Photo Recognition And Generative Adversarial Network Based Computational Experiments

This Project has received funding from the European Union' s Horizon 2020 research and innovation programme under grant agreement No 857627 (CIPHR)

J. Amudhavel

SENIOR RESEARCHER

UNIVERSITY OF TARTU

This Project has received funding from the European Union’ s Horizon 2020 research and innovation programme under grant agreement No 857627 (CIPHR)

J. Amudhavel

SENIOR RESEARCHER

Introductory Goals

1. Understand the different types of machine learning.
2. Understand the key concepts of supervised machine learning.
3. Learn how solving problems with ML is different from traditional approaches.

Types of ML Systems

1. ML systems fall into three distinct categories based on how they learn to make predictions:
   - ❖ Supervised learning
   - ❖ Unsupervised learning
   - ❖ Reinforcement learning

Supervised Learning

Supervised learning models can make predictions after seeing lots of data.

Ex: Student Learning.

Supervised Learning

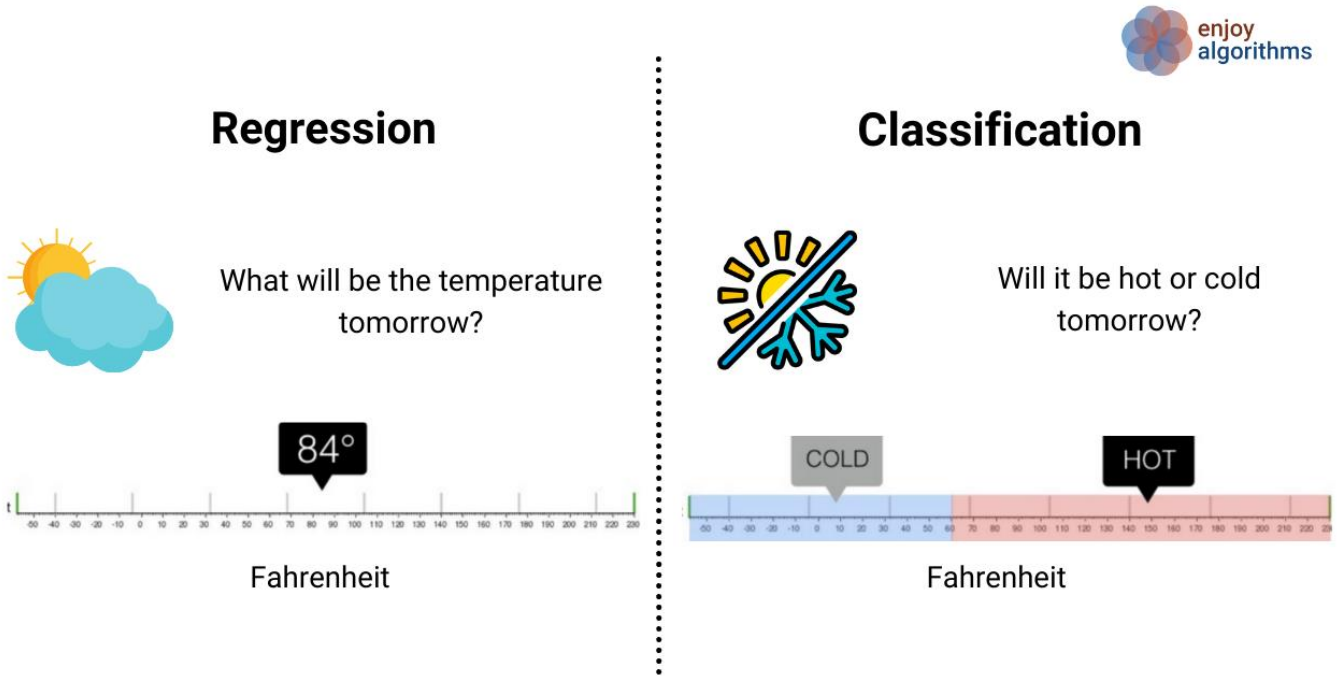
Regression

A regression model predicts a numeric value. For example, a weather model that predicts the amount of rain, in inches or millimeters, is a regression model.

Classification

Classification models predict the likelihood that something belongs to a category.

Supervised Learning



Source: https://www.enjoyalgorithms.com/blogs/classification-and-regression-in-machine-learning

Supervised Learning

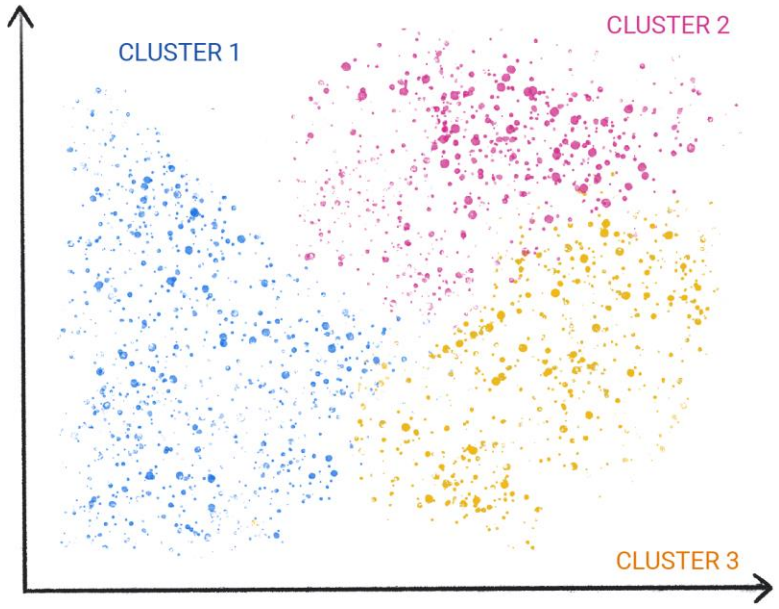
Classification models are divided into two groups:

Binary classification and

Multiclass classification.

Unsupervised Learning

Unsupervised learning model's goal is to identify meaningful patterns among the data.



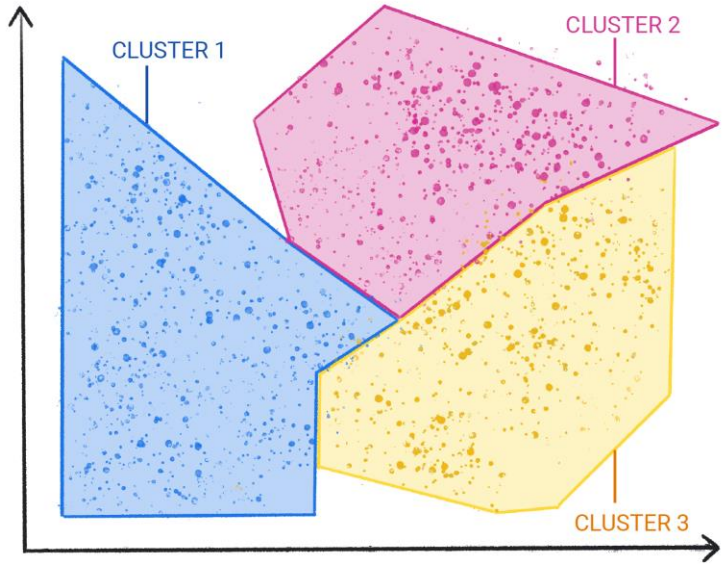
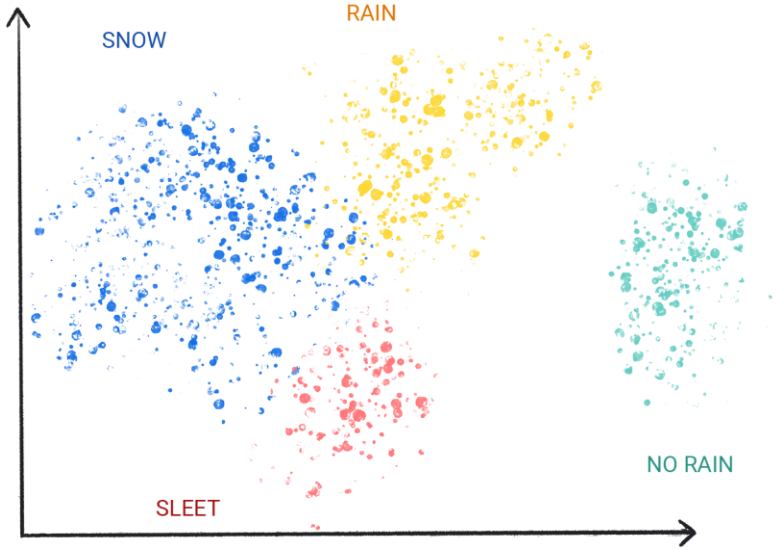


Image source: https://developers.google.com/machine-learning/intro-to-ml/what-is-ml

Unsupervised Learning

Unsupervised learning model's goal is to identify meaningful patterns among the data.



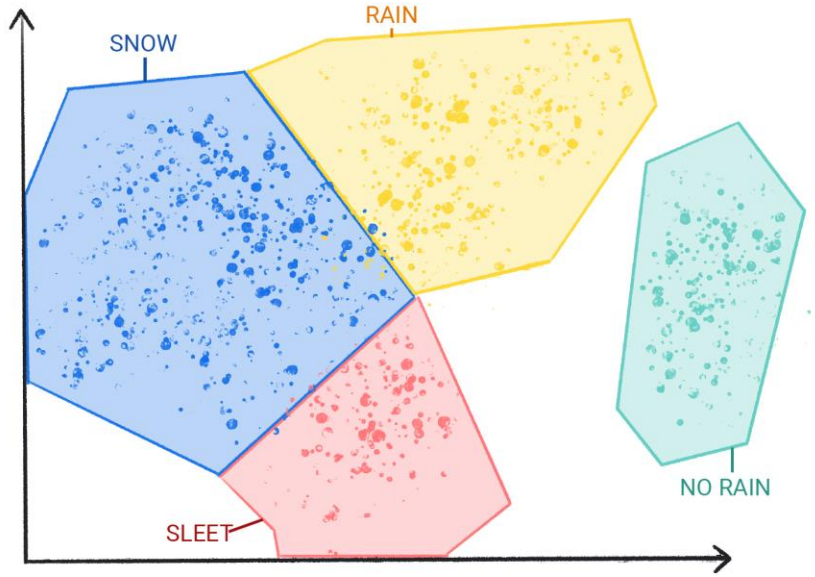
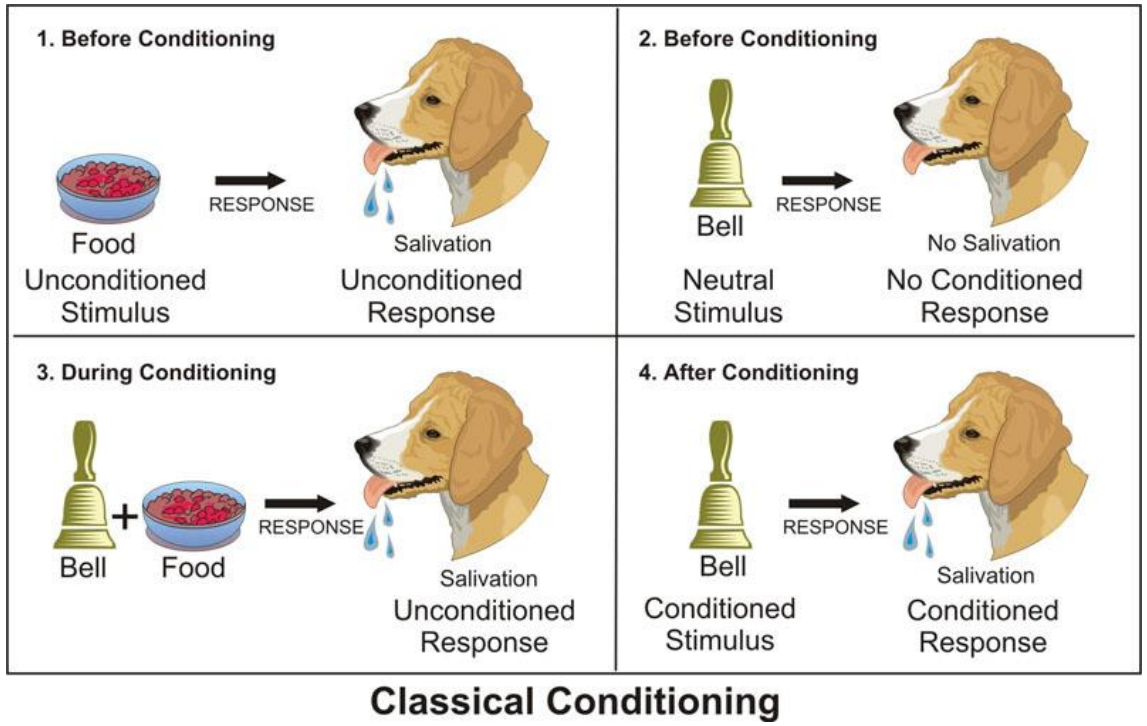


Image source: https://developers.google.com/machine-learning/intro-to-ml/what-is-ml

Reinforcement Learning

Reinforcement learning models make predictions by getting rewards or penalties based on actions performed within an environment.



Classical Conditioning

Image source: https://byronjaustin.files.wordpress.com/2015/05/dog-resized.jpg

Workaround

https://colab.research.google.com/github/google/eng-edu/blob/main/ml/cc/exercises/linear_regression_with_synthetic_data.ipynb?utm_source=mlcc&utm_campaign=colab-external&utm_medium=referral&utm_content=linear_regression_synthetic_tf2-colab&hl=en#scrollTo=_GMGgR6O54IN

https://colab.research.google.com/github/google/eng-edu/blob/main/ml/cc/exercises/linear_regression_with_a_real_dataset.ipynb?utm_source=mlcc&utm_campaign=colab-external&utm_medium=referral&utm_content=linear_regression_real_tf2-colab&hl=en#scrollTo=xRfxp_3yofe3

Intermediate Goals

1. To Understand the difference between generative and discriminative models.
2. Understand the roles of the generator and discriminator in a GAN system.
3. To understand loss functions in GAN Training.
4. Use the TF GAN library to make a GAN. – Hands on

Identify the Celebrities

GAN - Progressive

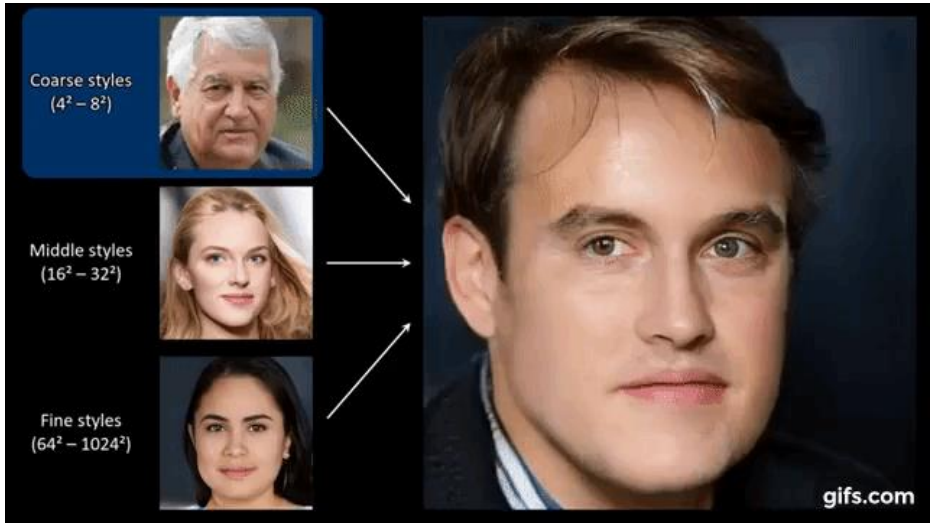
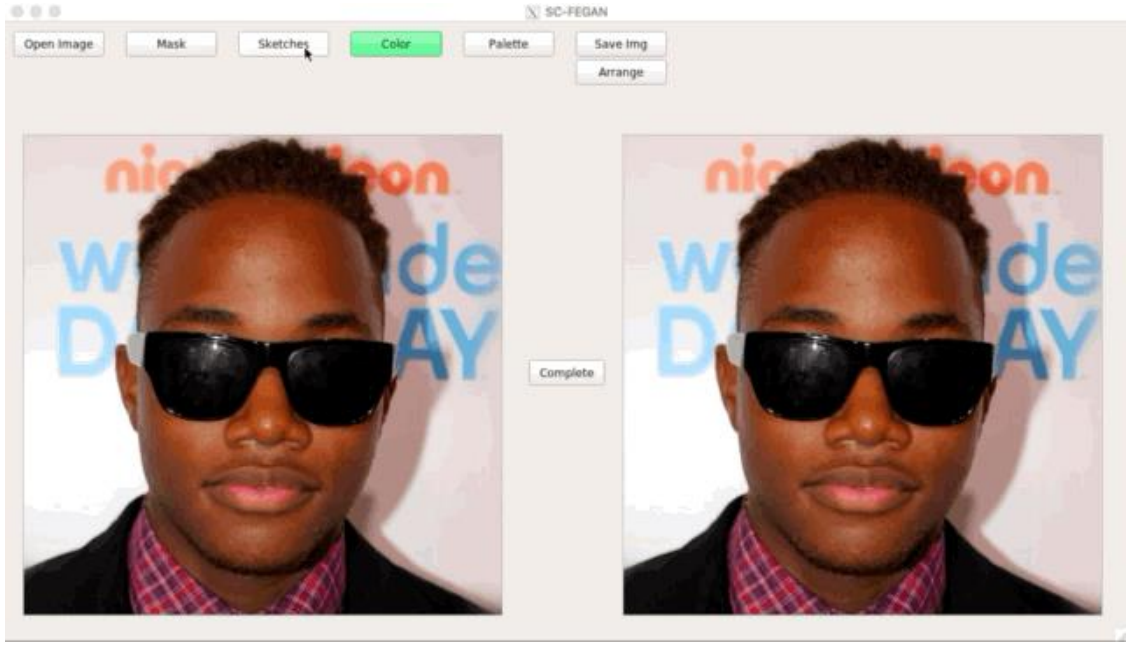


Image Credits: https://github.com/Puzer/stylegan-encoder/issues/1

Image Credits: https://www.outerplaces.com/science/item/19172-artificial-intelligence-faces-gan

GAN - Progressive



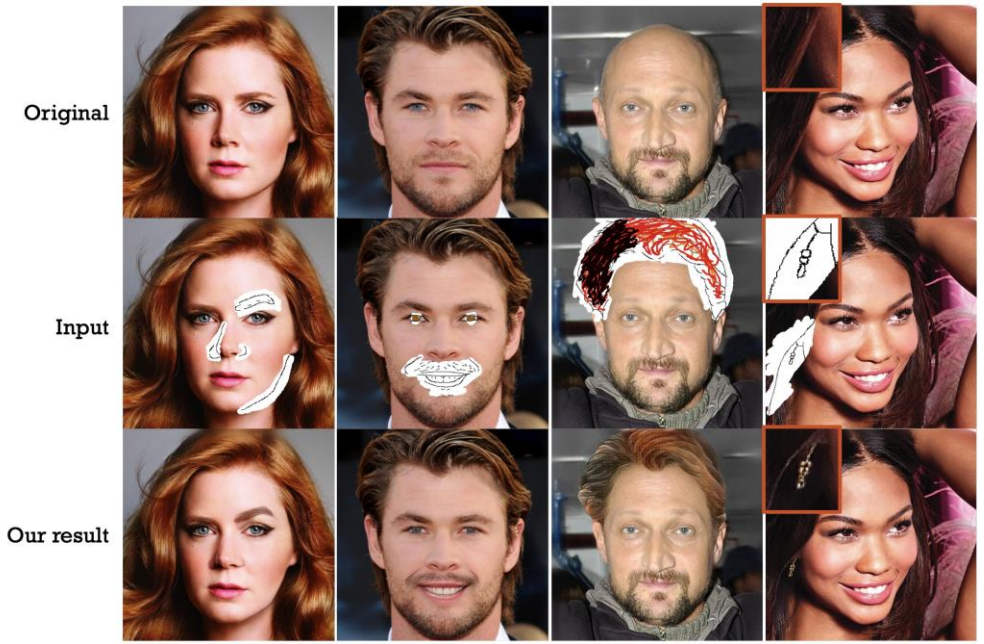
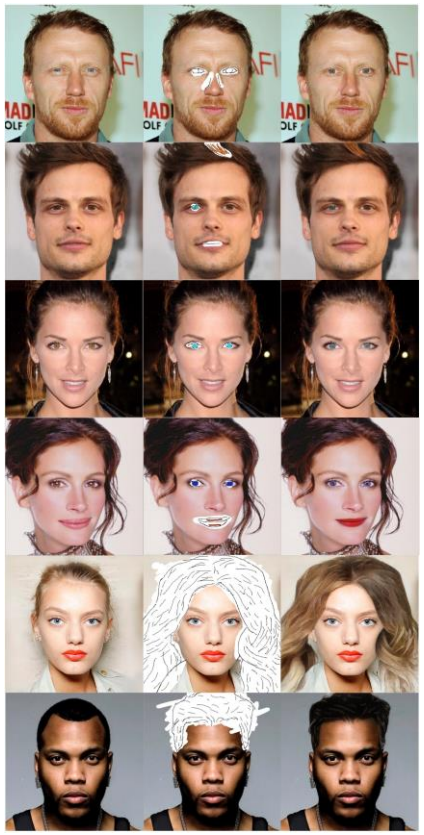


Image Credits: https://github.com/run-youngjoo/SC-FEGAN/raw/master/imgs/GUI.gif

Image Credits: https://github.com/run-youngjoo/SC-FEGAN/raw/master/imgs/teaser.jpg

GAN - Progressive



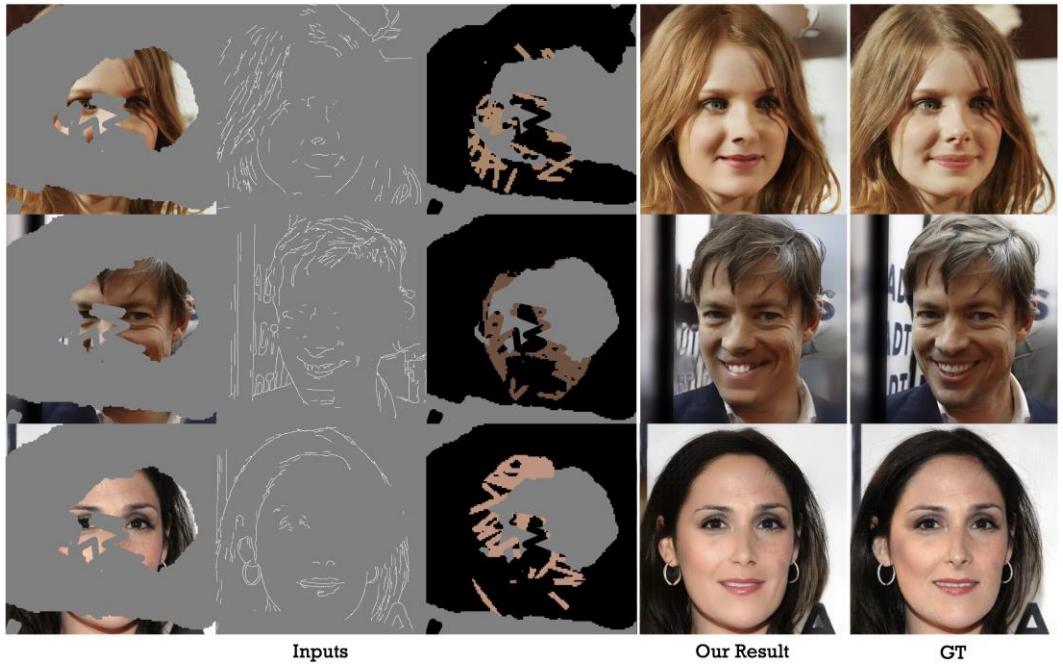


Image Credits: https://github.com/run-youngjoo/SC-FEGAN/blob/master/imgs/face_edit.jpg

Image Credits: https://github.com/run-youngjoo/SC-FEGAN/blob/master/imgs/restoration.jpg

GAN - Progressive

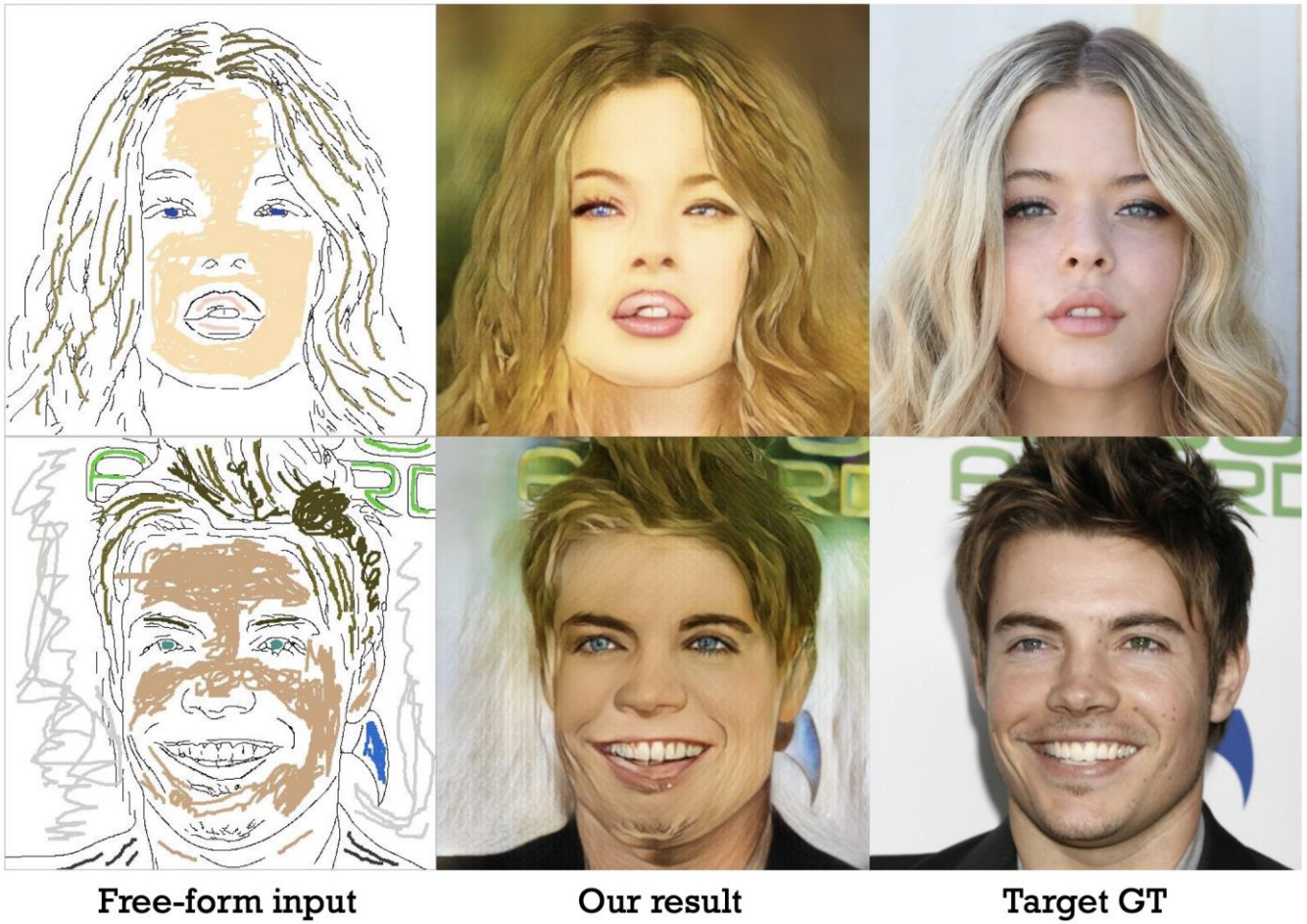


Image Credits: https://github.com/run-youngjoo/SC-FEGAN/blob/master/imgs/restoration2.jpg

GAN - Progressive

https://thispersondoesnotexist.com/
https://thisartworkdoesnotexist.com/
https://thiscatdoesnotexist.com/
https://thischemicaldoesnotexist.com/

GAN - ?

Generative adversarial networks (GANs) are *generative* models that create new data instances that resemble your training data.

GANs can create images that look like photographs of human faces, even though the faces don't belong to any real person.

Generative Model

1. Generative models can generate new data instances.

2. Discriminative models discriminate between different kinds of data instances.

- A generative model could generate new photos of animals that look like real animals, while a discriminative model could tell a dog from a cat.

GAN - Generative Model?

- GANs are just one kind of generative model.
- More formally, given a set of data instances X and a set of labels Y:
- **Generative** models capture the joint probability p(X, Y), or just p(X) if there are no labels.
- **Discriminative** models capture the conditional probability p(Y | X).
- A generative model includes the distribution of the data itself and tells you how likely a given example is.
- A discriminative model ignores the question of whether a given instance is likely, and just tells you how likely a label is to apply to the instance.

Generative Models – Hard?

Generative models tackle a more difficult task than analogous discriminative models. Generative models have to model *more*.

Why it is Hard?

A generative model for images might capture correlations like "things that look like boats are probably going to appear near things that look like water" and "eyes are unlikely to appear on foreheads." These are very complicated distributions.

In contrast, a discriminative model might learn the difference between "sailboat" or "not sailboat" by just looking for a few tell-tale patterns. It could ignore many of the correlations that the generative model must get right.

Generative Models –Brainstorming

1. A model returns a probability when you give it a data instance. Is this model a generative model or a discriminative model? - **5 Minutes**

GAN Structure

- A generative adversarial network (GAN) has two parts:
- The **generator** learns to generate plausible data. The generated instances become negative training examples for the discriminator.
- The **discriminator** learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.
- When training begins, the generator produces obviously fake data, and the discriminator quickly learns to tell that it's fake

Generated Data

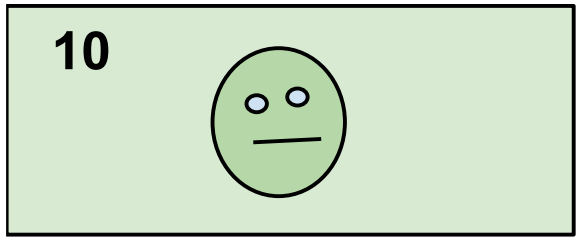
Discriminator

Real Data

FAKE

REAL

GAN Training – Process



FAKE

REAL

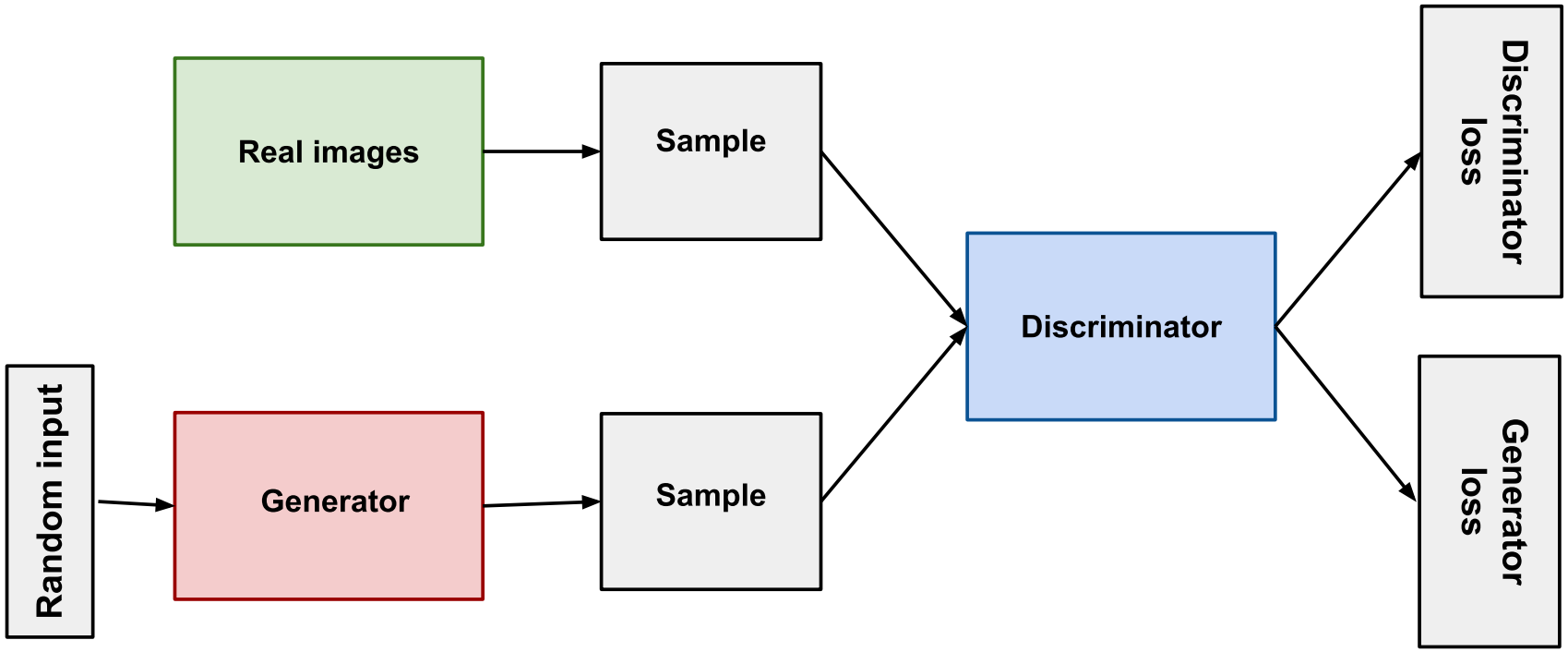
GAN Training – Process

REAL

REAL

GAN Training – Process

Structure of GAN



Structure of GAN

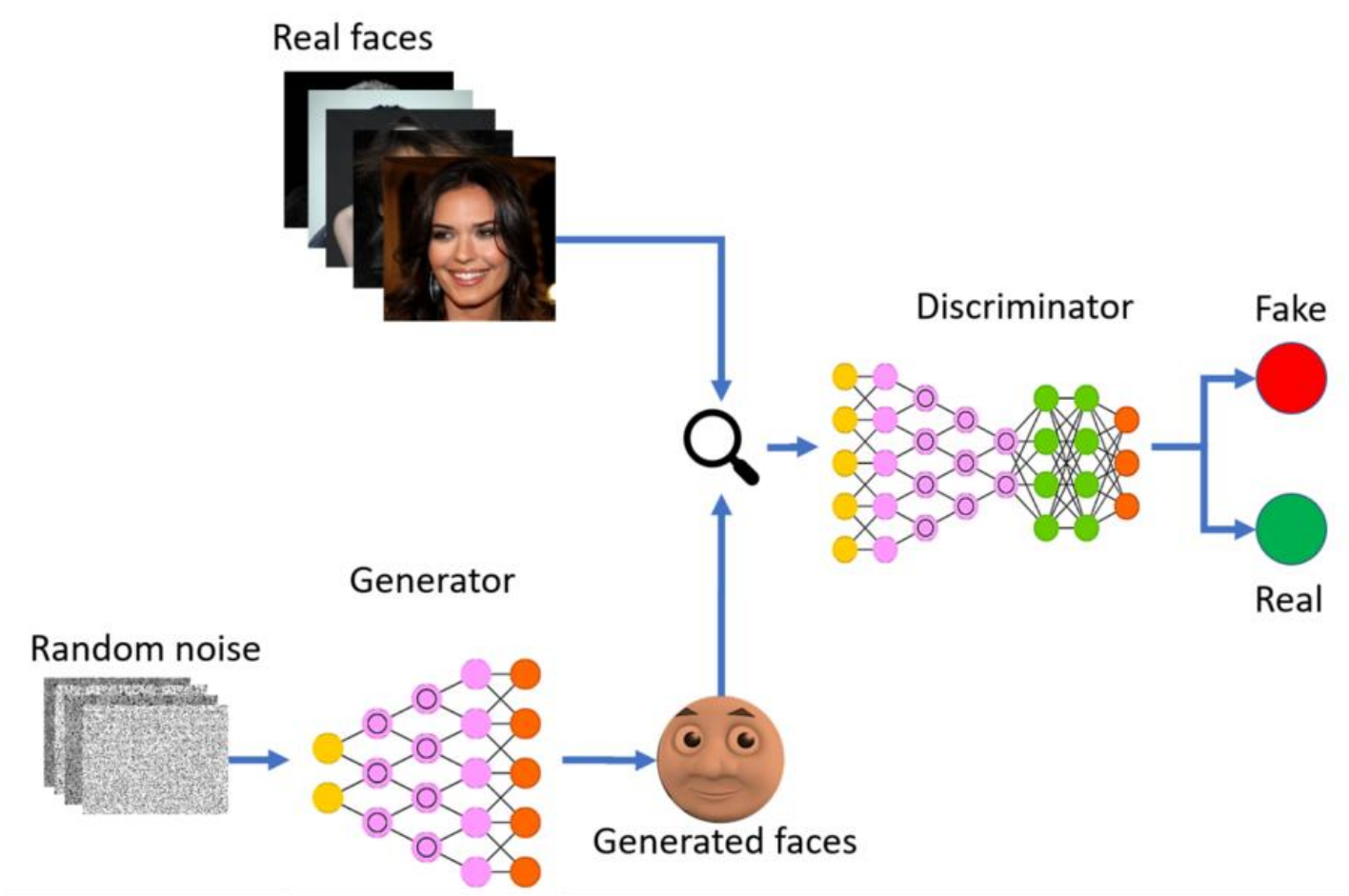


Image Credits: https://medium.com/sigmoid/a-brief-introduction-to-gans-and-how-to-code-them-2620ee465c30

Generator vs Discriminator

- A generative adversarial network (GAN) has two parts:
- The **generator** learns to generate plausible data. The generated instances become negative training examples for the discriminator.
- The **discriminator** learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.
- When training begins, the generator produces obviously fake data, and the discriminator quickly learns to tell that it's fake

The Discriminator

The discriminator in a GAN is simply a classifier. It tries to distinguish real data from the data created by the generator. It could use any network architecture appropriate to the type of data it's classifying.

1) Random input
2) Generator network, which transforms the random input into a data instance
3) Discriminator network, which classifies the generated data
4) Discriminator output
5) Generator loss, which penalizes the generator for failing to fool the discriminator

Generator

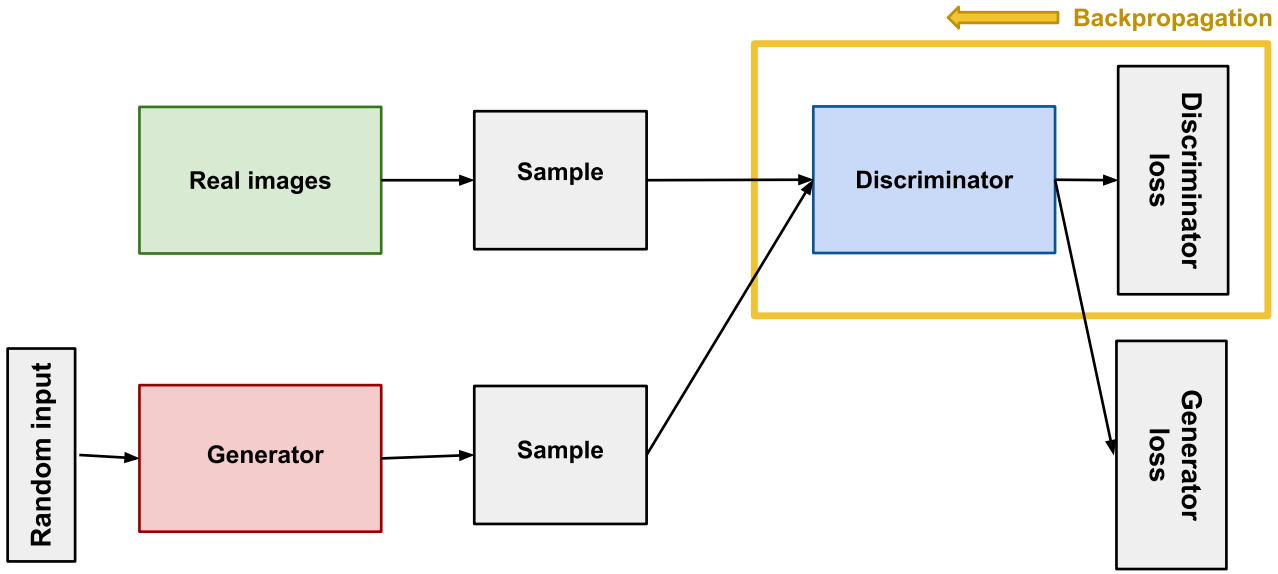


Figure 1: Backpropagation in discriminator training.

Discriminator Training Data

- ✓ The discriminator's training data comes from two sources:
- ✓ Real data instances, such as real pictures of people. The discriminator uses these instances as positive examples during training.
- ✓ Fake data instances created by the generator. The discriminator uses these instances as negative examples during training.

Training the Discriminator

1. The discriminator classifies both real data and fake data from the generator.
2. The discriminator loss penalizes the discriminator for misclassifying a real instance as fake or a fake instance as real.
3. The discriminator updates its weights through backpropagation from the discriminator loss through the discriminator network.

Generator

The generator part of a GAN learns to create fake data by incorporating feedback from the discriminator. It learns to make the discriminator classify its output as real.

Generator training requires tighter integration between the generator and the discriminator than discriminator training requires. The portion of the GAN that trains the generator includes:

1) Random input
2) Generator network, which transforms the random input into a data instance
3) Discriminator network, which classifies the generated data
4) Discriminator output
5) Generator loss, which penalizes the generator for failing to fool the discriminator

Generator

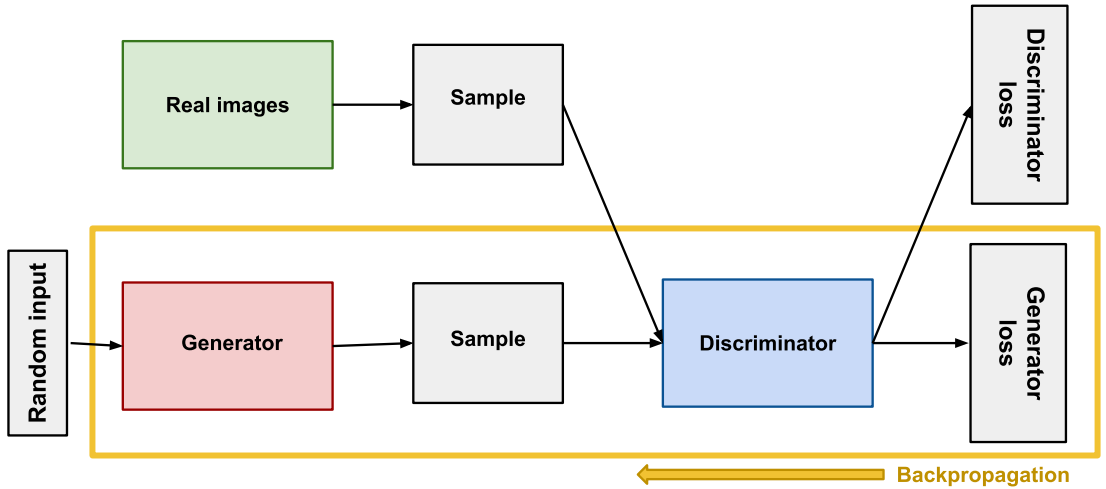


Figure 1: Backpropagation in generator training.

Using the Discriminator to Train the Generator

1. Sample random noise.
2. Produce generator output from sampled random noise.
3. Get discriminator "Real" or "Fake" classification for generator output.
4. Calculate loss from discriminator classification.
5. Backpropagate through both the discriminator and generator to obtain gradients.
6. Use gradients to change only the generator weights.

Using the Discriminator to Train the Generator

Minimax Loss

In the paper that introduced GANs, the generator tries to minimize the following function while the discriminator tries to maximize it:

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))]$$

In this function:

- `D(x)` is the discriminator's estimate of the probability that real data instance x is real.
- E_x is the expected value over all real data instances.
- `G(z)` is the generator's output when given noise z.
- `D(G(z))` is the discriminator's estimate of the probability that a fake instance is real.
- E_z is the expected value over all random inputs to the generator (in effect, the expected value over all generated fake instances G(z)).
- The formula derives from the cross-entropy between the real and generated distributions.

The generator can't directly affect the `log(D(x))` term in the function, so, for the generator, minimizing the loss is equivalent to minimizing `log(1 - D(G(z)))`.

In TF-GAN, see minimax_discriminator_loss and minimax_generator_loss for an implementation of this loss function.

Image Source: https://developers.google.com/machine-learning/gan/loss

Workaround

https://experiments.withgoogle.com/
https://floom.withgoogle.com/
https://sodar.withgoogle.com/
http://measureup.withgoogle.com/
https://quickdraw.withgoogle.com/
https://magenta.tensorflow.org/assets/sketch_rnn_demo/multi_vae.html
https://creatability.withgoogle.com/body-synth/
https://creatability.withgoogle.com/seeing-music/
https://creatability.withgoogle.com/keyboard
https://semiconductor.withgoogle.com/
https://experiments.withgoogle.com/interplay-mode/view/
http://xinyue.de/scribbling-speech.html
https://andymatuschak.org/scrying-pen/
https://magic-sketchpad.glitch.me/
https://www.autodraw.com/

Workaround

https://colab.research.google.com/drive/19haFgvhNwQQYGdqvoV9X_WLZjmlgM4Oi#scrollTo=83-azWpoYsDg

https://colab.research.google.com/github/google/eng-edu/blob/main/ml/cc/exercises/linear_regression_with_synthetic_data.ipynb?utm_source=mlcc&utm_campaign=colab-external&utm_medium=referral&utm_content=linear_regression_synthetic_tf2-colab&hl=en#scrollTo=_GMGgR6O54IN

https://colab.research.google.com/github/google/eng-edu/blob/main/ml/cc/exercises/linear_regression_with_a_real_dataset.ipynb?utm_source=mlcc&utm_campaign=colab-external&utm_medium=referral&utm_content=linear_regression_real_tf2-colab&hl=en#scrollTo=xRfxp_3yofe3

References

Theoretical:

https://developers.google.com/machine-learning/crash-course
https://developers.google.com/machine-learning/gan
https://ai.google/

UNIVERSITY OF TARTU

1632

unitartu

tartuuniversity