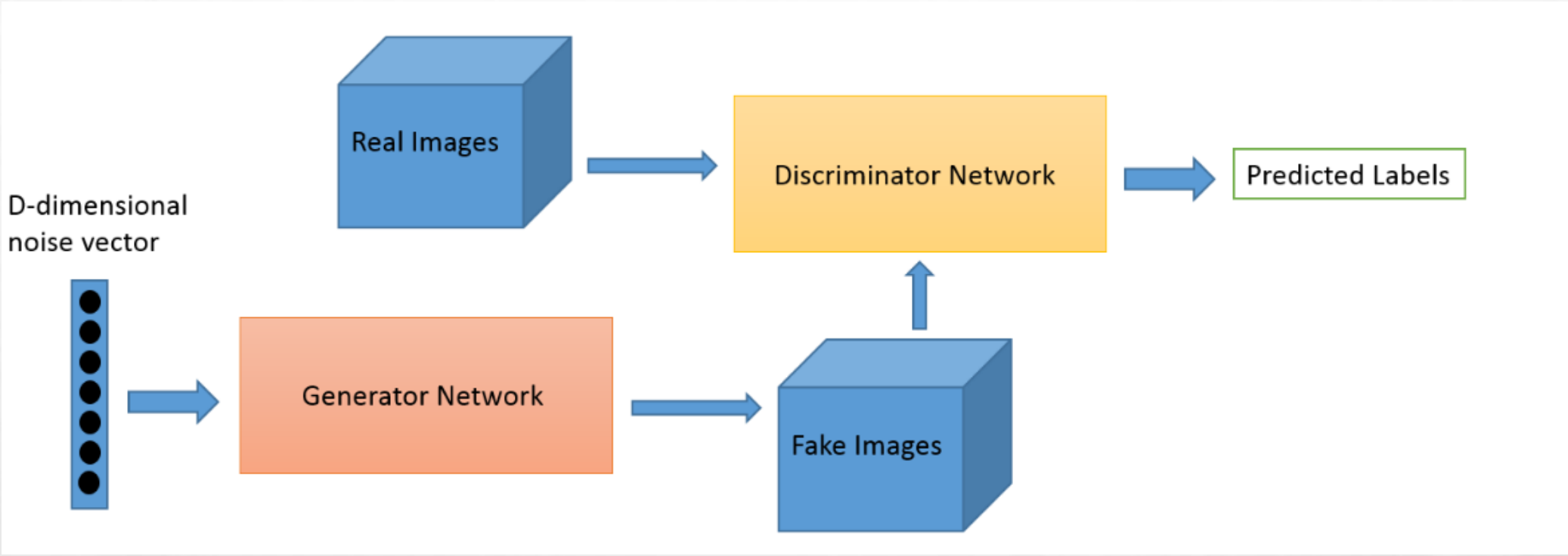
GENERATIVE ADVERSARIAL NETWORK

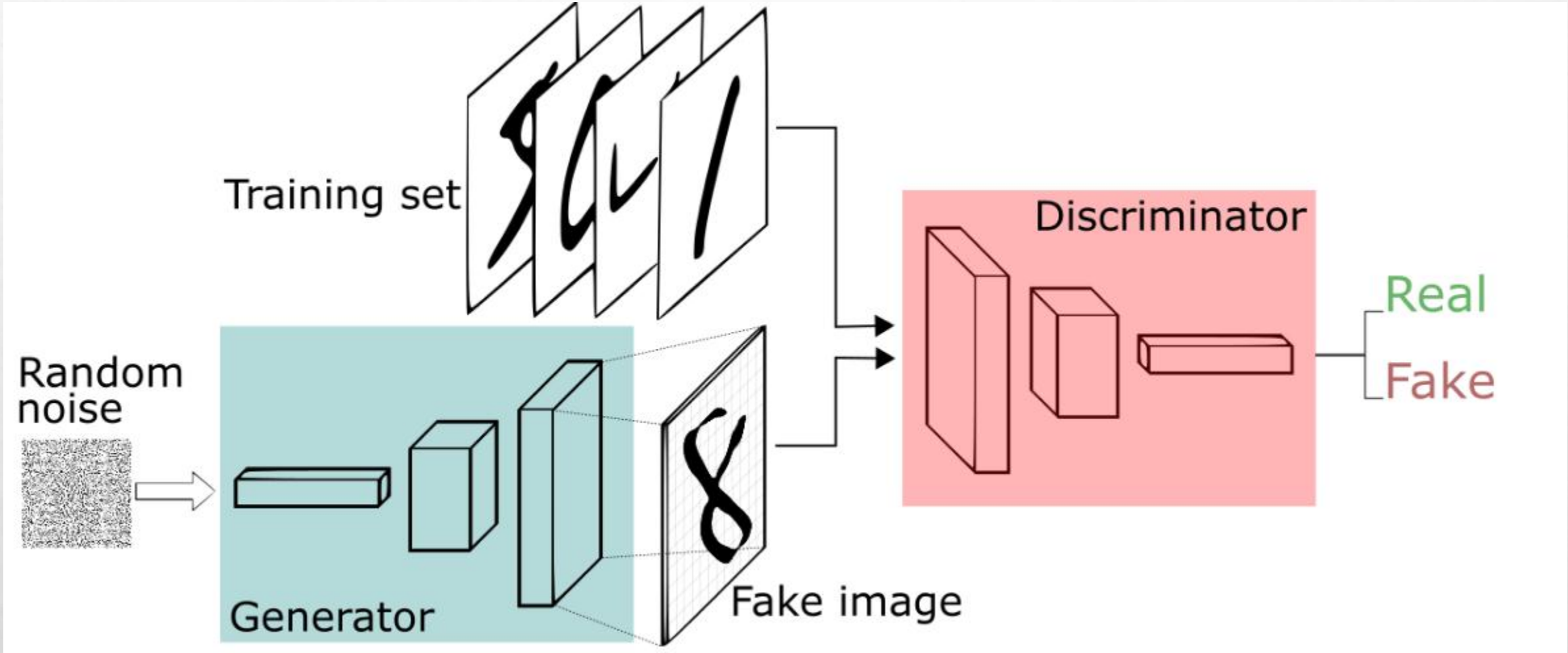
-- SX

CLASSIFICATIONS & USAGE

- Vanllia GAN
  - Black-white pictures
- Deep Convolutional GAN
  - Colored Pictures （three channals）
- Semi-supervisored Learning
  - Technique in which both labeled and unlabeled data are used to train a classifier



- **GANs are composed of two components, a generator and a discriminator.**
- **The discriminator has the task of determining whether a given image looks natural (ie, is an image from the dataset) or looks like it has been artificially created.**
- **The task of the generator is to create natural looking images that are similar to the original data distribution, images that look natural enough to fool the discriminator network.**


Training set
Discriminator
Real
Fake
Random noise
Generator
Fake image

ALGORITHMS

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Demo

Reference paper：

- https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
- https://github.com/uclaacmai/Generative-Adversarial-Network-Tutorial