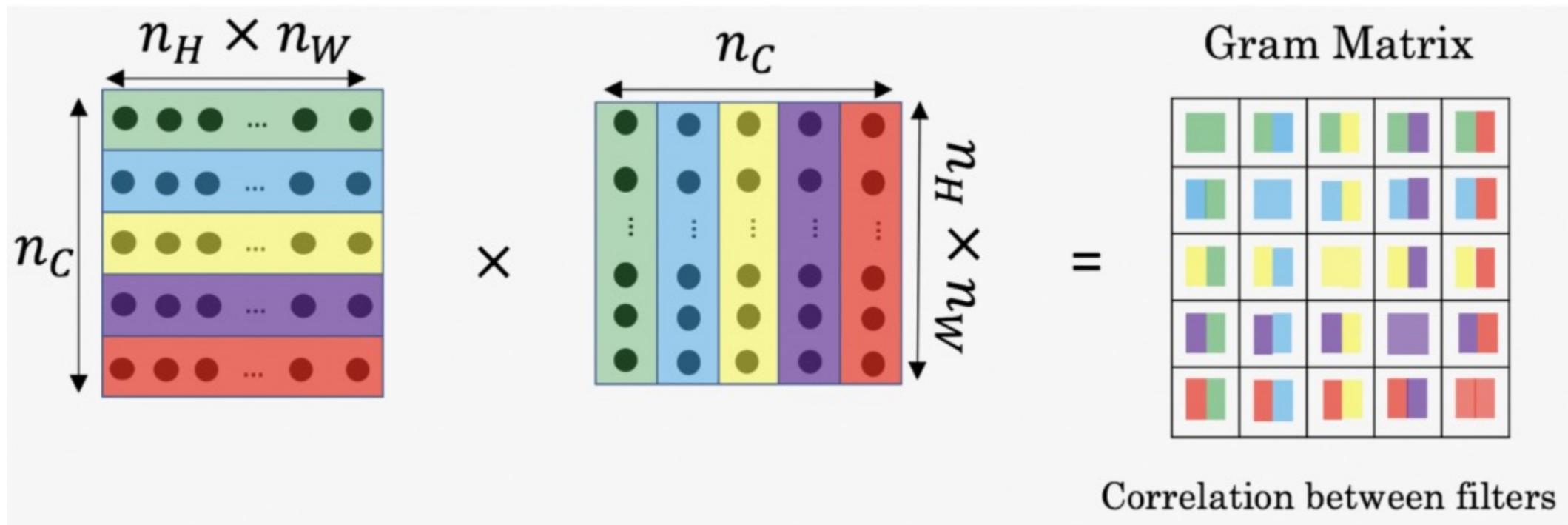
# BACKGROUND RESEARCH

$$G_{i,k}^l = \sum_k F_{i,k}^l F_{j,k}^l$$

Number of layers  
Original Matrix  
Gram Matrix  
Transpose Matrix  
 $i, j = \text{feature maps}$   
 $k = \text{channels}$

*The computation of the Gram matrices*

## BACKGROUND RESEARCH



# BACKGROUND RESEARCH

$$L_{style} = \sum_l \sum_{i,j} (\beta G_{i,j}^{s,l} - \beta G_{i,j}^{p,l})^2$$

Style Weight

Style Loss

Gram Matrix for Style Image

Gram Matrix for Target Image

*Style loss function defined with the gram matrices: **(Style – Target)**<sup>2</sup>*

## WHAT IS DONE

- VGG-19 pre-trained model was used.
- Gram matrices of the style and target were computed.
- Parameters of the target image were optimized with Adam.

## WHAT IS RECENTLY DONE

- Several bugs in image and video rendering is fixed.
- The quality of the target image is enhanced by experimenting with the hyper-parameters.
- Users can now specify an artist instead of providing the style image.
- Style transitions can now be saved as a video.

## EXPERIMENTS

- Different weights are given to the content and style losses.
- The learning rate and number of layers are changed.
- Various pre-trained models are considered.

# THE END PRODUCT

Content Image



Style Image



Target Image

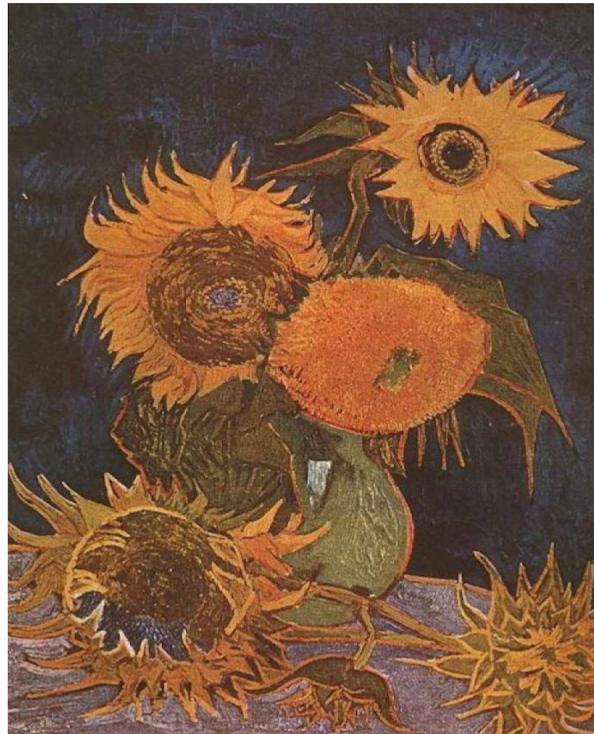


# THE END PRODUCT

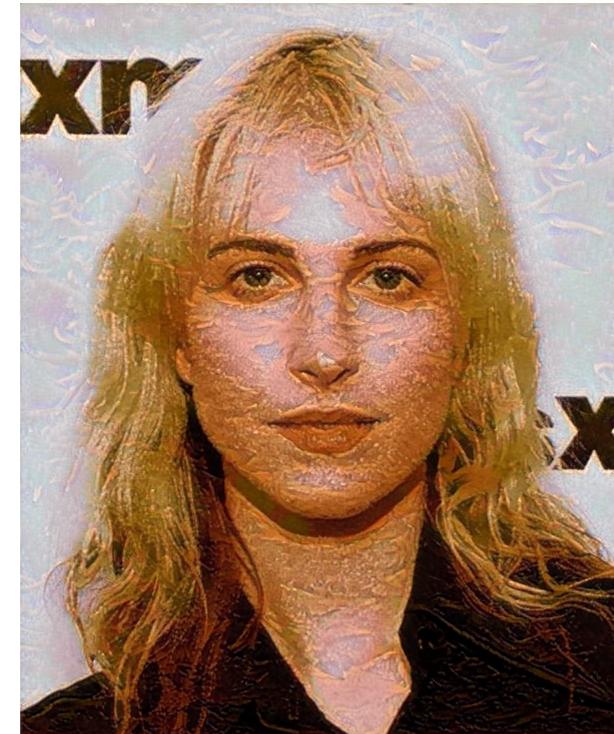
Content Image



Style Image



Target Image



# THE END PRODUCT

Content Image



Style Image



Target Image



# THE END PRODUCT

Content Image



Style Image



Target Image



# ALPHA RATIO EXPERIMENT

Content Image



Style Image



Target with sw: 1e6



Target with sw: 1e4



Target with sw: 1e2



Target with sw: 1e1



# LEARNING RATE EXPERIMENT

Will be stated in the final report.

## CONCLUSION

- The algorithm we used is essentially an optimization algorithm.
- There is no training process; model parameters change for each content & style pair.
- Consequently, what we do is an image processing task.

**DEMO AND VIDEO**

THANK YOU