

# CVPR2017 參加報告

---

名古屋大学  
榎本 憲二

# 自己紹介

榎本 憲二 (Kenji Enomoto)

博士前期課程1年

研究分野

- コンピュータビジョン, リモートセンシング

所属

- 名古屋大学 河口研究室

Webページ : <https://enomotokenji.github.io/>

# CVPRとは

---

The Conference on Computer Vision and Pattern Recognition

*CVPR is the premier annual computer vision event comprising the main conference and several co-located workshops and short courses. With its high quality and low cost, it provides an exceptional value for students, academics and industry researchers.*

開催期間: 7/21 – 7/26 (Workshop, Tutorial含む)

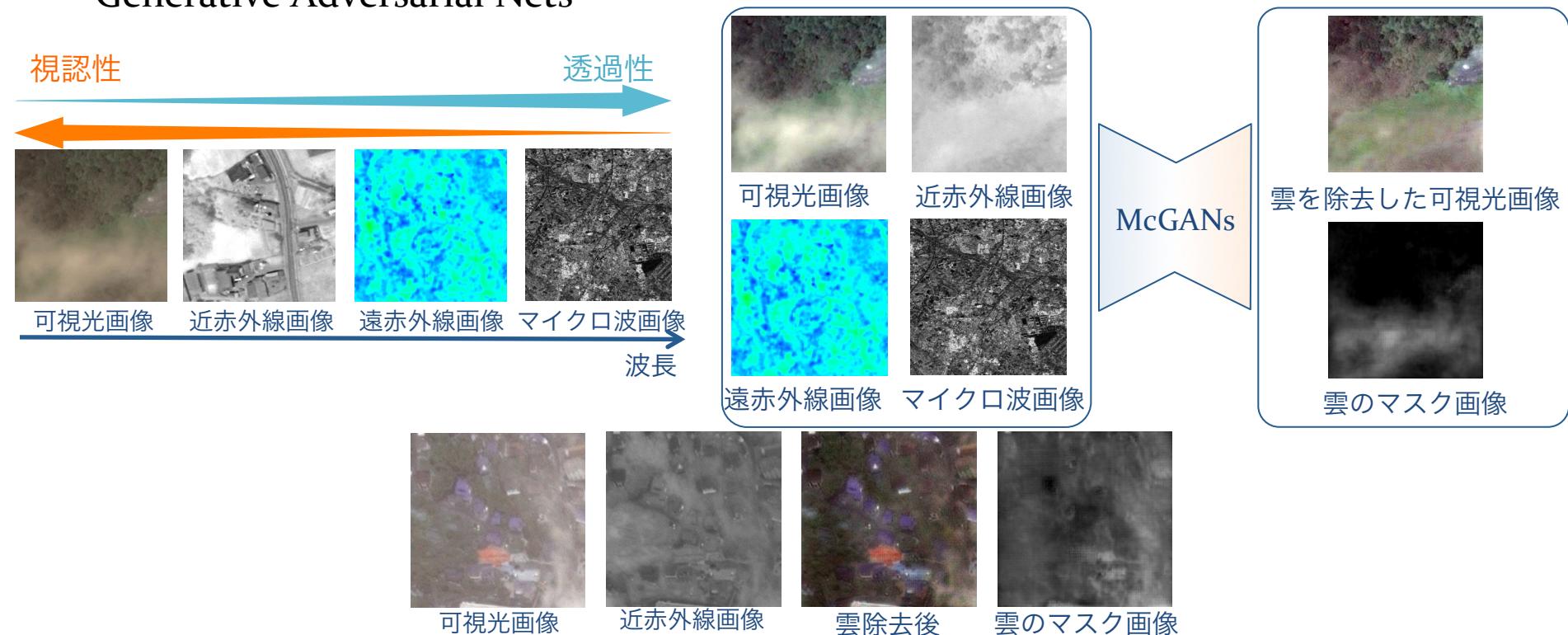
開催場所: ハワイ ホノルル

Papers: <http://openaccess.thecvf.com/CVPR2017.py>

YouTube: [https://www.youtube.com/channel/UCon76gicaarsN\\_Y9YShWwhw](https://www.youtube.com/channel/UCon76gicaarsN_Y9YShWwhw)

# 参加理由

## Filmy Cloud Removal on Satellite Imagery with Multispectral Conditional Generative Adversarial Nets



## 参加理由

---

### EARTHVISION: Large Scale Computer Vision for Remote Sensing Imagery

- CVPR併設workshop
- 主に衛星画像など人工衛星から取得されるデータを用いた画像処理
- 他の発表者はFacebook, MIT, Stanford, TUM...

### Best Paper Awards

- Robocodes: Towards Generative Street Addresses from Satellite Imagery
- Temporal Vegetation Modelling using Long Short-Term Memory Networks for Crop Identification from Medium-Resolution Multi-Spectral Satellite Images

## その他チュートリアル

---

### Theory and Application of Generative Adversarial Network

- 近年盛んに研究が行われているGANに関するTutorial
- 発表資料: [https://github.com/mingyuliutw/cvpr2017\\_gan\\_tutorial.git](https://github.com/mingyuliutw/cvpr2017_gan_tutorial.git)

### GANを使った研究について

- 画像生成は一般うけはものすごく良い
- 研究として使う場合は問題設定が重要
- 定量的評価が難しくなる場合がある
- Adversarial Trainingは強力な学習方法で今後も広く使われていくのではないか

# CVPRに参加した感想

---

8~9割はディープな研究

- ネットワークアーキテクチャが書いてないポスターはレア

いかに学習コストを下げるか

- Unsupervised, Weakly supervised
- Zero-shot, One-shot
- データセットをGANなどにより擬似的に生成

マルチタスクを扱うネットワーク

- UberNet

Notディープな研究ちらほら

- 個人的にものすごくかっこよく見える

# トップ会議に参加するメリット

---

最先端の技術をまとめて把握することができる

- ポスターを全て見て回るだけで1日が終わってしまう
- 発表されている素晴らしい研究たちが一年以上前に行われていた研究という焦りも...

なぜか自分の研究に関するアイデアがポンポン出てくる

偉い先生方とコンタクトがとれる

- 観光地だと普段より話してくれるかもしれない
- 「CVPRに来てるの。えらいね。」となって話しやすい(年齢制限あり?)

研究のモチベーションがMAXに

就職活動もできるかも

# 準備しておくといいこと

## 自分の研究の整理

- アイデアの想起につながる
- 問題点の洗い出し、問題点に対する解決案...
- 論文を書く



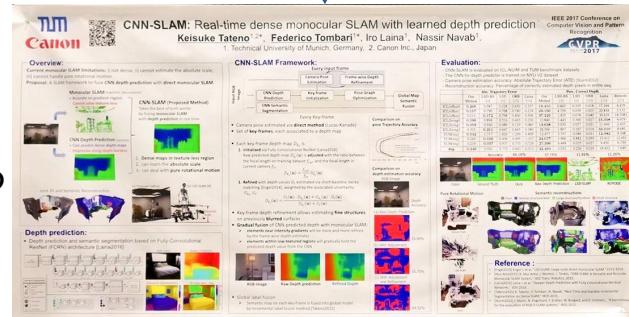
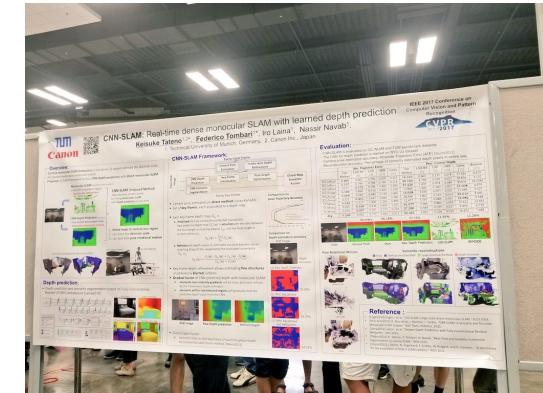
## Microsoft Pix

- 無音カメラ
- 基本的に情報収集は写真



## Office Lens

- 写真中の書類を自動で検知しトリミングしてくれる
- ポスターの撮影に便利



## トップ会議に通すには

---

Reviewerは基本的に落とすつもりなのでは？

- Valid submission: 2620
- Accepted: 783
- Reviewerの立場から自分の論文をみてみることが必要

絶対に落とせない論文はあるらしい

- 新規性あり
- 結果が良い
- いろんな問題やデータに適応できる

Accepted papersに共通して感じたことは論文としての完成度が高い

- 新規性
- 比較実験
- 定量的評価

ぜひ名古屋CV・PRML勉強会からトップ会議に通してほしい(ちなみに今年は11/15)

## 参考資料

---

CVPR2017速報スライド:

[http://hirokatsukataoka.net/temp/cvpaper.challenge/cvpr17\\_finalize.pdf](http://hirokatsukataoka.net/temp/cvpaper.challenge/cvpr17_finalize.pdf)

CVPR参加報告(CyberAgent):

<https://developers.cyberagent.co.jp/blog/archives/9582/>

CVPR2017 posterの写真(気になったやつだけ):

<https://paper.dropbox.com/doc/CVPR2017-poster-J3qRN57LZBpgygt7fzQZ1>

# Learning from Simulated and Unsupervised Images through Adversarial Training

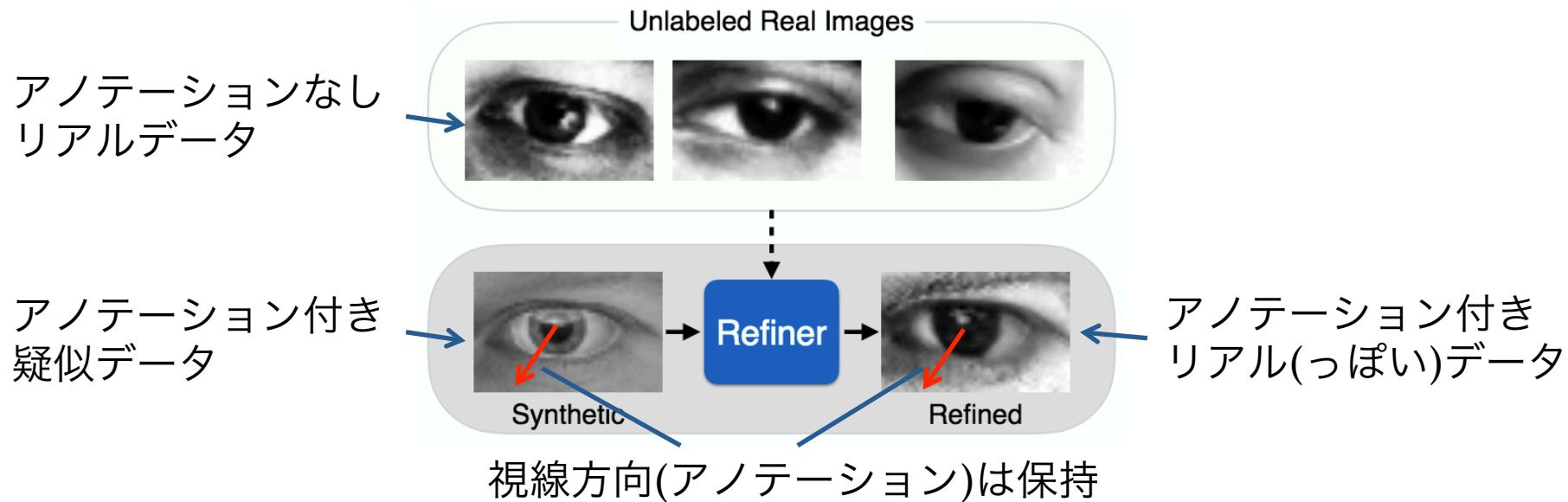
---

arXiv: <https://arxiv.org/abs/1612.07828>

# 概要

CVPR2017 Best Paper Award  
Appleが出したはじめての技術論文

GANを用いてアノテーション付き疑似データとアノテーションなしリアルデータからアノテーション付きリアル(っぽい)データを生成する技術



# S+U Learning with SimGAN

ポイント 1: self-regularization

$R_{\theta}$  : Refiner(GANのGeneratorに相当)

$D_{\phi}$  : Discriminator

$\tilde{\mathbf{x}} := R_{\theta}(\mathbf{x})$

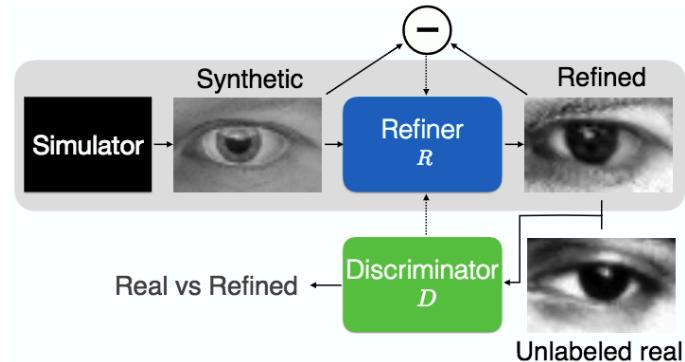
Discriminatorを更新する際のロス関数

$$\mathcal{L}_D(\phi) = - \sum_i \log(D_{\phi}(\tilde{\mathbf{x}}_i)) - \sum_j \log(1 - D_{\phi}(\mathbf{y}_j))$$

Refinerを更新する際のロス関数

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

Self-regularization: RefinedとSyntheticで大きすぎる違いが生じないようにする

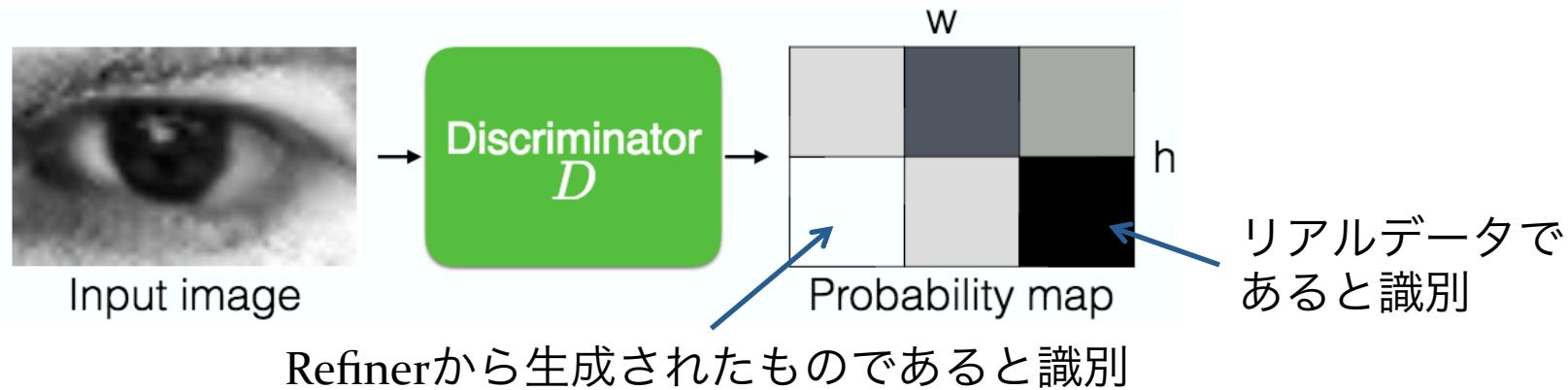


# Local Adversarial Loss

## ポイント 2: Local adversarial loss

Discriminatorは $w \times h$ のローカルバッチ全てに対してRefinerから生成されたかどうかの確率を出力

- Refined imageのどの部分をみても本物と識別できるDiscriminator  
どの部分をみられても本物と識別されない画像を生成するRefinerを学習



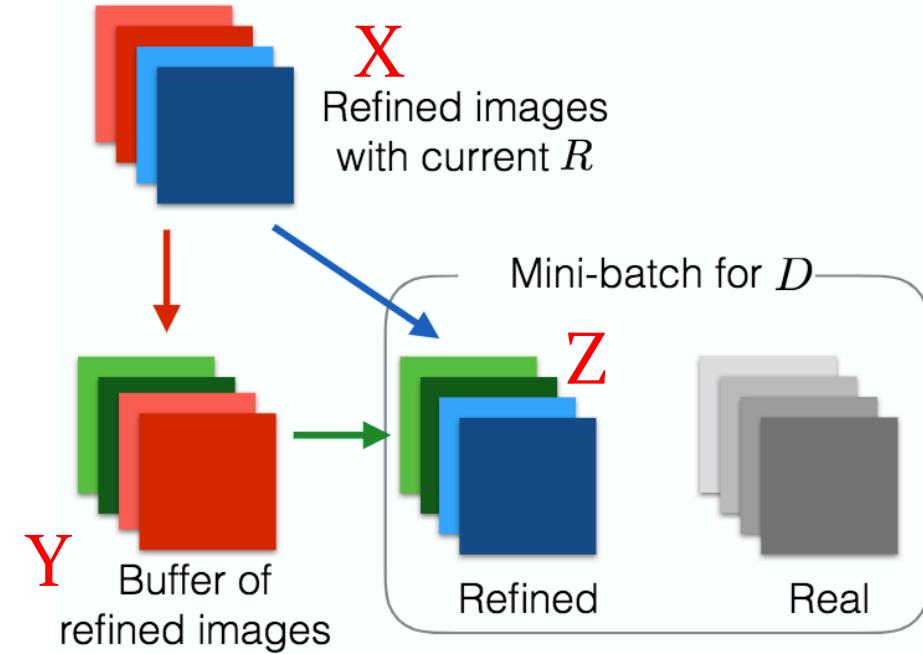
# Updating Discriminator using a History of Refined Images

## ポイント 3: Using a history of refined images

DiscriminatorにとってRefinerから生成されたものは常に'fake'画像

➤ 過去にRefinerから生成された画像全てについて識別できるべきだ

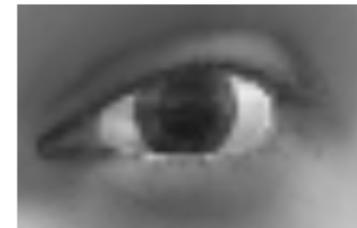
1. Yには過去に生成された画像がB枚格納されている
2. XとYから $b/2$ 枚ずつランダムに選択されZへ
3.  $b$ 枚のRefined imagesとReal imagesでDiscriminatorを学習
4. Xから $b/2$ 枚ランダムに選択されYへ
5. Yから $b/2$ 枚ランダムに選択され捨てられる



# Experiments

## 視線方向推定問題

- 視線方向を保持したままリアル(っぽい)画像を生成



## 手の姿勢推定問題

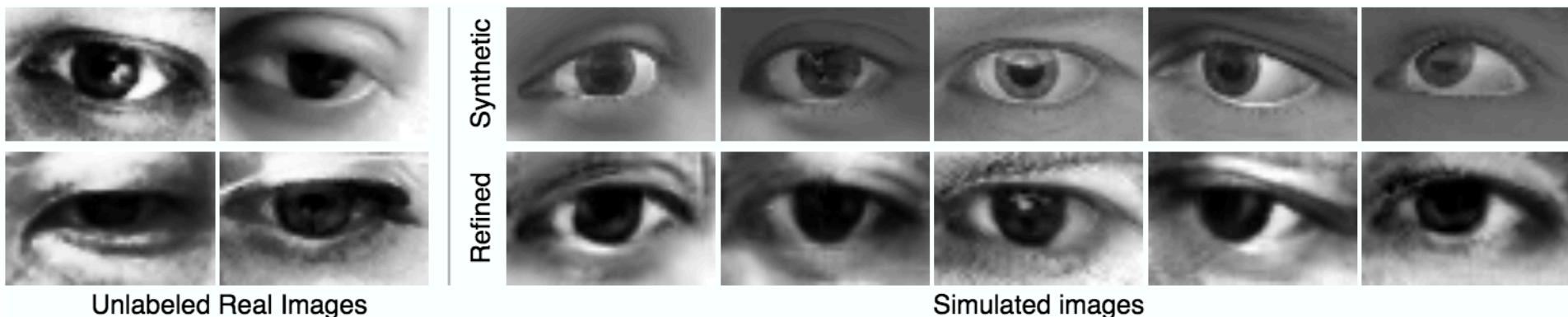
- 関節などの形状を保持したままリアル(っぽい)画像を生成



Synthetic

Real

# Visual Turing Test



Refined imageはSynthetic imageの視線方向を保持している

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

正解: 517/1000(51.7%)

不正解: 483/1000(48.3%)

# Other experimental results

Training data	% of images within $d$
Synthetic Data	62.3
Synthetic Data 4x	64.9
Refined Synthetic Data	69.4
Refined Synthetic Data 4x	<b>87.2</b>

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 sup

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	<b>7.8</b>

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

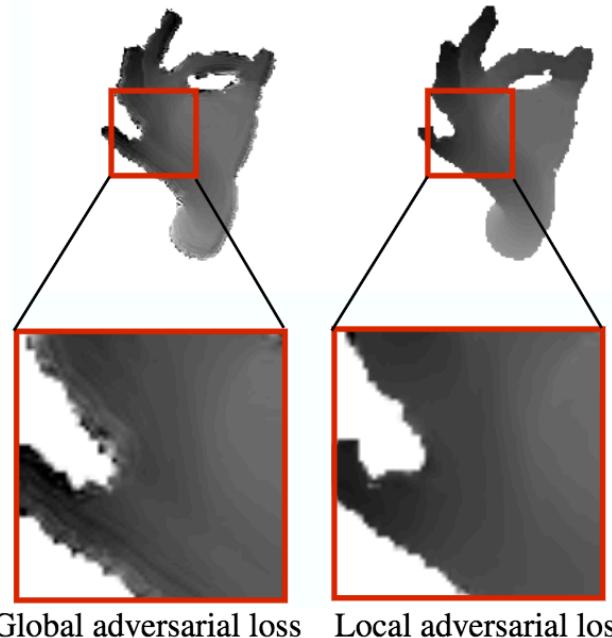
mpared to the state-of-the-art.

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	<b>83.3</b>

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. Training on

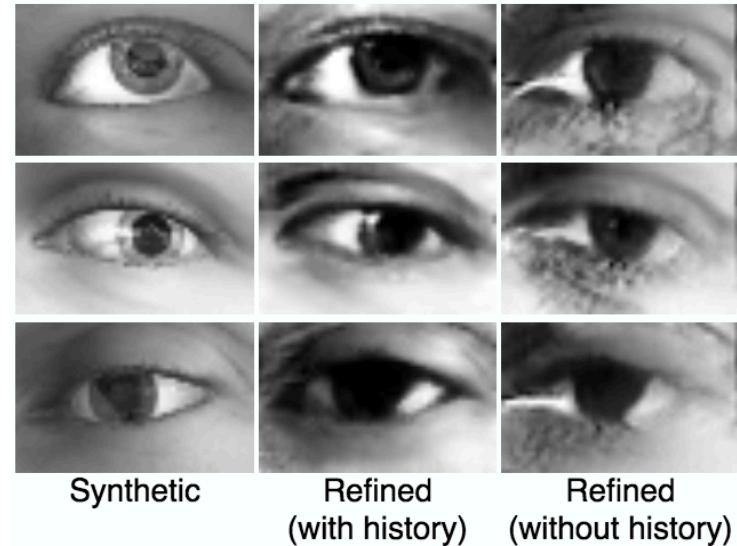
# Analysis of Modifications to Adversarial Training

Local adversarial loss



エッジ部分のノイズが不自然(らしい)

Updating Discriminator using a History of Refined Images



目のコーナー部分が不自然

# まとめ

---

## Self-regularization

- アノテーションを保持

## Local adversarial loss

- Refined imageのどの部分をみても本物と識別できるD  
どの部分をみられても本物と識別されないRを学習

## Using a history of refined images

- どんなRに生成された画像でも本物と識別できるDを学習

Refinerのネットワーク構造は問題によって自分で構築しなければいけない

論文としてとてもきれいにまとまっている印象

# Removing rain from single images via a deep detail network

---

Open access:

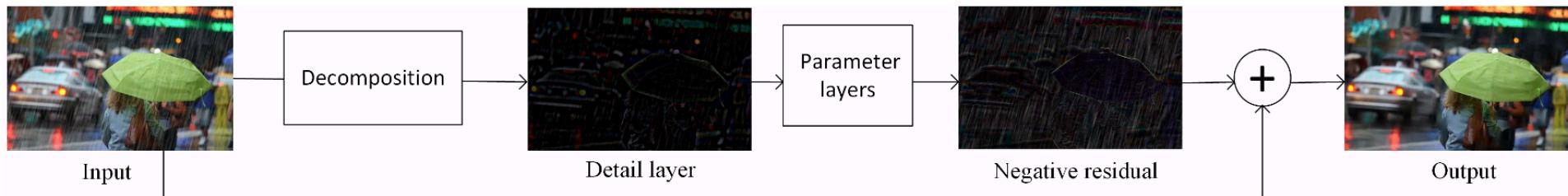
[http://openaccess.thecvf.com/content\\_cvpr\\_2017/papers/  
Fu\\_Removing\\_Rain\\_From\\_CVPR\\_2017\\_paper.pdf](http://openaccess.thecvf.com/content_cvpr_2017/papers/Fu_Removing_Rain_From_CVPR_2017_paper.pdf)

# 概要

単一画像から雨を除去

➤ 動画から雨を除去するよりも難しい

最終的に下の図さえ理解できればOK



# Contributions

---

1. negative residual mapping(neg-mapping)を利用  
ネットワークのsolution spaceを制限することで学習を容易にした
2. ResNetに直接雨画像を入力せず、画像の高周波成分のみを入力  
勾配消失を避けるネットワーク構造について議論し、より深いネットワークでより良い結果が得られたことを示した
3. 14000ペアの雨画像とクリーンな画像を合成により作成  
合成データで学習したが、実際の画像においても良い結果が得られることを示した

# Direct network

---

$$\mathcal{L} = \sum_i \|h(\mathbf{X}_i) - \mathbf{Y}_i\|_F^2 \quad (1)$$

X: 入力画像

Y: 正解画像

$h(\cdot)$ : Deep CNN architecture

$L_2$ ノルムが最小になるようにネットワークを学習すればいいだけでは？

- どうもうまくいかない
- Mapping rangeが広すぎる

$[0, 1]^D \rightarrow [0, 1]^D$  D: 全ピクセル数

- ディープラーニングを用いて直接画像を推定する場合どうしても勾配消失問題が発生してしまう(らしい)



(a) Ground truth



(b) Rainy image



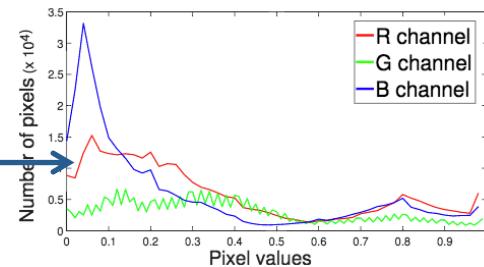
(c) Direct network

# Negative residual mapping

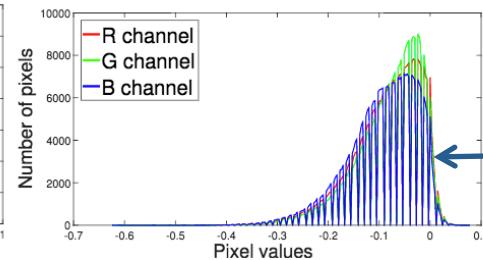
Mapping rangeが広すぎる問題を解決(出力のrangeを狭くする)

ネットワークに  $\mathbf{Y} - \mathbf{X}$ (residual)を推定させて、 $\mathbf{X}$ に足せばいいのではないか  
 ▷  $\mathbf{Y} - \mathbf{X}$ の多くのピクセル値は0に近い

0 - 1にほぼ一様に分布している



(e) Histogram of  $\mathbf{Y}$



(g) Histogram of  $\mathbf{Y} - \mathbf{X}$



大体-0.3 - 0に収まっている！

# Negative residual mapping

$$\mathcal{L} = \sum_i \|h(\mathbf{X}_i) + \mathbf{X}_i - \mathbf{Y}_i\|_F^2 \quad (2)$$

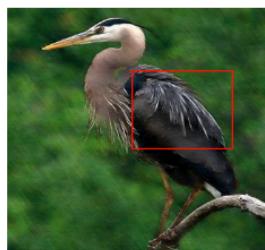
X: 入力画像

Y: 正解画像

$h(\cdot)$ : Y-Xを教師とするネットワーク

ResNetを利用して直接画像を推定した結果とNeg-mappingを利用した結果の比較

- ResNet: 雨模様が残っている, 羽がぼやけている
- Neg-mapping: skip connectionのおかげで  
物体の輪郭が保持できている  
が, まだ雨模様が残っている

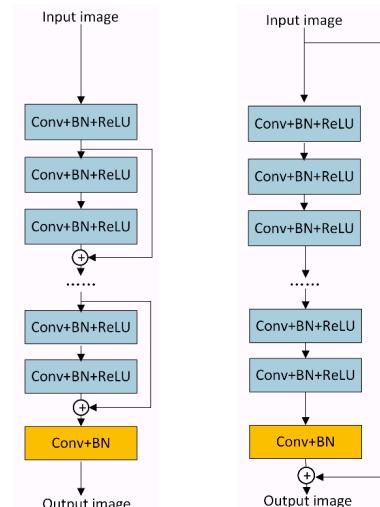


(a) Ground truth

(b) Rainy image

(d) ResNet

(e) Neg-mapping



ResNet

Neg-mapping

## Deep detail network

---

入力のrangeも狭くしてしまえ→ネットワークの入力を画像の高周波成分とする  
雨画像を高周波成分(detail)と低周波成分(base)の和としてモデリング

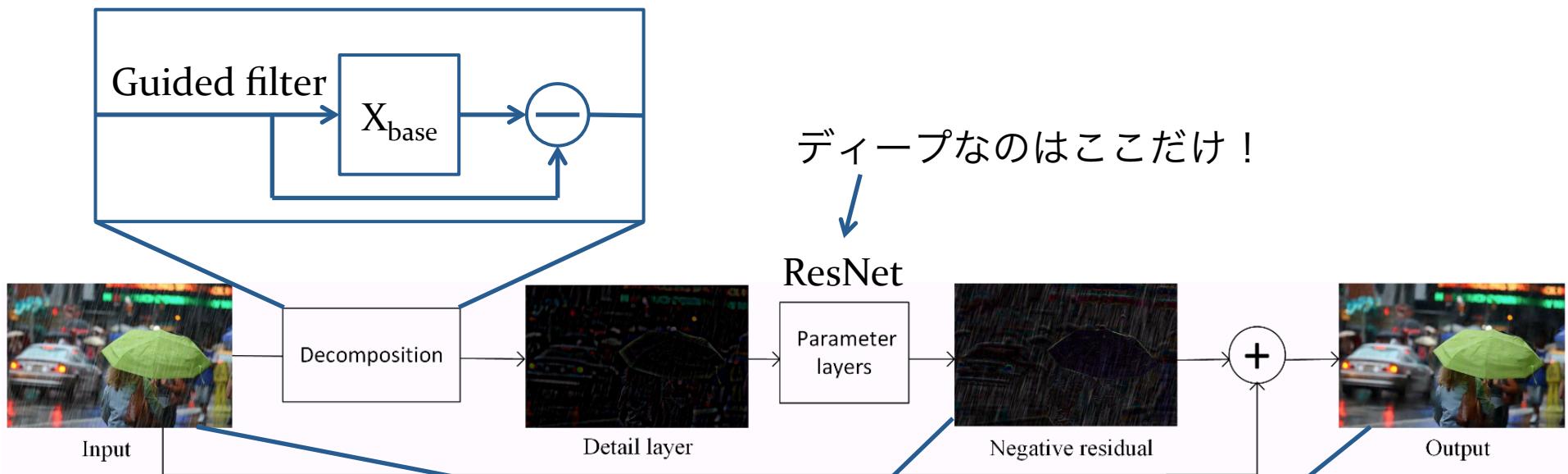
$$\mathbf{X} = \mathbf{X}_{\text{detail}} + \mathbf{X}_{\text{base}} \quad (3)$$

guided filterを用いて低周波成分(base)を抽出し、元画像から引く

$$\mathbf{X}_{\text{detail}} = \mathbf{X} - \mathbf{X}_{\text{base}}$$

この高周波成分(detail)からY-X(residual)を推定するネットワークを構築

# Deep detail network



$$\mathcal{L} = \sum_{i=1}^N \|f(\mathbf{X}_{i,\text{detail}}, \mathbf{W}, \mathbf{b}) + \underline{\mathbf{X}_i} - \underline{\mathbf{Y}_i}\|_F^2$$

# Training/Parameter setting

---

最適化手法: 確率的勾配降下法(SGD)

データセット: 1000枚の画像に14種類の雨を合成し14000組の画像ペアを作成  
9100枚をトレーニングデータ, 3900枚をテストデータ

深層学習フレームワーク: Caffe(Matlabコードあり)

The network depth: 26

Weight decay:  $10^{-10}$

Momentum: 0.9

Minibatch size: 20

Learning rate: 0.1(0~100K iter), 0.01(100K~200K iter), 0.001(200K~210K iter)

Filter size: 3×3

Filter numbers: 16

# Results on synthetic test data



SSIM

Images	Ground truth	Rainy image	Method [25]	Method [24]	Ours
girl	1	0.65	0.71	0.80	<b>0.90</b>
flower	1	0.69	0.77	0.81	<b>0.92</b>
umbrella	1	0.75	0.80	0.82	<b>0.86</b>
4,900 test images	1	$0.78 \pm 0.12$	$0.83 \pm 0.09$	$0.87 \pm 0.07$	<b><math>0.90 \pm 0.05</math></b>

# Results on real-world test data



(a) Rainy images

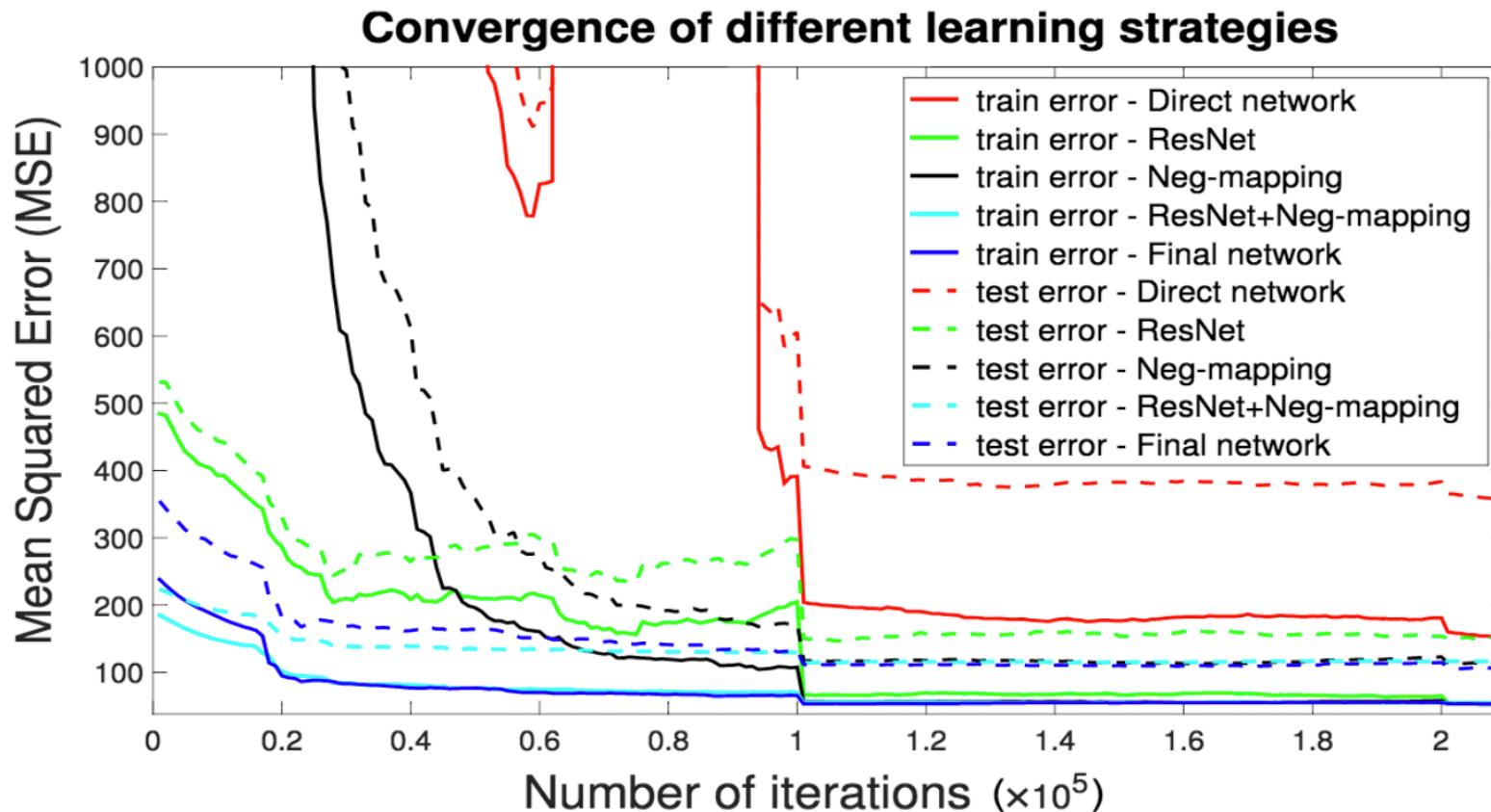
(b) Method [25]

(c) Method [24]

(d) Ours

Since no standard high-quality reference-free quantitative measure exists, we only show qualitative results for real-world data.

# Convergence of different network structures



# Network depth versus breadth

Table 3: Average SSIM using different network sizes.  
Filter numbers

	$a_1 = a_2 = 16$	$a_1 = a_2 = 32$	$a_1 = a_2 = 64$
$L = 14$	0.906	0.912	0.915
$L = 26$	0.916	0.920	0.920
$L = 50$	0.921	0.926	0.928

- Filter numbersよりdepthのほうが結果に影響している

Average SSIM on 100 synthetic testing images

depth	8	14	20	26	50
ResNet	0.896	0.904	0.909	0.907	0.917
Ours	0.896	0.906	0.915	0.916	0.921

- 差が最も大きいところを選んだ？

## Comparison with deep learning based method



(a) Rainy images



(b) [9] (SSIM=0.78)



(c) Ours (SSIM=0.86)

ディープラーニングを使った他の手法との比較

- やはりdirect network(画像から画像への直接的な写像)はあまりうまくいかないらしい
- 確かに図(b)は物体の輪郭がぼやけている

## Extension: noise and JPEG artifacts reduction

同様の手法でノイズ除去やJPEGの劣化修復もできる



## まとめ/所感

---

ネットワークの入出力画像のピクセル値がとりうる値を制限することで学習を安定させている

画像生成において、入力と得たい結果をネットワークに突っ込むだけでは良い結果は得られない

たしかに結果は良いが、定量的評価が合成画像に対してしか行えていない

わりとギリギリ通った論文？

ノイズ除去やJPEG劣化の復元が意外と汎用性を示しているのか？