

# 事前学習用画像データモジュール コンテストレポート

**FYSignate1009**

# 開発したモジュールの概要

項目	記入
工夫した点・新規性	<ul style="list-style-type: none"><li>・フラクタル組み合わせの改良(付録1) 既存OSSの組み合わせ方法は分類問題に適していない。 1クラス内で使用するIFSパラメータを固定、フラクタルが集合しやすいデータ拡張へ修正することで、複合フラクタルの複雑さとバリエーションを高めた。</li><li>・フラクタル着色方法の改良(付録2) 既存OSSのカラーフラクタルは不自然な色合いで、事前学習したモデルのフィルタも自然画像を用いた場合と異なる。 色付け方法を調整し、極端な配色を防ぐことで、自然画像を用いたモデルに近いフィルタを構築した。 予選のデータセット規模では自然画像を用いた場合と同程度の転移性能に達した(図5左)。</li><li>・Refined Data Augmentationの導入(付録3) 人工画像は自然画像に比べてバリエーションが低く、本選のデータセット規模ではかなり早い段階で学習が収束する。 それを防ぐために事前学習をRefined Data Augmentationの序盤フェーズとみなし、MixUpを導入した。 中規模データセット(500クラスx各500枚)において、最終的には自然画像と同程度の分類精度に達した(図5右)。</li></ul>

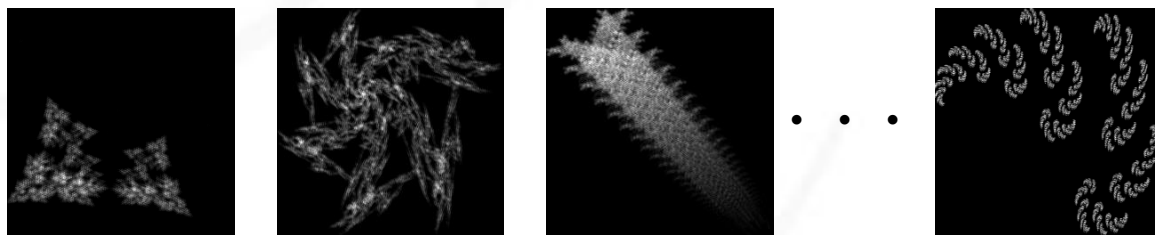
# 開発したモジュールの概要

項目	記入
既存OSSの改良 or 新規作成	既存OSSの改良 <a href="https://catalys1.github.io/fractal-pretraining/">https://catalys1.github.io/fractal-pretraining/</a>
既存OSSの改良の場合、 改良点・優位性	<ul style="list-style-type: none"><li>・色付け方法の改良</li><li>・フラクタル組み合わせ方法の改良</li><li>・MixUpの追加</li><li>・並列処理の追加</li></ul> MixUp元画像をスレッド並列で作成

# 提出モジュールの概要

generator.py/get\_params関数

必要なフラクタル数分のIFSパラメータを探索。



係数制限をかけて形状が整ったものに限定。  
IFSパラメータの探索方法は既存手法と同じ。

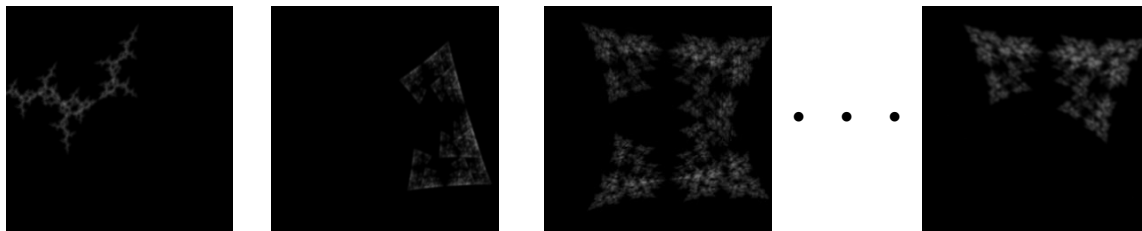
generator.py/generate関数

複数のフラクタルを組み合わせたMulti-Fractalクラスを2つ作成、MixUpする。

1. IFSパラメータの取得

1クラスあたりのフラクタル種類数(3)xベースクラス(2)=6のIFSパラメータを取得。

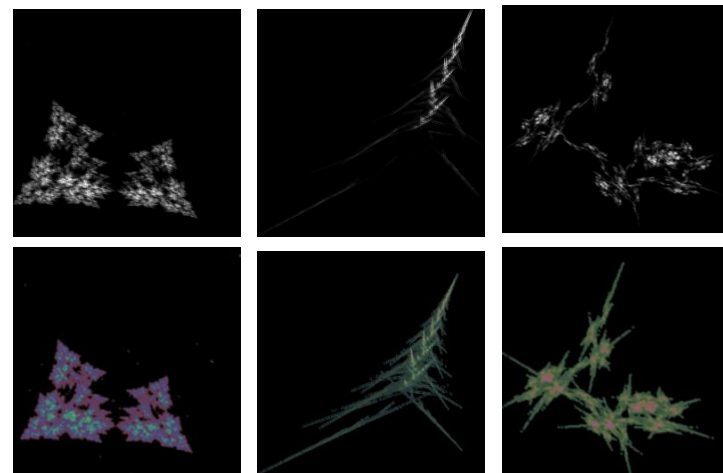
2. JITありでCacheサイズ分のフラクタル座標を計算。



パラメータにJIT(微小変動)を  
加えて形状変化。

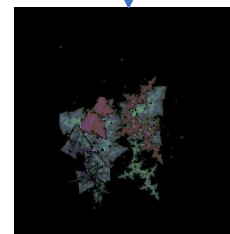
# 提出モジュールの概要

3. フラクタルのカラーリング。  
座標濃度(フラクタル座標->画像サイズ->Patchカウント->正規化)を元にHSV空間で疑似的に色付け。
4. 複数のフラクタル画像をデータ拡張、組み合わせ。
5. 背景画像を作成。  
Diamond-Squareを同様に色付け
6. 背景へ前景を重畳描画。
7. もう一方のベース画像も同様に作成。
8. 2つのベース画像をMixUp。



Cacheから選択。  
座標濃度をもとに色付け

データ拡張(Rotate,  
Flip, Resize, Random配置)  
を加えつつ、組み合わせ

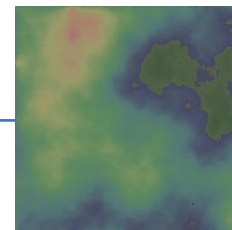
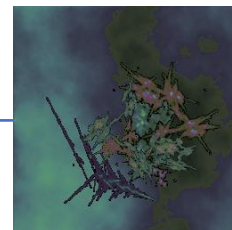
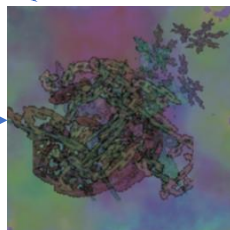
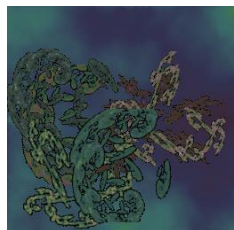


背景画像

MixUp画像

前景を埋め込み

もう一方のベース  
クラスも同様に作成



# 付録1(フラクタル組み合わせ方法の改良)

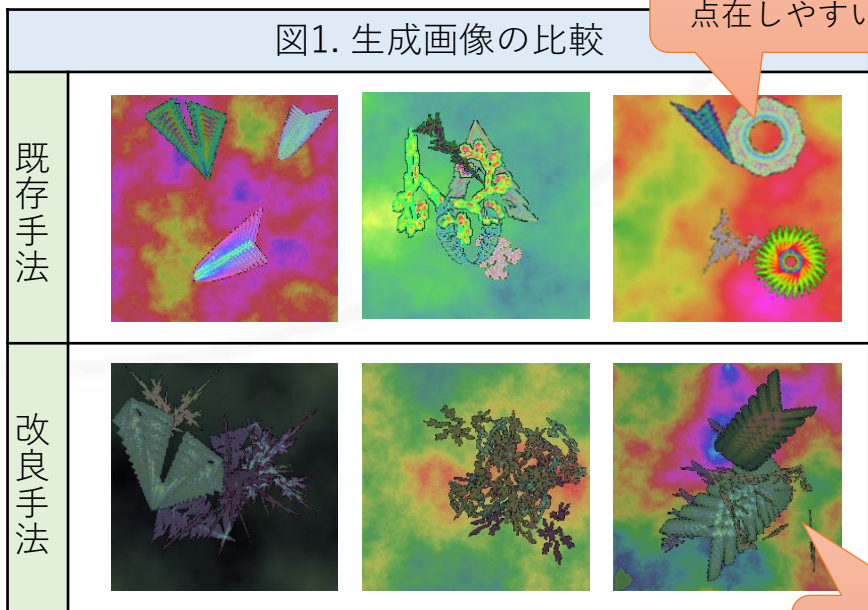
既存文献では複数のフラクタルを組み合わせたMulti-Instanceのデータセットが最も精度が高い。しかし、組み合わせるフラクタルは全クラスから無作為で選択し、縮小配置するため多ラベル分類や物体検出に向いている画像となる(図1上)。

フラクタルの集合を1つの物体とみなして、多クラス分類に適した画像を生成した(図1下)。

クラス毎に使用フラクタルの種類を固定、集合しやすいデータ拡張へ改良した(表1)。

フラクタルをデータ拡張して組み合わせることでクラスとしての複雑さとバリエーションを高めた。

図1. 生成画像の比較



フラクタルが  
点在しやすい

フラクタルの  
塊ができやすい

表1. MultiFractalの作成方法

変更点	既存手法	改良手法
IFSパラメータ	全クラスで共有	クラス毎に固有
JIT有無	無し	有り
見切れ有無	半分まで許容	無し
リサイズ係数	0.2-0.6	0.4-0.6
組み合わせ個数	1-5	4-6
画像サイズ	224	256
その他	-	Rotate追加

# 付録2(フラクタル着色方法の改良)

既存の色付け方法は全体的に明るい画像で、フラクタルも不自然に目立つ。  
HSV空間で色付けを行う関数を修正し、暗めでわかりづらい画像を生成した(図2中央)。  
事前学習モデルの第1畳み込み層(Conv1)を可視化すると、色変化への反応が低下した(図2右)。  
CCA類似度も自然画像(ImageNet500)を用いたものに近くなったため(表2)、改良手法ではより自然画像のエッジ抽出に適したフィルタが学習できたと思われる。  
(※後述のMixUpまで行くとさらに改善する。)

図2. 色付け方法の違い			
	着色方法	生成画像	First-layer Filters
既存手法	色相： 乱数初期値+ 座標濃度で変化 彩度、明度： 乱数固定値		
改良手法	色相、彩度、明度： 乱数初期値+ スケーリング係数+ 座標濃度で変化 パラメータ試行錯誤		

全体的に暗め

表2. 事前学習モデルのConv1のCCA類似度	
データセット種類	CCA類似度
ImageNet500	-
VisualAtom	0.123
MultiFractal(既存)	0.045
MultiFractal(改良)	0.063
MultiFractal(改良MixUp)	0.121

若干改善

※中規模データセット(500クラスx500枚)、ResNet50、10epochで事前学習した結果。  
※ImageNet500はImageNetの後半500クラスを各500枚に限定、256x256へリサイズしたもの。

カラフルさが減少



## 付録3(Refined Data Augmentationの導入)

人工データセットはバリエーションに乏しいため、中規模の事前学習では自然画像に比べてかなり早く学習が進む(図3)。

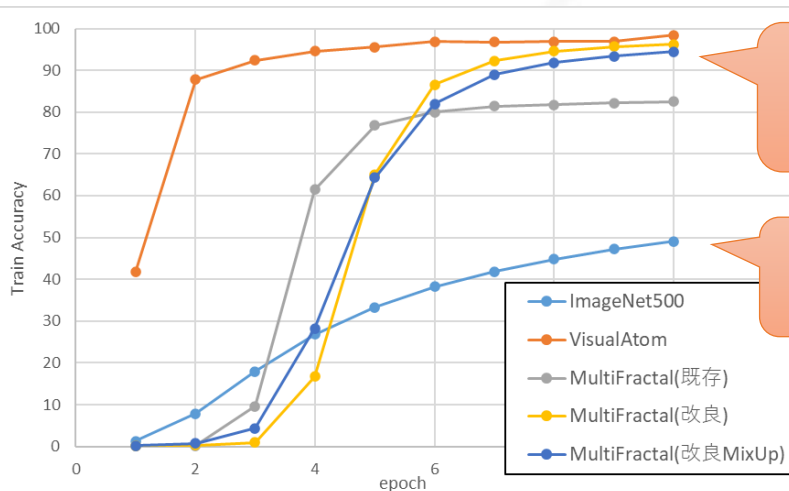
しかし、Multi-Fractalのパラメータ調整や通常の水データ拡張だけで複雑さを上げると、クラスの特徴が発散・曖昧になり、転移性能が悪くなる。

クラス内の一貫性を保ちつつ、多様性も確保するため、MixUpを行った(図4)。

MixUpは非常に効果的で、転移性能以外の観点でも良い結果が得られた(付録4)。

Refined Data Augmentationのように大域探索から局所探索へつなげることが出来たと思われる。

図3. 事前学習の正解率の推移



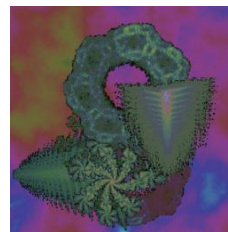
MixUp版は顕著な変化ないが、転移性能は向上

自然画像はゆっくり上昇

※事前学習の条件は前ページと同じ。

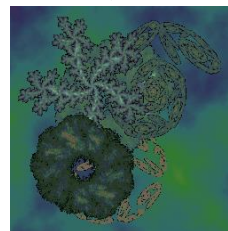
図4. MixUp画像の作成

Baseクラス1

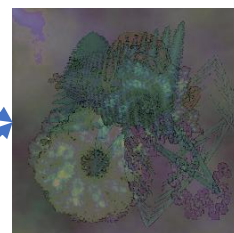


複雑さの上限あり

Baseクラス2



MixUpクラス



クラスの特徴を維持 + 多様性を確保

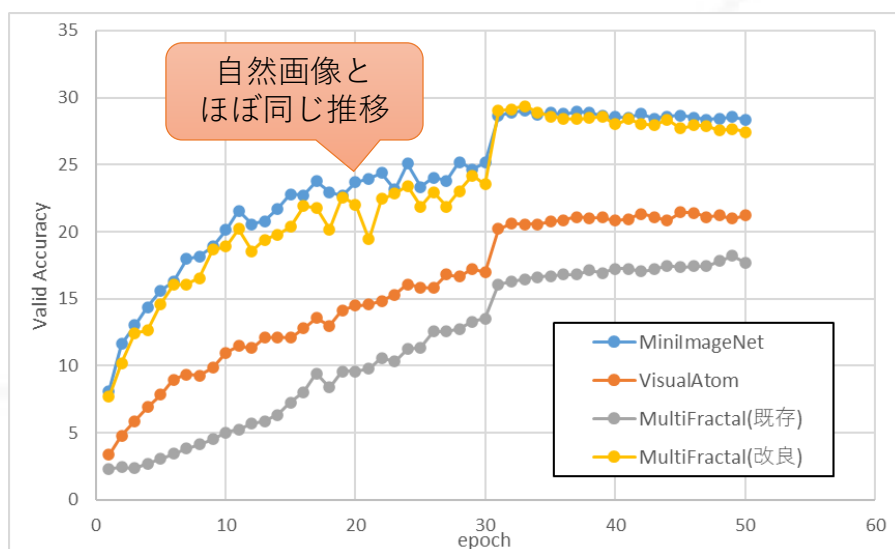


## 付録4(転移性能の評価)

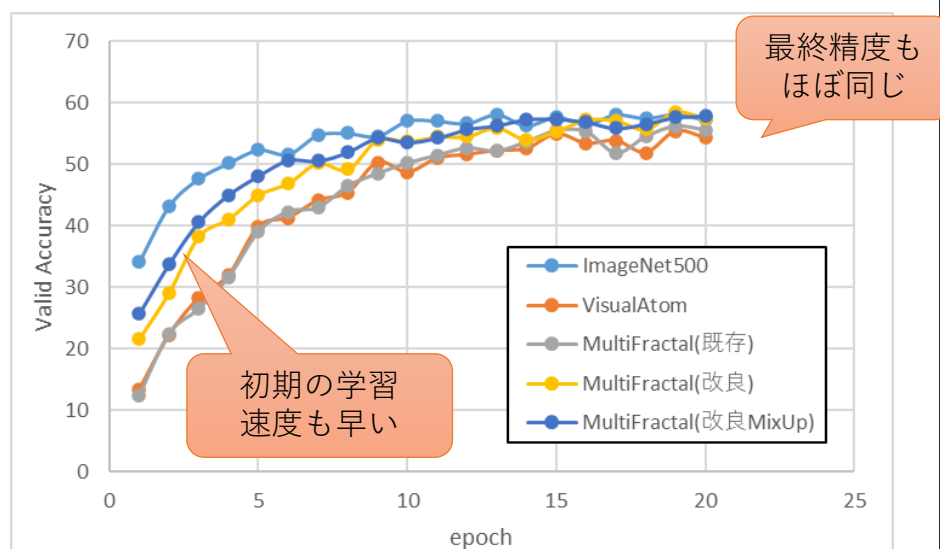
既存手法と提案手法で転移性能(速度と精度)の比較を行った。  
予選規模のデータセットでは自然画像を用いた場合と同程度の転移性能を達成した(図5左)。  
中規模のデータセットでも既存手法やVisualAtomを上回り、最終的な正解率も自然画像を用いた場合と同程度になった(図5右)。

図5. 事前学習データセットによる転移性能の比較(検証データの正解率の推移)

小規模検証(Food101, 100クラスx100枚)



中規模検証(ImageNet500\_1, 500クラスx500枚)



※小規模検証の事前学習はResNet50、50epoch。MiniImageNetとFood101は各クラス100枚に限定、256x256ヘリサイズ。

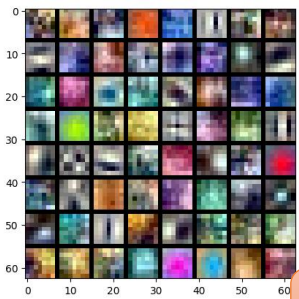
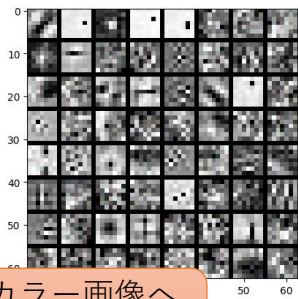
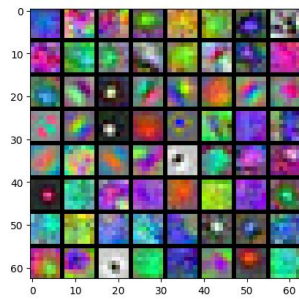
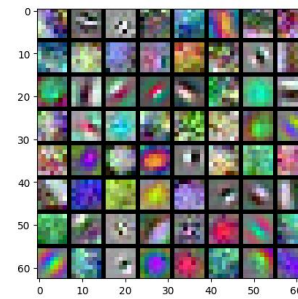
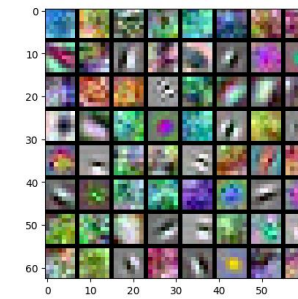
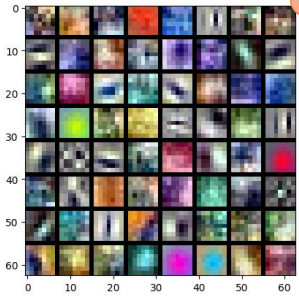
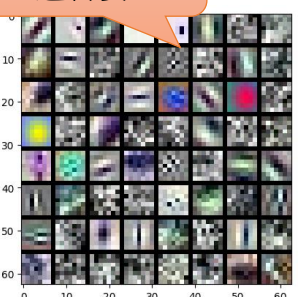
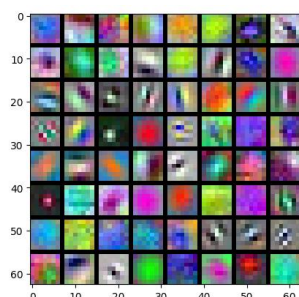
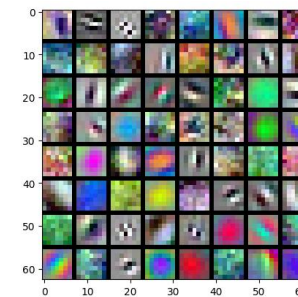
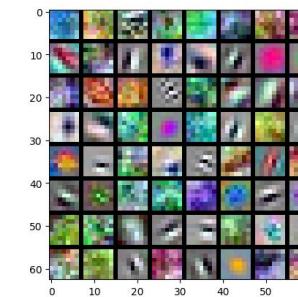
※中規模検証の事前学習の条件は前ページと同じ。ImageNet500\_1はImageNetの前半500クラスを各500枚に限定、256x256ヘリサイズ。

## 付録4(転移性能の評価)

転移学習前後での第1畳み込み層を比較した。

ImageNet500とMultiFractal系では明確な変化はないが、VisualAtomは色の反応が出ている(図6)。グレー画像で事前学習した場合、カラー画像に合わせてフィルタの再学習が必要となる。

図6. 転移学習前後の第1畳み込み層の比較

	ImageNet500	VisualAtom	MultiFractal (既存)	MultiFractal (改良)	MultiFractal (改良MixUp)
事前学習 モデル					
転移学習 モデル					

カラー画像へ  
適合要

## 付録4(転移性能の評価)

転移学習前後でResNet50の第1畳み込み層(Conv1)とLayer1-4のCCA類似度を比較した(図7)。ImageNet500とMultiFractal(改良、改良MixUp)のConv1はCCA類似度が高く、変化が少ない。事前学習のフィルタを再利用しながら、後段Layerの適合で効率良く転移学習を進めている。特にMultiFractal(改良MixUp)は転移先の自然画像のエッジ抽出に十分適したフィルタが構築できたと考えられる。

VisualAtomとMultiFractal(既存)はConv1も含めて全体的にCCA類似度が低い。グレイ画像や不自然なカラー画像で事前学習すると、フィルタの再学習が必要となり、初動も悪く最終的な分類精度も若干低い(図5右)。転移先がカラー画像の場合、今回の提案手法はある程度の有効性があると考えられる。

フィルタ  
の再利用

図7. 転移学習前後の各層のCCA類似度

