

論文紹介

Parallel Multiscale Autoregressive Density Estimation

東京大学大学院理学系研究科物理学専攻 藤堂研究室

中西 健

Paper information

論文名: Parallel Multiscale Autoregressive
Density Estimation

著者: Scott Reed, et. al. (DeepMind)

公開日: 10 Mar 2017

※スライド中の図表は特に記述のない限り上記の論文から引用

この論文を選んだ動機

- PixelCNNの高速化に興味があった

概要

自己回帰モデル で いくつかのピクセル間に条件付き独立性を仮定 することで、生成にかかる計算時間を $O(N)$ から $O(\log N)$ にした (N は画像のピクセル数)

→ 自己回帰モデルで大きな画像が作れるようになった

画像生成

画像生成の方法は主に三種類

- 変分推論 (VAEなど)
- 敵対的学習 (GAN)
- 自己回帰モデル ← 今回はこれ

自己回帰モデル

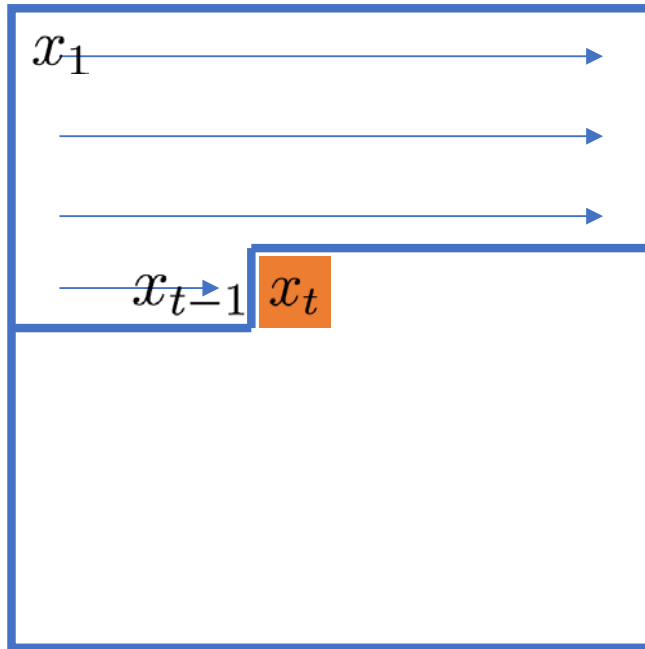
- 自己回帰モデル とは
 - 分布 $p(x_{1:T})$ を

$$p(x_{1:T}) = \prod_{t=1}^T p(x_t | x_{1:t-1})$$

のように書き下し、右辺の因子をNNなどでモデル化

画像における自己回帰モデル

- 画像における自己回帰モデル とは (e.g. PixelCNN)



$$p(x_{1:T}) = \prod_{t=1}^T p(x_t | x_{1:t-1})$$

- 画像の上から下に、行ごとに左から右に生成
- channel方向はRGBの順に生成
- 生成し終えたデータはすべて次の予測に使ってよい

画像における自己回帰モデル

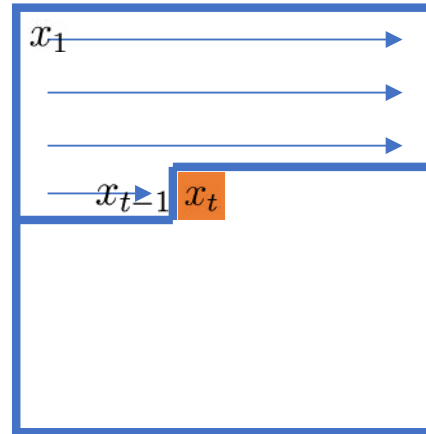
- 画像における自己回帰モデルの良い点
 - 画像の密度推定でSOTA
 - 学習が並列化できるので高速
- 画像における自己回帰モデルの悪い点
 - 生成に非常に時間がかかる
 - (参考)リアルタイム生成動画 <https://github.com/PrajitR/fast-pixel-cnn>
 - 画像のピクセル数を N として、生成にかかる時間は $O(N)$



本論文: 生成時間を $O(\log N)$ にした

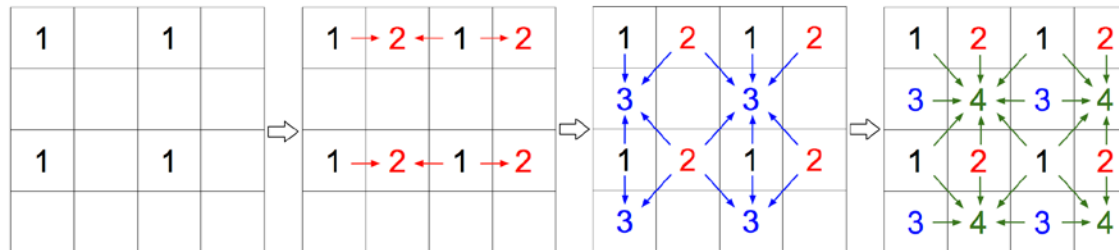
本論文の手法

PixelCNN



本論文の提案手法

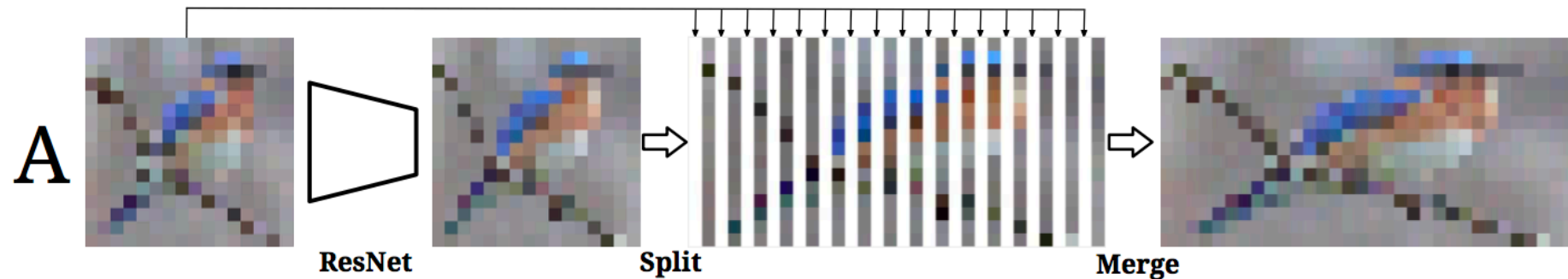
粗い画像から緻密な画像にしていく



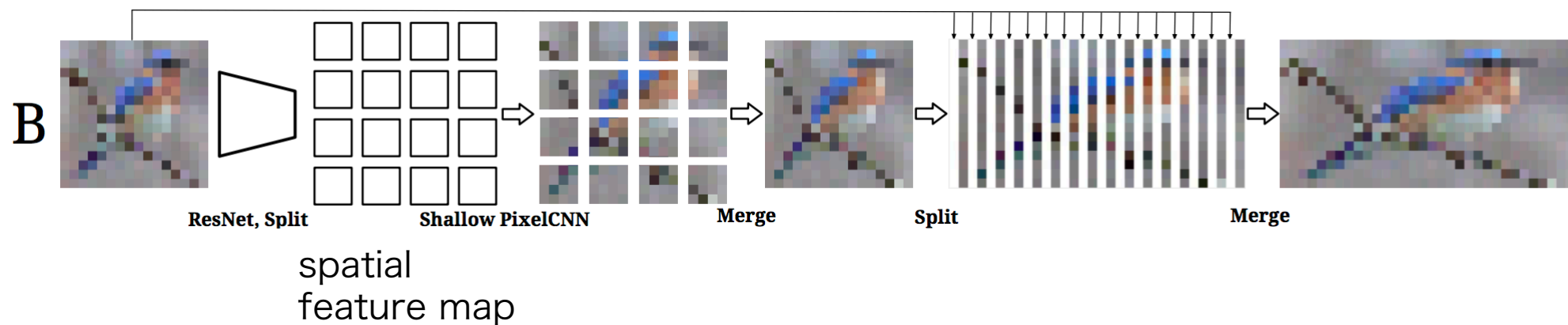
どちらもchannel方向はRGBの順に生成

本論文の手法

(A) Simplest version



(B) Sophisticated version



実験

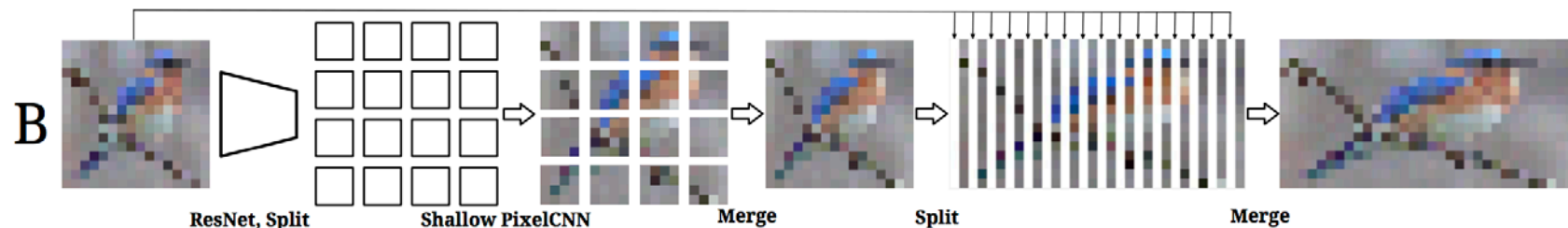
- クラス条件付き画像生成
 - Imagenetを使用
- キャプションからの画像生成 (今回は省略)
 - CUB(鳥の画像データセット)を使用
 - 他にもMPII, MS-COCOを用いて同様の実験をしている
- アクション条件付き動画生成 (今回は省略)
 - Robot Pushingを使用

クラス条件付き画像生成実験

dataset: ImageNet (1000クラス, 約100万枚)

Sophisticated versionのモデルで画像拡大

- 12層のResNet
- 4層のPixelCNN
- 隠れ層のユニット数はすべて256
- 8x8の画像から128x128まで拡大していく



結果

生成された画像

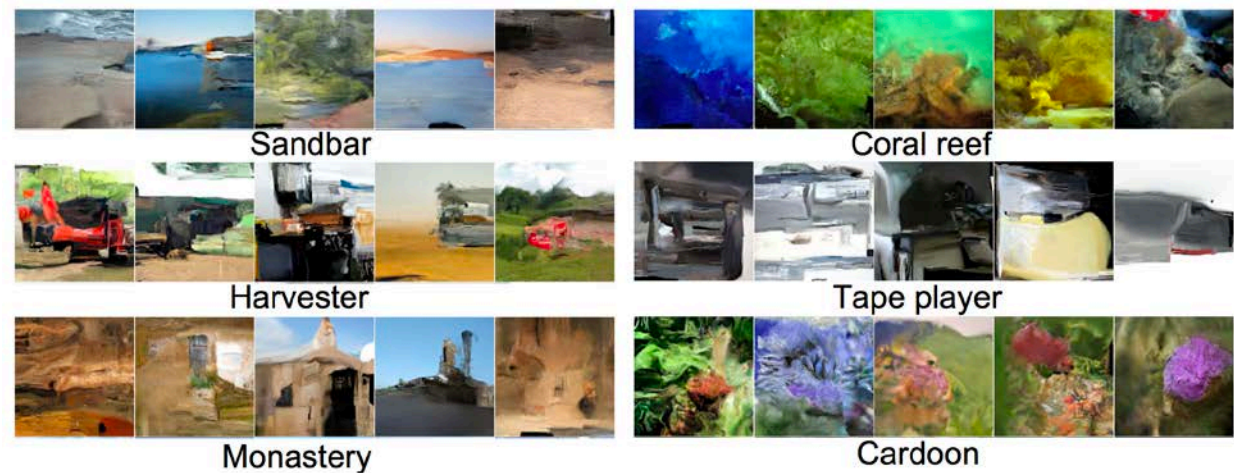


Figure 8. Class-conditional 128×128 samples from a model trained on ImageNet.

負の対数尤度

Model	32	64	128
PixelRNN	3.86 (3.83)	3.64(3.57)	-
PixelCNN	3.83 (3.77)	3.57(3.48)	-
Real NVP	4.28(4.26)	3.98(3.75)	-
Conv. DRAW	4.40(4.35)	4.10(4.04)	-
Ours	3.95(3.92)	3.70(3.67)	3.55(3.42)

)← 生成にかかる時間 $O(N)$

← 生成にかかる時間 $O(\log N)$

Table 3. ImageNet negative log-likelihood in bits per sub-pixel at 32×32 , 64×64 and 128×128 resolution.

- 自己回帰モデルでないモデルには勝っている

結果 (計算速度)

計算速度の比較

Model	scale	time	speedup
$O(N)$ PixelCNN	32	120.0	1.0×
$O(\log N)$ PixelCNN	32	1.17	102×
$O(\log N)$ PixelCNN, in-graph	32	1.14	105×

- 32x32の画像生成ですら100倍程度の高速化

まとめと展望

- PixelCNNの画像生成高速化方法を提案
 - 生成速度が $O(N) \rightarrow O(\log N)$ になった (N: ピクセル数)
- 応用先
 - テキストからの画像生成
 - ビデオ生成
 - 超解像

追加資料

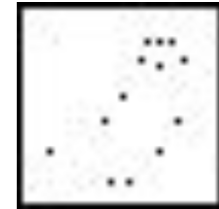
キャプションからの画像生成実験

- CUB
 - 200種の鳥の画像
 - 11788枚
 - 各画像に10個のキャプション
 - 各画像に15個のKeypoints

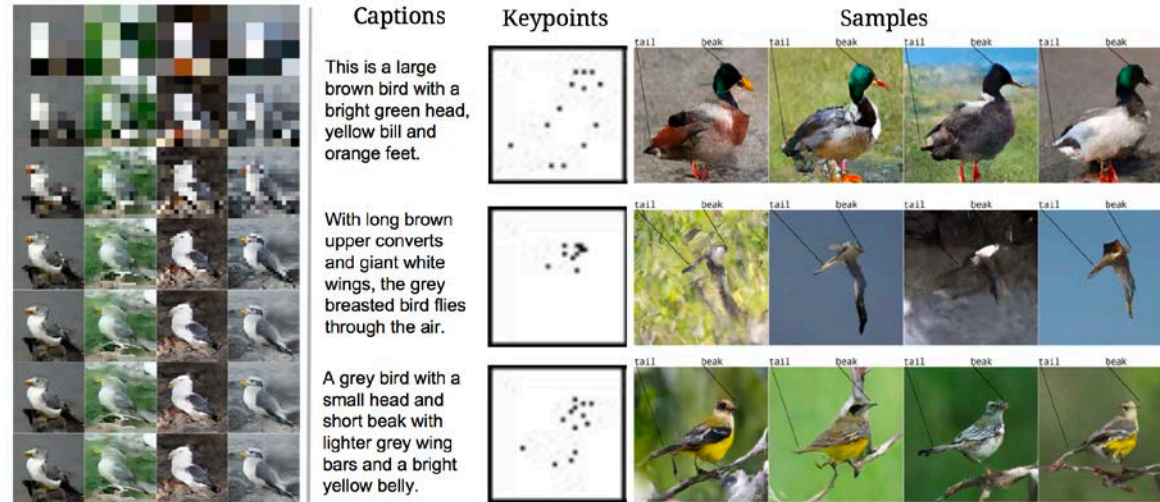
Captions

This is a large brown bird with a bright green head, yellow bill and orange feet.

Keypoints



結果



A white large bird with orange legs and gray secondaries and primaries, and a short yellow bill.

Figure 4. Text-to-image bird synthesis. The leftmost column shows the entire sampling process starting by generating 4×4 images, followed by six upscaling steps, to produce a 256×256 image. The right column shows the final sampled images for several other queries. For each query the associated part keypoints and caption are shown to the left of the samples.

CUB	Train	Val	Test
PixelCNN	2.91	2.93	2.92
Multiscale PixelCNN	2.98	2.99	2.98

Table 1. Text and structure-to image negative conditional log-likelihood in nats per sub-pixel.