Image2Image applications

Roberto Paredes

Centro de Investigación
Pattern Recognition and Human Language Technologies
Universidad Politécnica de Valencia

Index

- Image Segmentation
- Image Translation
- Image Style Transfer

Image Segmentation

- Pixel level annotation
- Some applications need class label assigned to each pixel instead of ROIs
- Classical patch based methods are very expensive
- Very sensitive parameter, patch size

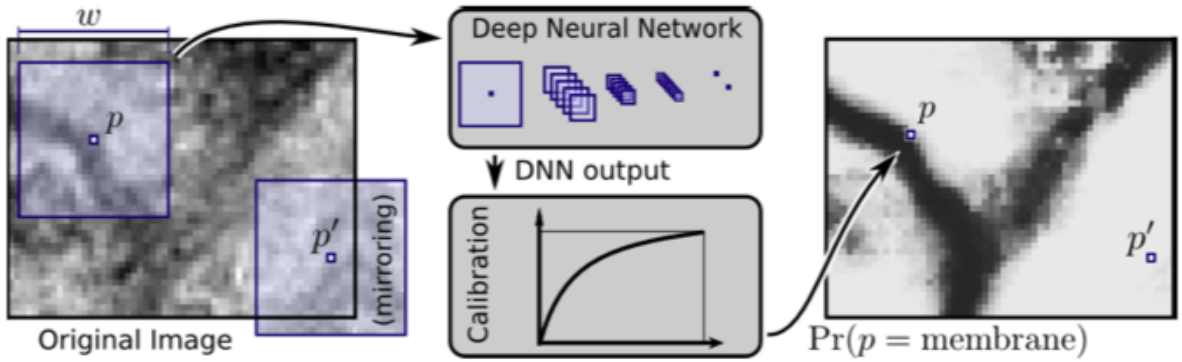

Image Segmentation

- Pixel level annotation
  `https://www.youtube.com/watch?v=OOT3UIXZztE`

Image Segmentation, U-Net

- U-Net: Convolutional Networks for Biomedical Image Segmentation
- A Fully Convolutional Network
- No fully connected layers
- Contractive and expansive paths, pooling and upsampling
- These contractive and expansive paths are more or less symmetric

Image Segmentation, U-Net

- U-net:

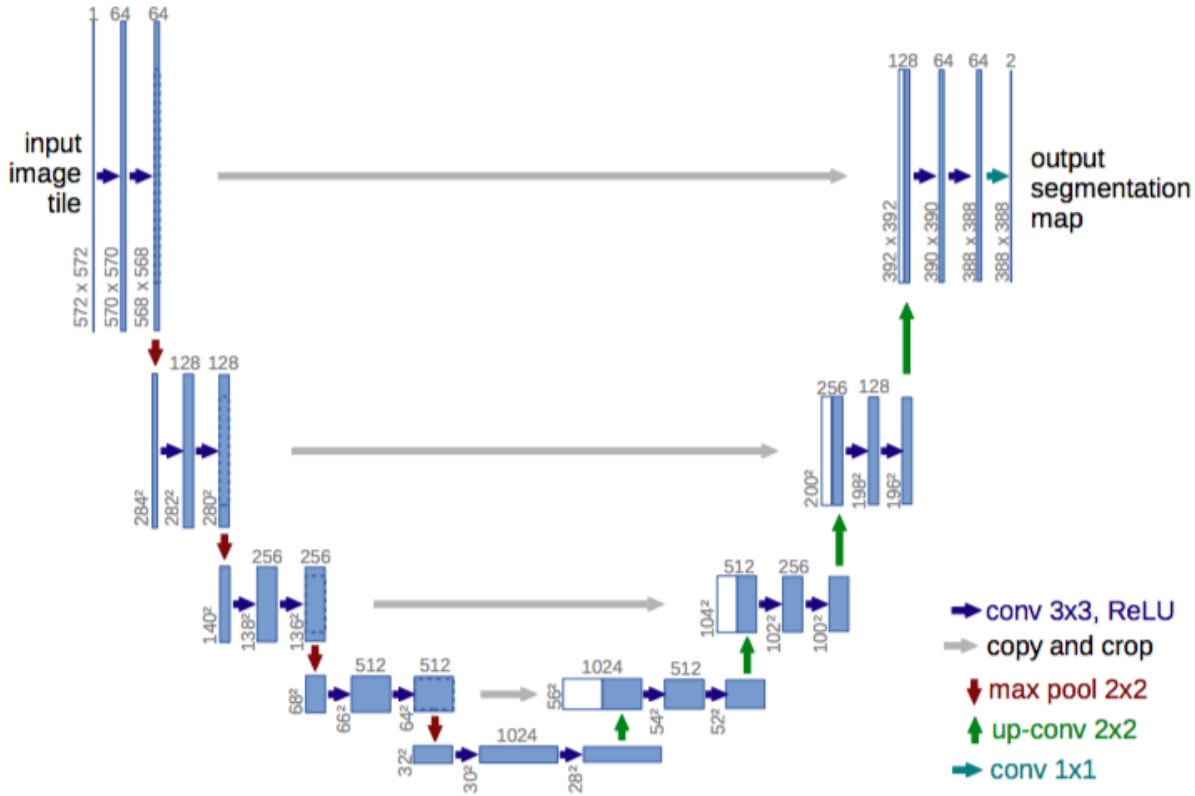

Image Segmentation, U-Net

- Segmentation of arbitrarily large images by an overlap tile strategy
- Prediction of the segmentation in the yellow area, requires image data within the blue area as input

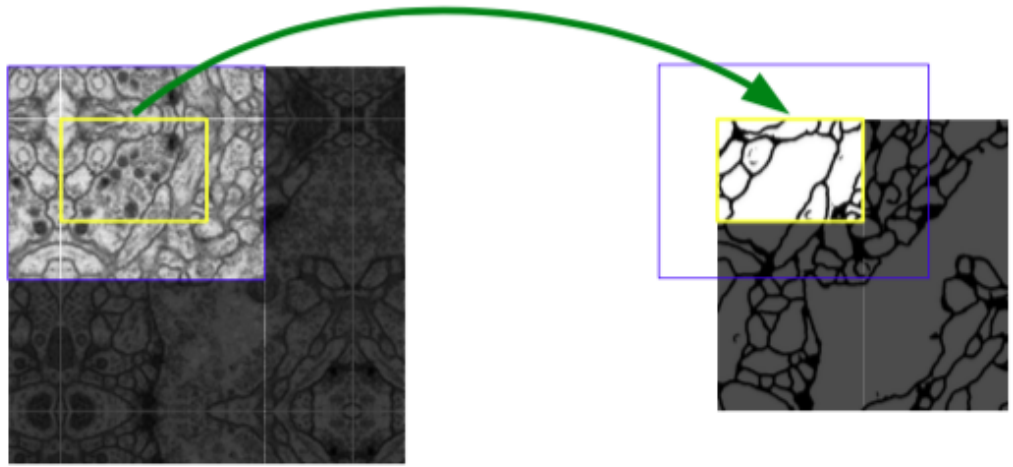

Image Segmentation, U-Net

- Output is a softmax
- Loss is a weighted cross-entropy to deal properly with the background pixels near to objects

$$loss = \sum_x (w(x) log(o_x * l_x)$$

- the weight is computed as:

$$w(x) = w_c(x) + w_0 * e^{\frac{(d_1(x)+d_2(x))^2}{2\sigma^2}}$$

- w_C balance the class frequencies, d_1 and d_2 denotes the distance to the nearest border and second nearest border
- while $w_0 = 10$ and $\sigma = 5$

Image Segmentation, U-Net

- Appropriate weighted loss for the background near to object borders

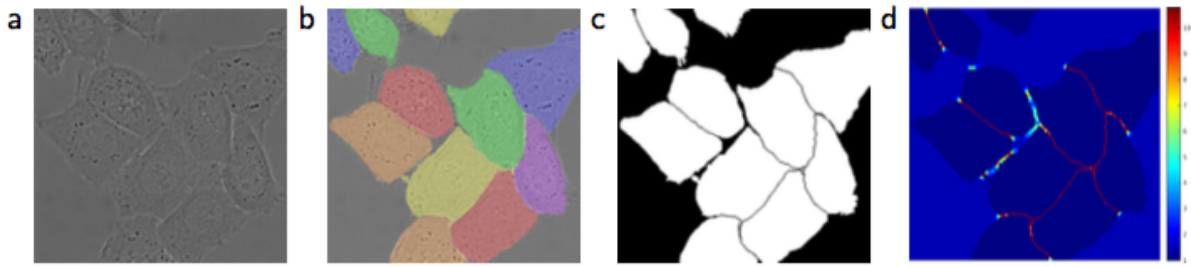

Image Segmentation, U-Net

- Appropriate Data Augmentation:
  - Shift
  - Rotation
  - Elastic deformations
  - Gray-level modifications

Image Segmentation, U-Net

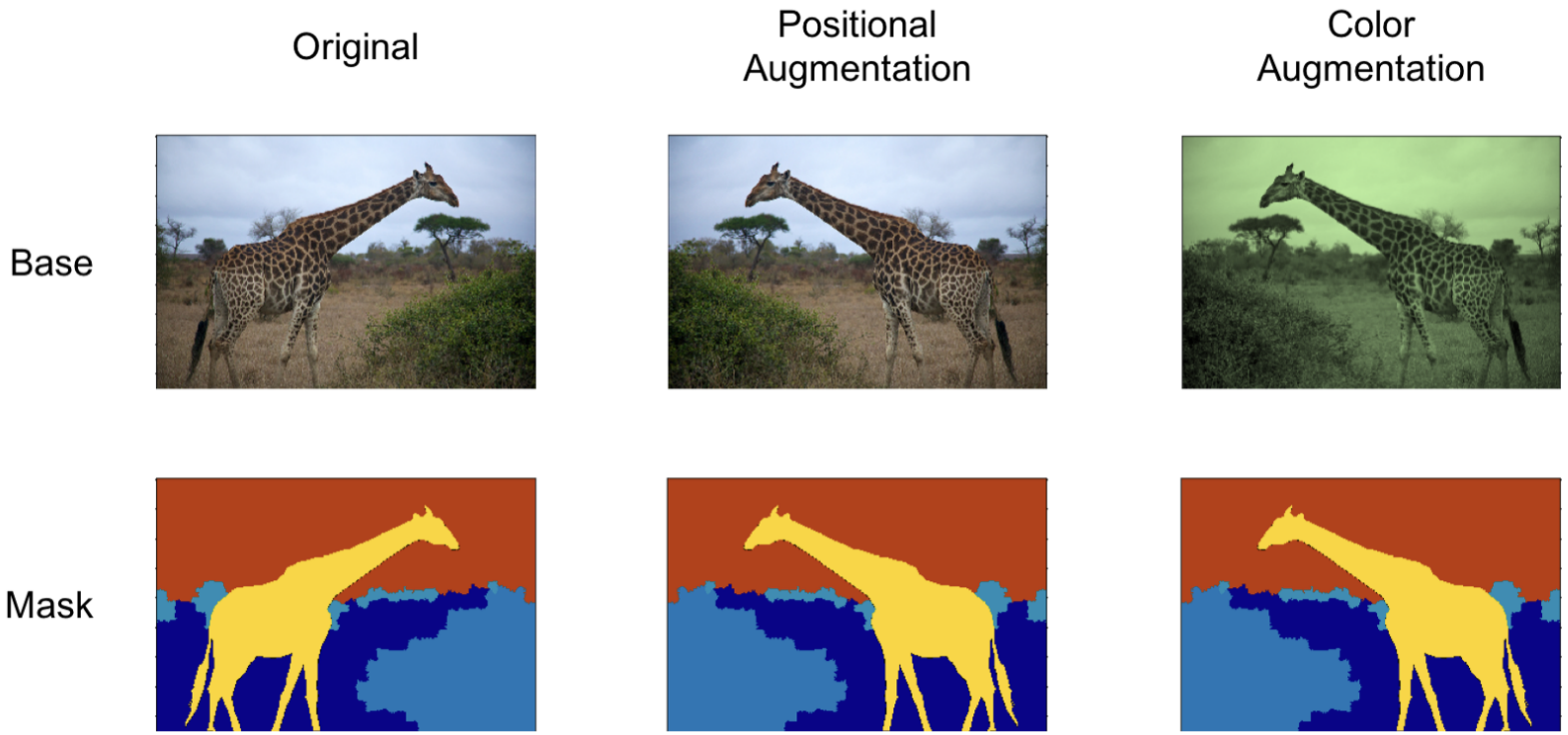

Image Segmentation, U-Net

- Results:

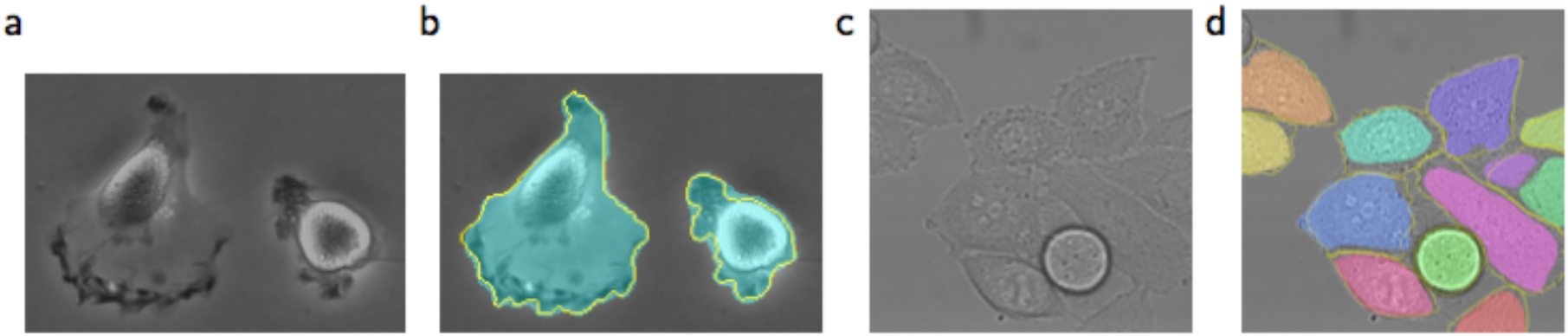

Image Segmentation, Mask-RCNN

- Mask-RCNN: intuitive extension of Faster R-CNN
- Predict segmentation masks on each Region of Interest (RoI) of the Faster R-CNN
- Predict a binary mask for each class independently, without competition among classes
- during training uses a multi-task loss on each sampled RoI as $L = L_{cls} + L_{box} + L_{mask}$
- Let's see again Faster R-CNN

Image Segmentation, Mask-RCNN

- Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
- An unified network

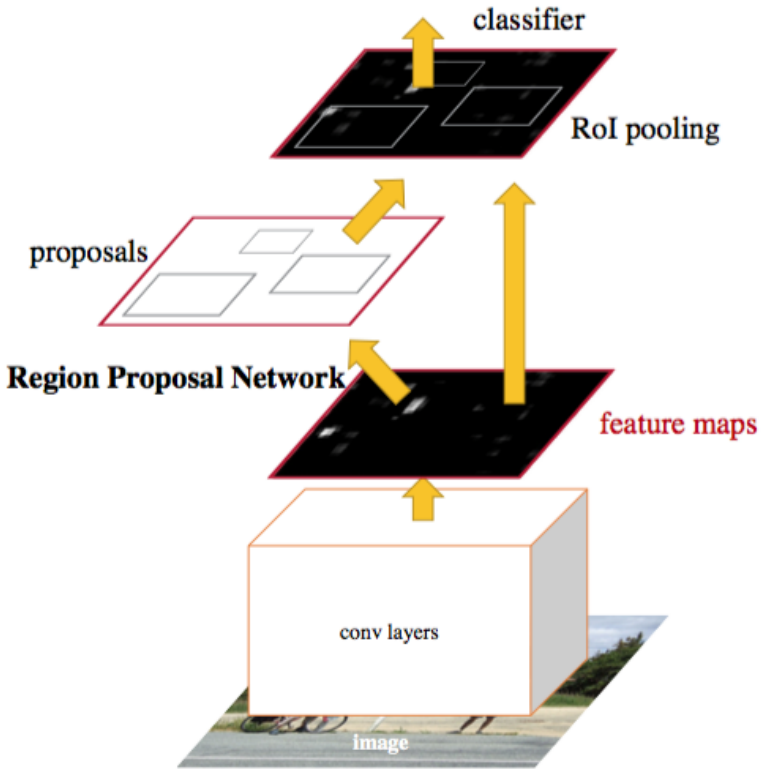

Image Segmentation, Mask-RCNN

- Faster R-CNN two modules:
  1. A deep fully convolutional network that proposes regions (RPN)
  2. A Fast-RCNN over the proposed regions
- An unified framework where the RPN module tells the Fast R-CNN module where to look

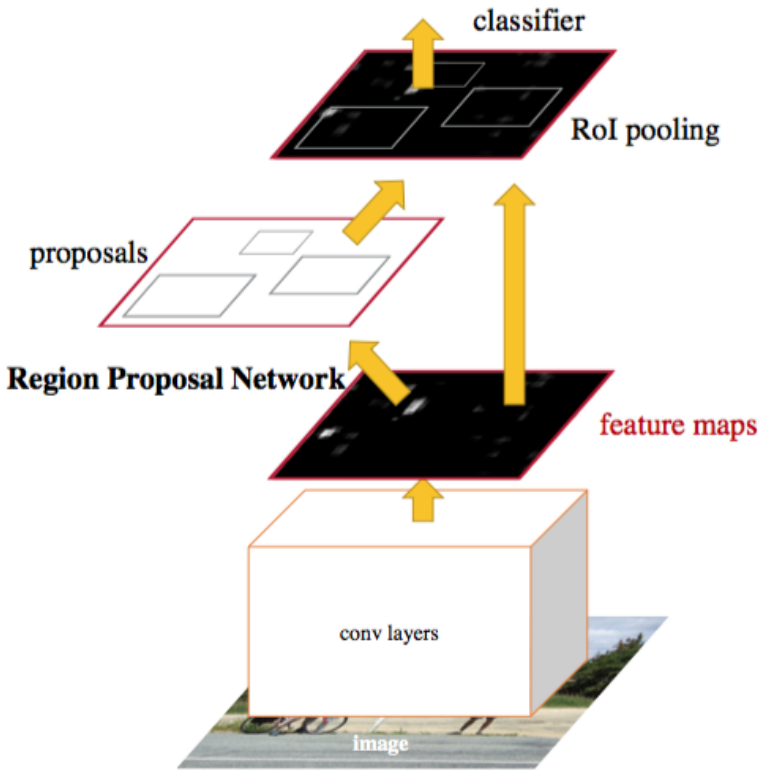

Image Segmentation, Mask-RCNN

- Training RPN:
  - Objectness loss
  - Regression box loss (only for positive samples)
- Training both, RPN and Fast R-CNN:
  - Alternating training (iterative alternating process)
  - Approximate joint training (joint RPN and Fast R-CNN)
  - Non-approximate joint training (using a RoI pooling layer that is differentiable)
  - Step Alternating Training. (not share (Both) - share cnn (RPN) - share cnn (Faster R-CNN))

Image Segmentation, Mask-RCNN

- The new loss L_{mask} is a binary cross-entropy
- A sigmoid is used in the output layer
- For an RoI associated with ground-truth class k, L_{mask} is only defined on the k-th mask
- Other mask outputs do not contribute to the loss

Image Segmentation, Mask-RCNN

- Results

Image Translation

- Image-to-Image Translation with Conditional Adversarial Networks

Image Translation

- Image-to-Image Translation with Conditional Adversarial Networks

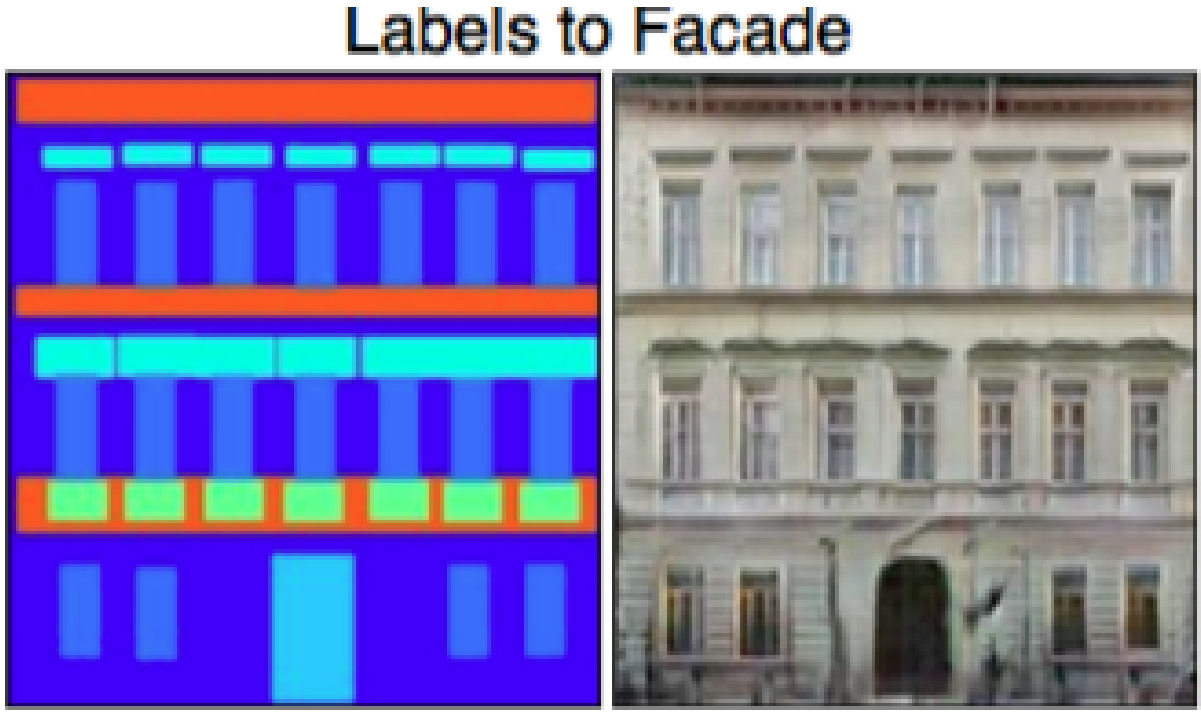

Image Translation

- Image-to-Image Translation with Conditional Adversarial Networks

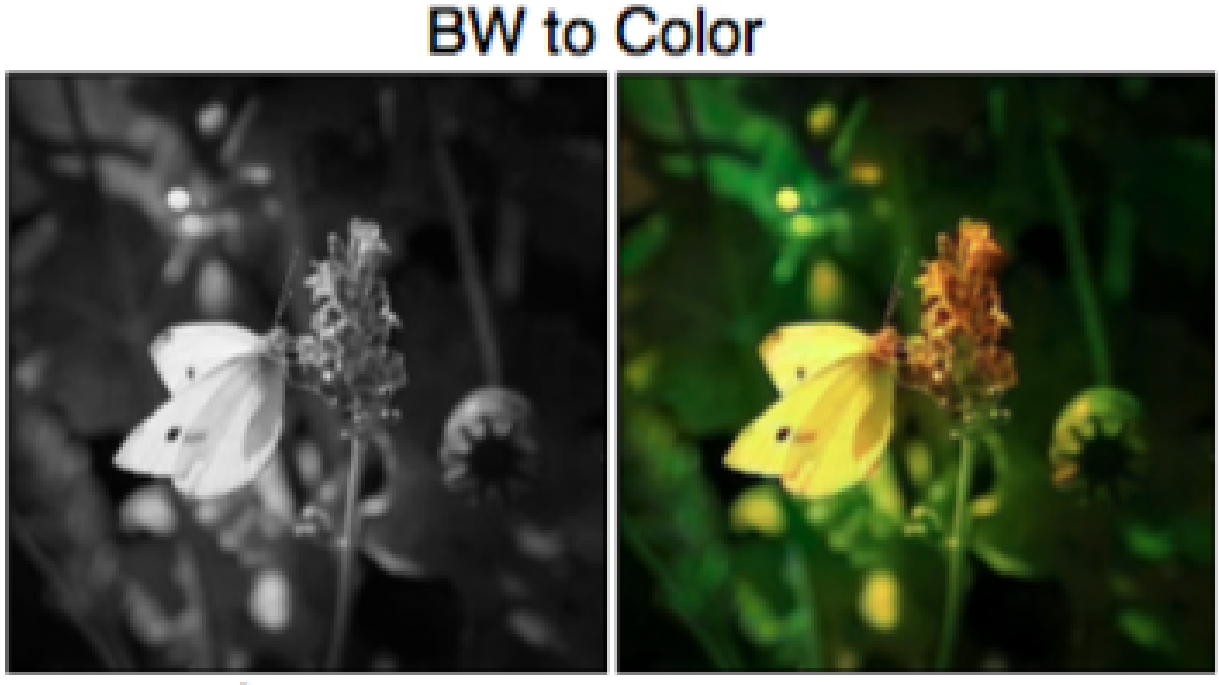

Image Translation

- Image-to-Image Translation with Conditional Adversarial Networks

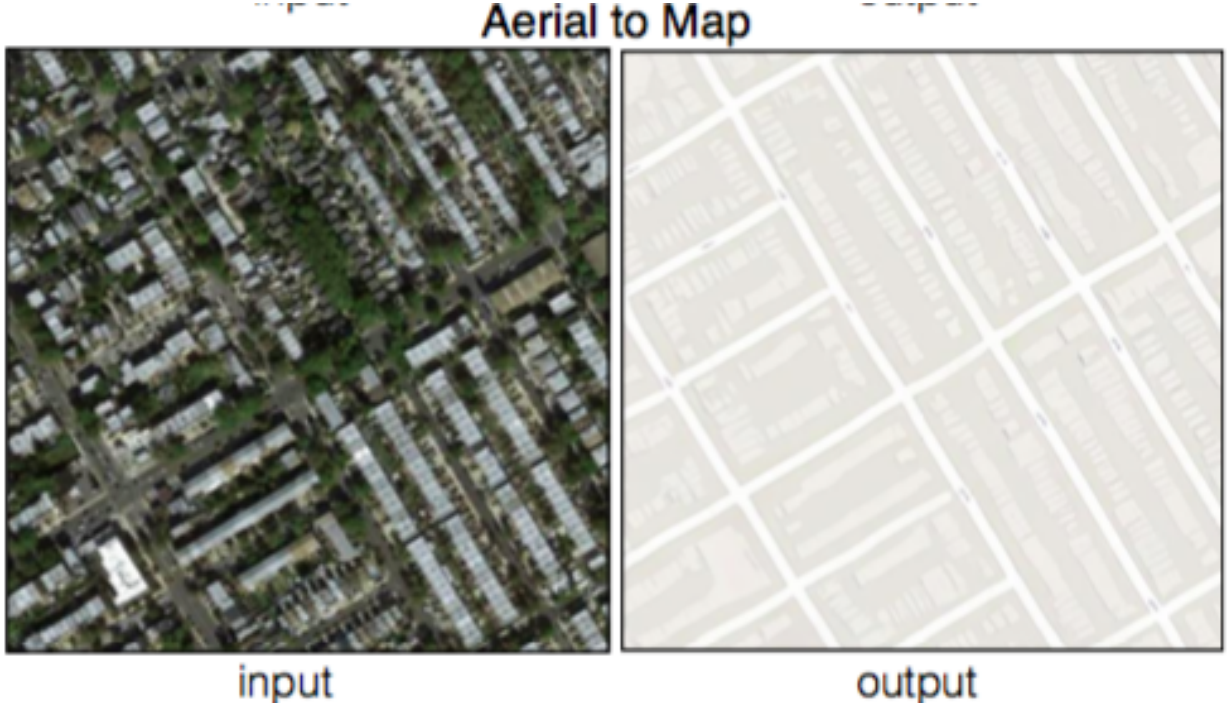

Image Translation

- Image-to-Image Translation with Conditional Adversarial Networks

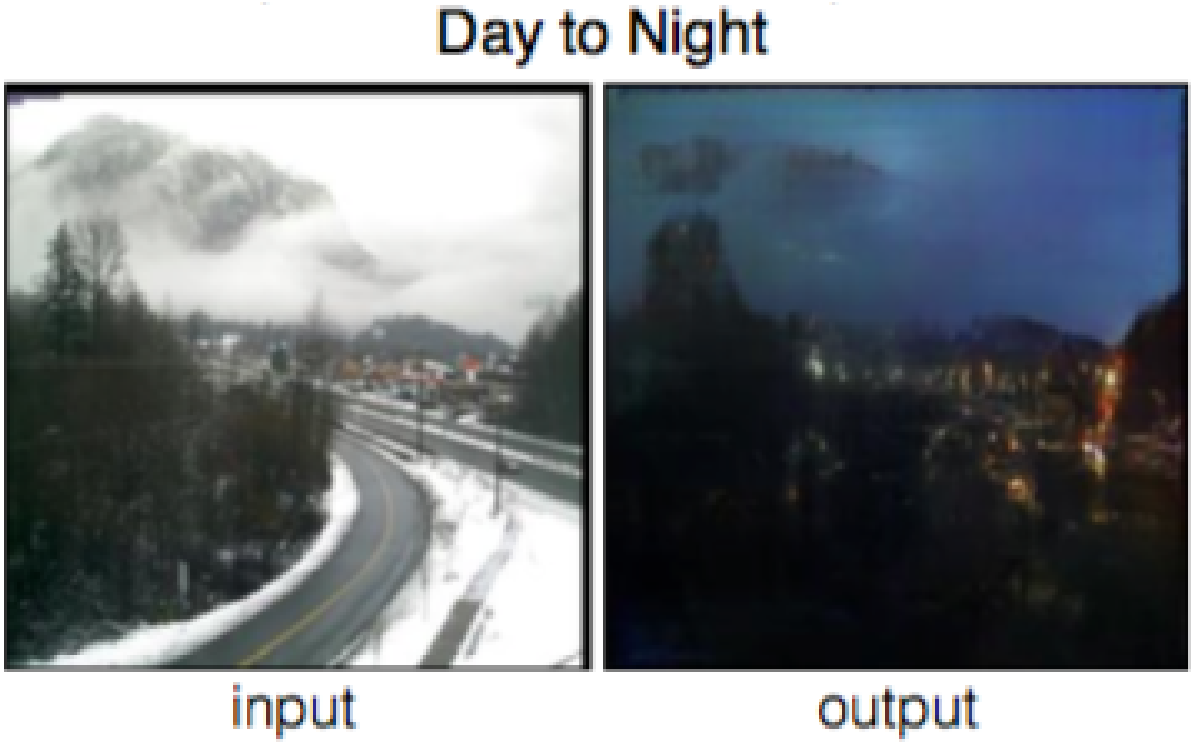

Image Translation

- Image-to-Image Translation with Conditional Adversarial Networks

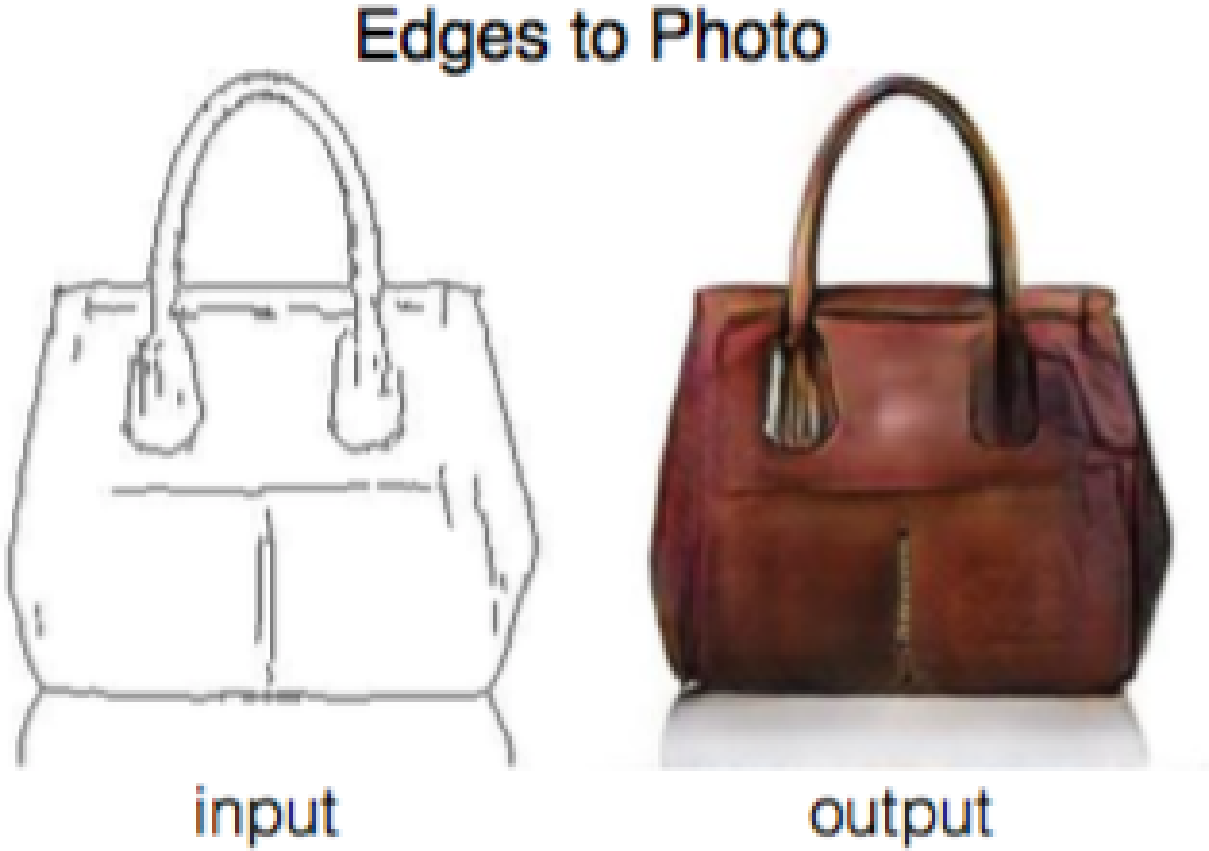

Image Translation

- In order to *translate* an image to obtain a desirable output, sometimes is better to rely on generative approaches to obtain realistic results
- Remember the blurry images obtained from classical sum of quadratic error losses
- Therefore, GAN seems a good framework to obtain realistic images, however these images must be conditioned to the source
- Conditional GANs is the appropriate framework

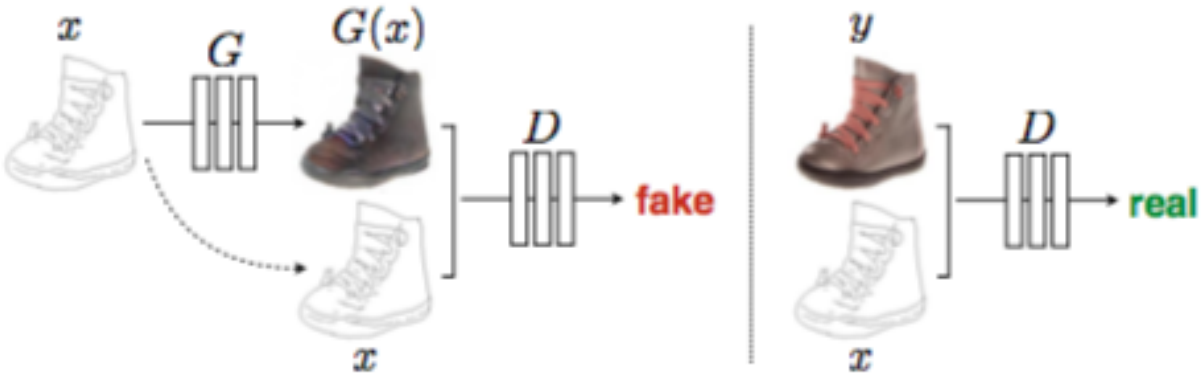

Image Translation

- Generator take (x, z) generate an image y
- GAN

$$L(G, D) = \mathbb{E}_y[logD(y)] + \mathbb{E}_z[log(1 - D(G(x, z))]$$

- Conditional-GAN

$$L_c(G, D) = \mathbb{E}_{x,y}[logD(x, y)] + \mathbb{E}_{x,z}[log(1 - D(x, G(x, z))]$$

- Some benefits observed adding an extra term:

$$L_{L1}(G) = \mathbb{E}_{x,y,z}[|| \, y - G(x, z) \, ||_1]$$

$$G^* = \arg\min_G \arg\max_D \{L_c(G, D) + \lambda L_{L1}(G)\}$$

Image Translation

- Generator is:
  - Conventional encoder-decoder (conv)
  - U-Net

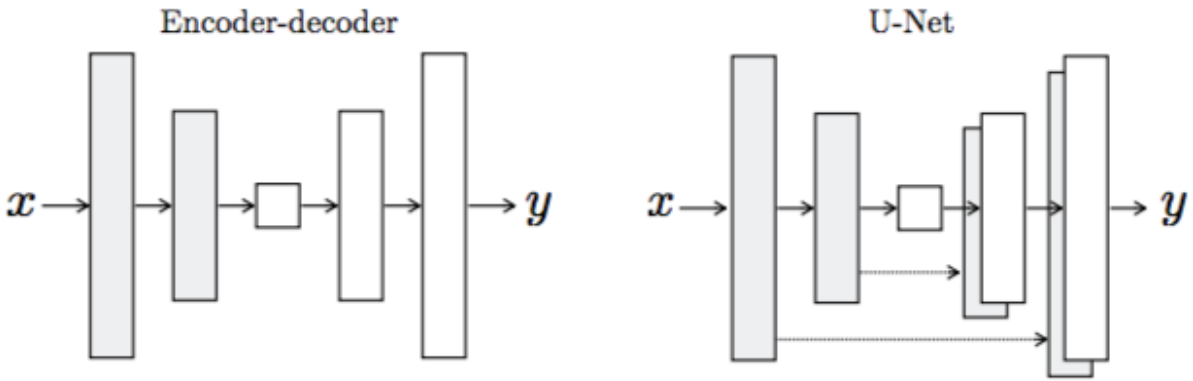

Image Translation

- Discriminator works a patch level (patchGAN)
- Each $N \times N$ patch of the image is evaluated
- The discriminator tries to classify if each $N \times N$ patch in an image is real or fake
- This can be done convolutionally across the image
- Finally averaging all responses to provide the ultimate output of D
- Such a discriminator effectively models the image as a Markov random field
- Assuming independence between pixels separated by more than a patch diameter

Image Translation

- Results depending on the loss function

Image Translation

- Results depending on the Generator

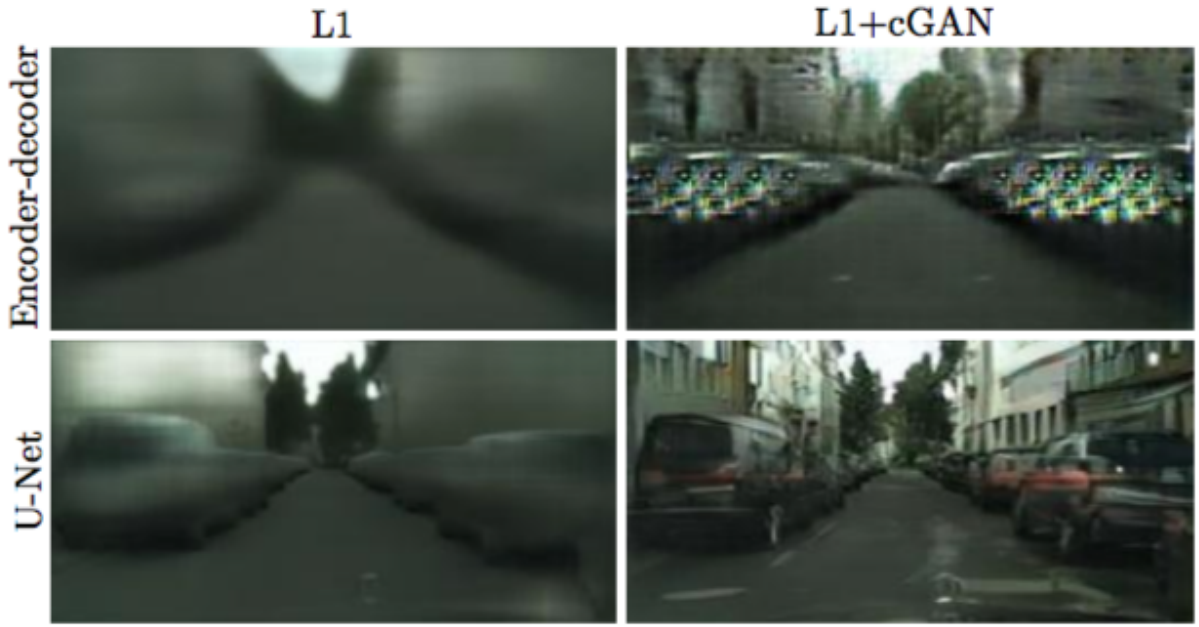

Image Translation

- Results are specific for the training dataset

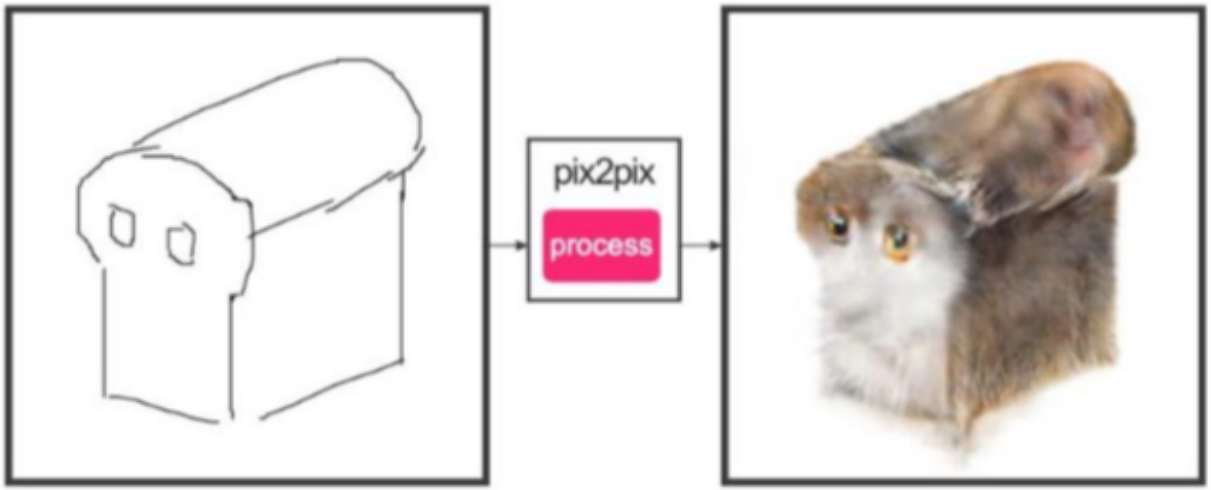

Image Style Transfer

Image Style Transfer

- VGG-19: 16 Conv + 5 pooling
- Each conv layer l is a feature map with N_l filters and M_l size, $M_l = h_l \times w_l$
- F_{ij}^l is the activation of the i^{th} filter at position j in layer l
- Content Loss:

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

  where P^l is the feature map of the original image and F^l is the feature map of a random (noise) generated image X
- Idea: By gradient descent modify X to have a similar content than P

Image Style Transfer

- Style Loss, Gram matrix G^l

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}$$

- then, define the Style Loss as:

$$L_{style}(Q, X) = \frac{1}{4N_l^2 M_l^2} \sum_{ij} w_l (G^l_{ij} - A^l_{ij})^2$$

  where w_l is a weight for each layer, G^l_{ij} is the gram matrix of the original style image while A^l_{ij} is the gram matrix of the random generated image

- Idea: By gradient descent modify X to have a similar style than P

Image Style Transfer

- Total Loss:

$$L_{total} = \alpha L_{content}(P, X) + \beta L_{style}(Q, X)$$

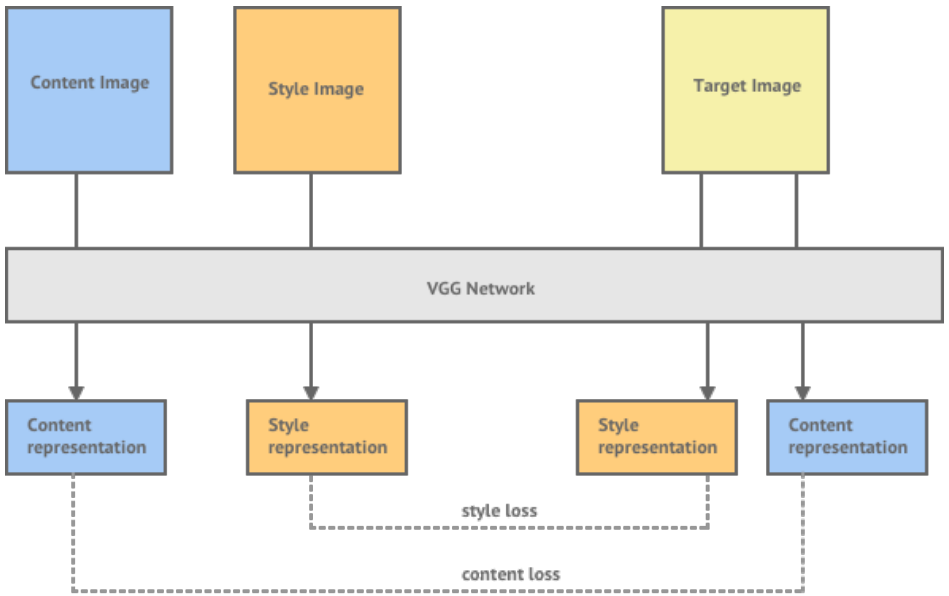

Image Style Transfer

- An example
  `https://cdn-images-1.medium.com/max/1600/1*r2T1RTjGMyCvYwY3EIOp1Q.gif`