S+U Learning through ANs

- Pranjit Kalita

- (from paper) Learning from Simulated and Unsupervised Images through Adversarial Training
- Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, Russ Webb
- Apple Inc., November 2016

Motivation

- Tractability of training models on synthetic images (rather than real ones)
- Avoids the need for expensive annotations on real images
- Gap between synthetic and real image distributions
- Simulated + Unsupervised (S+U) Learning
  - Improves realism of simulator's output
  - Retains annotation information from synthetic images

Basic Framework

- Given a simulator S
  - Assume annotated/labeled synthetic images
- Goal :
  - Improve S's output through an adversarial net (based on GAN)
- Why ?
  - Reduce human effort in data collection
  - Preserve annotation/labeling information from synthetic image o/p of S
  - Computational efficiency for ML algorithms without need to label images manually

SimGAN – Adversarial Nets for training image samples

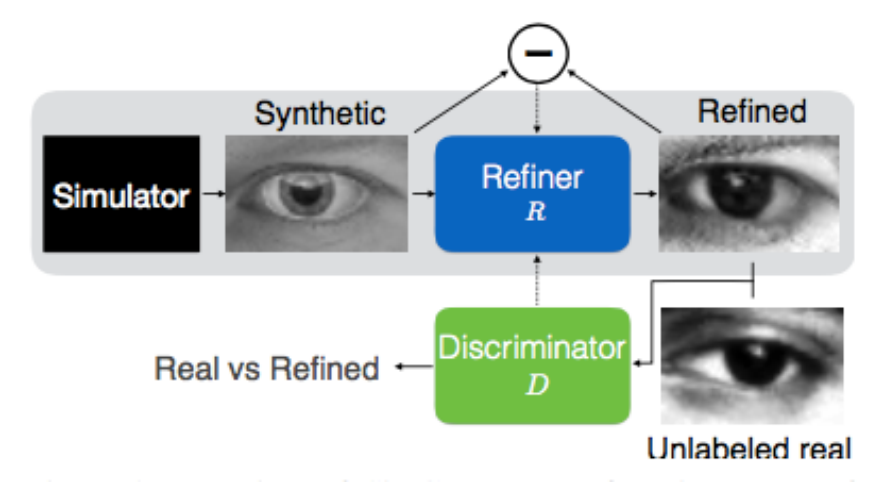


Figure 1: Overview of SimGAN
R & D updated alternately

Difference from GANs

- Adversarial Network (Generator and Discriminator) similar to GANs
- Synthetic images instead of random vectors
- Key architectural differences –
  - Adding a 'self-regularization' term
  - Local adversarial loss
  - Updating discriminator through history of refined images

Brief overview of GANs

- GANs refer to the following minimax game between Generator (G) and Discriminator (D) –

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Minibatch SGD training of GANs

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

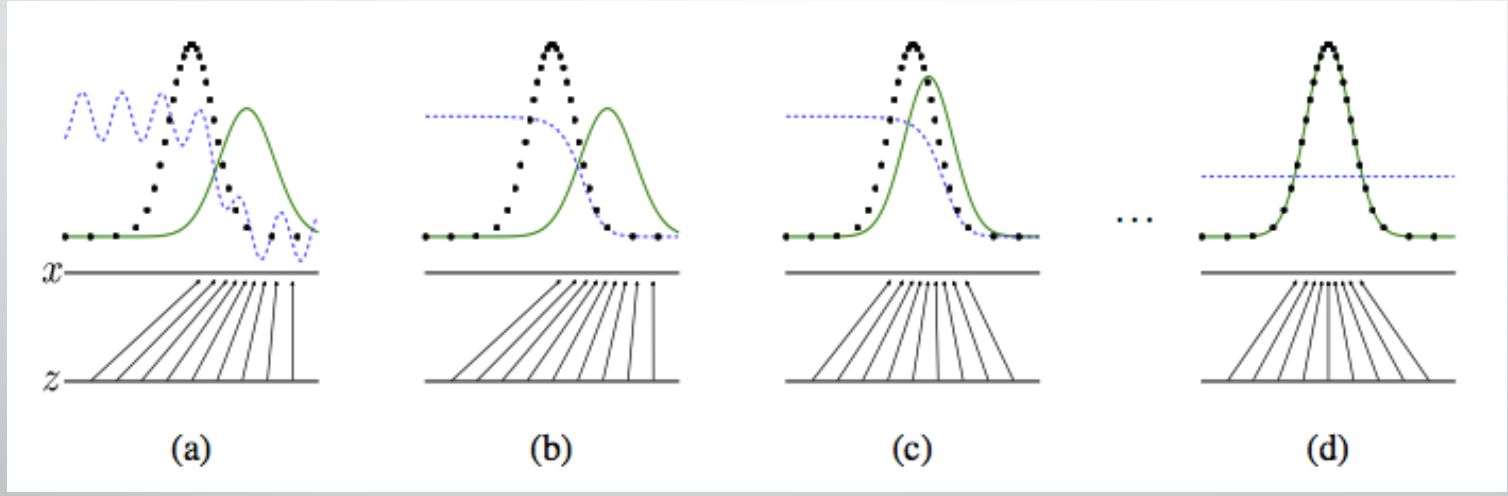
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

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

end for

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

GAN converging steps of p_g and p_{data}



Theoretical Results on GAN

- **Global Optimality of $p_g = p_{data}$**
- **Convergence of SGD GAN algorithm**

What is S+U Learning?

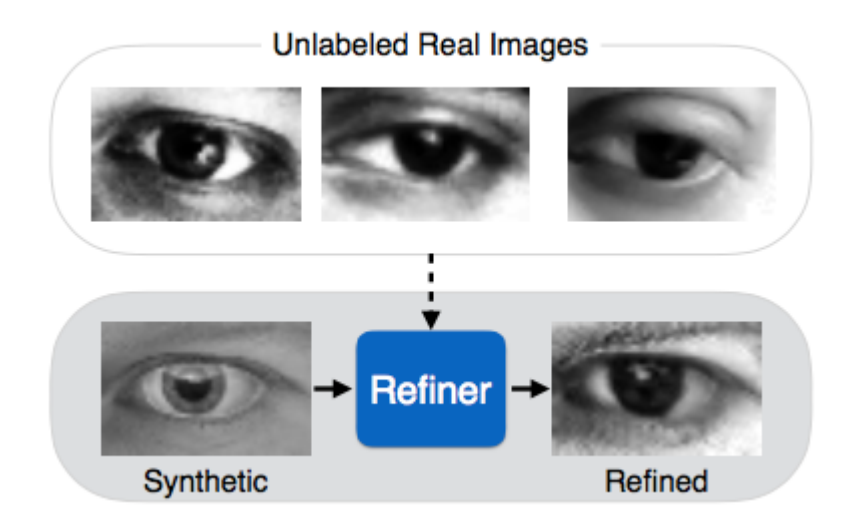


Figure 2: **S+U Learning** : The task is to learn a model that improves the realism of synthetic images from a simulator using unlabeled real data, while preserving the annotation information.

Issues with synthetic images for training

- Not realistic enough, lack of generalization on real image noise sources
  - Ex 1: Skin texture, Iris region for gaze estimation
  - Ex 2: Non-smooth depth boundaries of human for hand-pose estimation not modeled
- Improve simulator?
  - Computationally expensive
  - Renderer design takes a lot of hard work
- Lack of realism may cause models to overfit to '*unrealistic*' details
  - Too perfect!

Synthetic simulated vs. Refined simulated I

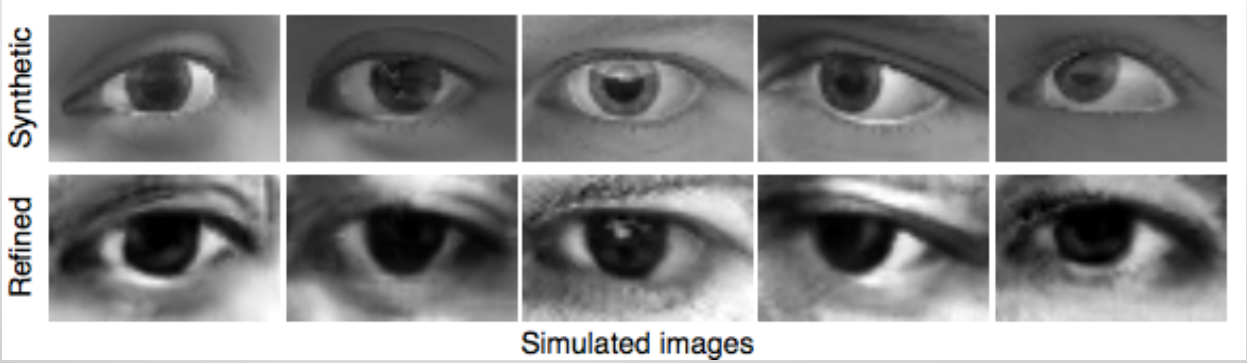


Figure 3 : Difference between synthetic and refined simulated for eye gaze estimation

Synthetic simulated vs. Refined simulated II

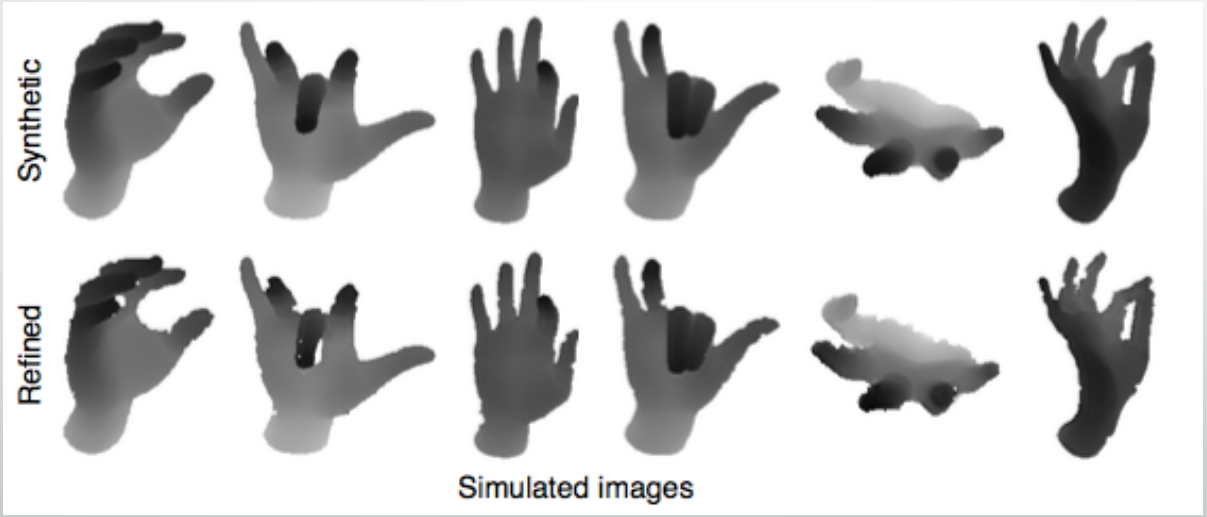


Figure 4 : Difference between synthetic and refined simulated for hand pose estimation

Proposed Approach

- **Goal :** Improve realism of simulated images using unlabeled real data (Figure 1).
- Improves efficiency by retaining annotation information of synthetic images while removing need for data collection
  - Ex : Preserve gaze direction in Figure 1
- Also ensure images generate without artifacts

SimGAN (*improvements over GANs*)

- Objective : Refines images from simulator using a neural net called 'Refiner Network (R)'
- Adding realism : R is trained using an adversarial loss, s.t. refined images indisguishable from real ones using another neural net called 'Discriminator Network (D)'
- Preserving annotations : Adversarial loss complemented with self-regularization loss penalizing large changes between synthetic & refined
- Global structure preserved and local adversarial loss protected by operating on a pixel level instead of holistically modifying the image content through a fully connected auto-encoder
- GAN framework produces artifacts
  - Two competing nn's with competing goals known to be unstable
  - SimGAN limits the discriminator's receptive field to local regions
    - Results in multiple local adversarial losses per image
- Stability of Training : D is updated using history of refined images instead of current R

S+U Learning with SimGAN

- Refined image is defined by – $\tilde{\mathbf{x}} := R_{\theta}(\mathbf{x}).$
- Refined image should look like a real image while preserving annotations.
- Minimize combination of two losses : $\mathcal{L}_R(\boldsymbol{\theta}) = \sum_i \ell_{\text{real}}(\boldsymbol{\theta}; \tilde{\mathbf{x}}_i, \mathcal{Y}) + \lambda \ell_{\text{reg}}(\boldsymbol{\theta}; \tilde{\mathbf{x}}_i, \mathbf{x}_i),$
- First part adds realism to synthetic images
- Second part preserves annotations by minimizing difference between synthetic and refined images

Adversarial Loss with Self-Regularization

- Adversarial loss used in training R is responsible for "fooling" D into classifying refined images as real
- Following GAN approach, modeled as a two-player minimax game, and update R and D alternatively
- D's loss function : $\mathcal{L}_D(\phi) = -\sum_i \log(D_\phi(\tilde{\mathbf{x}}_i)) - \sum_j \log(1 - D_\phi(\mathbf{y}_j)).$
- First term is probability of input being a synthetic image
- Second term probability of being a real one
- <u>Implementation :</u>
  - ConvNet whose last layer outputs the probability of the sample being a refined image
  - Each mini-batch consists of randomly samples refined synthetic images and real images
  - SGD step on mini-batch loss gradient

Realism Loss of R used by D

$$\ell_{\text{real}}(\boldsymbol{\theta}; \tilde{\mathbf{x}}_i, \mathcal{Y}) = -\sum_i \log(1 - D_{\boldsymbol{\phi}}(R_{\boldsymbol{\theta}}(\mathbf{x}_i))).$$

- Objective is to minimize this loss function
- Refiner forces the Discriminator to classify refined images as real
- Part of the min of the minimax game between G & D of GAN

How to preserve annotations?

- In addition to generate realistic images, annotations of synthetic images must be preserved
- This is where the self-regularization term comes in
  - Ex : hand pose estimation in the location of joints should not change in refined, gaze direction in gaze estimation (we will see this in results section)
- The overall loss function for the refiner is –

$$\mathcal{L}_R(\boldsymbol{\theta}) = -\sum_i \log(1 - D_{\boldsymbol{\phi}}(R_{\boldsymbol{\theta}}(\mathbf{x}_i))) + \lambda \|R_{\boldsymbol{\theta}}(\mathbf{x}_i) - \mathbf{x}_i\|_1,$$

Algorithm of SimGAN

Algorithm 1: Adversarial training of refiner network $R_{\boldsymbol{\theta}}$

Input: Sets of synthetic images $\mathbf{x}_i \in \mathcal{X}$, and real images $\mathbf{y}_j \in \mathcal{Y}$, max number of steps ($T$), number of discriminator network updates per step (K_d), number of generative network updates per step (K_g).

Output: ConvNet model $R_{\boldsymbol{\theta}}$.

for $t = 1, \ldots, T$ **do**

  for $k = 1, \ldots, K_g$ **do**

    1. Sample a mini-batch of synthetic images $\mathbf{x}_i$.

    2. Update $\boldsymbol{\theta}$ by taking a SGD step on mini-batch loss $\mathcal{L}_R(\boldsymbol{\theta})$ in (4) .

  end

  for $k = 1, \ldots, K_d$ **do**

    1. Sample a mini-batch of synthetic images $\mathbf{x}_i$, and real images $\mathbf{y}_j$.

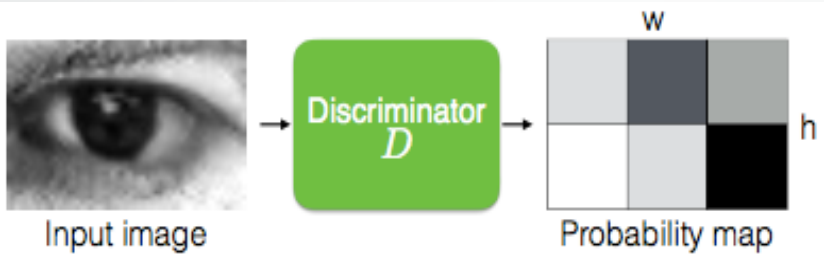
    2. Compute $\tilde{\mathbf{x}}_i = R_{\boldsymbol{\theta}}(\mathbf{x}_i)$ with current $\boldsymbol{\theta}$.

    3. Update $\boldsymbol{\phi}$ by taking a SGD step on mini-batch loss $\mathcal{L}_D(\boldsymbol{\phi})$ in (2).

  end

end

Local Adversarial Loss



- Does not introduce artifacts
  - While training D, R tends to over-emphasize certain image features to fool D
  - This leads to drifting and creating artifacts
- Local patch sampled from the refined image should have similar stats to real image patch
- Thus, D classifies all local image patches separately
  - Limits receptive field
  - Many samples per image for learning D
  - Improves training of R due to multiple 'realism loss'

Local Adversarial Loss in Hand Pose Estimation

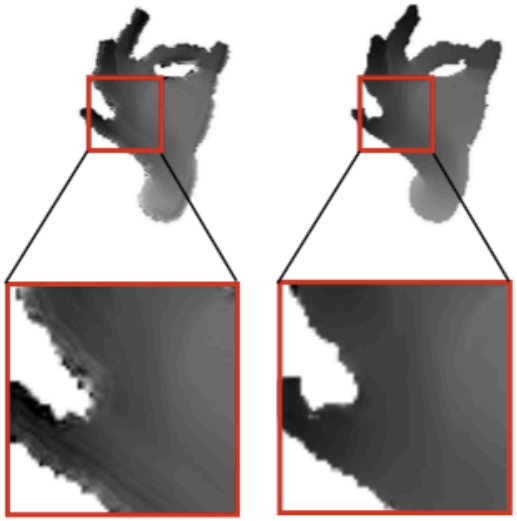


Figure 8. Importance of using a local adversarial loss. (Left) an example image that has been generated with a standard 'global' adversarial loss on the whole image. The noise around the edge of the hand contains obvious unrealistic depth boundary artifacts. (Right) the same image generated with a local adversarial loss that looks significantly more realistic.

Update D using a history of refined images

- Problems with training D only with refined images –
  - Diverging of adversarial training
  - R might re-introduce forgotten artifacts
- Proposed method –
  - Update D using history of refined images, instead of the ones in the current batch
  - Algorithm modified to have buffer of refined images generated by previous networks
  - If B is size of buffer and b is mini-batch size in Algorithm
    - For each iteration of D, computer loss by sampling b/2 images from current R, and b/2 from the buffer to update parameters for D's loss function

Illustration of using history of refined images

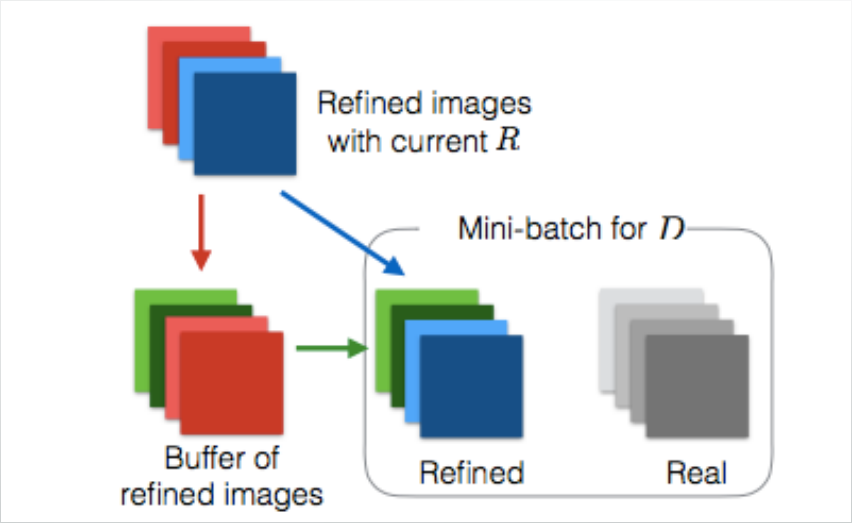


Figure 3: Using history of refined images from previous refined and current R (note the division)

History of Refined Images in Eye Gaze Estimation

Figure 9. Using a history of refined images for updating the discriminator. (Left) synthetic images; (middle) result of using the history of refined images; (right) result without using a history of refined images (instead using only the most recent refined images). We observe obvious unrealistic artifacts, especially around the corners of the eyes.

Experimental Results on Eye Gaze Estimation

- Qualitative Results

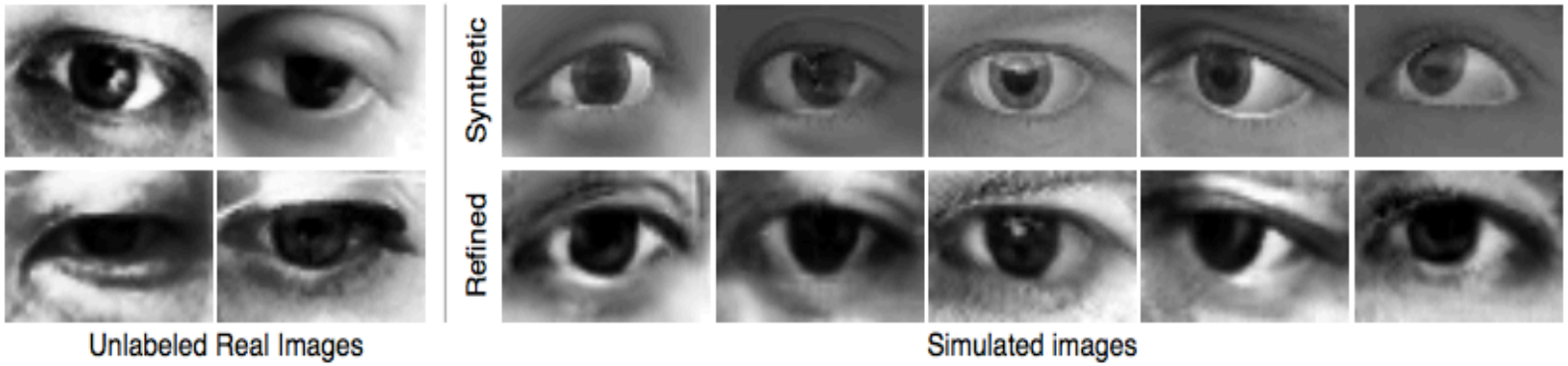


Figure 5. Example output of SimGAN for the UnityEyes gaze estimation dataset [40]. (Left) real images from MPIIGaze [43]. Our refiner network does not use any label information from MPIIGaze dataset at training time. (Right) refinement results on UnityEye. The skin texture and the iris region in the refined synthetic images are qualitatively significantly more similar to the real images than to the synthetic images. More examples are included in the supplementary material.

Visual Turing Test Results

	Selected as real	Selected as synt
Ground truth real	224	276
Ground truth synt	207	293

Table 1. Results of the 'Visual Turing test' user study for classifying real vs refined images. Subjects were asked to distinguish between refined synthetic images (output from our method) and real images (from MPIIGaze). The average human classification accuracy was 51.7%, demonstrating that the automatically generated refined images are visually very hard to distinguish from real images.

Quantitative Results of Eye Gaze Estimation

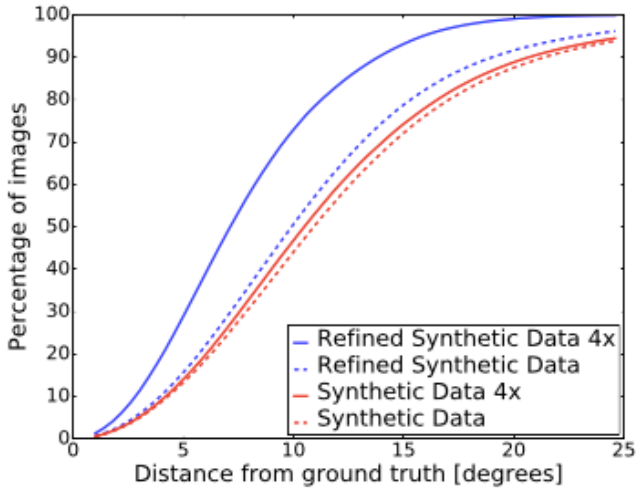


Figure 7. Quantitative results for appearance-based gaze estimation on the MPIIGaze dataset with real eye images. The plot shows cumulative curves as a function of degree error as compared to the ground truth eye gaze direction, for different numbers of training examples of synthetic and refined synthetic data. Gaze estimation using the refined images instead of the synthetic images results in significantly improved performance.

Comparison of SimGAN on different training data sources

Training data	% of images within d
Synthetic Data	62.3
Synthetic Data 4x	64.9
Refined Synthetic Data	69.4
Refined Synthetic Data 4x	**87.2**

Table 2. Comparison of a gaze estimator trained on synthetic data and the output of SimGAN. The results are at distance $d = 7$ degrees from ground truth. Training on the refined synthetic output of SimGAN outperforms training on synthetic data by 22.3%, without requiring supervision for the real data.

Comparison of SimGAN to state-of-the-art

Method	R/S	Error
Support Vector Regression (SVR) [30]	R	16.5
Adaptive Linear Regression ALR) [21]	R	16.4
Random Forest (RF) [33]	R	15.4
kNN with UT Multiview [43]	R	16.2
CNN with UT Multiview [43]	R	13.9
k-NN with UnityEyes [40]	S	9.9
CNN with UnityEyes Synthetic Images	S	11.2
CNN with UnityEyes Refined Images	S	**7.8**

Table 3. Comparison of SimGAN to the state-of-the-art on the MPIIGaze dataset of real eyes. The second column indicates whether the methods are trained on Real/Synthetic data. The error the is mean eye gaze estimation error in degrees. Training on refined images results in a 2.1 degree improvement, a relative 21% improvement compared to the state-of-the-art.

In the future…

- Investigate refined videos
- Model noise distribution to generate multiple refined images for each synthetic image

The End.

- Thank you for your attendance.