



# EAI 6020: AI System Technologies

Week 6:  
FACE AGING USING Cycle-GAN  
(Cycle-Generative adversarial networks)

Submitted To:  
Prof. Siddharth Rout, Faculty Lecturer

Submitted By:  
Abhilash Dikshit  
Murtaza Vora  
Gunjan Paladiya

Academic Term: Winter 2024  
Northeastern University, Vancouver, BC, Canada  
Master of Professional Studies in Analytics

March 30, 2024

## FACE AGING USING Cycle-GAN (Cycle-Generative adversarial networks)

### I. Abstract:

The study explores the potential of using CycleGAN for age progression, a technique that aesthetically renders facial images to simulate the effect of aging. Different datasets and components of datasets were examined, and various techniques like hyperparameter tuning, fine-tuning, and transfer learning were employed to expedite the training process. The performance of the models was evaluated both quantitatively and qualitatively.

### II. Introduction:

- **Purpose:** Age progression is essential for various applications, including entertainment and forensic science.
- **Challenges:** The complexity arises from variations in facial expressions, photographic settings, and the diverse physical environments people grow up in.
- **Solution:** A deep learning model based on CycleGAN is proposed to generate aging effects without the need for paired data.

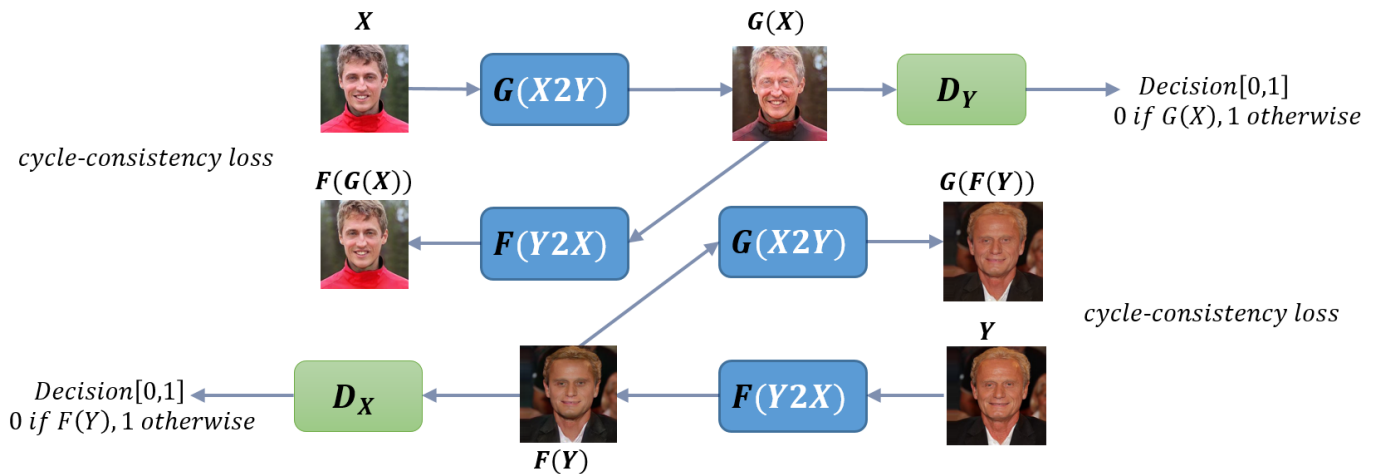
### III. Related Work:

- **Historical Approaches:** Earlier methods focused on specific facial features, while recent works adopted more holistic approaches.
- **Advancements:** Generative Adversarial Nets (GANs) have shown promise due to their simplicity, implementation ease, and less stringent training requirements.

### IV. Dataset and Features:

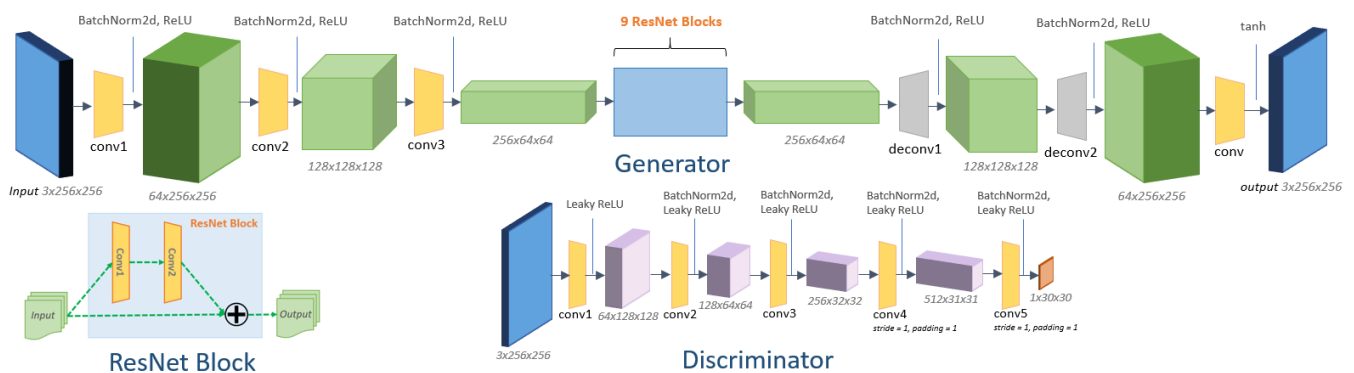
- **Datasets Explored:**
  - IMDB-WIKI: 500k+ face images with age and gender labels.
  - Cross-Age Celebrity Dataset (CACD): 163k+ images of 2,000 celebrities.
- **Data Processing:**
  - IMDB-WIKI: Only Wikipedia's 62k images were used after filtering.

- CACD: 2,200 images were randomly selected for each age group.



## V. Methods and Implementations:

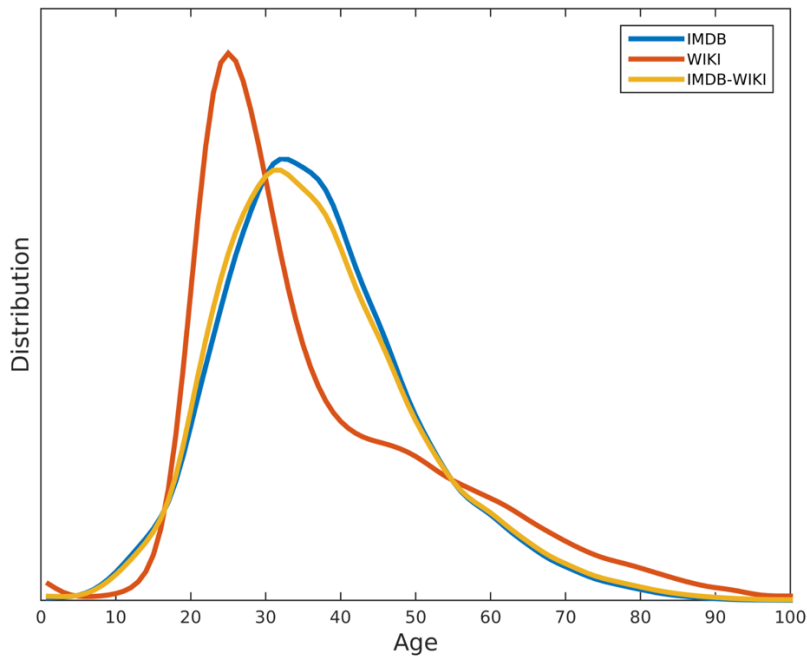
- **Model Structure:** CycleGAN consists of two generators ( $G$ ,  $F$ ) and two discriminators ( $D_X$ ,  $D_Y$ ).
- **Implementation:**
  - Generators: Encoder, Transformer (9 ResNet Blocks), Decoder.
  - Discriminator: 5 down sampling layers.
- **Objective:** Minimize the adversarial losses and cycle consistency losses to ensure meaningful mappings between domains.



## VI. Experiments and Results:

- **Data Source Analysis:**
  - IMDB-WIKI outperformed CACD significantly, likely due to the professional

settings of CACD images.



## - Data Composition Analysis:

- Gender separation improved performance and eliminated gender-biased traits.

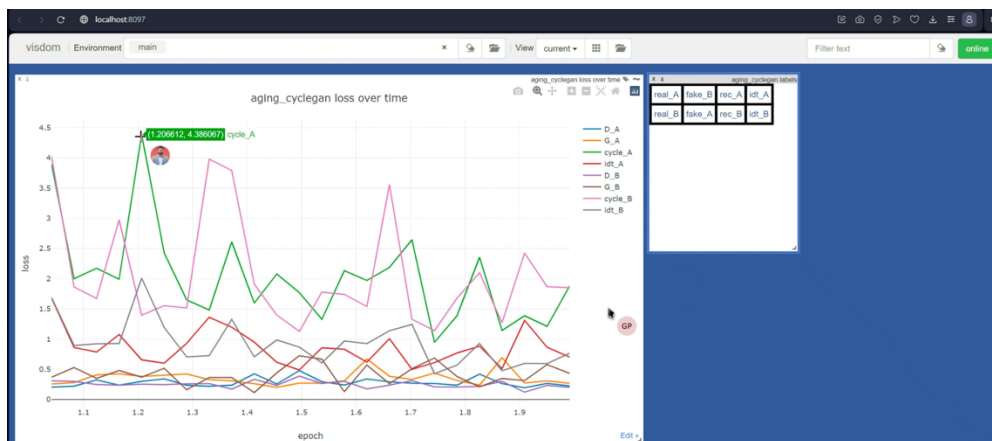
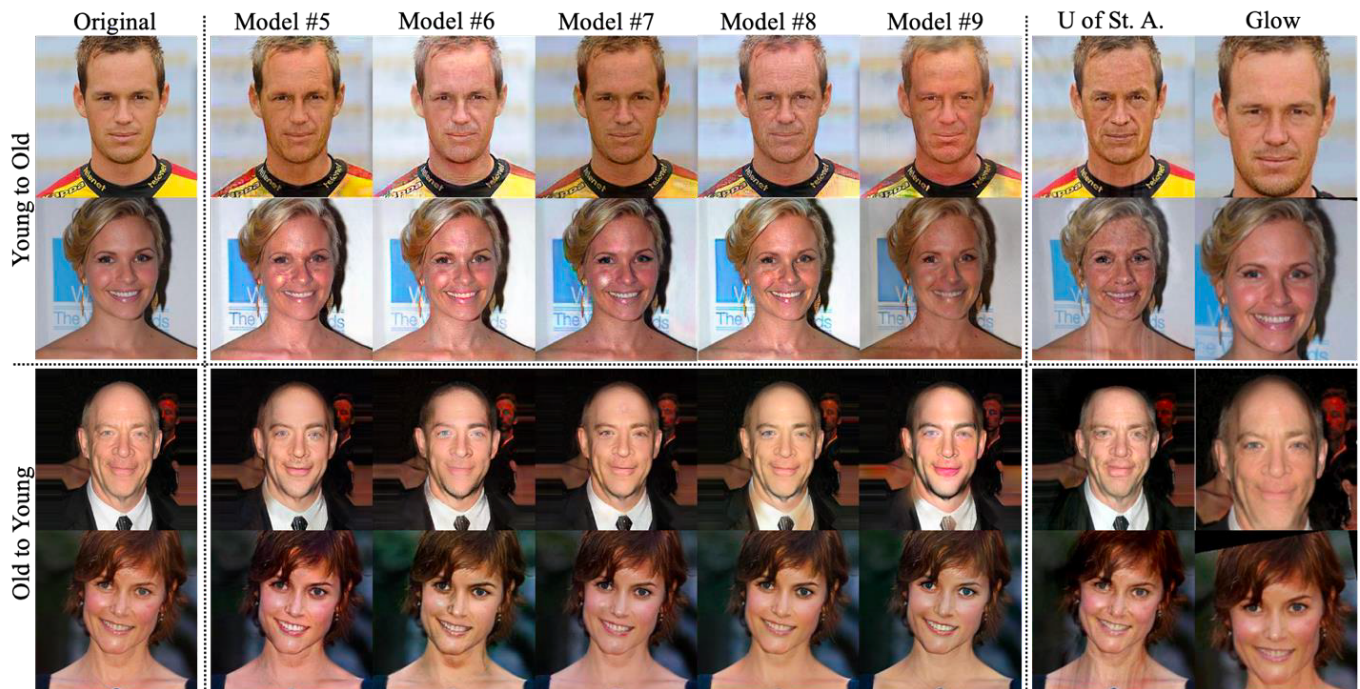
## - Speed-up Techniques:

- Hyperparameter tuning, transfer learning, and fine-tuning were explored to reduce training time without compromising quality.

```

MINGW64/g/faceaging-by-cycleGAN
no_isgan: False
norm: instance
num_threads: 4
output_nc: 3
phase: train
pool_size: 50
pretrained_model_epoch: 1
pretrained_model_name:
pretrained_model_subname:
print_freq: 100
resize_or_crop: resize_and_crop
save_epoch_freq: 5
save_latest_freq: 5000
serial_batches: False
suffix:
update_html_freq: 1000
use_pretrained_model: False
verbose: False
----- End -----
dataset [UnalignedDataset] was created
training images = 2420
initialize network with normal
initialize network with normal
initialize network with normal
initialize network with normal
model [cycleGANmodel] was created
----- Networks initialized -----
Network_G_A Total number of parameters : 11.378 M
Network_G_A Total number of trainable parameters : 11.378 M
Network_G_B Total number of parameters : 11.378 M
Network_G_B Total number of trainable parameters : 11.378 M
Network_D_A Total number of parameters : 2.765 M
Network_D_A Total number of trainable parameters : 2.765 M
Network_D_B Total number of parameters : 2.765 M
Network_D_B Total number of trainable parameters : 2.765 M
-----
create web directory ./checkpoints/aging-cycleGAN/web...
epoch: 1, iters: 100, time: 0.300, data: 8.396 G_A: 0.204 G_A: 0.257 cycle_A: 3.896 idt_A: 1.679 D_B: 0.310 G_B: 0.368 cycle_B: 4.026 idt_B: 1.690
epoch: 1, iters: 200, time: 0.291, data: 0.000 D_A: 0.222 G_A: 0.220 cycle_A: 1.999 idt_A: 0.862 D_B: 0.300 G_B: 0.350 cycle_B: 1.862 idt_B: 0.894
epoch: 1, iters: 300, time: 0.320, data: 0.000 D_A: 0.333 G_A: 0.407 cycle_A: 2.172 idt_A: 0.785 D_B: 0.248 G_B: 0.350 cycle_B: 1.673 idt_B: 1.190
epoch: 1, iters: 400, time: 0.390, data: 0.000 D_A: 0.237 G_A: 0.423 cycle_A: 1.991 idt_A: 1.075 D_B: 0.238 G_B: 0.481 cycle_B: 2.973 idt_B: 1.193
epoch: 1, iters: 500, time: 0.423, data: 0.000 D_A: 0.289 G_A: 0.382 cycle_A: 4.386 idt_A: 0.656 D_B: 0.253 G_B: 0.369 cycle_B: 1.395 idt_B: 1.193
epoch: 1, iters: 600, time: 0.330, data: 0.000 D_A: 0.342 G_A: 0.407 cycle_A: 2.426 idt_A: 0.602 D_B: 0.246 G_B: 0.517 cycle_B: 1.533 idt_B: 1.193
epoch: 1, iters: 700, time: 0.337, data: 0.000 D_A: 0.237 G_A: 0.423 cycle_A: 1.648 idt_A: 0.931 D_B: 0.260 G_B: 0.162 cycle_B: 1.517 idt_B: 0.705
epoch: 1, iters: 800, time: 0.434, data: 0.000 D_A: 0.219 G_A: 0.329 cycle_A: 1.482 idt_A: 1.363 D_B: 0.283 G_B: 0.364 cycle_B: 3.978 idt_B: 0.727
epoch: 1, iters: 900, time: 0.320, data: 0.000 D_A: 0.236 G_A: 0.310 cycle_A: 2.608 idt_A: 1.197 D_B: 0.173 G_B: 0.365 cycle_B: 3.791 idt_B: 1.325
epoch: 1, iters: 1000, time: 0.338, data: 0.000 D_A: 0.424 G_A: 0.261 cycle_A: 1.596 idt_A: 0.950 D_B: 0.335 G_B: 0.114 cycle_B: 1.210 idt_B: 0.707
epoch: 1, iters: 1100, time: 0.342, data: 0.001 D_A: 0.257 G_A: 0.201 cycle_A: 2.077 idt_A: 0.616 D_B: 0.233 G_B: 0.441 cycle_B: 1.398 idt_B: 0.985
epoch: 1, iters: 1200, time: 0.706, data: 0.000 D_A: 0.478 G_A: 0.272 cycle_A: 1.769 idt_A: 0.489 D_B: 0.389 G_B: 0.724 cycle_B: 1.128 idt_B: 0.868
epoch: 1, iters: 1300, time: 0.351, data: 0.000 D_A: 0.235 G_A: 0.270 cycle_A: 1.228 idt_A: 0.855 D_B: 0.264 G_B: 0.668 cycle_B: 1.777 idt_B: 0.605
epoch: 1, iters: 1400, time: 0.338, data: 0.001 D_A: 0.240 G_A: 0.308 cycle_A: 2.133 idt_A: 0.832 D_B: 0.301 G_B: 0.130 cycle_B: 1.740 idt_B: 0.968
epoch: 1, iters: 1500, time: 0.332, data: 0.000 D_A: 0.340 G_A: 0.669 cycle_A: 1.971 idt_A: 0.619 D_B: 0.176 G_B: 0.570 cycle_B: 1.540 idt_B: 0.926
epoch: 1, iters: 1600, time: 0.689, data: 0.000 D_A: 0.292 G_A: 0.384 cycle_A: 2.186 idt_A: 1.006 D_B: 0.238 G_B: 0.262 cycle_B: 3.157 idt_B: 1.137
epoch: 1, iters: 1700, time: 0.334, data: 0.000 D_A: 0.269 G_A: 0.332 cycle_A: 2.645 idt_A: 0.501 D_B: 0.312 G_B: 0.509 cycle_B: 1.328 idt_B: 1.243
epoch: 1, iters: 1800, time: 0.320, data: 0.000 D_A: 0.264 G_A: 0.437 cycle_A: 1.946 idt_A: 0.610 D_B: 0.209 G_B: 0.686 cycle_B: 1.140 idt_B: 0.427
epoch: 1, iters: 1900, time: 0.329, data: 0.000 D_A: 0.336 G_A: 0.316 cycle_A: 1.386 idt_A: 0.770 D_B: 0.206 G_B: 0.382 cycle_B: 1.677 idt_B: 0.569
epoch: 1, iters: 2000, time: 0.614, data: 0.001 D_A: 0.420 G_A: 0.238 cycle_A: 2.353 idt_A: 0.882 D_B: 0.217 G_B: 0.210 cycle_B: 2.097 idt_B: 0.927
epoch: 1, iters: 2100, time: 0.331, data: 0.000 D_A: 0.782 G_A: 0.691 cycle_A: 1.142 idt_A: 0.511 D_B: 0.301 G_B: 0.348 cycle_B: 1.270 idt_B: 0.476
epoch: 1, iters: 2200, time: 0.331, data: 0.001 D_A: 0.195 G_A: 0.274 cycle_A: 1.387 idt_A: 1.309 D_B: 0.122 G_B: 0.314 cycle_B: 2.425 idt_B: 0.598
epoch: 1, iters: 2300, time: 0.326, data: 0.001 D_A: 0.266 G_A: 0.308 cycle_A: 1.211 idt_A: 0.864 D_B: 0.235 G_B: 0.578 cycle_B: 1.867 idt_B: 0.593
epoch: 1, iters: 2400, time: 0.600, data: 0.003 D_A: 0.225 G_A: 0.287 cycle_A: 1.879 idt_A: 0.701 D_B: 0.201 G_B: 0.493 cycle_B: 1.850 idt_B: 0.746
end of epoch 1 / 1 Time Taken: 692 sec
learning rate = 0.0000000
mingw64 MINGW64 /g/faceaging-by-cycleGAN (master)

```



#	Source	Mix	Epochs	Preloaded?	Freeze until	G Size	Max	Avg	10+	15+	20+
0	CACD	All	200	N/A	N/A	9 blocks	25.8	6.7	22%	5.5%	1.7%
1	WIKI	All	200	N/A	N/A	9 blocks	31.2	8.8	37%	14%	5.8%
2	WIKI	Female	200	N/A	N/A	9 blocks	19.5	4.6	7.1%	2.5%	0.0%
3	WIKI	Male	200	N/A	N/A	9 blocks	27.3	10.3	50%	19%	5.1%
4	WIKI	Male	200	N/A	N/A	6 blocks	N/A	N/A	N/A	N/A	N/A
5	WIKI	All	200	horse2zebra	8th block	9 blocks	27.4	11.0	55%	20%	6.3%
6	WIKI	All	200	summer2winter	8th block	9 blocks	25.0	8.9	36%	10%	1.7%
7	WIKI	All	200	monet2photo	8th block	9 blocks	20.1	6.6	15%	2.5%	0.4%
8	WIKI	Male	100	horse2zebra	N/A	9 blocks	25.8	9.9	46%	12%	1.3%
9	WIKI	Male	100	Model #2	N/A	9 blocks	32.8	10.3	51%	18%	6.0%

## VII. Conclusion:

In conclusion, the AgingGAN study successfully demonstrated the potential of CycleGAN in generating quality age progression images, addressing a challenging task in the realm of deep learning and computer vision. Through meticulous exploration of datasets, the study found that the IMDB-WIKI dataset outperformed the Cross-Age Celebrity Dataset (CACD), likely due to differences in image settings and professional photography. Additionally, gender-specific data composition proved crucial in enhancing model performance and eliminating gender-biased traits in the generated images. Various techniques, including hyperparameter tuning, transfer learning, and fine-tuning, were explored to expedite the training process without compromising the quality of the aging effects. However, attempts to reduce the size of the generator network proved futile, highlighting the intricate balance between model complexity and performance. Overall, the study offers valuable insights into the optimization of age progression models and underscores the importance of dataset selection and composition in achieving desirable outcomes.

## VIII. References:

- Face aging with conditional generative adversarial networks. (2017, September 1). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/8296650>
- Ai, N. (2022, April 7). GAN for Face Aging problem | Neurond AI | Medium. Medium. <https://neurondai.medium.com/what-does-your-face-look-like-in-the-next-few-years-gans-for-face-aging-problems-8568299adfd>
- Wikipedia contributors. (2024, March 20). Generative adversarial network. Wikipedia. [https://en.wikipedia.org/wiki/Generative\\_adversarial\\_network](https://en.wikipedia.org/wiki/Generative_adversarial_network)
- Jiechen. (n.d.). GitHub - jiechen2358/FaceAging-by-cycleGAN. GitHub. <https://github.com/jiechen2358/FaceAging-by-cycleGAN/tree/master?tab=readme-ov-file>