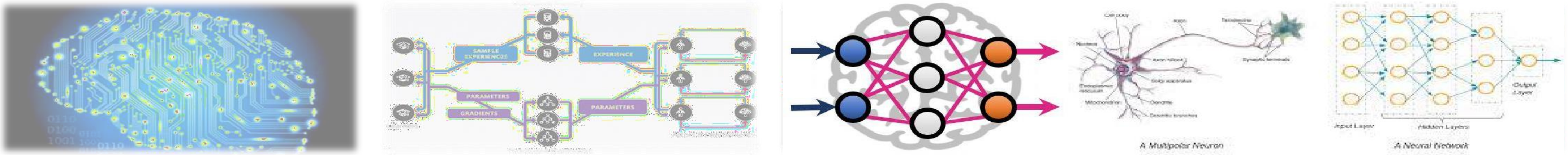


Hypernym Discovery Search

Hypernym Discovery 조사

Geonil Yun,
rjsdlf45@gmail.com

2021. 12. 21.

Overview

- **Hypernym Discovery**
  - MAP ,MRR metric function evaluation
  - Comparison corpus preprocessing

- **Scene Text Recognition**
  - PGNet
  - Issue

Hypernym Discovery

- **Score evaluation**

Result | **AP** | **MAP**

① Test

1	True	True	True
2	True	True	True

[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]

[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]

→ **1.0**

② Test

1	True	True	False
2	True	True	False

[1.0, 1.0, 0.66, 0.66, 0.66, 0.66, 0.66, 0.66, 0.66 , 0.66 , 0.66 , 0.66 , 0.66 , 0.66 , 0.66]

[1.0, 1.0, 0.66, 0.66, 0.66, 0.66, 0.66, 0.66, 0.66 , 0.66 , 0.66 , 0.66 , 0.66 , 0.66 , 0.66]

→ **0.711**

③ Test

1	True	False	False
2	False	True	False

[1.0, 0.5, 0.33, 0.33 , 0.33 ,0.33 , 0.33 ,0.33 , 0.33 ,0.33 , 0.33 ,0.33 , 0.33 ,0.33 , 0.33]

[0.0, 0.5, 0.33, 0.33 , 0.33 ,0.33 , 0.33 ,0.33 , 0.33 ,0.33 , 0.33 ,0.33 , 0.33 ,0.33 , 0.33]

→ **0.355**

Hypernym Discovery

- **Score evaluation**

Result | **RR** | **MRR**

① Test

1	True	True	True
2	True	True	True

RR: 1, 1 → MRR: **1.0**

MAP	1.0
MRR	1.0
P@1	1.0

② Test

1	True	True	False
2	True	True	False

RR: 1, 1 → MRR: **1.0**

MAP	0.71
MRR	1.0
P@1	1.0

③ Test

1	True	False	False
2	False	True	False

RR: 1, 1/2 → MRR: **0.75**

MAP	0.35
MRR	0.75
P@1	0.33

Hypernym Discovery

- **Preprocessing**

  - **Implementation**

    - Use original corpus of UMBC

" The DeLorme PN-20 represents a new breed of GPS device. Do you… "

↓ sentence tokenizing

" The DeLorme PN-20 represents a new breed of GPS devices. "
"Do you know your family members?"
…

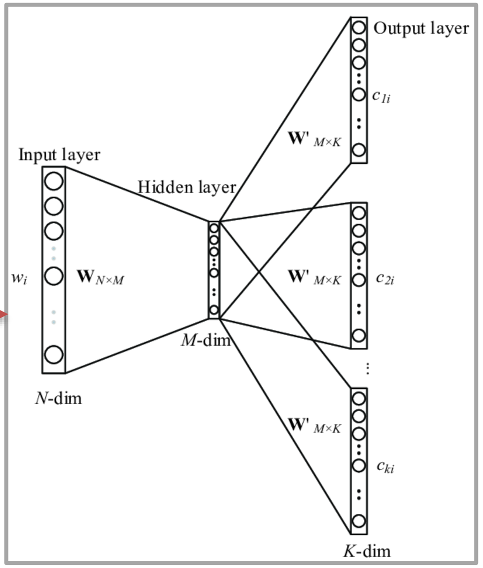
↓ word tokenizing

" The", "DeLorme" "PN-20" "represents", …

⊕ combination words (single word to trigram)
convert to lower case

vocabulary

training



Skip gram model

Hypernym Discovery

- **Preprocessing**

  - **NLP_HZ : Nearest Neighbor Approach**

① Lowercase all the words for corpus and train/gold/test data.
② Concatenate the phrase with underline

executive president → **executive_president**

Multiple phrase in one sentence.

"~ , vice executive president ~" → "~ , vice **executive_president** ~"

"~ , vice executive president ~" → "~ , **vice_executive_president** ~"

Hypernym Discovery

- **Preprocessing**

  - **UMDuluth-CS8761**

    - Use pre-processed corpus (POS tagging)
    - Search words with *noun* part-of-speech tag and bigram or trigram phrases with a noun head word

  - **Normalized Corpus**

    - POS tagged input corpus is processed per paragraph
    - Converted to **lower-case text**
    - Obtain **bigram, trigram noun phrases** by using **POS tags** of each word
    - Removing **punctuation marks** and words with POS tags other than ***noun, verb, adverb*** or ***adjective***

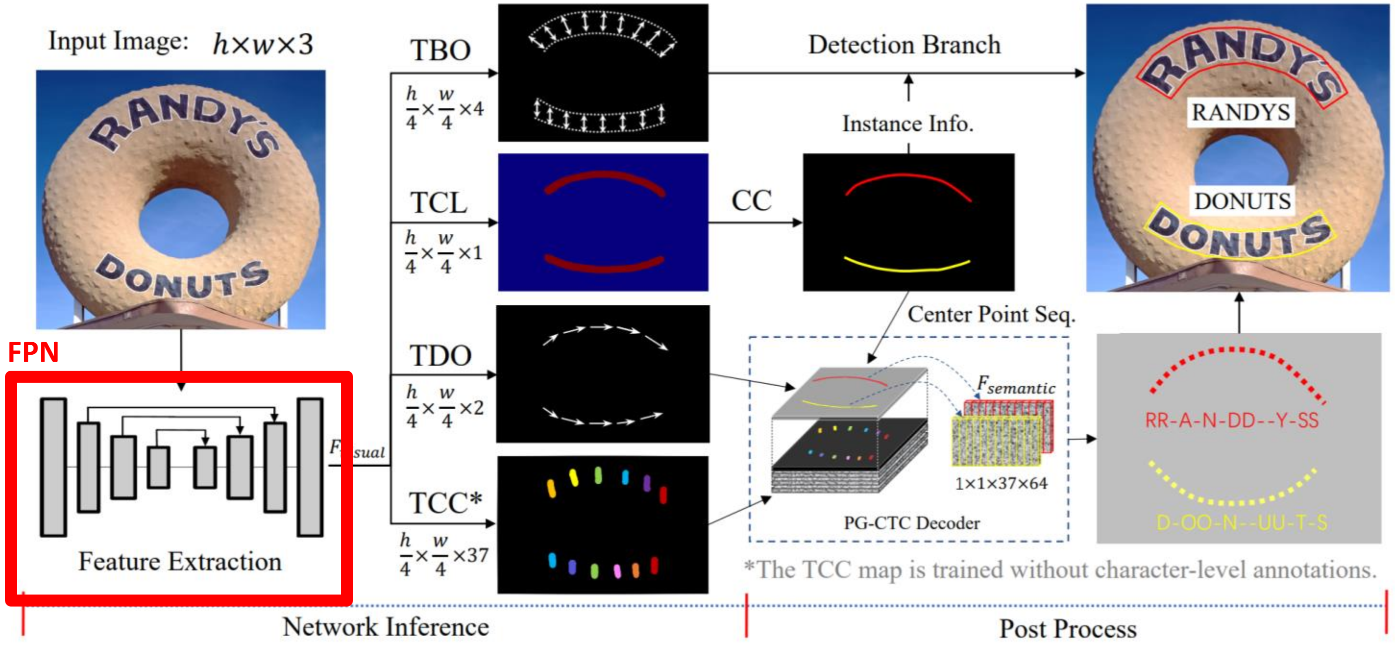
Future

- **Future works**

  - Increase window size of embeddings
  - Removing punctuation mark and other words

Scene Text Recognition

- **PGNet(Accuracy, Efficient)**
  - **Text Spotting**

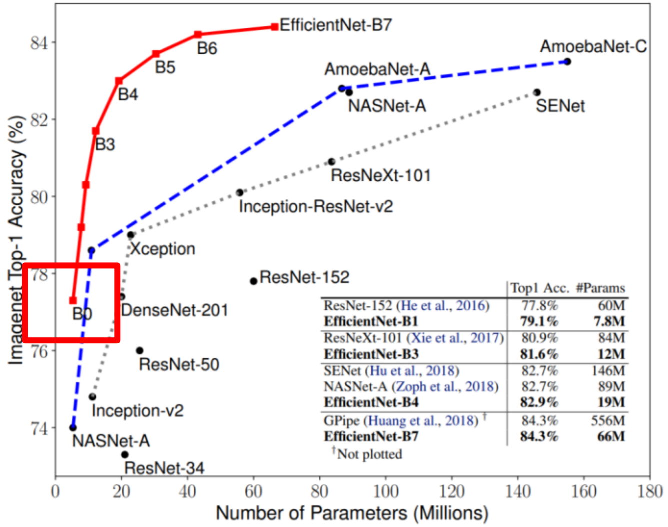


- PGNet-**Accuracy** : FPN backbone for **ResNet50**
- PGNet-**Efficient** : FPN backbone for **EfficentNet-B0**

Scene Text Recognition

- **PGNet(Accuracy, Efficient)**
  - **EfficientNet**

Name	#Params	Top-1 Acc.	Pretrained
efficient-b0	5.3M	76.3	O
efficient-b1	7.8M	78.8	O
efficient-b2	9.2M	79.8	O
efficient-b3	12M	81.1	O
efficient-b4	19M	82.6	O
efficient-b5	30M	83.3	O
efficient-b6	43M	84.0	O
efficient-b7	66M	84.4	O



  - Efficient-B0 for Backbone FPN

Scene Text Recognition

- **PGNet(Accuracy, Efficient)**
  - **EfficientNet**

Table 1. **EfficientNet-B0 baseline network** – Each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. Notations are adopted from equation 2.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
EfficientNet-B0	**77.1%**	**93.3%**	**5.3M**	**1x**	**0.39B**	**1x**
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	**79.1%**	**94.4%**	**7.8M**	**1x**	**0.70B**	**1x**
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	**80.1%**	**94.9%**	**9.2M**	**1x**	**1.0B**	**1x**
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	**81.6%**	**95.7%**	**12M**	**1x**	**1.8B**	**1x**
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	**82.9%**	**96.4%**	**19M**	**1x**	**4.2B**	**1x**
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	**83.6%**	**96.7%**	**30M**	**1x**	**9.9B**	**1x**
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	**84.0%**	**96.8%**	**43M**	**1x**	**19B**	**1x**
EfficientNet-B7	**84.3%**	**97.0%**	**66M**	**1x**	**37B**	**1x**
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

We omit ensemble and multi-crop models (Hu et al., 2018), or models pretrained on 3.5B Instagram images (Mahajan et al., 2018).

Scene Text Recognition

- **PaddlePaddle (ISSUE)**

  - **Parallel Distributed Deep Learning : Baidu**
  - **Deep Learning Framework**

Porting

PaddlePaddle → PyTorch

- Porting for used in embedded board
- PaddlePaddle to Pytorch or Tensorflow(Lite)
- Training totaltext on paddlepaddle PGNet, not working
- Implement feature extractor FPN with pytorch

Thank you