Image-to-image translation
Outline

• Paired image-to-image translation: pix2pix
• Unpaired image-to-image translation: CycleGAN
• Extensions, applications
Paired image-to-image translation

Pix2pix

- Produce modified image $y$ conditioned on input image $x$ (note change of notation)
  - Generator receives $x$ as input
  - Discriminator receives an $x, y$ pair and has to decide whether it is real or fake
Pix2pix: Generator

- Generator architecture: U-Net (no $z$ used as input)
Pix2pix: Generator

- Generator architecture: U-Net (no $z$ used as input)

Encode: convolution $\rightarrow$ BatchNorm $\rightarrow$ ReLU

Decode: transposed convolution $\rightarrow$ BatchNorm $\rightarrow$ ReLU

Figure source
Pix2pix: Generator

Effect of adding skip connections to the generator

<table>
<thead>
<tr>
<th>Encoder-decoder</th>
<th>L1</th>
<th>L1+cGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-Net</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pix2pix: Generator loss

- GAN loss plus L1 reconstruction penalty

\[ G^* = \arg \min_G \max_D \mathcal{L}_{GAN}(G, D) + \lambda \sum_i \| y_i - G(x_i) \|_1 \]

Generated output
\[ G(x_i) \] should be close to ground truth target \( y_i \)
Pix2pix: Generator loss

- GAN loss plus L1 reconstruction penalty

\[ G^* = \arg \min_G \max_D \mathcal{L}_{GAN}(G, D) + \lambda \sum_i \|y_i - G(x_i)\|_1 \]
Pix2pix: Discriminator

- Given input image $x$ and second image $y$, decide whether $y$ is a ground truth target or produced by the generator.
Pix2pix: Discriminator

• “PatchGAN” architecture: output a 30x30 map where each value (0 to 1) represents the quality of the corresponding section of the output image, average to obtain final discriminator loss

• Implemented as FCN, effective patch size can be increased by increasing the depth

Figure source
Pix2pix: Discriminator

- “PatchGAN” architecture: output a 30x30 map where each value (0 to 1) represents the quality of the corresponding section of the output image, average to obtain final discriminator loss
- Implemented as FCN, effective patch size can be increased by increasing the depth

Effect of discriminator patch size on generator output

L1  1×1  16×16  70×70  286×286
Pix2pix: Results

- Translating between maps and aerial photos
Pix2pix: Results

- Translating between maps and aerial photos
- Human study:

<table>
<thead>
<tr>
<th>Loss</th>
<th>Photo $\rightarrow$ Map</th>
<th>Map $\rightarrow$ Photo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>% Turkers labeled real</td>
<td>% Turkers labeled real</td>
</tr>
<tr>
<td>L1</td>
<td>2.8% ± 1.0%</td>
<td>0.8% ± 0.3%</td>
</tr>
<tr>
<td>L1+cGAN</td>
<td>6.1% ± 1.3%</td>
<td>18.9% ± 2.5%</td>
</tr>
</tbody>
</table>
Pix2pix: Results

- Semantic labels to scenes
Pix2pix: Results

- Semantic labels to scenes
  - Evaluation: FCN score – the higher the quality of the output, the better the FCN should do at recovering the original semantic labels

<table>
<thead>
<tr>
<th>Loss</th>
<th>Per-pixel acc.</th>
<th>Per-class acc.</th>
<th>Class IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>0.42</td>
<td>0.15</td>
<td>0.11</td>
</tr>
<tr>
<td>GAN</td>
<td>0.22</td>
<td>0.05</td>
<td>0.01</td>
</tr>
<tr>
<td>cGAN</td>
<td>0.57</td>
<td>0.22</td>
<td>0.16</td>
</tr>
<tr>
<td>L1+GAN</td>
<td>0.64</td>
<td>0.20</td>
<td>0.15</td>
</tr>
<tr>
<td>L1+cGAN</td>
<td><strong>0.66</strong></td>
<td><strong>0.23</strong></td>
<td><strong>0.17</strong></td>
</tr>
<tr>
<td>Ground truth</td>
<td>0.80</td>
<td>0.26</td>
<td>0.21</td>
</tr>
</tbody>
</table>
Pix2pix: Results

- Scenes to semantic labels
Pix2pix: Results

- Scenes to semantic labels
  - Accuracy is worse than that of regular FCNs or generator with L1 loss

<table>
<thead>
<tr>
<th>Loss</th>
<th>Per-pixel acc.</th>
<th>Per-class acc.</th>
<th>Class IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>0.86</td>
<td>0.42</td>
<td>0.35</td>
</tr>
<tr>
<td>cGAN</td>
<td>0.74</td>
<td>0.28</td>
<td>0.22</td>
</tr>
<tr>
<td>L1+cGAN</td>
<td>0.83</td>
<td>0.36</td>
<td>0.29</td>
</tr>
</tbody>
</table>
Pix2pix: Results

- Semantic labels to facades
Pix2pix: Results

- Day to night
Pix2pix: Results

- Edges to photos
Pix2pix: Results

• pix2pix demo

#edges2cats by Christopher Hesse

Background removal by Kaihu Chen
Sketch → Pokemon by Bertrand Gondouin
“Do as I do” by Brannon Dorsey

Palette generation by Jack Qiao

Sketch → Portrait by Mario Klingemann

Sketch by Yann LeCun

Sketch by Ivy Tsai
Pix2pix: Limitations

• Visual quality could be improved
• Requires $x, y$ pairs for training
• Does not model conditional distribution $P(y|x)$, returns a single mode instead
Outline

• Paired image-to-image translation: pix2pix
• Unpaired image-to-image translation: CycleGAN
Unpaired image-to-image translation

- Given two unordered image collections $X$ and $Y$, learn to “translate” an image from one into the other and vice versa.

Unpaired image-to-image translation

- Given two unordered image collections $X$ and $Y$, learn to “translate” an image from one into the other and vice versa.

CycleGAN

- Given: domains $X$ and $Y$
- Train two generators $F$ and $G$ and two discriminators $D_X$ and $D_Y$
  - $G$ translates from $X$ to $Y$, $F$ translates from $Y$ to $X$
  - $D_X$ recognizes images from $X$, $D_Y$ from $Y$
  - Cycle consistency: we want $F(G(x)) \approx x$ and $G(F(y)) \approx y$

![CycleGAN Diagram]

- $D_X$ - $G$ - $D_Y$
- $X$ - $G$ - $Y$ - $F$
CycleGAN: Architecture

- Generators (based on Johnson et al., 2016):

- Discriminators: PatchGAN on 70 x 70 patches
CycleGAN: Loss

- Requirements:
  - $G$ translates from $X$ to $Y$, $F$ translates from $Y$ to $X$
  - $D_X$ recognizes images from $X$, $D_Y$ from $Y$
  - We want $F(G(x)) \approx x$ and $G(F(y)) \approx y$

- CycleGAN discriminator loss: LSGAN

\[
\mathcal{L}_{GAN}(D_Y) = \mathbb{E}_{y \sim p_{data}(y)}[(D_Y(y) - 1)^2] + \mathbb{E}_{x \sim p_{data}(x)}[D_Y(G(x))^2]
\]

\[
\mathcal{L}_{GAN}(D_X) = \mathbb{E}_{x \sim p_{data}(x)}[(D_X(x) - 1)^2] + \mathbb{E}_{y \sim p_{data}(y)}[D_X(F(y))^2]
\]

- CycleGAN generator loss:

\[
\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}[D_Y(G(x) - 1)^2] + \mathbb{E}_{y \sim p_{data}(y)}[D_X(F(y) - 1)^2]
\]

\[
+ \mathbb{E}_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)}[\|G(F(y)) - y\|_1]
\]
CycleGAN

- Illustration of cycle consistency:
CycleGAN: Results

- Translation between maps and aerial photos
CycleGAN: Results

- Other pix2pix tasks
CycleGAN: Results

- Scene to labels and labels to scene
  - Worse performance than pix2pix due to lack of paired training data

<table>
<thead>
<tr>
<th>Loss</th>
<th>Per-pixel acc.</th>
<th>Per-class acc.</th>
<th>Class IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoGAN [32]</td>
<td>0.40</td>
<td>0.10</td>
<td>0.06</td>
</tr>
<tr>
<td>BiGAN/ALI [9, 7]</td>
<td>0.19</td>
<td>0.06</td>
<td>0.02</td>
</tr>
<tr>
<td>SimGAN [46]</td>
<td>0.20</td>
<td>0.10</td>
<td>0.04</td>
</tr>
<tr>
<td>Feature loss + GAN</td>
<td>0.06</td>
<td>0.04</td>
<td>0.01</td>
</tr>
<tr>
<td>CycleGAN (ours)</td>
<td><strong>0.52</strong></td>
<td><strong>0.17</strong></td>
<td><strong>0.11</strong></td>
</tr>
<tr>
<td>pix2pix [22]</td>
<td>0.71</td>
<td>0.25</td>
<td>0.18</td>
</tr>
</tbody>
</table>

Table 2: FCN-scores for different methods, evaluated on Cityscapes labels→photo.

<table>
<thead>
<tr>
<th>Loss</th>
<th>Per-pixel acc.</th>
<th>Per-class acc.</th>
<th>Class IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoGAN [32]</td>
<td>0.45</td>
<td>0.11</td>
<td>0.08</td>
</tr>
<tr>
<td>BiGAN/ALI [9, 7]</td>
<td>0.41</td>
<td>0.13</td>
<td>0.07</td>
</tr>
<tr>
<td>SimGAN [46]</td>
<td>0.47</td>
<td>0.11</td>
<td>0.07</td>
</tr>
<tr>
<td>Feature loss + GAN</td>
<td>0.50</td>
<td>0.10</td>
<td>0.06</td>
</tr>
<tr>
<td>CycleGAN (ours)</td>
<td><strong>0.58</strong></td>
<td><strong>0.22</strong></td>
<td><strong>0.16</strong></td>
</tr>
<tr>
<td>pix2pix [22]</td>
<td>0.85</td>
<td>0.40</td>
<td>0.32</td>
</tr>
</tbody>
</table>

Table 3: Classification performance of photo→labels for different methods on cityscapes.
CycleGAN: Results

- Tasks for which paired data is unavailable
CycleGAN: Results

- Style transfer
CycleGAN: Failure cases

Input | Output
--- | ---
apple | orange
zebra | horse
winter | summer
dog | cat
cat | dog
Monet | photo
photo | Ukiyo-e
photo | Van Gogh
iPhone photo | DSLR photo
CycleGAN: Failure cases

Input

Output

horse $\rightarrow$ zebra
CycleGAN: Limitations

- Cannot handle shape changes (e.g., dog to cat)
- Can get confused on images outside of the training domains (e.g., horse with rider)
- Cannot close the gap with paired translation methods
- Does not account for the fact that one transformation direction may be more challenging than the other
Outline

- Paired image-to-image translation: pix2pix
- Unpaired image-to-image translation: CycleGAN
- Extensions, applications
Multimodal image-to-image translation

Toward Multimodal Image-to-Image Translation, NIPS 2017
High-resolution, high-quality pix2pix

T.-C. Wang et al., High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs, CVPR 2018
High-resolution, high-quality pix2pix

- Two-scale generator architecture (up to 2048 x 1024 resolution)

First train *global generator* network (G1) on lower-res images

Then append higher-res *enhancer network* (G2) blocks and train G1 and G2 jointly

T.-C. Wang et al., *High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs*, CVPR 2018
Human generation conditioned on pose

https://carolineec.github.io/everybody_dance_now/

Human generation conditioned on pose

Figure 3: (Top) **Training**: Our model uses a pose detector $P$ to create pose stick figures from video frames of the target subject. We learn the mapping $G$ alongside an adversarial discriminator $D$ which attempts to distinguish between the “real” correspondences $(x_t, x_{t+1}), (y_t, y_{t+1})$ and the “fake” sequence $(x_t, x_{t+1}), (G(x_t), G(x_{t+1}))$. (Bottom) **Transfer**: We use a pose detector $P$ to obtain pose joints for the source person that are transformed by our normalization process $Norm$ into joints for the target person for which pose stick figures are created. Then we apply the trained mapping $G$.

Human generation conditioned on pose

https://carolineec.github.io/everybody_dance_now/