



به نام خدا



دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
مدل‌های مولد عمیق

تمرین دوم

سید محمد جزایری	نام و نام خانوادگی
810101399	شماره دانشجویی
1404/09/01	تاریخ ارسال گزارش

فهرست

- سوال اول.....
2.....
2.....
بخش اول.....

2.....	زیر بخش اول)
3.....	زیر بخش دوم)
3.....	بخش دوم.....
3.....	زیر بخش اول)
4.....	زیر بخش دوم)
5.....	زیر بخش سوم)
5.....	سوال دوم.....
5.....	بخش اول.....
5.....	زیر بخش اول)
6.....	زیر بخش دوم)
7.....	زیر بخش سوم)
8.....	بخش دوم.....
8.....	زیر بخش اول)
8.....	زیر بخش سوم)
11.....	زیر بخش چهارم)
14.....	مراجع.....

سؤال اول

بخش اول

(زیر بخش اول)

طبق اسلاید ها می دانیم:

$$P_X(x) = P_Z(f^{-1}(x)) \left| \frac{\partial f^{-1}(x)}{\partial x} \right|$$

$$1. \quad Z = \frac{X-1}{2} \Rightarrow \frac{\partial f^{-1}(x)}{\partial x} = \frac{1}{2}$$

$$0 \leq Z \leq 1 \Rightarrow 1 \leq X \leq 3$$

$$P_X(x) = P_Z\left(\frac{X-1}{2}\right) \left| \frac{1}{2} \right| = 1 * \frac{1}{2} = \frac{1}{2}$$

$$2. \quad Z = \log X \Rightarrow \frac{\partial f^{-1}(x)}{\partial x} = \frac{1}{x}$$

$$0 \leq Z \leq 2 \Rightarrow 1 \leq X \leq e^2$$

$$P_X(x) = P_Z(\log X) \left| \frac{1}{x} \right| = \frac{1}{2} * \frac{1}{x} = \frac{1}{2x}$$

(زیر بخش دوم)

از اسلاید ها می دانیم که ماتریس ژاکوبین اینگونه محاسبه می شود:

$$\begin{vmatrix} \frac{\partial z_1}{\partial x_1} & \frac{\partial z_1}{\partial x_2} \\ \frac{\partial z_2}{\partial x_1} & \frac{\partial z_2}{\partial x_2} \end{vmatrix}$$

1.

$$\begin{vmatrix} 1 & 0 \\ m'(x_1) & 1 \end{vmatrix}$$

چونکه ماتریس پایین مثلثی است پس دترمینان آن برابر با ضرب درایه های قطر اصلی هست، پس برابر 1 است.

2.

$$\begin{vmatrix} 1 & 0 \\ s'(x_1)x_2 + t'(x_1) & s(x_1) \end{vmatrix}$$

مشابه استدلال قبلی دترمینان ما برابر (x_1) است.

بخش دوم

زیر بخش اول)

بخش MADE در ابتدا initialize می شود مثلاً ابعاد ورودی، خروجی، لایه آخر هست یا خیر، اتصالات و.... هر لایه یک نمونه از MaskedLinear است که در صورت مسئله تعریف شده است و خطی ساده نیست. فعال ساز هر لایه تابع ReLU است. در نهایت شبکه عصبی را به صورت sequential تعریف می کنیم تا شبکه feed forward بماند. این بخش t_i و $logs_i$ را به گونه‌ای محاسبه می کند و برای اینکه تنها به x های قبلی وابسته باشد یک ماسک بولی M بر روی ماتریس وزن‌های W لایه خطی اعمال می‌شود. وزن W_{ij} (اتصال از نورون ورودی i به نورون خروجی j) تنها در صورتی فعال (غیرصفر) است که درجه m_j بزرگتر یا مساوی درجه m_i باشد.

در بخش MAF تابع موردنظر را اینگونه تعریف کرده‌ایم.

$$z_i = x_i e^{logs(x_{<i})} + t(x_{<i}) \quad , \quad x_i = (z_i - t_i(x_{<i})) e^{-log(s_i(x_{<i}))}$$

این بخش بعد از MADE اجرا می‌شود.

به دلیل ساختار اتورگرسیو ژاکوبین مجموع عناصر قطر اصلی است.

$$\log|J(f)| = \sum_i logs_i \quad loss = -\frac{1}{N} \sum_i (P_z(z) + \sum_i logs_i)$$

برای جلوگیری از انفجار گرادیان و واگرایی مدل در اوایل آموزش، از تکنیک Clamping استفاده شده است،

خروجی خام log-scale از MADE به صورت $logs = clamp(logs_{raw}, -5, 5)$ محدود می‌شود.

این محدودسازی تضمین می‌کند که عامل مقیاس‌دهی واقعی e^{logs} بین e^{-5} و e^5 باقی بماند و از واگرایی Log-Determinant جلوگیری شود.

زیر بخش دوم)

1. مجموع زمان آموزش: 1199.7 ثانیه مجموع زمان تولید عکس: 140.4 ثانیه

$$\frac{time_{gen}}{time_{train}} = \frac{140.4}{1199.7/(100 * 73 batches)} \approx 854$$

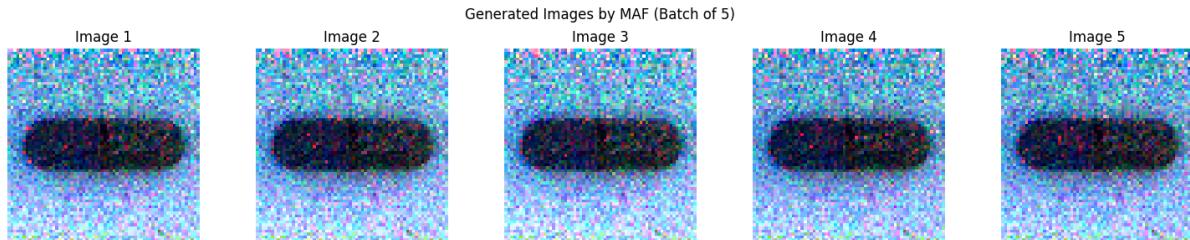
همانگونه که می‌بینیم نسبت خیلی بزرگ است.

2. چرا که z_i ها (ابعاد) بصورت موازی تولید می‌شوند اما x_i ها (ابعاد) به صورت سری تولید می‌شوند و شبکه عصبی هم بر روی آن‌ها اعمال می‌شود.

3. تصاویر

شده:

تولید



با توجه به ماهیت تصاویر و مدل مورد استفاده و نحوه آموزش این خروجی طبیعی است.

4. در مدل های IAF شبکه عصبی بر روی z اعمال می شود و z ها بصورت سری تولید می شوند اما x ها بصورت موازی تولید می شوند تبدیل z به x سریعتر است اما تبدیل x به z زمانبر است.

زیر بخش سوم

1. از معیار دقت (Accuracy) استفاده نمی شود؛ زیرا دیتاست های Anomaly Detection به شدت نامتوازن هستند (تعداد تصاویر نرمال بسیار بیشتر از تصاویر ناهمجارت). دقت بالا در این حالت لزوماً نشان دهنده عملکرد خوب در تشخیص ناهمجارت نیست.

معیار هایی که استفاده می شوند: AUROC (Area Under the Receiver Operating Characteristic) و AUPRC (Area Under the Precision-Recall Curve) (Curve

2. Anomaly Score یک مقدار عددی است که میزان غیرعادی بودن یک نمونه را نشان می دهد. هر چه این مقدار بالاتر باشد، احتمال ناهمجارت بودن آن نمونه بیشتر است.

در مدل های Normalizing Flow، نمره ناهمجارت معمولاً به عنوان Negative Log Likelihood تصویر ورودی تحت مدل آموزش دیده در نظر گرفته می شود.

$$Anomaly score = - \log P_X(x)$$

3. مدل NF روی تصاویر نرمال آموزش داده می شود. هدف، یادگیری توزیع این تصاویر است. در زمان تست، برای یک تصویر ورودی تصویر محاسبه می شود. از آنجایی که مدل NF بر روی داده های نرمال آموزش دیده، انتظار می رود که تصاویر نرمال دارای NLL پایین و تصاویر ناهمجارت دارای NLL بالا باشند. این NLL به عنوان Anomaly Score استفاده می شود.

$$4. AUROC = 0.7782 \quad ARPRC = 0.9422$$

5. خیر. NF ها مدل های دقیق چگالی هستند که برای نگاشت x به z و بالعکس طراحی شده اند، و هدف اصلی آن ها بازسازی تصویر نیست. در NF ها، معمولاً تبدیل معکوس $X \rightarrow Z^{-1}$ برای تولید تصویر استفاده می شود، اما این عمل یک فرایند بازسازی از فضای پنهان نیست، بلکه نمونه برداری برای توزیع اصلی است. از طرفی، برخلاف VAE که یک مدل مولد-بازسازنده است، NF ها ذاتاً یک مکانیزم بازسازی برای محاسبه اختلاف ندارند. امتیاز ناهمجارت در NF بر اساس احتمال (Likelihood) تصویر تحت توزیع یادگرفته شده محاسبه می شود، نه بر اساس خطای بازسازی.

سؤال دوم

بخش اول

زیر بخش اول)

1. چالش اصلی، نیاز به داده‌های جفت‌شده (Paired Data) است. برای مثال، برای تبدیل یک عکس به نقاشی، Pix2Pix نیاز به مجموعه‌ای از عکس‌ها و نقاشی‌های متناظر دارد که دقیقاً محتوای هندسی یکسانی را نمایش دهد. جمع آوری این داده‌ها در خیلی از موارد خیلی سخت و حتی غیر ممکن است مثل تصاویری که ونگوگ نقاشی کرده باشد.

2. ایده اصلی cycleGAN برای رفع این محدودیت قید سازگاری چرخه‌ای است. CycleGAN برای دور زدن محدودیت داده‌های جفت‌شده، از دو نگاشت (مولد) $F: Y \rightarrow X$ و $G: X \rightarrow Y$ استفاده می‌کند. قید سازگاری چرخه‌ای بیان می‌کند که اگر یک تصویر x از دامنه X به دامنه Y تبدیل کنیم (تبدیل $G(x) \rightarrow x$) و سپس تصویر حاصل را دوباره به دامنه X برگردانیم (تبدیل $(G(x)) \rightarrow F(G(x))$ ، تصویر نهایی باید به تصویر اولیه x بسیار شبیه باشد. صورت بندی ریاضی (این عبارت به زیان اصلی GAN اضافه می‌شود):

$$Loss_{cycle}(G, F) = E_{x \sim P_{data}(x)} [\|F(G(x)) - x\|_1] + E_{y \sim P_{data}(y)} [\|G(F(y)) - y\|_1]$$

3. این قید حیاتی است زیرا به تهایی، آموزش تخاصمی می‌تواند نگاشتهایی را پیدا کند که توزیع تصاویر تولیدی را به توزیع هدف نزدیک کند، اما تضمین نمی‌کند که محتوای اصلی و ساختار هندسی تصویر حفظ شود. با حذف این قید، مدل ممکن است یک Mode Collapse را یاد بگیرد که در آن همه تصاویر ورودی به یک زیرمجموعه کوچک از فضای تصویر هدف نگاشت می‌شوند. به عبارت دیگر، مولد می‌تواند بدون حفظ محتوای ورودی، هر تصویری را به یک تصویر هدف که فربیندهنده Discriminator است، تبدیل کند (مثلاً همه اسبهارا به یک گورخر خاص تبدیل کند).

$$4. Loss(G, F, X, Y) = Loss_{GAN}(G, D_Y, X, Y) + Loss_{GAN}(F, D_X, Y, X) + \lambda Loss_{cycle}(G, F)$$

یک ابر پارامتر است که وزن یا اهمیت نسبی قید سازگاری چرخه‌ای ($Loss_{cycle}$) را در برابر زیان تخاصمی ($Loss_{GAN}$) تعیین می‌کند.

تأثیر λ بزرگ: اگر مقدار λ بیش از حد بزرگ انتخاب شود، مدل بر روی ارضای قید سازگاری چرخه‌ای مرکز می‌شود. این امر باعث می‌شود که نگاشتهای G و F تقریباً به نگاشتهای همانی (Identity Mappings) تبدیل شوند تا خطای خطا را به حداقل برسانند. در نتیجه، خروجی مدل (تصاویر تبدیل شده) تغییر سبک کافی را نشان خواهد داد و شبیه تصاویر ورودی باقی می‌مانند.

5. زیان همانی تضمین می‌کند که اگر یک تصویر از دامنه هدف (مثلاً Y) به عنوان ورودی به مولد تبدیل کنند و به آن دامنه $(G: X \rightarrow Y)$ داده شود، آن تصویر به صورت ایده‌آل بدون تغییر باقی بماند. صورت بندی ریاضی:

$$Loss_{id}(G, F) = E_{x \sim P_{data}(x)} [\|F(x) - x\|_1] + E_{y \sim P_{data}(y)} [\|G(y) - y\|_1]$$

فرض کنید می‌خواهیم اسبهای قهوه‌ای را به گورخرهای رامراه تبدیل کنیم. اگر از تصویر یک اسب قهوه‌ای به عنوان ورودی به مولد $F: Y \rightarrow X$ (تبدیل گورخر به اسب) استفاده شود، انتظار می‌رود که رنگ و بافت تغییر نکند و تصویر همان اسب قهوه‌ای باقی بماند. زیان همانی از بروز تغییرات غیرضروری در رنگ‌های ورودی جلوگیری می‌کند. بدون این زیان، مولد ممکن است یاد بگیرد که رنگ‌های ورودی را به طور دلخواه تغییر دهد (مثلاً رنگ

اسب قهوه‌ای را به آبی تغییر دهد) تا تبدیلش به دامنه هدف بهتر به نظر برسد. این زیان تضمین می‌کند که فقط ویژگی‌های مرتبط با سبک و بافت دامنه هدف تغییر کند.

زیر بخش دوم

1. دلایل تئوری: تبدیل تصویر به تصویر، دو نگاشت متقاوت را شامل می‌شود: $Y \rightarrow X$ و $X \rightarrow Y$. این دو نگاشت می‌توانند سیار متفاوت باشند. برای مثال، تبدیل "عکس به نقاشی" با "نقاشی به عکس" از لحاظ پیچیدگی یادگیری نگاشت‌ها یکسان نیستند. استفاده از دو مولد G برای $Y \rightarrow X$ و F برای $X \rightarrow Y$ امکان یادگیری نگاشت‌های مستقل و بهینه‌تر را برای هر جهت فراهم می‌کند.

دلایل عملی: قید سازگاری چرخه‌ای به صراحت به دوتابع نیاز دارد تا یک چرخه بسته ($x \approx F(G(x))$) را ایجاد کند. استفاده از یک مولد واحد برای هر دو جهت، انعطاف‌پذیری مدل را برای یادگیری نگاشت‌های پیچیده در هر دو جهت محدود می‌کند.

2. Pix2Pix: به دلیل استفاده از داده‌های جفت‌شده (Supervised)، مولد Pix2Pix از معماری U-Net استفاده می‌کند. U-Net از اتصالات پرشی (Skip Connections) استفاده می‌کند که اطلاعات با فرکانس بالا (High-frequency information) را مستقیماً از لایه‌های رمزگذار (Encoder) به لایه‌های رمزگشای (Decoder) منتقل می‌کند. این امر برای حفظ جزئیات دقیق هندسی بین تصویر ورودی و هدف جفت‌شده حیاتی است.

CycleGAN: به دلیل استفاده از داده‌های جفت‌نشده (Unpaired)، مولد CycleGAN از معماری ResNet-based (شامل بلوك‌های Residual) استفاده می‌کند. این معماری به مدل اجازه می‌دهد تا بر روی تغییرات سبک (Style/Texture) تمرکز کند، در حالی که ساختار کلی تصویر حفظ می‌شود. فاقد اتصالات پرشی مستقیم U-Net است که برای حفظ جزئیات دقیق، نیاز به سختگیری ResNet Pix2Pix را ندارد، بلکه بر روی تبدیل‌های سطح بالاتر (Higher-level transformations) تمرکز دارد.

3. به جای اینکه تمیزدهنده کل تصویر را به عنوان "واقعی" یا "جهلی" دست‌بندی کند، PatchGAN تصویر را به وصله‌های کوچک (N^*N پیکسل) تقسیم می‌کند و برای هر وصله یک خروجی "واقعی/جهلی" تولید می‌کند. خروجی PatchGAN یک ماتریس N^*N است که هر عنصر آن نشان‌دهنده نظر تمیزدهنده درباره واقعی بودن یک وصله از تصویر ورودی است. در نهایت، میانگین این خروجی‌ها برای تعیین نهایی تصویر استفاده می‌شود. این روش باعث می‌شود که مدل بر روی حفظ بافت و جزئیات با فرکانس بالا تمرکز کند و در عین حال پارامترهای کمتری نسبت به یک تمیزدهنده کامل داشته باشد.

4. در آموزش تمیزدهنده، به جای استفاده مستقیم از آخرین تصاویر تولید شده توسط مولد، از یک بافر برای نگهداری تصاویری که مولد در طول زمان تولید کرده است، استفاده می‌شود. تمیزدهنده با ترکیبی از تصاویر جدید و تصاویر قدیمی (مثلًا ۵ تا ۰ تصویر) آموزش داده می‌شود. این مکانیزم مشکل نوسان در فرآیند آموزش GAN را حل می‌کند. بدون بافر، تمیزدهنده ممکن است خیلی سریع به آخرین تغییرات مولد عادت کند و در مقابل، مولد نیز ممکن است به سرعت یک تصویر واحد را یاد بگیرد که تمیزدهنده فعلی را فریب دهد، که منجر به ناپایداری و واگرایی می‌شود. بافر باعث می‌شود که تمیزدهنده با یک مجموعه متنوع‌تر و تاریخی از تصاویر جعلی مواجه شود و آموزش آن پایدارتر گردد.

زیر بخش سوم

1. CycleGAN در وظایفی که نیاز به تغییر شکل‌های بزرگ در ساختار هندسی اصلی تصویر دارند، مانند تبدیل عکس پرتره به عکس تمام قد یا تغییرات زاویه دید چشمگیر، عملکرد ضعیفی دارد. این مدل بیشتر بر روی تغییر سبک، بافت و رنگ تمرکز دارد. قید سازگاری چرخه‌ای ($x \approx F(G(x))$) برای $G(x) \rightarrow F(G(x))$ تلاش می‌کند تا اطمینان حاصل کند که اطلاعات محتوای ورودی حفظ شود. اگر مولد، تغییرات هندسی بزرگی را یاد بگیرد (مانند تبدیل یک اسب به یک اسب کاملاً متفاوت با حالت بدنی متفاوت)، نگاشت

معکوس (F) به سختی می‌تواند تصویر اصلی را بازسازی کند. به عبارت دیگر، قید سازگاری چرخه‌ای به عنوان یک جریمه برای هر گونه تغییر هندسی عمدی که غیرقابل بازگشت باشد، عمل می‌کند و از یادگیری چنین نگاشتهایی جلوگیری می‌کند.

بخش دوم

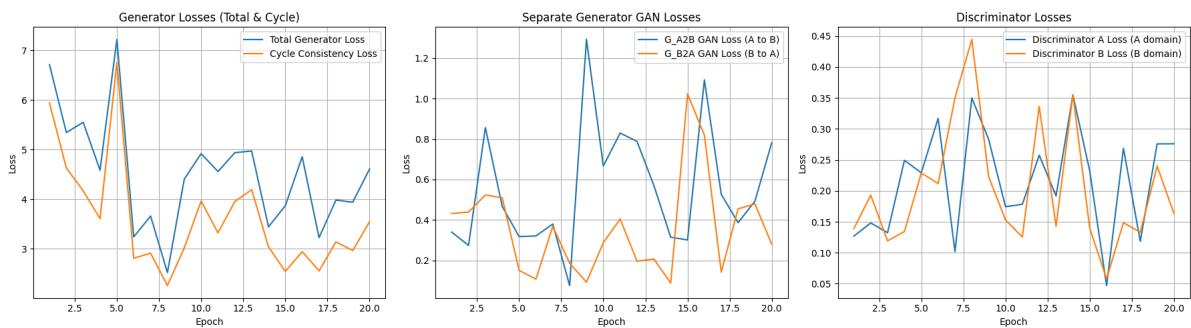
(زیر بخش اول)



زیر بخش سوم

ابر پارامترها همان ابر پارامترهای مقاله هستند و تغییری روی آن ها اعمال نکردم.

```
IMAGE_SIZE = 128
NUM_RESIDUAL_BLOCKS = 6
BATCH_SIZE = 1
LR = 0.0002
BETA1 = 0.5
LAMBDA_CYCLE = 10.0
LAMBDA_IDENTITY = 0.5 * LAMBDA_CYCLE
NUM_EPOCHS = 20
```



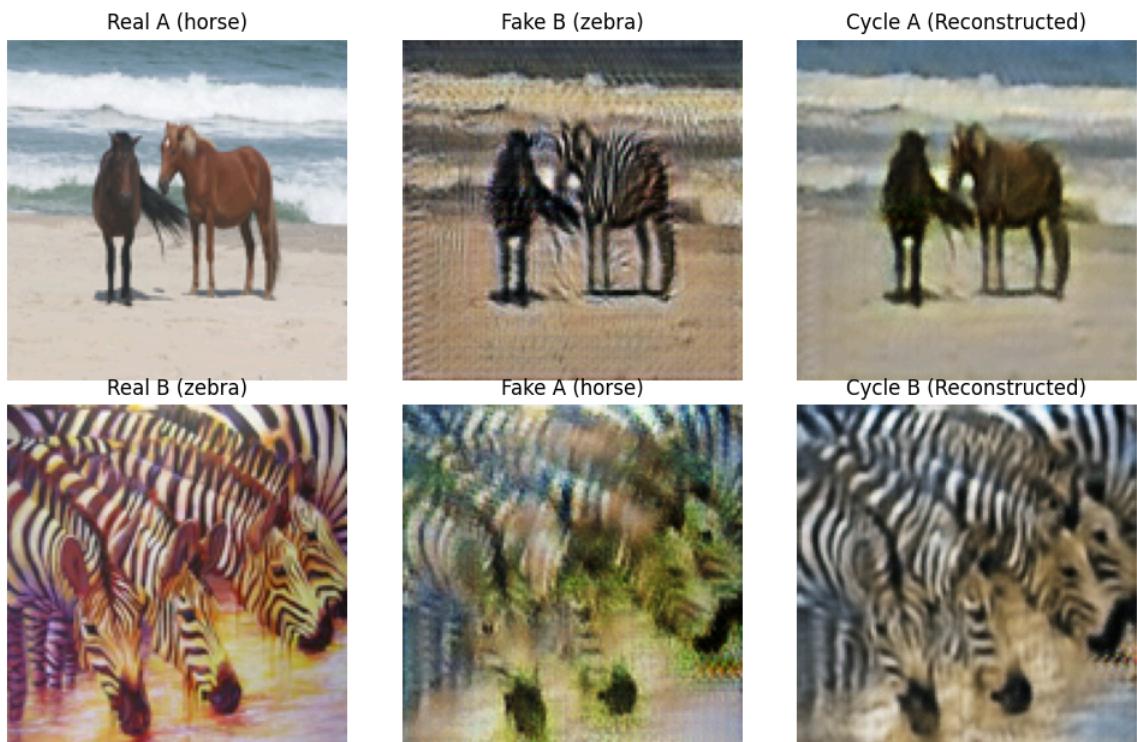
نکته: اسب است و گُوره خر.

همانگونه که مشاهده می شود آموزش مولد ها شبیه به هم نیست و علاوه بر آن پایدار هم نیست چرا که در این 20 دور مدام در حال نوسان هستند و قابل پیش بینی هم بود. طبقبند ها هم در ابتدا کار راحتی داشتند اما در دور هشتم زیان آنها خیلی افزایش یافت چرا که مولد ها در آن حوالی به زیان کمی رسیدند. یک جور می توان گفت که زیان مولد ها و طبقبند ها بر عکس هم است. زیان هماهنگی هم بسیار شبیه به زیان مولد ها است چرا که مسئولیت آن بر عهده آنهاست و تغییراتش شبیه به تغییرات مجموع آنهاست.

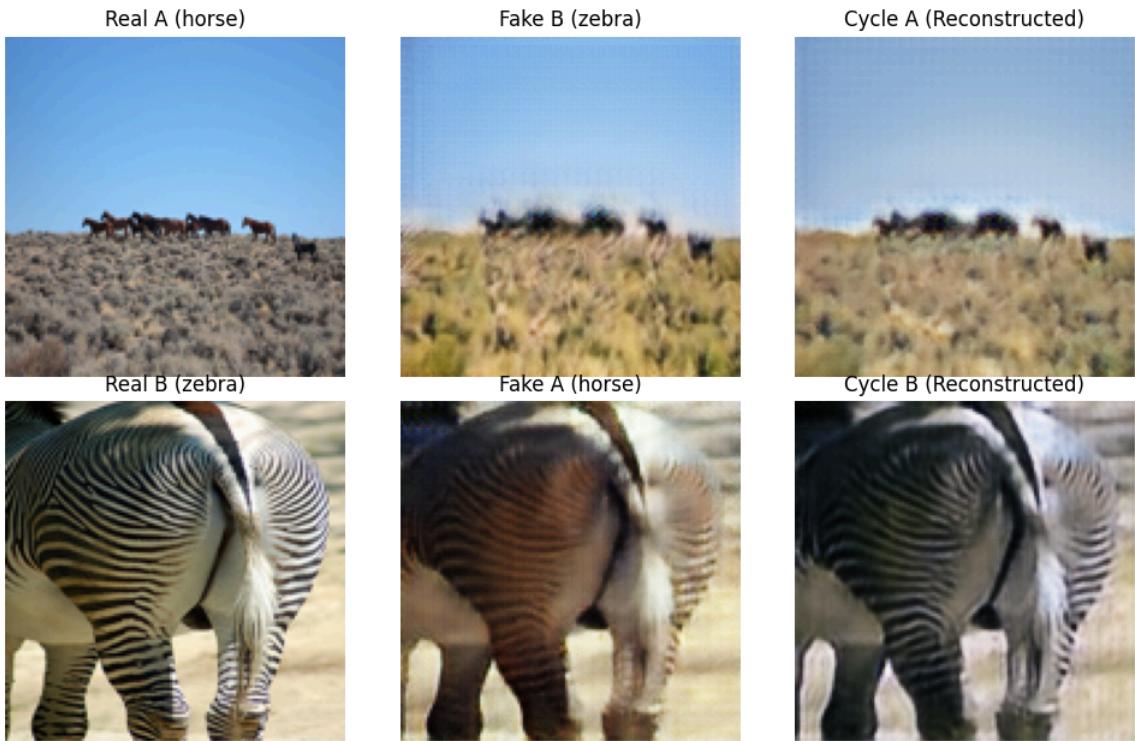
CycleGAN Results - Epoch 5



CycleGAN Results - Epoch 10



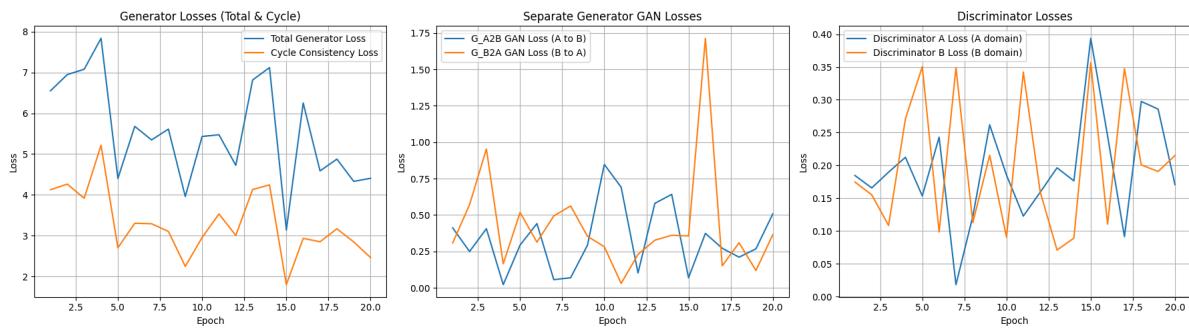
CycleGAN Results - Epoch 20



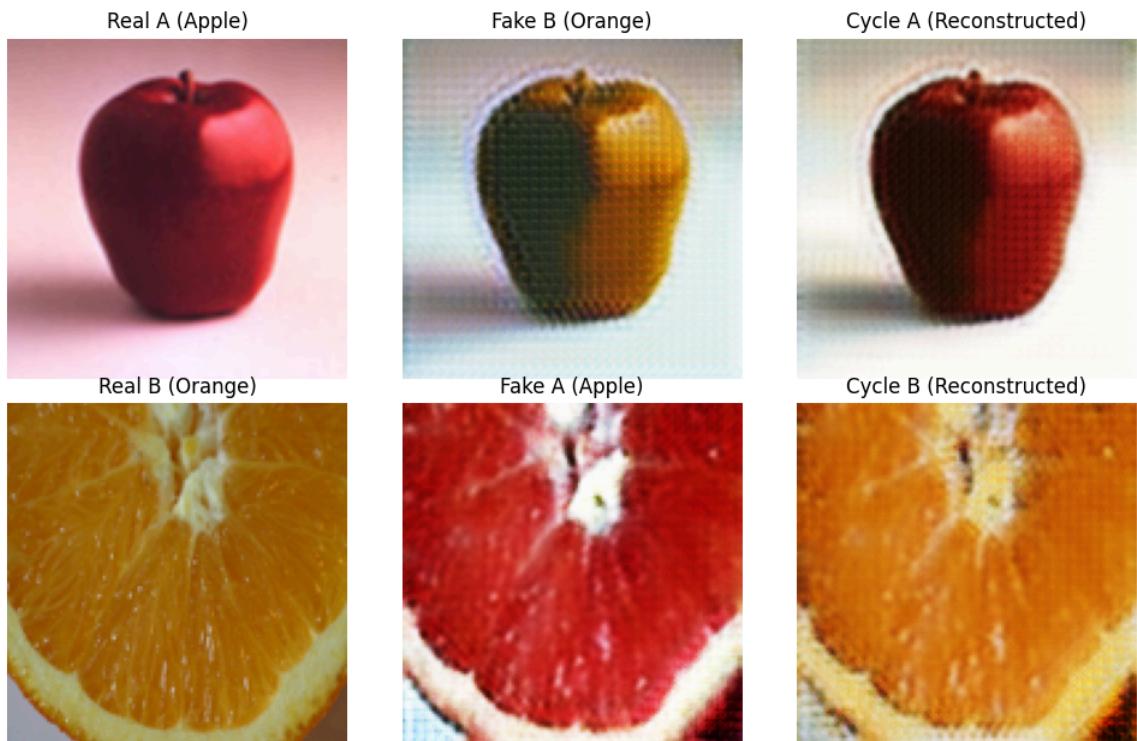
مدل در ابتدا فقط به رنگ توجه می کند و با تغییر آن حتی محیط دور و بر را هم تغییر می دهد اما در ادامه بر این مشکل غلبه می کند و بهبود می یابد. همانطور که از نمودار هم واضح بود بازسازی به مرور بهتر می شود و مدل هم بهتر یاد می گیرد که شیء مورد نظر را بهتر از محیط اطراف جدا کند.

(زیر بخش چهارم)
در تصاویر زیر A سبب است و B پرتوان.

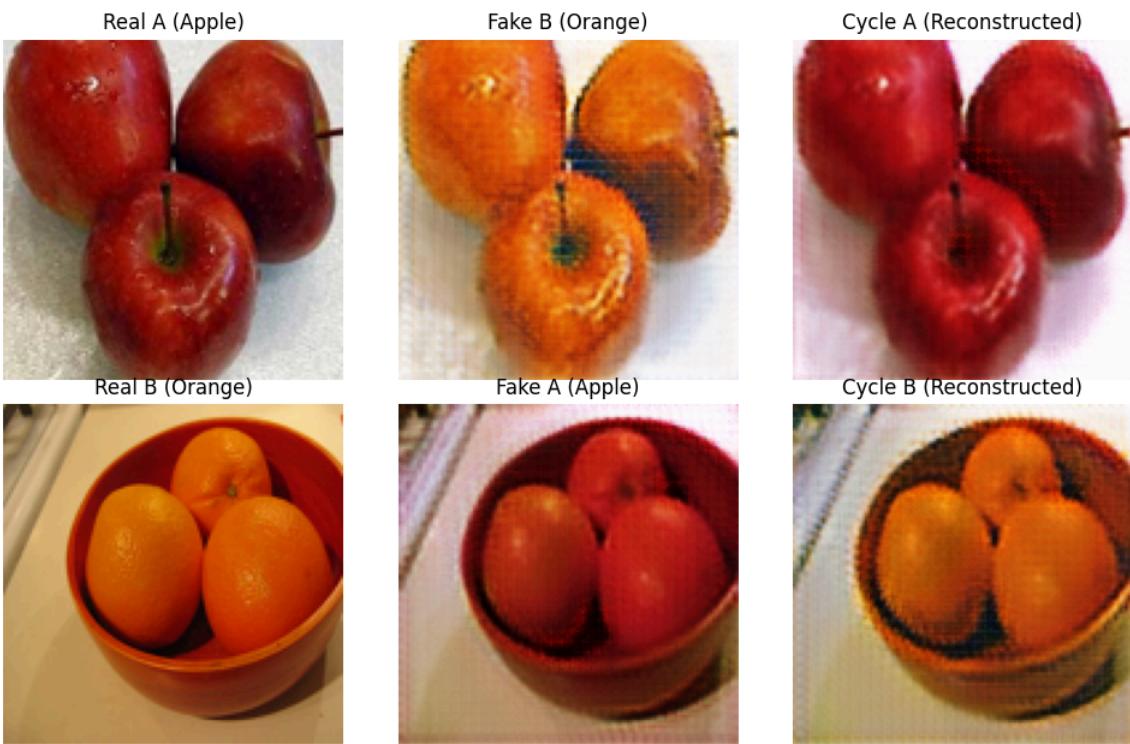
زیان مولدها در این مورد چنان تغییری نکرده است و همچنان در حال نوسان هستند اما فرکانس تغییرات طبقه‌بند ها به شدت بالا رفته و نوسان می کنند این بدین دلیل است که ما هر بار 50 عکس به آن ها می دهیم و تغییرات با عکس های بیشتری اعمال می شوند پس شدت آن ها هم بیشتر است. تصاویر تولیدی هم مانند دفعه قبل اول با رنگ شروع کردند ولی بافت اطراف را هم تغییر می دهند اما تا 200 دور هم ما هنوز شاهد اصلاح این رویه نیستیم.



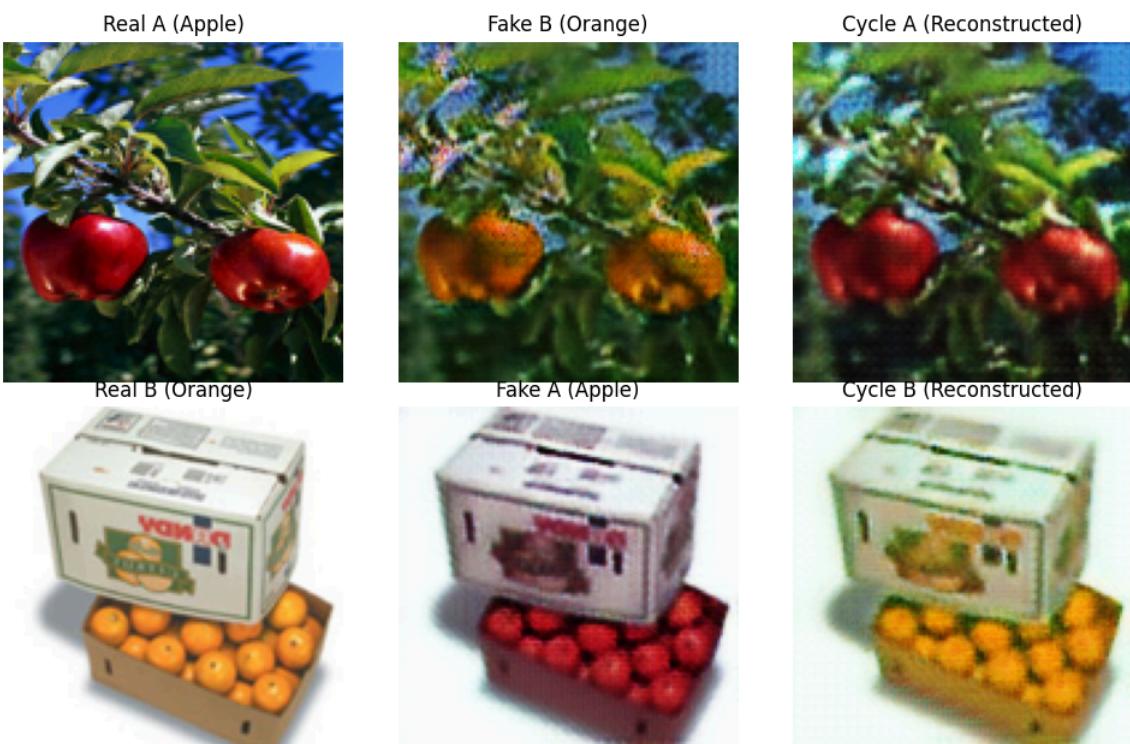
CycleGAN Results - Epoch 5



CycleGAN Results - Epoch 10



CycleGAN Results - Epoch 20



مراجع