



セグメンテーション, 教師無し 学習, Zero Shot

最新技術を知る → 各自が創造力, 問題解決力を発揮する

URL: <https://www.kkaneko.jp/a/2023.html>

金子邦彦



セグメンテーション



セグメンテーションは、**画像の画素が、どの物体や領域に属しているかを識別すること**



セマンティック・セグメンテーション
画素単位で、
領域名を示すラベルを割り当て



Transformer

- **自然言語処理**のために考案された技術
- **言語モデル**（GPT, BERT など）で利用されている。 **画像認識**にも応用が始まっている。
- **Attention Mechanizm（注意機構）**を特徴。 **入力データの各部分**が、 **どの程度重要であるか**を自動的に決定。 **重要な部分に重点を置いて、特徴量を抽出**
- 2017年発表

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention Is All You Need, arXiv:1706.03762, 2017.



DiNAT-L

- **画像認識**に使われる**ディープニューラルネットワーク**
- **Transformer** の技術をベース
- 特徴：動的な Attention を特徴. 入力に応じて Attention を動的に変更できる. **より重要な特徴を抽出可能.**
 - 犬の画像：自動的に犬の顔や目にフォーカス
- 2022年発表

Kai Liu, Tianyi Wu, Cong Liu, Guodong Guo, Dynamic Group Transformer: A General Vision Transformer Backbone with Dynamic Group Attention, arXiv:2203.03937, 2022.

DiNAT-L を実行できるのデモサイト



- オンラインのサイト

URL: <https://huggingface.co/spaces/shi-labs/OneFormer>

- セグメンテーションの種類：パノプティック, インスタンス, セマンティック
- データセット：COCO (133 クラス), Cityscapes (19 クラス), ADE20K (150クラス)



文献: Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, Humphrey Shi, OneFormer: One Transformer to Rule Universal Image Segmentation, arXiv:2211.06220, 2022.

DiNAT-L (教師有り学習) の動作結果



シヨシ

COCO データセットで学習済み. DiNAT-L, セマンティックセグメンテーション



元画像



実行結果



元画像



実行結果

DiNAT-L (教師有り学習) の動作結果



COCO データセットで学習済み. DiNAT-L, セマンティックセグメンテーション



元画像



実行結果



元画像



実行結果



学習の種類

- 教師有り学習

正解ラベル付きのデータセットで学習。

(例) 「犬」という**正解ラベル**の付いた画像を使用。大量のデータを利用することで、高い精度を達成

- 教師無し学習

正解ラベルの無いデータセットで学習。

(例) **正解ラベル**の無い画像を使用。データのパターンを自己学習することにより、特徴抽出、クラスタリング、次元削減などを実行。

- Few Shot

少数の正解ラベル付きのデータセットで学習。

(例) 「犬」という**正解ラベル**の付いた画像数枚を使用

- Zero Shot

正解ラベル付きのデータセットで学習。正解ラベルにない新しいラベルについての認識能力を持つ

(例) 「犬」という**正解ラベル**の付いた画像が無くても「犬」を識別

CutLER



- **CutLER**は、**物体検出**と**インスタンスセグメンテーション**を行うための**教師無し学習アルゴリズム**
- **物体の部分が存在するべき領域を推定**することで、**物体の位置を検出**。これにより、教師あり学習による物体の位置検出が不要に。
- 元の画像から物体の部分を復元することにより、インスタンスセグメンテーションを実行。
- 2023年発表

Wang, Xudong and Girdhar, Rohit and Yu, Stella X and Misra, Ishan, Cut and Learn for Unsupervised Object Detection and Instance Segmentation, arXiv preprint arXiv:2301.11320, 2023.

CutLER (教師無し) の動作結果



元画像



実行結果



元画像



実行結果

CutLER (教師無し) の動作結果



元画像



実行結果



元画像



実行結果

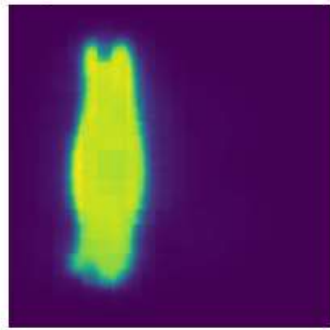
CLIPSeg



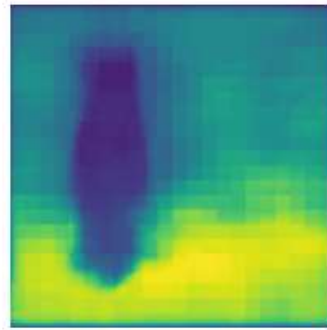
- **Zero Shot** の**セグメンテーション**を行うためのディープニューラルネットワーク
- **CNN** を使用：画像の特徴抽出
- **Transformer** を使用：画像とテキストの関連性の学習
- 画像の特徴量と、文章の特徴量を組み合わせての**セグメンテーション**を行う。
- 2021年発表

Timo Lüddecke, Alexander S. Ecker, Image Segmentation Using Text and Image Prompts, arXiv:2112.10003, 2021.

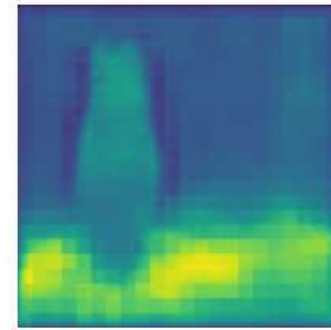
CLIPSeg (Zero Shot) の動作結果



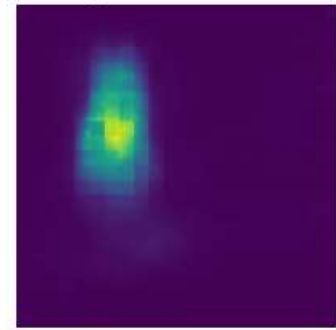
dog



ice

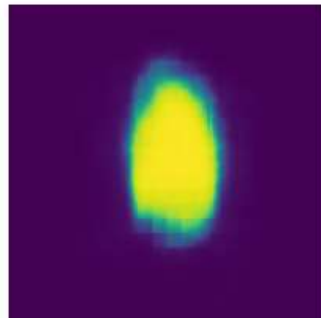


cold



running

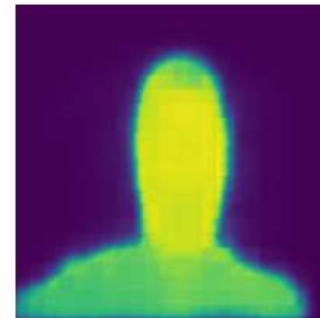
元画像



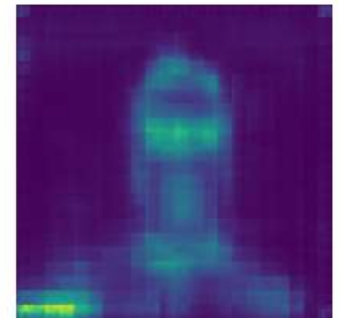
man



happy



man



woman

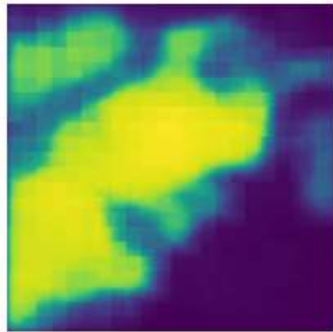
元画像

実行結果

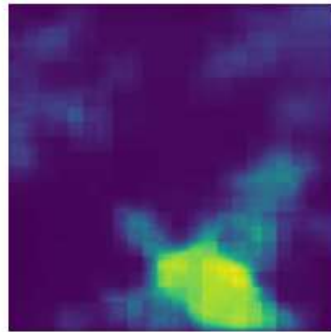
CLIPSeg (Zero Shot) の動作結果



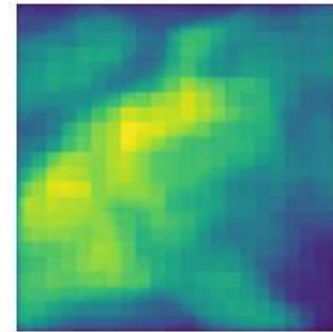
元画像



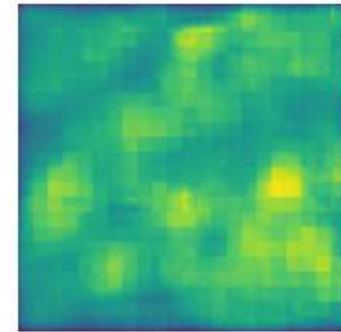
building



road



developed

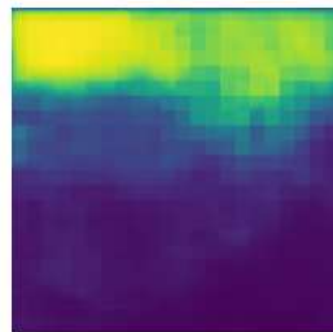


beautiful

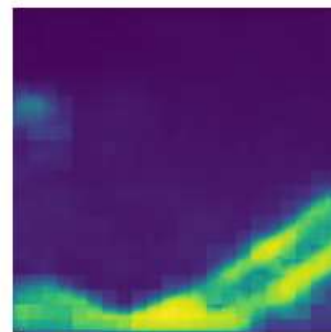
実行結果



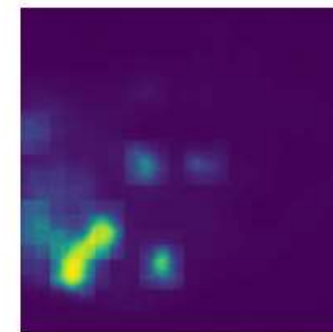
元画像



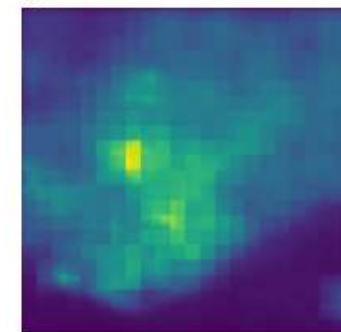
mountain



river



sand



grass

実行結果



- DiNAT-L, Swin-L オンラインイデモ

<https://huggingface.co/spaces/shi-labs/OneFormer>

- CutLER インストールして, コマンドで使用

<https://www.kkaneko.jp/ai/win/cutler.html>

- CLIPSeg

インストールは `python -m pip install transformers`

プログラムは, 次のページで公開されているものを使用

<https://huggingface.co/blog/clipseg-zero-shot>

CLIPSeg のプログラム



```
from PIL import Image
import requests
```

```
fname = "sample1.jpg"
prompts = ["face", "human"]
```

```
from transformers import CLIPSegProcessor, CLIPSegForImageSegmentation
```

```
processor = CLIPSegProcessor.from_pretrained("CIDAS/clipseg-rd64-refined")
model = CLIPSegForImageSegmentation.from_pretrained("CIDAS/clipseg-rd64-refined")
```

```
image = Image.open(fname)
import torch
```

```
inputs = processor(text=prompts, images=[image] * len(prompts), padding="max_length", return_tensors="pt")
# predict
with torch.no_grad():
    outputs = model(**inputs)
preds = outputs.logits.unsqueeze(1)
```

```
import matplotlib.pyplot as plt
```

```
image
```

```
_, ax = plt.subplots(1, len(prompts) + 1, figsize=(3*(len(prompts) + 1), 4))
[a.axis('off') for a in ax.flatten()]
ax[0].imshow(image)
[ax[i+1].imshow(torch.sigmoid(preds[i][0])) for i in range(len(prompts))];
[ax[i+1].text(0, -15, prompt) for i, prompt in enumerate(prompts)];
```




- AI には、**さまざまな種類**が存在
- AIを試して**失敗した場合**でも、「AIは役に立たない」「AIの**精度が悪い**」と決めつけることは**避ける**.
根拠をしっかりと確認.
- AIの種類によって、得意とすることや想定外のことが異なる。
さまざまな種類を知る。**チャレンジ精神**も重要. .
- AIの学習には時間がかかる。時間をかける価値がある。時間をかけて、より良い結果を得ましょう。