AIRND.CENTER
artificial intelligence Innovation

إعداد الفريق العلمي:

بمركز أبحاث الذكاء الاصطناعي (آيرند)

إشراف المهندس: عبدالله بن إبراهيم الحجي

AIRND.CENTER

مركز آيرند - تعزيز أبحاث الذكاء الاصطناعي

اسم البحث:

التعلم من الصور المحاكاة وغير المشرف عليها من خلال التدريب العدائي

Learning from Simulated and Unsupervised Images through Adversarial Training

Apple Inc

إعداد الفريق العلمي:

بمركز أبحاث الذكاء الاصطناعي (آيرند)

إشراف المهندس: عبدالله بن إبراهيم الحجي

تاريخ التقرير: 12/16/2024

تاريخ البحث: 7/19/2017

اختار مركز أبحاث الذكاء الاصطناعي (أيرند) هذا البحث ليقدم تلخيصاً عنه يبرز أهميته ويقربه للباحثين

قدمت شركة Apple في هذا البحث طريقة مبتكرة تجمع بين **البيانات المحاكاة** و**البيانات الحقيقية غير المشرف عليها** لتحسين تدريب الشبكات العصبية المستخدمة في التعرف على الصور. الطريقة المقترحة تُعرف باسم **Simulated+Unsupervised (S+U)**، وهي تعتمد على **الشبكات التوليدية العدائية (Generative Adversarial Networks - GAN)**.

يقوم هذا النهج على فكرة أساسية وهي أن البيانات المحاكاة (Simulation Data) يمكن أن تكون مفيدة جدًا في تدريب الشبكات العصبية، لكن هذه البيانات وحدها قد تفتقر إلى الواقعية الكافية لتقديم نتائج دقيقة. لذلك، يتم دمج البيانات المحاكاة مع بيانات حقيقية غير مشرف عليها، مما يسمح للشبكات العصبية بتحسين أدائها العام في التعرف على الصور.

النقاط الرئيسية في البحث:

أ) مفهوم: S+U (Simulated+Unsupervised)

- تجمع الطريقة بين نوعين من البيانات:
  1. **البيانات المحاكاة :**وهي صور يتم إنشاؤها باستخدام تقنيات المحاكاة الرقمية.
  2. **البيانات غير المشرف عليها :**وهي صور حقيقية بدون تسميات أو علامات محددة.
- باستخدام **شبكاتGAN** ، يتم تقليل الفجوة بين البيانات المحاكاة والبيانات الحقيقية، مما يجعل البيانات المحاكاة تبدو أكثر واقعية للشبكة العصبية.

ب) استخدام شبكات GAN:

- يتم توظيف الشبكات التوليدية العدائية (GAN) كأداة للربط بين البيانات المحاكاة والحقيقية.
- تتكون شبكات GAN من نموذجين:
  1. **المولد :(Generator)** يقوم بإنشاء بيانات محاكاة أكثر واقعية.
  2. **المميز :(Discriminator)** يحاول التمييز بين البيانات الحقيقية والبيانات التي تم إنشاؤها بواسطة المولد.
- الهدف هو تحسين واقعية البيانات المحاكاة لدرجة تجعل الشبكة العصبية غير قادرة على التمييز بينها وبين البيانات الحقيقية.

ج) تحسين التعرف على الصور:

- الطريقة المقترحة تسهم في تحسين أداء الشبكات العصبية في **التعرف على الصور**، خاصة في الحالات التي تكون فيها البيانات المسمى (Labeled Data) محدودة أو غير كافية.
- تعمل هذه الطريقة على تقليل الحاجة إلى بيانات مرقمة أو مصنفة يدويًا، مما يقلل من الجهد والتكلفة في إعداد البيانات للتدريب.

أهمية البحث:

أ) حل مشكلة نقص البيانات المشرفة:

- تعتمد الشبكات العصبية الحديثة بشكل كبير على البيانات المصنفة بدقة، والتي تتطلب جهدًا بشريًا كبيرًا لإعدادها.
- يقدم هذا البحث حلاً عمليًا من خلال دمج البيانات المحاكاة مع البيانات غير المشرفة لتقليل الاعتماد على البيانات المصنفة يدويًا.

ب) تحسين أداء الشبكات العصبية:

- يساعد النهج المقترح في تحسين دقة النماذج المستخدمة في التعرف على الصور، خاصة في المجالات التي تحتاج إلى بيانات متنوعة وواقعية.

ج) دعم استخدام البيانات المحاكاة:

- يبرز البحث دور البيانات المحاكاة كأداة فعالة لتدريب النماذج، خاصة عندما تكون البيانات الحقيقية محدودة أو صعبة الجمع.

التطبيقات المحتملة:

أ) التعرف على الصور في تطبيقات مختلفة:

- مثل التعرف على الوجوه، الأجسام، أو الكائنات في الصور.
- يمكن استخدام الطريقة في أنظمة المراقبة الأمنية أو تطبيقات الهواتف الذكية.

ب) القيادة الذاتية:

- يمكن استخدام البيانات المحاكاة لتدريب أنظمة الرؤية الخاصة بالمركبات ذاتية القيادة، مع تقليل الاعتماد على جمع بيانات الطرق الحقيقية.

ج) الواقع المعزز والافتراضي:

- تحسين النماذج المستخدمة في تطبيقات الواقع الافتراضي والمعزز التي تتطلب فهمًا دقيقًا للصور والكائنات.

د) التطبيقات الطبية:

- يمكن استخدام الطريقة لتحسين أنظمة التعرف على الصور الطبية، مثل صور الأشعة السينية أو التصوير بالرنين المغناطيسي.

القيود والتحديات:

أ) جودة البيانات المحاكاة:

- رغم أن استخدام GAN يساعد في تحسين واقعية البيانات المحاكاة، إلا أن البيانات الناتجة قد لا تكون مثالية دائمًا.

ب) تعقيد التدريب:

- تدريب شبكات GAN يتطلب موارد حاسوبية كبيرة وفترات زمنية طويلة.

ج) تعميم النموذج:

- قد تواجه الطريقة صعوبات في التعميم على مجالات أو تطبيقات جديدة تختلف عن البيانات المستخدمة في التدريب.

البحث: التعلم من الصور المحاكاة وغير المشرف عليها من خلال التدريب العدائي (2017)

Tags: #AI #Artficatial_Intelligent #Airnd_Center #Training #Simulation

كلمات مفتاحية: #الذكاء_الاصطناعي #مركز_ابحاث_الذكاء_الاصطناعي #آيرند #التدريب #المحاكاة

This paper has been submitted for publication on November 15, 2016.

Learning from Simulated and Unsupervised Images through Adversarial Training

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, Russ Webb
Apple Inc
{a_shrivastava,tpf,otuzel,jsusskind,wenda_wang,rwebb}@apple.com

Abstract

With recent progress in graphics, it has become more tractable to train models on synthetic images, potentially avoiding the need for expensive annotations. However, learning from synthetic images may not achieve the desired performance due to a gap between synthetic and real image distributions. To reduce this gap, we propose Simulated+Unsupervised (S+U) learning, where the task is to learn a model to improve the realism of a simulator's output using unlabeled real data, while preserving the annotation information from the simulator. We develop a method for S+U learning that uses an adversarial network similar to Generative Adversarial Networks (GANs), but with synthetic images as inputs instead of random vectors. We make several key modifications to the standard GAN algorithm to preserve annotations, avoid artifacts, and stabilize training: (i) a 'self-regularization' term, (ii) a local adversarial loss, and (iii) updating the discriminator using a history of refined images. We show that this enables generation of highly realistic images, which we demonstrate both qualitatively and with a user study. We quantitatively evaluate the generated images by training models for gaze estimation and hand pose estimation. We show a significant improvement over using synthetic images, and achieve state-of-the-art results on the MPIIGaze dataset without any labeled real data.

1. Introduction

Large labeled training datasets are becoming increasingly important with the recent rise in high capacity deep neural networks [4, 20, 48, 48, 1, 24, 17]. However, labeling such large datasets is expensive and time-consuming. Thus, the idea of training on synthetic instead of real images has become appealing because the annotations are automatically available. Human pose estimation with Kinect [35] and, more recently, a plethora of other tasks have been tackled using synthetic data [43, 42, 29, 34]. However, learning from synthetic images can be problematic due to a gap between synthetic and real image distributions – synthetic data is often not realistic enough, leading the network to learn details only present in synthetic images and failing to generalize well on real images. One solution to closing this gap is to improve the simulator. However, increasing the realism is often computationally expensive, the content modeling takes a lot of hard work, and even the best rendering algorithms may still fail to model all the characteristics of real images. This lack of realism may cause models to overfit to 'unrealistic' details in the synthetic images.

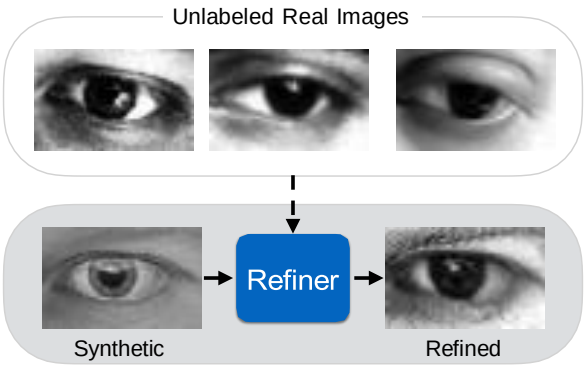

Figure 1. Simulated+Unsupervised (S+U) learning. The task is to learn a model that improves the realism of synthetic images from a simulator using unlabeled real data, while preserving the annotation information.

In this paper, we propose Simulated+Unsupervised (S+U) learning, where the goal is to improve the realism of synthetic images from a simulator using unlabeled real data. The improved realism enables the training of better machine learning models on large datasets without any data collection or human annotation effort. In addition to adding realism, S+U learning should preserve annotation information for training of machine learning models – *e.g.* the gaze direction in Figure 1 should be preserved. Moreover, since machine learning models can be sensitive to artifacts in the synthetic data,

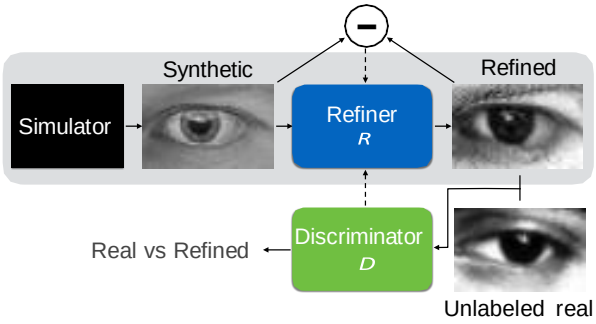

Figure 2. Overview of SimGAN. We refine the output of the simulator with a refiner neural network, R, that minimizes the combination of a local adversarial loss and a 'self-regularization' term. The adversarial loss 'fools' a discriminator network, D, that classifies an image as real or refined. The self-regularization term minimizes the image difference between the synthetic and the refined images. The refiner network and the discriminator network are updated alternately.

S+U learning should generate images without artifacts.

We develop a method for S+U learning, which we term SimGAN, that refines synthetic images from a simulator using a neural network which we call the 'refiner network'. Figure 2 gives an overview of our method: a synthetic image is generated with a black box simulator and is refined using the refiner network. To add realism, we train our refiner network using an adversarial loss, similar to Generative Adversarial Networks (GANs) [8], such that the refined images are indistinguishable from real ones using a discriminative network. To preserve the annotations of synthetic images, we complement the adversarial loss with a self-regularization loss that penalizes large changes between the synthetic and refined images. Moreover, we propose to use a fully convolutional neural network that operates on a pixel level and preserves the global structure, rather than holistically modifying the image content as in *e.g.* a fully connected encoder network. The GAN framework requires training two neural networks with competing goals, which is known to be unstable and tends to introduce arti- facts [32]. To avoid drifting and introducing spurious artifacts while attempting to fool a single stronger discriminator, we limit the discriminator's receptive field to local regions instead of the whole image, resulting in multiple local adversarial losses per image. More- over, we introduce a method for improving the stability of training by updating the discriminator using a history of refined images rather than only the ones from the current refiner network.

Contributions:

1. We propose S+U learning that uses unlabeled real data to refine the synthetic images.
2. We train a refiner network to add realism to synthetic images using a combination of an adversarial loss and a self-regularization loss.
3. We make several key modifications to the GAN training framework to stabilize training and prevent the refiner network from producing artifacts.
4. We present qualitative, quantitative, and user study experiments showing that the proposed framework significantly improves the realism of the simulator output. We achieve state-of-the-art results, without any human annotation effort, by training deep neural networks on the refined output images.

1.1. Related Work

The GAN framework learns two networks (a generator and a discriminator) with competing losses. The goal of the generator network is to map a random vector to a realistic image, whereas the goal of the discriminator is to distinguish the generated from the real images. The GAN framework was first introduced by Goodfellow *et al*. [8] to generate visually realistic images and, since then, many improvements and interesting applications have been proposed [32]. Wang and Gupta [41] use a Structured GAN to learn surface normals and then combine it with a Style GAN to generate natural indoor scenes. Im *et al*. [13] propose a recurrent generative model trained using adversarial training. The recently proposed iGAN [49] enables users to change the im- age interactively on a natural image manifold. CoGAN by Liu *et al*. [21] uses coupled GANs to learn a joint distribution over images from multiple modalities without requiring tuples of corresponding images, achieving this by a weight-sharing constraint that favors the joint distribution solution. Chen *et al*. [2] propose InfoGAN, an information-theoretic extension of GAN, that allows learning of meaningful representations. Tuzel *et al*. [39] tackled image super-resolution for face images with GANs. Li and Wand [19] propose a Markovian GAN for efficient texture synthesis. Lotter *et al*. [22] use adversarial loss in an LSTM network for visual sequence prediction. Yu *et al*. [45] propose the SeqGAN framework that uses GANs for reinforcement learning. Yoo *et al*. [44] tackle pixel-level semantic transfer learning with GANs. Style transfer [7] is also closely related to our work. Many recent works have explored related problems in the domain of generative models, such as PixelRNN [40] that predicts pixels sequentially with an RNN with a softmax loss. The generative networks focus on generating images using a random noise vector; thus, in contrast to our method, the generated images do not have any annotation information that can be used for training a machine learning model.

Many efforts have explored using synthetic data for

various prediction tasks, including gaze estimation [43], text detection and classification in RGB images [9, 15], font recognition [42], object detection [10, 27], hand pose estimation in depth images [38, 37], scene recognition in RGB-D [11], semantic segmentation of urban scenes [31], and human pose estimation [26, 3, 18, 14, 28, 30]. Gaidon *et al*. [5] show that pre-training a deep neural network on synthetic data leads to improved performance. Our work is complementary to these approaches, where we improve the realism of the simulator using unlabeled real data.

Ganin and Lempitsky [6] use synthetic data in a domain adaptation setting where the learned features are invariant to the domain shift between synthetic and real images. Wang *et al*. [42] train a Stacked Con- volutional Auto-Encoder on synthetic and real data to learn the lower-level representations of their font detec- tor ConvNet. Zhang *et al*. [46] learn a Multichannel Autoencoder to reduce the domain shift between real and synthetic data. In contrast to classical domain adaptation methods that adapt the features with respect to a specific prediction task, we bridge the gap between image distributions through adversarial training. This approach allows us to generate realistic training images which can be used to train any machine learning model, potentially for multiple tasks.

Johnson *et al*. [16] transfer the style from a set of real images to the synthetic image by co-segmenting and then identifying similar regions. This approach requires users to select the top few matches from an image database. In contrast, we propose an end-to-end solution that does not require user intervention at inference time.

2. S+U Learning with SimGAN

The goal of Simulated+Unsupervised learning is to use a set of unlabeled real images $\mathbf{y}_i \in Y$ to learn a refiner $R_\theta(\mathbf{x})$ that refines a synthetic image $\mathbf{x}$, where θ are the function parameters. Let the refined image be denoted by $\tilde{\mathbf{x}}$, then $\tilde{\mathbf{x}} := R_\theta(\mathbf{x})$. The key requirement for S+U learning is that the refined image $\tilde{\mathbf{x}}$ should look like a real image in appearance while preserving the annotation information from the simulator.

To this end, we propose to learn θ by minimizing a combination of two losses:

$$L_R(\theta) = \sum_i l_{\text{real}}(\theta; \mathbf{x}_i, Y) + \lambda l_{\text{reg}}(\theta; \mathbf{x}_i), \qquad (1)$$

where $\mathbf{x}_i$ is the i^{th} synthetic training image. The first part of the cost, l_{real}, adds realism to the synthetic images, while the second part, l_{reg}, preserves the annotation information. In the following sections, we expand this formulation and provide an algorithm to optimize for θ.

2.1. Adversarial Loss with Self-Regularization

To add realism to the synthetic image, we need to bridge the gap between the distributions of synthetic and real images. An ideal refiner will make it impossible to classify a given image as real or refined with high confidence. This need motivates the use of an adversarial discriminator network, D_φ, that is trained to classify images as real vs refined, where φ are the parameters of the discriminator network. The adversarial loss used in training the refiner network, R, is responsible for 'fooling' the network D into classifying the refined images as real. Following the GAN approach [8], we model this as a two-player minimax game, and update the refiner network, R_θ, and the discriminator network, D_φ, alternately. Next, we describe this intuition more precisely.

The discriminator network updates its parameters by minimizing the following loss:

$$L_D(\varphi) = -\sum_i \log(D_\varphi(\tilde{\mathbf{x}}_i)) - \sum_j \log(1 - D_\varphi(\mathbf{y}_j)). \qquad (2)$$

This is equivalent to cross-entropy error for a two class classification problem where $D_\varphi(.)$ is the probability of the input being a synthetic image, and $1 - D_\varphi(.)$ that of a real one. We implement D_φ as a ConvNet whose last layer outputs the probability of the sample being a refined image. For training this network, each mini-batch consists of randomly sampled refined synthetic images $\tilde{\mathbf{x}}_i$'s and real images $\mathbf{y}_j$'s. The target labels for the cross-entropy loss layer are 0 for every $\mathbf{y}_j$, and 1 for every $\tilde{\mathbf{x}}_i$. Then φ for a mini-batch is updated by taking a stochastic gradient descent (SGD) step on the mini-batch loss gradient.

In our implementation, the realism loss function l_{real} in (1) uses the trained discriminator D as follows:

$$l_{\text{real}}(\theta; \mathbf{x}_i, Y) = -\log(1 - D_\varphi(R_\theta(\mathbf{x}_i))). \qquad (3)$$

By minimizing this loss function, the refiner forces the discriminator to fail classifying the refined images as synthetic. In addition to generating realistic images, the refiner network should preserve the annotation information of the simulator. For example, for gaze estimation the learned transformation should not change the gaze direction, and for hand pose estimation the location of the joints should not change. This restriction is an essential ingredient to enable training a machine learning model that uses the refined images with the simulator's annotations. For this purpose, we propose using a self-regularization loss that minimizes per-pixel difference between a feature transform of the synthetic and refined images, $l_{\text{reg}} = \|\psi(\tilde{\mathbf{x}}) - \mathbf{x}\|_1$, where ψ is the mapping from image space to a feature space, and $\|.\|_1$ is the L1 norm. The feature transform can be an identity map

Algorithm 1: Adversarial training of refiner network R_{θ}

Input: Sets of synthetic images $\mathbf{x}_i \in \mathsf{X}$, and real images $\mathbf{y}_j \in \mathsf{Y}$, 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_{θ}.

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 $\mathsf{L}_R(\boldsymbol{\theta})$ 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 $\tilde{\mathbf{x}}_i = R_{\theta}(\mathbf{x}_i)$ with current $\boldsymbol{\theta}$.
    3. Update $\boldsymbol{\varphi}$ by taking a SGD step on mini-batch loss $\mathsf{L}_D(\boldsymbol{\varphi})$ in (2).

  end

end

($\psi(\mathbf{x}) = \mathbf{x}$), image derivatives, mean of color channels, or a learned transformation such as a convolutional neural network. In this paper, unless otherwise stated, we used the identity map as the feature transform. Thus, the overall refiner loss function (1) used in our implementation is:

$$\mathsf{L}_R(\boldsymbol{\theta}) = -\sum_i \log(1 - D_{\boldsymbol{\varphi}}(R_{\boldsymbol{\theta}}(\mathbf{x}_i))) + \lambda \| \psi(R_{\boldsymbol{\theta}}(\mathbf{x}_i)) - \psi(\mathbf{x}_i) \|_1. \quad (4)$$

We implement $R_{\boldsymbol{\theta}}$ as a fully convolutional neural net without striding or pooling, modifying the synthetic image on a pixel level, rather than holistically modifying the image content as in *e.g.* a fully connected encoder network, thus preserving the global structure and annotations. We learn the refiner and discriminator parameters by minimizing $\mathsf{L}_R(\boldsymbol{\theta})$ and $\mathsf{L}_D(\boldsymbol{\varphi})$ alternately. While updating the parameters of $R_{\boldsymbol{\theta}}$, we keep $\boldsymbol{\varphi}$ fixed, and while updating $D_{\boldsymbol{\varphi}}$, we fix $\boldsymbol{\theta}$. We summarize this training procedure in Algorithm 1.

2.2. Local Adversarial Loss

Another key requirement for the refiner network is that it should learn to model the real image characteristics without introducing any artifacts. When we train a single strong discriminator network, the refiner network tends to over-emphasize certain image features to fool the current discriminator network, leading to drifting and producing artifacts. A key observation is that

Figure 3. Illustration of local adversarial loss. The discriminator network outputs a $w \times h$ probability map. The adversarial loss function is the sum of the cross-entropy losses over the local patches.

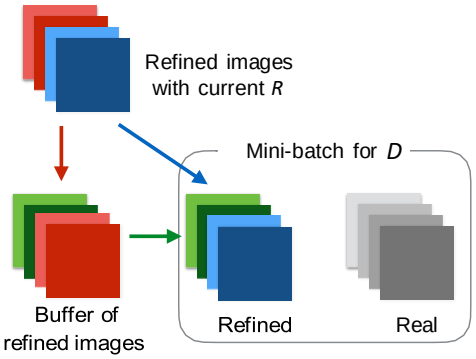

Figure 4. Illustration of using a history of refined images. See text for details.

any local patch sampled from the refined image should have similar statistics to a real image patch. Therefore, rather than defining a global discriminator network, we can define a discriminator network that classifies all local image patches separately. This division not only limits the receptive field, and hence the capacity of the discriminator network, but also provides many samples per image for learning the discriminator network. The refiner network is also improved by having multiple 'realism loss' values per image.

In our implementation, we design the discriminator D to be a fully convolutional network that outputs $w \times h$ dimensional probability map of patches belonging to the fake class, where $w \times h$ are the number of local patches in the image. While training the refiner network, we sum the cross-entropy loss values over $w \times h$ local patches, as illustrated in Figure 3.

2.3. Updating Discriminator using a History of Refined Images

Another problem of adversarial training is that the discriminator network only focuses on the latest refined images. This lack of memory may cause (i) divergence of the adversarial training, and (ii) the refiner network re-introducing the artifacts that the discriminator has forgotten about. Any refined image generated by the refiner network at any time during the entire training procedure is a 'fake' image for the discriminator. Hence, the discriminator should be able to classify all these images as fake. Based on this observation, we introduce

Figure 5. Example output of SimGAN for the UnityEyes gaze estimation dataset [43]. (Left) real images from MPIIGaze [47]. 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.

a method to improve the stability of adversarial training by updating the discriminator using a history of refined images, rather than only the ones in the current mini-batch. We slightly modify Algorithm 1 to have a buffer of refined images generated by previous networks. Let B be the size of the buffer and b be the mini-batch size used in Algorithm 1. At each iteration of discriminator training, we compute the discriminator loss function by sampling $b/2$ images from the current refiner network, and sampling an additional $b/2$ images from the buffer to update parameters $\boldsymbol{\varphi}$. We keep the size of the buffer, B, fixed. After each training iteration, we randomly replace $b/2$ samples in the buffer with the newly generated refined images. This procedure is illustrated in Figure 4.

In contrast to our approach, Salimans *et al*. [32] used a running average of the model parameters to stabilize the training. Note that these two approaches are complementary and can be used together.

3. Experiments

We evaluate our method for appearance-based gaze estimation in the wild on the MPIIGaze dataset [43, 47], and hand pose estimation on the NYU hand pose dataset of depth images [38]. We use a fully convolutional refiner network with ResNet blocks for all of our experiments.

3.1. Appearance-based Gaze Estimation

Gaze estimation is a key ingredient for many human computer interaction (HCI) tasks. However, estimating the gaze direction from an eye image is challenging, especially when the image is of low quality, *e.g.* from a laptop or a mobile phone camera – annotating the eye images with a gaze direction vector is challenging even for humans. Therefore, to generate large amounts of annotated data, several recent approaches [43, 47] train their models on large amounts of synthetic data. Here, we show that training with the refined synthetic images generated by SimGAN significantly outperforms the state-of-the-art for this task.

Synthetic Refined Sample real

Figure 6. Self-regularization in feature space for color images.

The gaze estimation dataset consists of 1.2M synthetic images from the UnityEyes simulator [43] and 214K real images from the MPIIGaze dataset [47] – samples shown in Figure 5. MPIIGaze is a very challenging eye gaze estimation dataset captured under extreme illumination conditions. For UnityEyes, we use a single generic rendering environment to generate training data without any dataset-specific targeting.

Qualitative Results : Figure 5 shows examples of synthetic, real and refined images from the eye gaze dataset. As shown, we observe a significant qualitative improvement of the synthetic images: SimGAN successfully captures the skin texture, sensor noise and the appearance of the iris region in the real images. Note that our method preserves the annotation information (gaze direction) while improving the realism.

Self-regularization in Feature Space: When the synthetic and real images have significant shift in the distribution, a pixel-wise L1 difference may be restrictive. In such cases, we can replace the identity map with an alternative feature transform. For example, in Figure 6, we use the mean of RGB channels for color image refinement. As shown, the network trained using this feature transform is able to generate realistic color images. Note that in our quantitative experiments we still use grayscale images because gaze estimation is better tackled in grayscale due to added invariance [43, 47].

'Visual Turing Test': To quantitatively evaluate the visual quality of the refined images, we designed a simple user study where subjects were asked to classify images as real or refined synthetic. Each subject was shown a random selection of 50 real images and 50 re-

	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. The average human classification accuracy was 51.7% (chance = 50%).

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 output of SimGAN outperforms training on synthetic data by 22.3%.

fined images in a random order, and was asked to label the images as either real or refined. The subjects were constantly shown 20 examples of real and refined images while performing the task. The subjects found it very hard to tell the difference between the real images and the refined images. In our aggregate analysis, 10 subjects chose the correct label 517 times out of 1000 trials ($p = 0.148$), meaning they were not able to reliably distinguish real images from synthetic. Table 1 shows the confusion matrix. In contrast, when testing on original synthetic images vs real images, we showed 10 real and 10 synthetic images per subject, and the subjects chose correctly 162 times out of 200 trials ($p \leq 10^{-8}$), which is significantly better than chance.

Quantitative Results: We train a simple convolutional neural network (CNN) similar to [47] to predict the eye gaze direction (encoded by a 3-dimensional vector for x, y, z) with l_2 loss. We train on UnityEyes and test on MPIIGaze. Figure 7 and Table 2 compare the performance of a gaze estimation CNN trained on synthetic data to that of another CNN trained on refined synthetic data, the output of SimGAN. We observe a large improvement in performance from training on the SimGAN output, a 22.3% absolute percentage improvement. We also observe a large improvement by using more training data – here 4x refers to 100% of the training dataset. The quantitative evaluation confirms the value of the qualitative improvements observed in Figure 5, and shows that machine learning models generalize significantly better using SimGAN.

Table 3 shows a comparison to the state-of-the-art. Training the CNN on the refined images outperforms the state-of-the-art on the MPIIGaze dataset, with a relative improvement of 21%. This large improvement shows the practical value of our method in many HCI tasks.

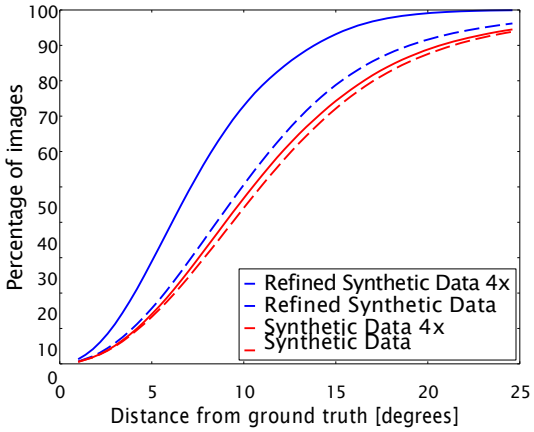

Figure 7. Quantitative results for appearance-based gaze estimation on the MPIIGaze dataset with real eye images. The plot shows cumulative curves as a function of degree error as compared to the ground truth eye gaze direction, for different numbers of training examples of data.

Method	R/S	Error
Support Vector Regression (SVR) [33]	R	16.5
Adaptive Linear Regression ALR) [23]	R	16.4
Random Forest (RF) [36]	R	15.4
kNN with UT Multiview [47]	R	16.2
CNN with UT Multiview [47]	R	13.9
k-NN with UnityEyes [43]	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.

Preserving Ground Truth: To quantify that the ground truth gaze direction doesn't change significantly, we manually labeled the ground truth pupil centers in 100 synthetic and refined images by fitting an ellipse to the pupil. This is an approximation of the gaze direction, which is difficult for humans to label accurately. The absolute difference between the estimated pupil center of synthetic and corresponding refined image is quite small: 1.1 ± 0.8px (eye width=55px).

Implementation Details: The refiner network, R_θ, is a residual network (ResNet) [12]. Each ResNet block consists of two convolutional layers containing 64 feature maps. An input image of size 55×35 is convolved with 3×3 filters that output 64 feature maps. The output is passed through 4 ResNet blocks. The output of the last ResNet block is passed to a 1×1 convolutional layer producing 1 feature map corresponding to the re-

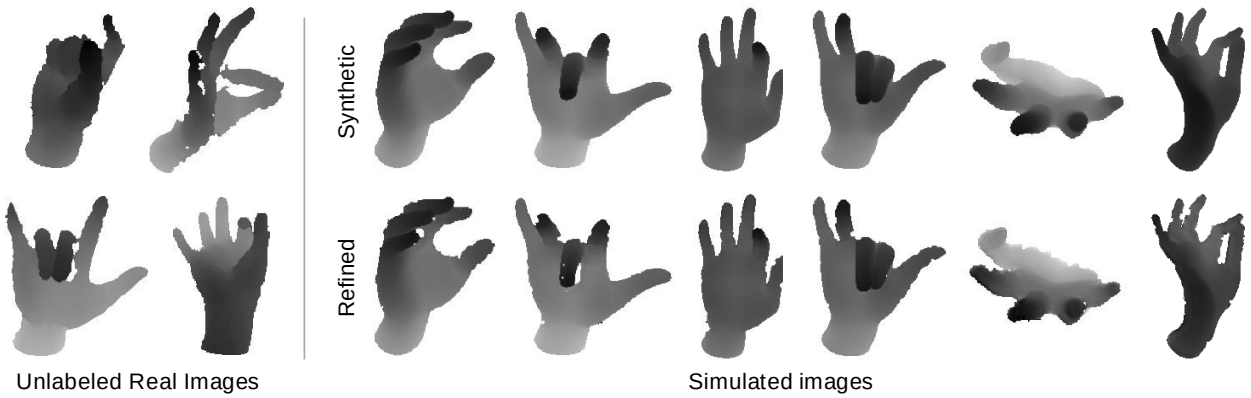

Figure 8. Example refined test images for the NYU hand pose dataset [38]. (Left) real images, (right) synthetic images and the corresponding refined output images from the refiner network. The major source of noise in the real images is the non-smooth depth boundaries that the refiner networks learns to model.

fined synthetic image.

The discriminator network, D_{φ}, contains 5 convolution layers and 2 max-pooling layers as follows: (1) Conv3x3, stride=2, feature maps=96, (2) Conv3x3, stride=2, feature maps=64, (3) MaxPool3x3, stride=1, (4) Conv3x3, stride=1, feature maps=32, (5) Conv1x1, stride=1, feature maps=32, (6) Conv1x1, stride=1, feature maps=2, (7) Softmax.

Our adversarial network is fully convolutional, and has been designed such that the receptive field of the last layer neurons in R_{θ} and D_{φ} are similar. We first train the R_{θ} network with just self-regularization loss for $1,000$ steps, and D_{φ} for 200 steps. Then, for each update of D_{φ}, we update R_{θ} twice, *i.e.* K_d is set to 1, and K_g is set to 50 in Algorithm 1.

The eye gaze estimation network is similar to [47], with some changes to enable it to better exploit our large synthetic dataset. The input is a 35×55 grayscale image that is passed through 5 convolu- tional layers followed by 3 fully connected layers,
the last one encoding the 3-dimensional gaze vector: (1) Conv3x3, feature maps=32, (2) Conv3x3, feature maps=32, (3) Conv3x3, feature maps=64, (4) Max-Pool3x3, stride=2, (5) Conv3x3, feature maps=80, (6) Conv3x3, feature maps=192, (7) MaxPool2x2, stride=2, (8) FC9600, (9) FC1000, (10) FC3, (11) Euclidean loss. All networks are trained with a constant 0.001 learning rate and 512 batch size, until the validation error converges.

3.2. Hand Pose Estimation from Depth Images

Next, we evaluate our method for hand pose estimation in depth images. We use the NYU hand pose dataset [38] that contains $72,757$ training frames and $8,251$ testing frames captured by 3 Kinect cameras – one frontal and 2 side views. Each depth frame is la- beled with hand pose information that has been used to create a synthetic depth image. We pre-process the data by cropping the pixels from real images using the synthetic images. The images are resized to 224×224 before passing them to the ConvNet.

Qualitative Results: Figure 8 shows example output of SimGAN on the NYU hand pose test set. The main source of noise in real depth images is from depth discontinuity at the edges, which the SimGAN is able to learn without requiring any label information.

Quantitative Results: We train a fully convolutional hand pose estimator CNN similar to Stacked Hourglass Net [25] on real, synthetic and refined synthetic images of the NYU hand pose training set, and evaluate each model on all *real* images in the NYU hand pose test set. We train on the same 14 hand joints as in [38]. Many state-of-the-art hand pose estimation methods are customized pipelines that consist of several steps. We use only a single deep neural network to analyze the effect of improving the synthetic images to avoid bias due to other factors. Figure 9 and Table 4 present quantitative results on NYU hand pose. Training on refined synthetic data – the output of SimGAN which does not require any labeling for the real images – outperforms the model trained on real images with supervision, by 8.8%. The proposed method also outperforms training on synthetic data. We also observe a large improvement as the number of synthetic training examples is increased – here 3x corresponds to training on all views.

Implementation Details: The architecture is the same as for eye gaze estimation, except the input image size is 224×224, filter size is 7×7, and 10 ResNet blocks are used. The discriminative net D_{φ} is: (1) Conv7x7, stride=4, feature maps=96, (2) Conv5x5, stride=2, feature maps=64, (3) MaxPool3x3, stride=2, (4) Conv3x3, stride=2, feature maps=32, (5) Conv1x1, stride=1, feature maps=32, (6) Conv1x1, stride=1, feature maps=2,

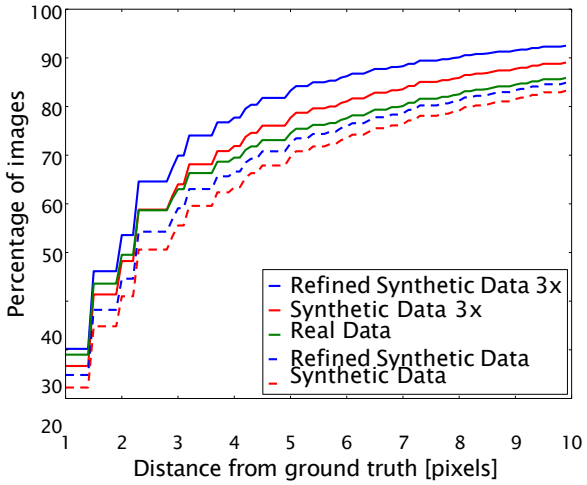

Figure 9. Quantitative results for hand pose estimation on the NYU hand pose test set of real depth images [38]. The plot shows cumulative curves as a function of distance from ground truth keypoint locations, for different numbers of training examples of synthetic and refined images.

Training data	% of images within d
Synthetic Data	69.7
Refined Synthetic Data	72.4
Real Data	74.5
Synthetic Data 3x	77.7
Refined Synthetic Data 3x	**83.3**

Table 4. Comparison of a hand pose estimator trained on synthetic data, real data, and the output of SimGAN. The results are at distance $d = 5$ pixels from ground truth.

(7) Softmax. We train the R_θ network first with just self-regularization loss for 500 steps and D_φ for 200 steps; then, for each update of D_φ we update R_θ twice, *i.e.* K_d is set to 1, and K_g is set to 2 in Algorithm 1.

For hand pose estimation, we use the Stacked Hourglass Net of [25] 2 hourglass blocks, and an output heatmap size 64×64. We augment at training time with random $[-20, 20]$ degree rotations and crops.

3.3. Ablation Study

First, we analyzed the effect of using history of refined images during training. As shown in Figure 10, using the history of refined images (second column) prevents severe artifacts observed while training without the history (third column). This results in an increased gaze estimation error of 12.2 degrees without the history, in comparison to 7.8 degrees with the history.

Next, we compare local vs global adversarial loss during training. A global adversarial loss uses a fully connected layer in the discriminator network, classifying the whole image as real vs refined. The local adversarial loss removes the artifacts and makes the generated image significantly more realistic, as seen in Figure 11.

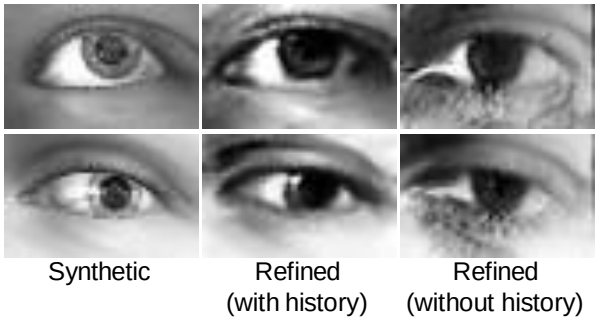

Figure 10. Using a history of refined images for updating the discriminator. (Left) synthetic images; (middle) result of using the history of refined images; (right) result without using a history of refined images (instead using only the most re- cent refined images). We observe obvious unrealistic artifacts, especially around the corners of the eyes.

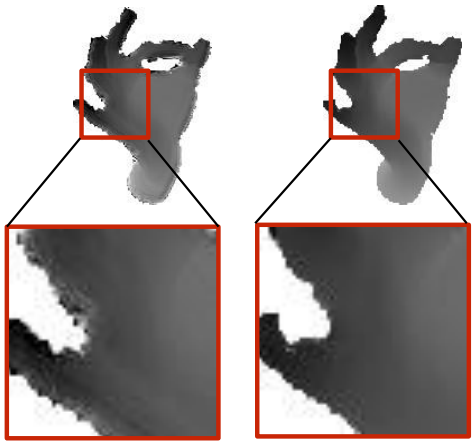

Figure 11. Importance of using a local adversarial loss. (Left) an example image that has been generated with a standard 'global' adversarial loss on the whole image. The noise around the edge of the hand contains obvious unrealistic depth boundary artifacts. (Right) the same image generated with a local adversarial loss that looks significantly more realistic.

4. Conclusions and Future Work

We have proposed Simulated+Unsupervised learning to add realism to the simulator while preserving the annotations of the synthetic images. We described SimGAN, our method for S+U learning, that uses an adversarial network and demonstrated state-of-the-art results without any labeled real data. In future, we intend to explore modeling the noise distribution to generate more than one refined image for each synthetic image, and investigate refining videos rather than single images.

Acknowledgement: We are grateful to our colleagues Barry Theobald, Carlos Guestrin, Ruslan Salakhutdinov, Abhishek Sharma and Yin Zhou for their valuable inputs.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Nat- sev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*, 2016.

[3] T. Darrell, P. Viola, and G. Shakhnarovich. Fast pose estimation with parameter sensitive hashing. In *Proc. CVPR*, 2015.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. CVPR*, 2009.

[5] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proc. CVPR*, 2016.

[6] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[7] L. Gatys, A. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, 2016.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. NIPS*, 2014.

[9] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. *Proc. CVPR*, 2016.

[10] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Proc. ECCV*, 2014.

[11] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. SceneNet: Understanding real world indoor scenes with synthetic data. In *Proc. CVPR*, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[13] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic. Generating images with recurrent adversarial networks. *http://arxiv.org/abs/ 1602.05110*, 2016.

[14] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2014.

[15] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 116(1):1–20, 2016.

[16] M. K. Johnson, K. Dale, S. Avidan, H. Pfister, W. T. Freeman, and W. Matusik. Cg2real: Improving the realism of computer generated images using a large collection of photographs. *IEEE Transactions on Visualization and Computer Graphics*, 17(9):1273–1285, 2011.

[17] I. Krasin, T. Duerig, N. Alldrin, A. Veit, S. Abu-El-Haija, S. Belongie, D. Cai, Z. Feng, V. Fer- rari, V. Gomes, A. Gupta, D. Narayanan, C. Sun, G. Chechik, and K. Murphy. OpenImages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2016.

[18] Y. LeCun, F. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proc. CVPR*, 2004.

[19] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proc. ECCV*, 2016.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proc. ECCV*, 2014.

[21] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Proc. NIPS*, 2016.

[22] W. Lotter, G. Kreiman, and D. Cox. Unsupervised learning of visual structure using predictive generative networks. *arXiv preprint arXiv:1511.06380*, 2015.

[23] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Adaptive linear regression for appearance-based gaze estimation. *PAMI*, 36(10):2033–2046, 2014.

[24] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. In *Proc. ECCV*, 2016.

[25] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. *arXiv preprint arXiv:1603.06937*, 2016.

[26] D. Park and D. Ramanan. Articulated pose estimation with tiny synthetic videos. In *Proc. CVPR*, 2015.

[27] X. Peng, B. Sun, K. Ali, and K. Saenko. Learning deep object detectors from 3d models. In *Proc. ICCV*, 2015.

[28] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In *Proc. CVPR*, 2012.

[29] W. Qiu and A. Yuille. UnrealCV: Connecting computer vision to Unreal Engine. *arXiv preprint arXiv:1609.01326*, 2016.

[30] G. Rogez and C. Schmid. MoCap-guided data augmentation for 3d pose estimation in the wild. *arXiv preprint arXiv:1607.02046*, 2016.

[31] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proc. CVPR*, 2016.

[32] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

[33] T. Schneider, B. Schauerte, and R. Stiefelha- gen. Manifold alignment for person independent appearance-based gaze estimation. In *Proc. ICPR*, 2014.

[34] A. Shafaei, J. Little, and M. Schmidt. Play and learn: Using video games to train computer vision models. In *Proc. BMVC*, 2016.

[35] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *PAMI*, 35(12):2821–2840, 2013.

[36] Y. Sugano, Y. Matsushita, and Y. Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *Proc. CVPR*, 2014.

[37] J. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *Proc. CVPR*, 2015.

[38] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graphics*, 2014.

[39] O. Tuzel, Y. Taguchi, and J. Hershey. Global-local face upsampling network. *arXiv preprint arXiv:1603.07235*, 2016.

[40] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[41] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *Proc. ECCV*, 2016.

[42] Z. Wang, J. Yang, H. Jin, E. Shechtman, A. Agarwala, J. Brandt, and T. Huang. Deepfont: Identify your font from an image. In *Proc. ACMM*, 2015.

[43] E. Wood, T. Baltrušaitis, L. Morency, P. Robinson, and A. Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proc. ACM Symposium on Eye Tracking Research & Applications*, 2016.

[44] D. Yoo, N. Kim, S. Park, A. Paek, and I. Kweon. Pixel-level domain transfer. In *Proc. ECCV*, 2016.

[45] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. *arXiv preprint arXiv:1609.05473*, 2016.

[46] X. Zhang, Y. Fu, A. Zang, L. Sigal, and G. Agam. Learning classifiers from synthetic data using a multichannel autoencoder. *arXiv preprint arXiv:1503.03163*, 2015.

[47] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *Proc. CVPR*, 2015.

[48] Y. Zhang, K. Lee, and H. Lee. Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. In *Proc. ICML*, 2016.

[49] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. Efros. Generative visual manipulation on the natural image manifold. In *Proc. ECCV*, 2016.

Additional Experiments

Qualitative Experiments for Appearance-based Gaze Estimation

Dataset: The gaze estimation dataset consists of 1.2M synthetic images from eye gaze synthesizer UnityEyes [43] and 214K real images from the MPIIGaze dataset [47] – samples shown in Figure 12. MPIIGaze is a very challenging eye gaze estimation dataset captured under extreme illumination conditions. For UnityEyes we use a single generic rendering environment to generate training data without any dataset-specific targeting.

Qualitative Results: In Figure 13, we show many examples of synthetic, and refined images from the eye gaze dataset. We show many pairs of synthetic and refined in multiple rows. The top row contains synthetic images, and the bottom row contains corresponding refined images. As shown, we observe a significant qualitative improvement of the synthetic images: SimGAN successfully captures the skin texture, sensor noise and the appearance of the iris region in the real images. Note that our method preserves the annotation information (gaze direction) while improving the realism.

Qualitative Experiments for Hand Pose Estimation

Dataset: Next, we evaluate our method for hand pose estimation in depth images. We use the NYU hand pose dataset [38] that contains $72,757$ training frames and $8,251$ testing frames. Each depth frame is labeled with hand pose information that has been used to create a synthetic depth image. We pre-process the data by cropping the pixels from real images using the synthetic images. Figure 14 shows example real depth images from the dataset. The images are resized to 224×224 before passing them to the refiner network.

Quantative Results: We show examples of synthetic and refined hand depth images in Figure 15 from the test set. We show our results in multiple pairs of rows. The top row in each pair, contains synthetic depth image, and the bottom row shows the corresponding refined image using the proposed SimGAN approach. Note the realism added to the depth boundary in the refined images, compare to the real images in Figure 14.

Convergence Experiment

To investigate the convergence of our method, we visualize intermediate results as training progresses. As shown in Figure 16, in the beginning, the refiner network learns to predict very smooth edges using only the self-regularization loss. As the adversarial loss is enabled, the network starts adding artifacts at the depth boundaries. However, as these artifacts are not the same as real images, the discriminator easily learns to differentiate between the real and refined images. Slowly the network starts adding realistic noise, and after many steps, the refiner generates very realistic-looking images. We found it helpful to train the network with a low learning rate and for a large number of steps. For NYU hand pose we used lr=0.0002 in the beginning, and reduced to 0.00005 after $600,000$ steps.

Figure 12. Example real images from MPIIGaze dataset.

Figure 13. Qualitative results for automatic refinement of simulated eyes. The top row (in each set of two rows) shows the synthetic eye image, and the bottom row shows the corresponding refined image.

Figure 14. Example real test images in the NYU hand dataset.

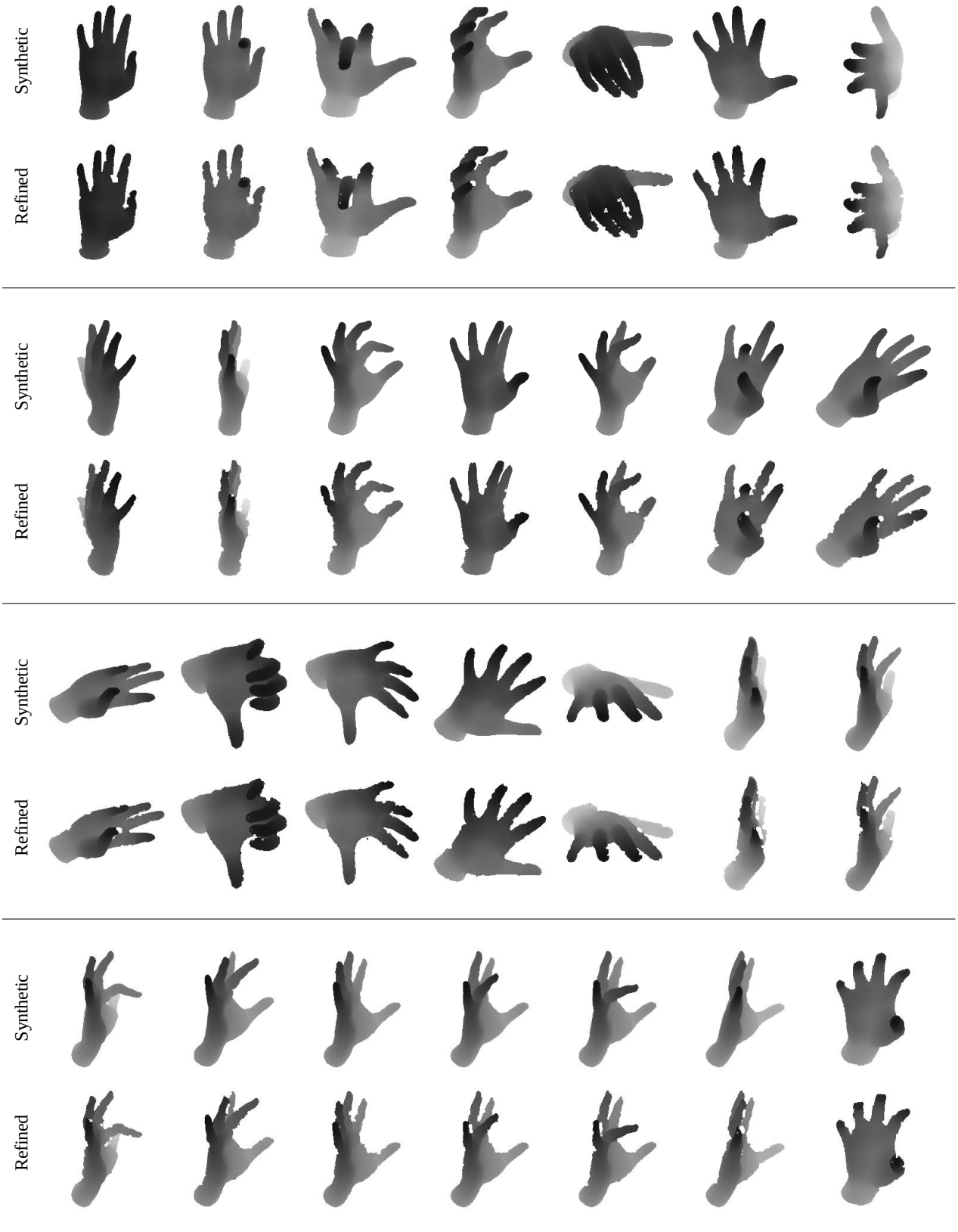

Figure 15. Qualitative results for automatic refinement of NYU hand depth images. The top row (in each set of two rows) shows the synthetic hand image, and the bottom row is the corresponding refined image. Note how realistic the depth boundaries are compared to real images in Figure 14.

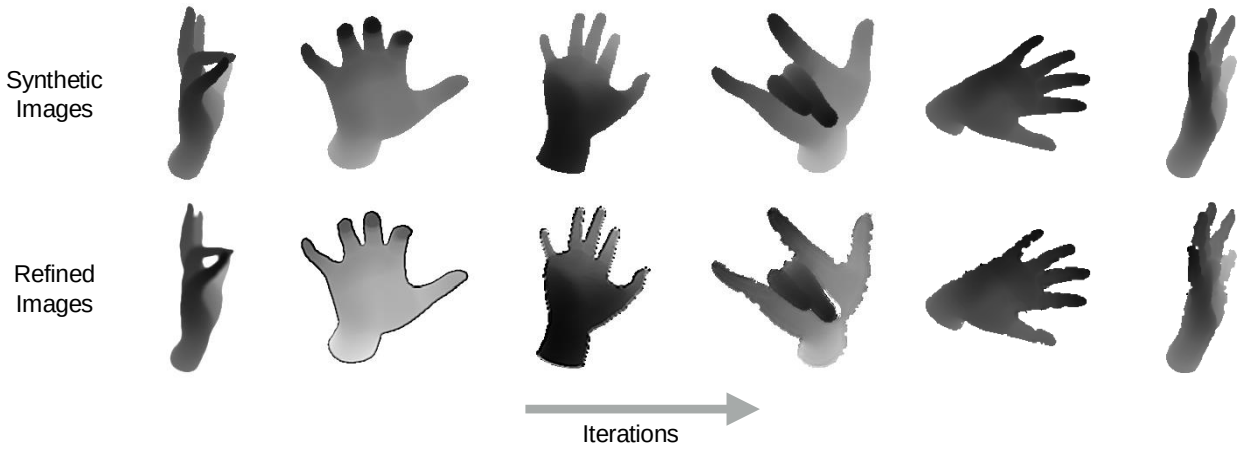

Figure 16. SimGAN output as a function of training iterations for NYU hand pose. Columns correspond to increasing training iterations. First row shows synthetic images, and the second row shows corresponding refined images. The first column is the result of training with l_1 image difference for 300 steps; the later rows show the result when trained on top of this model. In the beginning the adversarial part of the cost introduces different kinds of unrealistic noise to try beat the adversarial network D_{φ}. As the dueling between R_{θ} and D_{φ} progresses, R_{θ} learns to model the right kind of noise.