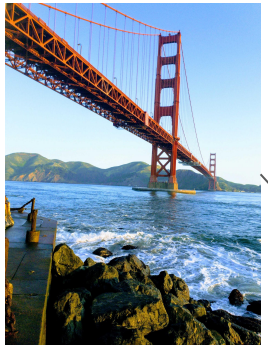
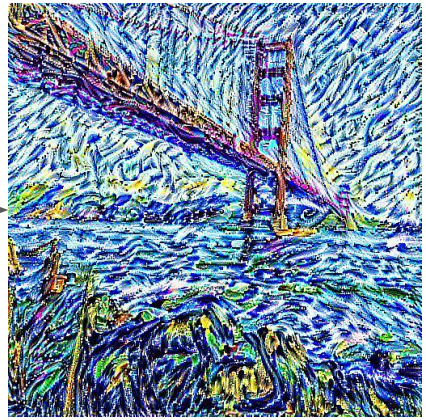
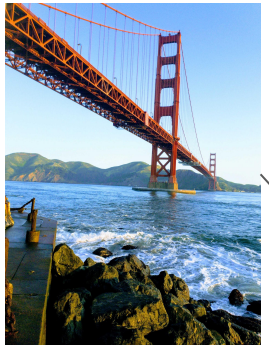


Model



In style transfer you update the input to minimize a cost function not a model!

Apply optimization on the input!

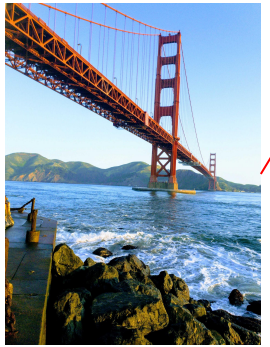


Constant

Constant

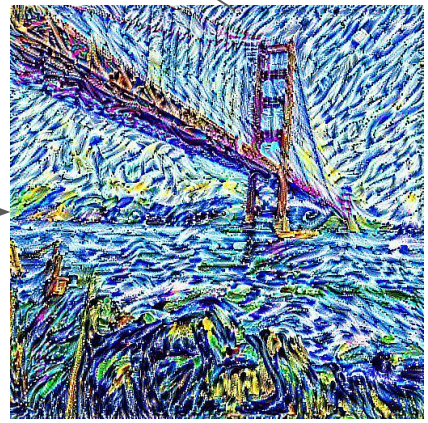
Constant

Model



Input

Variable



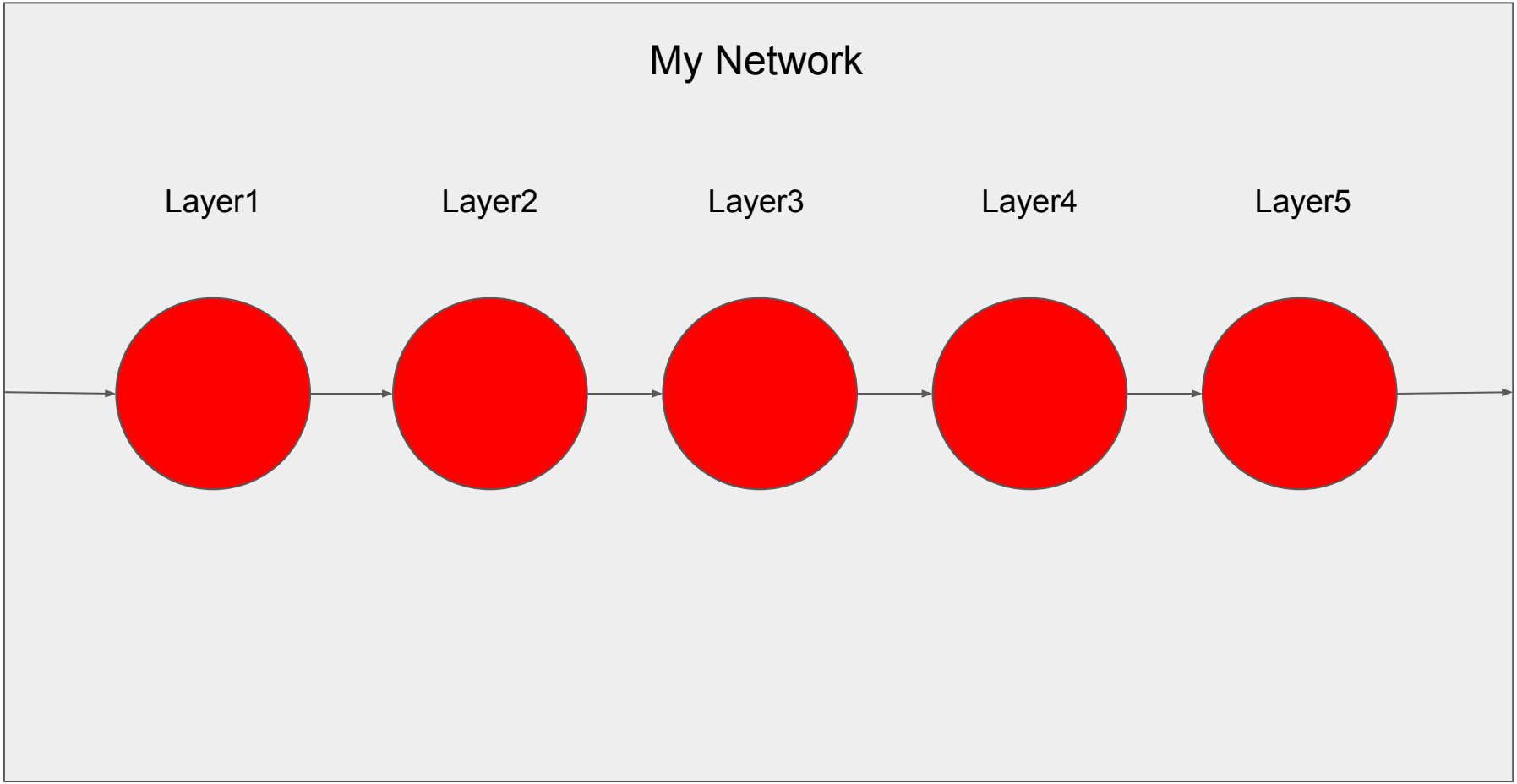
Machine Learning is ***all*** optimization

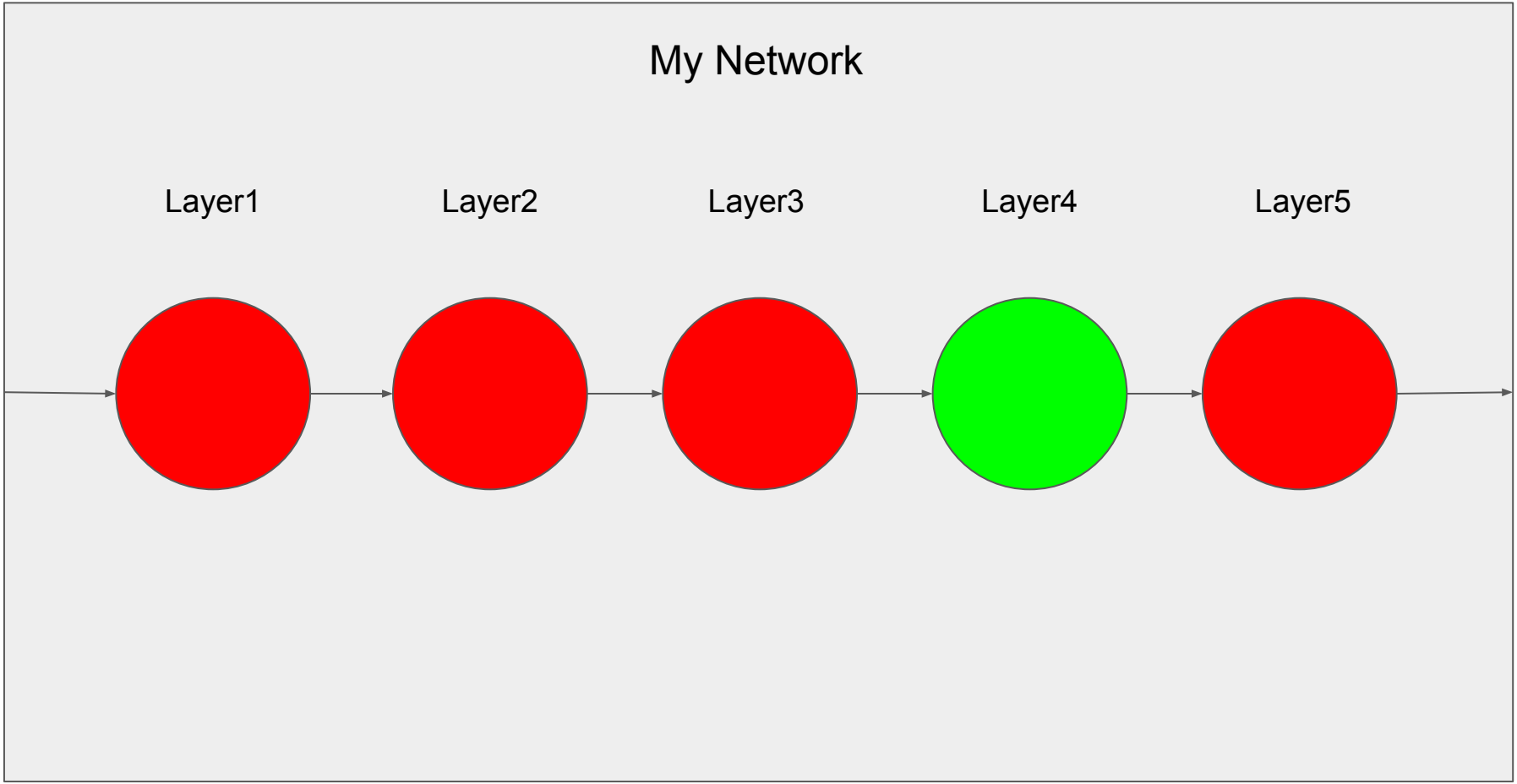
Loss = beta*style_loss + alpha*content_loss

Machine Learning is ***all*** optimization

Loss = beta*style_loss + alpha*content_loss

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{content} + \beta\mathcal{L}_{style}$$


My Network
Layer1
Layer2
Layer3
Layer4
Layer5


My Network
Layer1
Layer2
Layer3
Layer4
Layer5

Constant

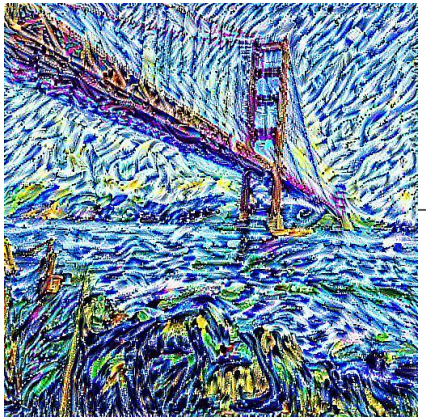
Constant

Model

Style Features

Style Loss

Constant

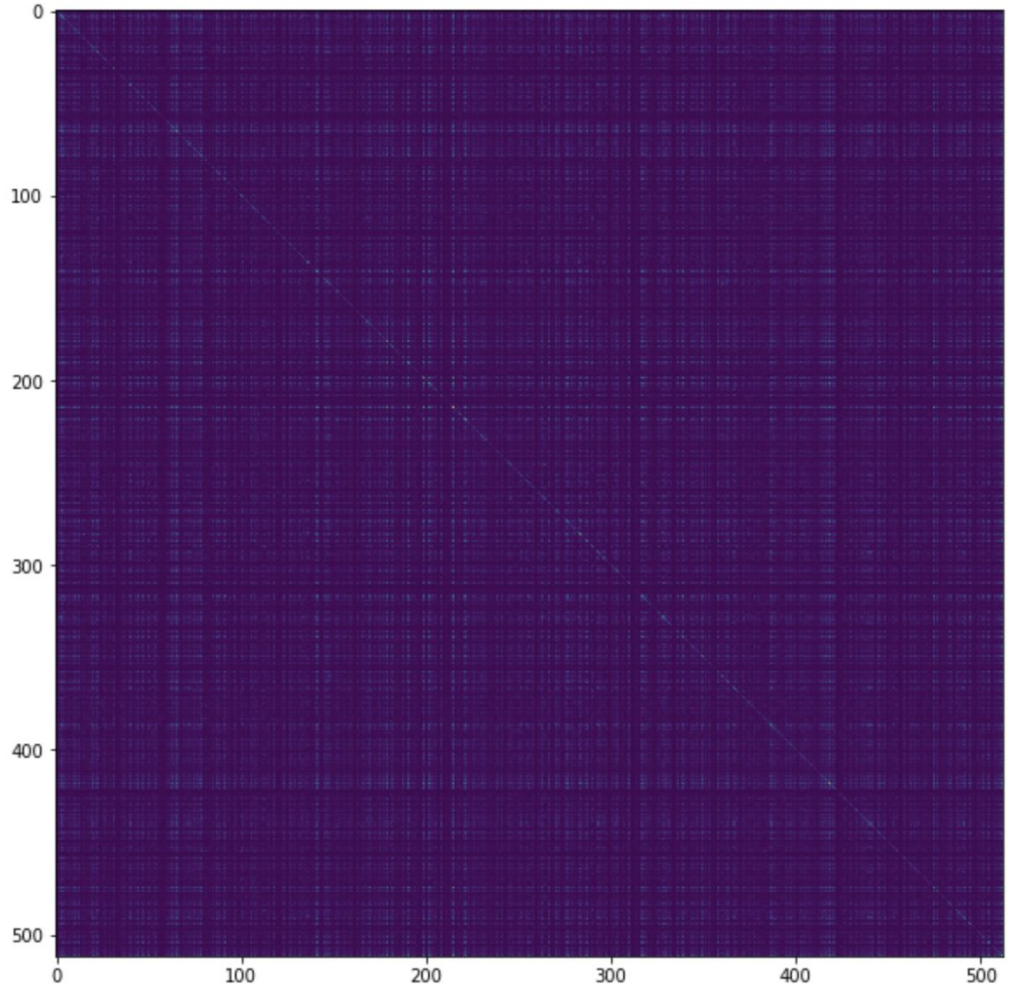


Model

Style Features

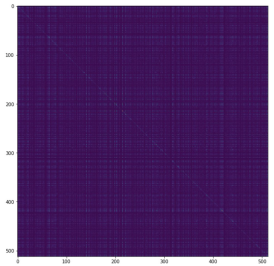
Gram Matrix:
The Matrix of all possible inner products.
“How close is every feature to every other feature?”

$$G_{ij} = \sum_k F_{ik} F_{jk}$$



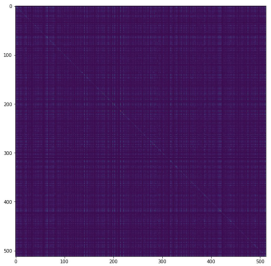
Style Loss

Style Image
Style



-

Input
Style



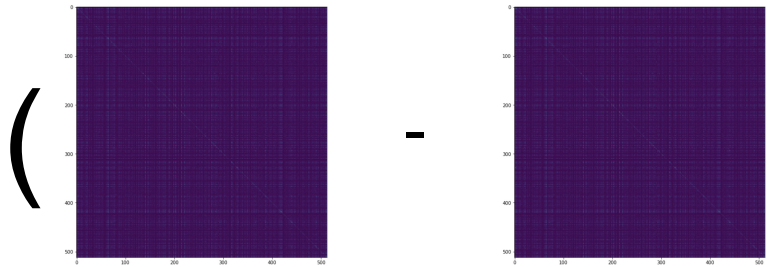
$)^2$

(We add these for all layers that we are interested in)

Style Loss

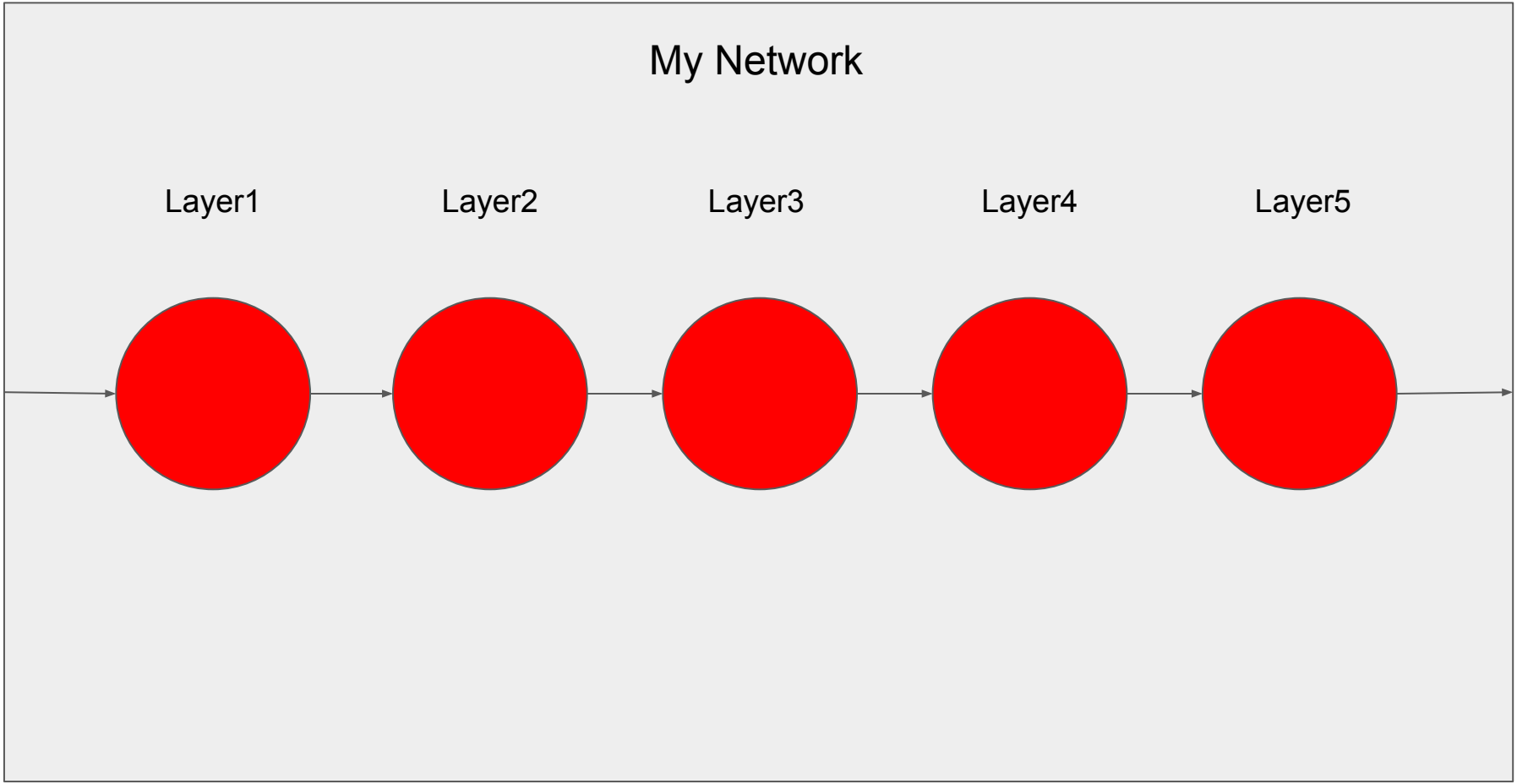
Style Image
Style

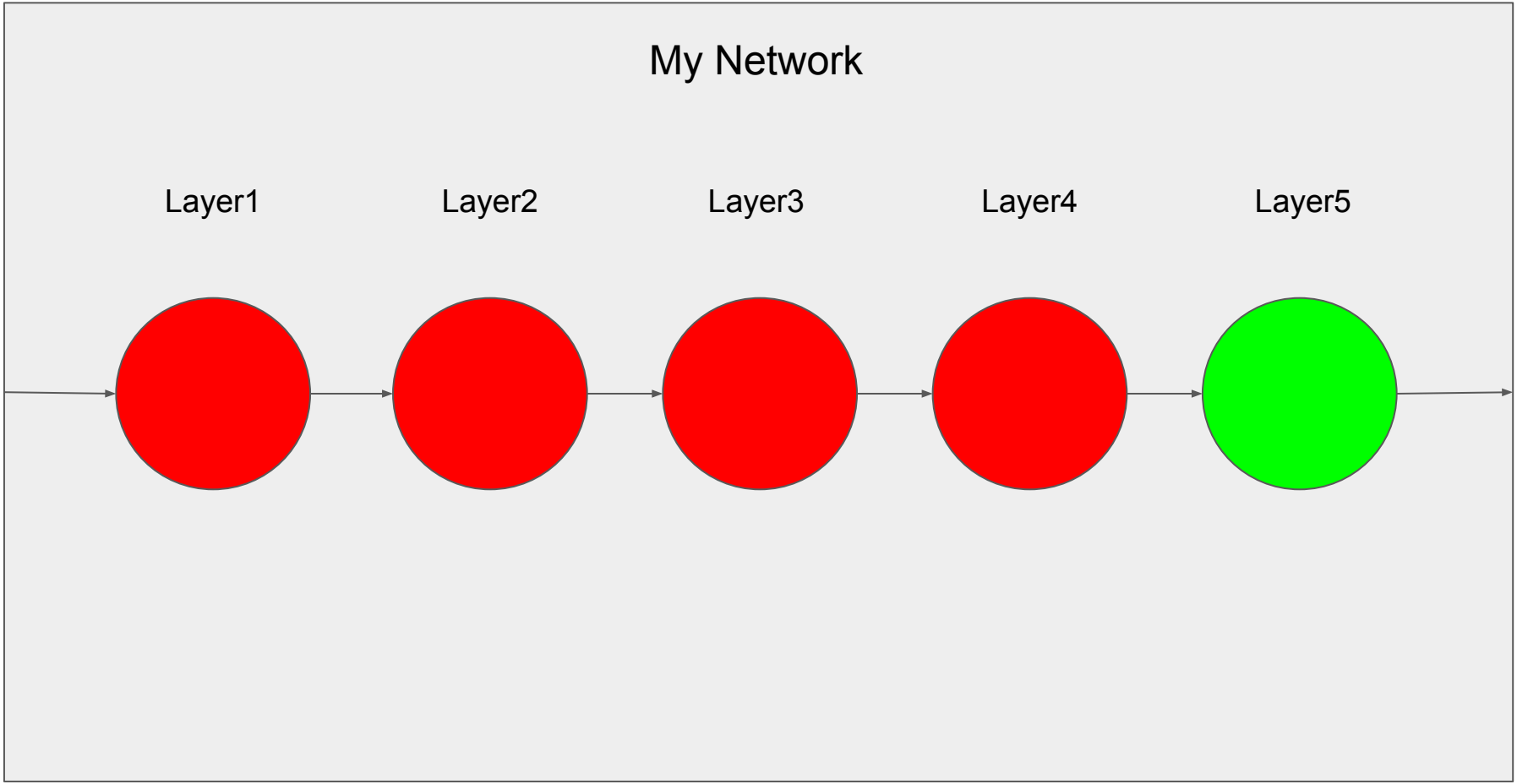
Input
Style



(We add these for all layers that we are interested in)

$$\mathcal{L}_{style} = \frac{1}{2} \sum_{l=0}^{L} (G_{ij}^{l} - A_{ij}^{l})^2$$


My Network
Layer1
Layer2
Layer3
Layer4
Layer5


My Network
Layer1
Layer2
Layer3
Layer4
Layer5


Constant
Constant
Model
Content Features
Content
Loss
Constant
Model
Content Features

Content Loss

$$\mathcal{L}_{content} = \frac{1}{2} \sum_{i,j} (F_{ij}^{l} - P_{ij}^{l})^2$$

Just subtract the filters in the late stage of a well trained model. Those signals should tell you what the object is.
The content image and the output image should be recognizable as the object if the network recognizes them as the same object.

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style}$$