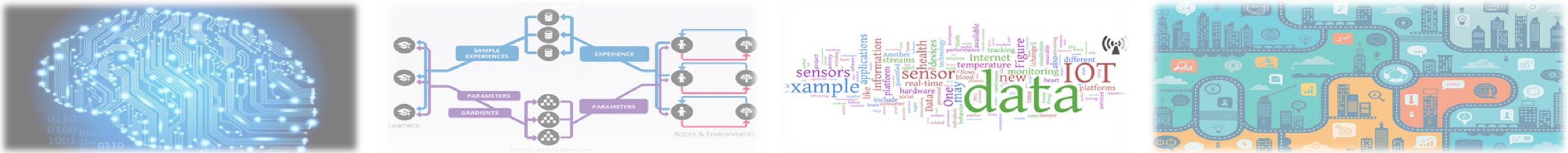


Stock Price Prediction

Tae-Won Lee

Overall Progress

AutoML

Problem

- Stock Price Prediction
- Low Accuracy

Related Works

- CNN-BiLSTM-AM
- AdvLSTM
- DTML

Experiments

- KDD 17 Dataset

Data 개수 : 2518
Feature : 6개
(Open, High, Low, Close, Volume, Adj Close)

- Metric

Accuracy

Analysis

- Need more information (Multimodal)

Proposal

- Multimodal Learning

Post-experimental analysis

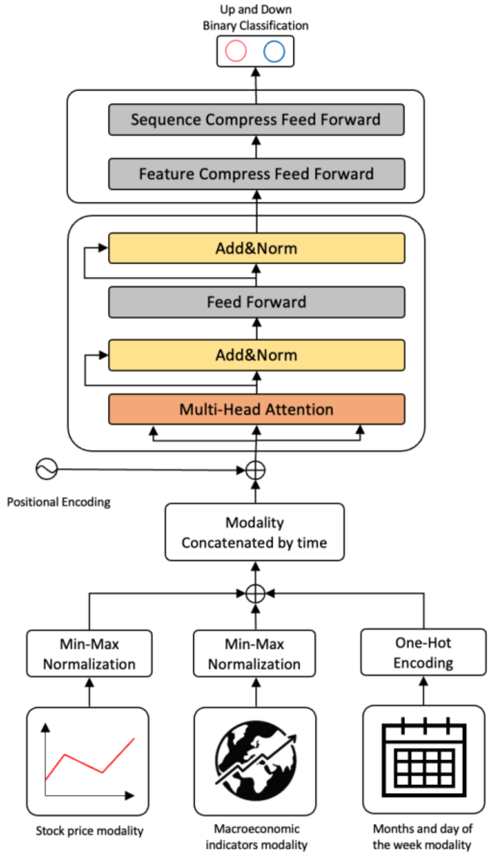
- Accuracy improvement (0.506->0.517)
- Ranking mean improvement (2.48->1.72)

Conclusion

- Accuracy improvement
- Statistically significant

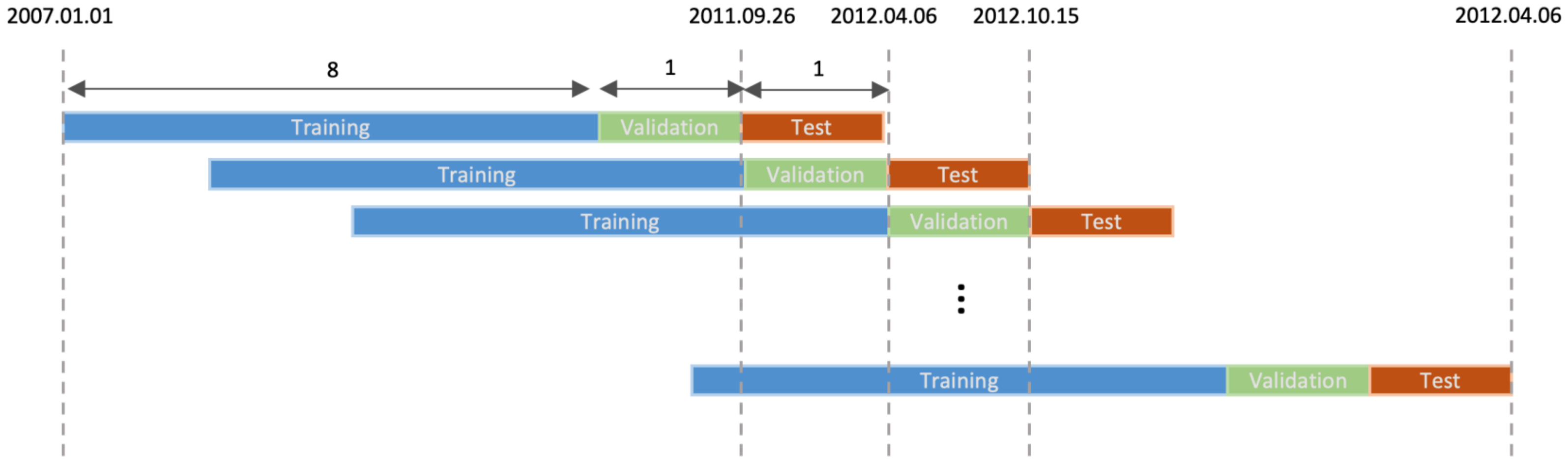
Progress of writing a Thesis

- Multimodal Transformer for stock price prediction



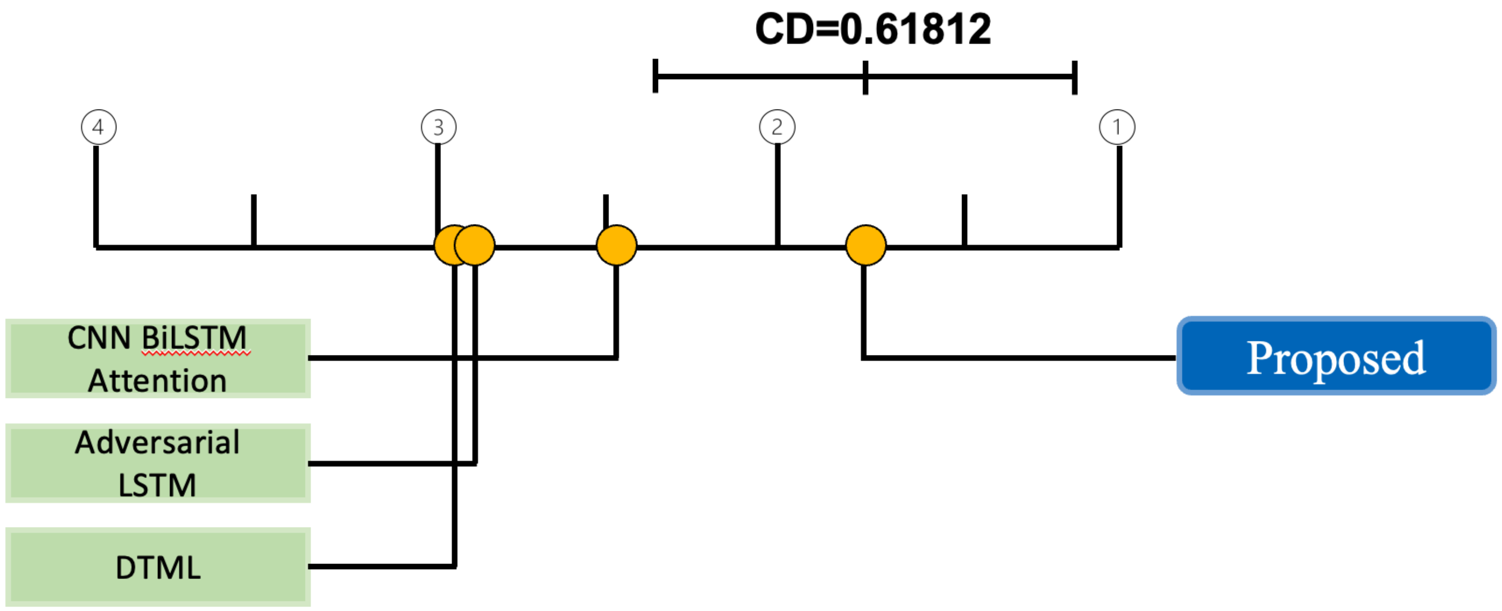
Progress of writing a Thesis

- Blocked Time series cross validation



Progress of writing a Thesis

- Bonferroni-Dunn test



Progress of writing a Thesis

- Experinet Results

TABLE 1. Classification accuracy

Ticker in dataset	Proposed Method	CNN-Attention BiLSTM [22]	Adversarial LSTM [26]	DTML [25]
AAPL	**54.47%**	48.28%	47.65%	52.03%
AMZN	52.08%	49.68%	**53.12%**	51.25%
BA	**53.75%**	52.18%	51.56%	50.62%
BAC	**54.27%**	48.43%	50.78%	51.56%
BHP	48.54%	48.12%	**50.00%**	47.81%
BRK-B	49.06%	49.21%	**50.00%**	47.18%
CHL	47.91%	50.93%	50.78%	**52.34%**
CMCSA	51.66%	**52.03%**	48.75%	49.06%
CVX	**50.72%**	50.31%	49.68%	48.12%
D	51.77%	**52.81%**	50.31%	51.09%
DCM	**53.43%**	48.90%	48.43%	52.65%
DIS	**52.91%**	51.25%	51.09%	48.75%
DOW	**51.87%**	46.40%	48.28%	50.00%
DUK	51.77%	**55.46%**	47.34%	48.28%
EXC	**52.91%**	50.78%	46.87%	49.53%
GE	**52.60%**	50.46%	52.18%	51.56%
GOOGL	51.66%	52.81%	50.93%	**52.96%**
HD	51.45%	52.03%	**52.81%**	50.46%
INTC	52.50%	**53.12%**	48.59%	51.09%
JNJ	**52.60%**	49.21%	48.28%	52.50%
JPM	**51.45%**	50.62%	50.31%	50.15%
KO	51.25%	**52.34%**	47.96%	47.34%
MA	**52.81%**	50.15%	49.21%	51.25%
MMM	**51.97%**	46.87%	51.25%	47.96%
MO	54.47%	**55.46%**	51.25%	48.90%
MRK	**50.62%**	49.84%	49.37%	49.37%
MSFT	50.10%	51.25%	49.53%	**53.43%**
NGG	**53.85%**	51.40%	51.56%	53.75%
NTT	50.93%	49.06%	51.56%	**56.71%**
NVS	**51.87%**	51.25%	50.78%	49.37%
ORCL	51.14%	**52.96%**	50.46%	50.93%
PEP	**51.45%**	48.90%	50.31%	48.75%
PFE	49.27%	**50.62%**	47.81%	48.59%
PG	51.66%	47.34%	**52.50%**	48.43%
PTR	**52.81%**	52.50%	48.43%	52.18%
RDS-B	**52.70%**	51.71%	49.68%	48.75%
RIO	50.10%	49.37%	51.71%	**52.34%**
SO	**52.29%**	51.40%	51.71%	50.00%
SPY	**55.62%**	52.34%	50.31%	47.81%
SYT	48.54%	49.68%	50.46%	**51.25%**
T	51.45%	50.46%	50.93%	**53.75%**
TM	**51.14%**	47.18%	50.00%	48.12%
TOT	51.56%	**52.18%**	52.03%	48.43%
UNH	**52.50%**	51.40%	51.56%	46.40%
UPS	**51.97%**	47.81%	49.84%	51.09%
VALE	51.35%	**51.87%**	49.21%	49.68%
VZ	**55.10%**	52.81%	51.40%	46.40%
WFC	**52.39%**	48.75%	47.65%	52.18%
WMT	50.41%	50.00%	**51.56%**	47.65%
XOM	46.87%	**51.56%**	48.90%	44.84%

Progress of writing a Thesis

- 논문 수정 방안

  - 실험 결과 Chapter 구체화
    - 통계 테스트 결과에 대한 절차 설명 추가
    - Accuracy에 대한 t-test 추가
    - 실험결과에 대한 정성적인 분석

Appendix

Experiment Result

- Red : High Accuracy
- Asterisk (**) : Low Accuracy

- Accuracy (Index : NASDAQ) – 1/3

Data	Proposed -Early fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]
AAPL	0.544792	0.52031	**0.47656	0.48281
AMZN	0.520833	0.51250	0.53125	**0.49688
BA	0.5375	**0.50625	0.51563	0.52188
BAC	0.542708	0.51563	0.50781	**0.48438
BHP	0.485417	**0.47813	0.50000	0.48125
BRK-B	0.490625	**0.47188	0.50000	0.49219
CHL	**0.479167	0.52344	0.50781	0.50938
CMCSA	0.516667	**0.49063	0.48750	0.52031
CVX	0.507292	**0.48125	0.49688	0.50313
D	0.517708	0.51094	**0.50313	0.52813
DCM	0.534375	0.52656	**0.48438	0.48906
DIS	0.529167	**0.48750	0.51094	0.51250
DOW	0.51875	0.50000	0.48281	**0.46406
DUK	0.517708	0.48281	0.47344	0.55469
EXC	0.529167	0.49531	**0.46875	0.50781
GE	0.526042	0.51563	0.52188	**0.50469
GOOGL	0.516667	0.52969	0.50938	0.52813
HD	0.514583	**0.50469	0.52813	0.52031

Experiment Result

- Red : High Accuracy
- Asterisk (**) : Low Accuracy

- Accuracy (Index : NASDAQ) – 2/3

Data	Proposed -Early fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]
INTC	0.525	0.51094	**0.48594	0.53125
JNJ	0.526042	0.52500	**0.48281	0.49219
JPM	0.514583	0.50156	**0.50313	0.50625
KO	0.5125	**0.47344	0.47969	0.52344
MA	0.528125	0.51250	**0.49219	0.50156
MMM	0.519792	0.47969	0.51250	**0.46875
MO	0.544792	**0.48906	0.51250	0.55469
MRK	0.50625	*0.49375	*0.49375	0.49844
MSFT	0.501042	0.53438	0.49531	0.51250
NGG	0.538542	0.53750	0.51563	**0.51406
NTT	0.509375	0.56719	0.51563	**0.49063
NVS	0.51875	**0.49375	0.50781	0.51250
ORCL	0.511458	0.50938	**0.50469	0.52969
PEP	0.514583	**0.48750	0.50313	0.48906
PFE	0.492708	0.48594	**0.47813	0.50625
PG	0.516667	0.48438	0.52500	**0.47344
PTR	0.528125	0.52188	**0.48438	0.52500
RDS-B	0.527083	**0.48750	0.49688	0.51719

Experiment Result

- Red : High Accuracy
- Asterisk (**) : Low Accuracy

- Accuracy (Index : NASDAQ) – 3/3

Data	Proposed -Early fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM -AM [8]
RIO	0.501042	0.52344	0.51719	0.49375
SO	0.522917	**0.50000	0.51719	0.51406
SPY	0.55625	**0.47813	0.50313	0.52344
SYT	**0.485417	0.51250	0.50469	0.49688
T	0.514583	0.53750	0.50938	0.50469
TM	0.511458	0.48125	0.50000	**0.47188
TOT	0.515625	**0.48438	0.52031	0.52188
UNH	0.525	**0.46406	0.51563	0.51406
UPS	0.519792	0.51094	0.49844	**0.47813
VALE	0.513542	0.49688	0.49219	0.51875
VZ	0.551042	**0.46406	0.51406	0.52813
WFC	0.523958	0.52188	**0.47656	0.48750
WMT	0.504167	**0.47656	0.51563	0.50000
XOM	0.46875	**0.44844	0.48906	0.51563

Experiment Result

- Red : High Accuracy
- Asterisk (**) : Low Accuracy

- Accuracy (Index : NASDAQ) – early fusion (1/3)

Data	Proposed -Early fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]	Proposed -Early fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]
SPY	0.55625	0.478125	0.503125	0.523438	1	4	3	2
VZ	0.551042	0.464063	0.514063	0.528125	1	4	3	2
AAPL	0.544792	0.520313	0.476563	0.482813	1	2	4	3
MO	0.544792	0.489063	0.5125	0.554688	2	4	3	1
BAC	0.542708	0.515625	0.507813	0.484375	1	2	3	4
NGG	0.538542	0.5375	0.515625	0.514063	1	2	3	4
BA	0.5375	0.50625	0.515625	0.521875	1	4	3	2
DCM	0.534375	0.526563	0.484375	0.489063	1	2	4	3
DIS	0.529167	0.4875	0.510938	0.5125	1	4	3	2
EXC	0.529167	0.495313	0.46875	0.507813	1	3	4	2
MA	0.528125	0.5125	0.492188	0.501563	1	2	4	3
PTR	0.528125	0.521875	0.484375	0.525	1	3	4	2
RDS-B	0.527083	0.4875	0.496875	0.517188	1	4	3	2
GE	0.526042	0.515625	0.521875	0.504688	1	3	2	4
JNJ	0.526042	0.525	0.482813	0.492188	1	2	4	3
INTC	0.525	0.510938	0.485938	0.53125	2	3	4	1
UNH	0.525	0.464063	0.515625	0.514063	1	4	2	3
WFC	0.523958	0.521875	0.476563	0.4875	1	2	4	3

Experiment Result

- Red : High Accuracy
- Asterisk (**) : Low Accuracy

- Accuracy (Index : NASDAQ) – early fusion (2/3)

Data	Proposed -Early fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]	Proposed -Early fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]
SO	0.522917	0.5	0.517188	0.514063	1	4	2	3
AMZN	0.520833	0.5125	0.53125	0.496875	2	3	1	4
MMM	0.519792	0.479688	0.5125	0.46875	1	3	2	4
UPS	0.519792	0.510938	0.498438	0.478125	1	2	3	4
DOW	0.51875	0.5	0.482813	0.464063	1	2	3	4
NVS	0.51875	0.49375	0.507813	0.5125	1	4	3	2
D	0.517708	0.510938	0.503125	0.528125	2	3	4	1
DUK	0.517708	0.482813	0.473438	0.554688	2	3	4	1
CMCSA	0.516667	0.490625	0.4875	0.520313	2	3	4	1
GOOGL	0.516667	0.529688	0.509375	0.528125	3	1	4	2
PG	0.516667	0.484375	0.525	0.473438	2	3	1	4
TOT	0.515625	0.484375	0.520313	0.521875	3	4	2	1
HD	0.514583	0.504688	0.528125	0.520313	3	4	1	2
JPM	0.514583	0.501563	0.503125	0.50625	1	4	3	2
PEP	0.514583	0.4875	0.503125	0.489063	1	4	2	3
T	0.514583	0.5375	0.509375	0.504688	2	1	3	4
VALE	0.513542	0.496875	0.492188	0.51875	2	3	4	1
KO	0.5125	0.473438	0.479688	0.523438	2	4	3	1

Experiment Result

- Red : High Accuracy
- Asterisk (**) : Low Accuracy

- Accuracy (Index : NASDAQ) – early fusion (3/3)

Data	Proposed -Early fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]	Proposed -Early fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]
ORCL	0.511458	0.509375	0.504688	0.529688	2	3	4	1
TM	0.511458	0.48125	0.5	0.471875	1	3	2	4
NTT	0.509375	0.567188	0.515625	0.490625	3	1	2	4
CVX	0.507292	0.48125	0.496875	0.503125	1	4	3	2
MRK	0.50625	0.49375	0.49375	0.498438	1	3	3	2
WMT	0.504167	0.476563	0.515625	0.5	2	4	1	3
MSFT	0.501042	0.534375	0.495313	0.5125	3	1	4	2
RIO	0.501042	0.523438	0.517188	0.49375	3	1	2	4
PFE	0.492708	0.485938	0.478125	0.50625	2	3	4	1
BRK-B	0.490625	0.471875	0.5	0.492188	3	4	1	2
BHP	0.485417	0.478125	0.5	0.48125	2	4	1	3
SYT	0.485417	0.5125	0.504688	0.496875	4	1	2	3
CHL	0.479167	0.523438	0.507813	0.509375	4	1	3	2
XOM	0.46875	0.448438	0.489063	0.515625	3	4	2	1

Data	Proposed -Early fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]	Proposed -Early fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]
NASDAQ	0.517563	0.500969	0.501375	0.506344	1.72	2.92	2.86	2.48

Experiment Result

- 특정 모델이 잘 한 종목

early fusion		
SPY	SPDR S&P 500	S&P 500지수를 이용한 ETF
VZ	버라이즌 커뮤니케이션스	통신산업 관련기업(통신사)
AAPL	애플	휴대폰 및 컴퓨터
MO	알트리아그룹	담배(말보로), 와인 판매
BAC	뱅크오브아메리카	은행
NGG	National Grid plc	전기 가스 등 에너지 회사(수송)
BA	보잉	항공기 제조
DCM		
DIS	월트 디즈니	애니메이션 제작사
EXC	Exelon Corporation	에너지 회사
MA	마스터카드	카드회사
PTR	PetroChina Company Limited	중국 에너지 회사
RDS-B		
GE	제너럴 일렉트릭	전력 인프라 기업
JNJ	존슨앤존슨	제약사
INTC	인텔	종합반도체회사
UNH	유나이티드헬스	보험회사
WFC	웰스파고	은행

Related Work

- Modality1 data

data	
Stock price data (7개)	High, Low, Open, Volume, Adj Close, upndown, change
Technical indicator (10개)	위의 Stock data로부터 만들어진 10개의 feature - Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. Expert Systems with Applications. Yakup Kara, Melek Acar Boyacioglu, Ömer Kaan Baykan. 2011

- Modality2 data

data	
Month Data (12개)	월 정보 데이터를 One-Hot Encoding을 사용하여 추가
Day of the week Data (5개)	요일 정보를 One-Hot Encoding을 사용하여 추가

Related Work

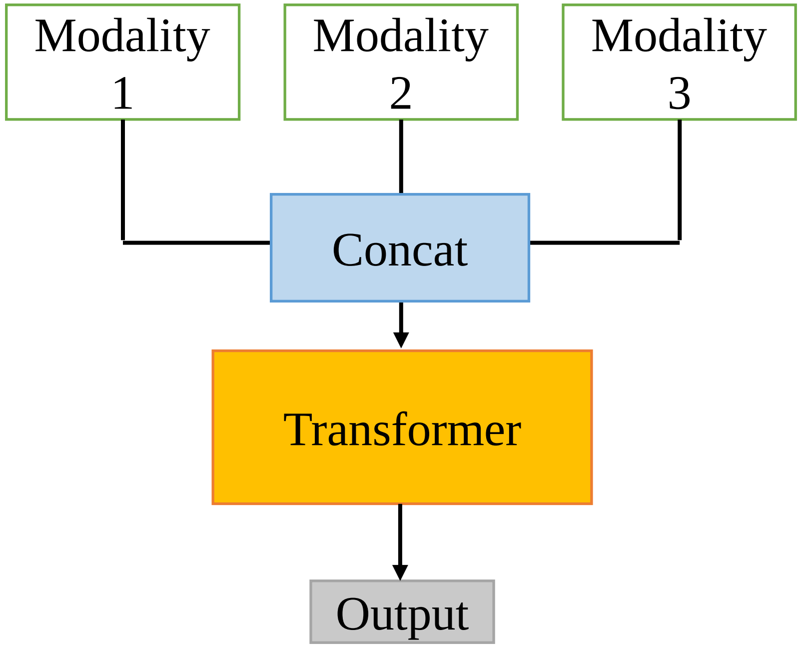
- Modality3 data

data	
nasdaq100	나스닥 상위 100개 종목의 가치의 합을 지수화 한 것
미국 2년물 채권 수익률	‘’
미국 10년물 채권 수익률	‘’
미국 30년물 채권 수익률	‘’
미국 달러 지수	‘’
WTI Oil Price	미국 텍사스산 원유 가격

- 3개의 Modality
- 각 17개,17개, 6개의 feature
- 총 40개를 입력으로 사용

Proposal: *Multimodal Learning*

- Early Fusion (or data fusion)



1)

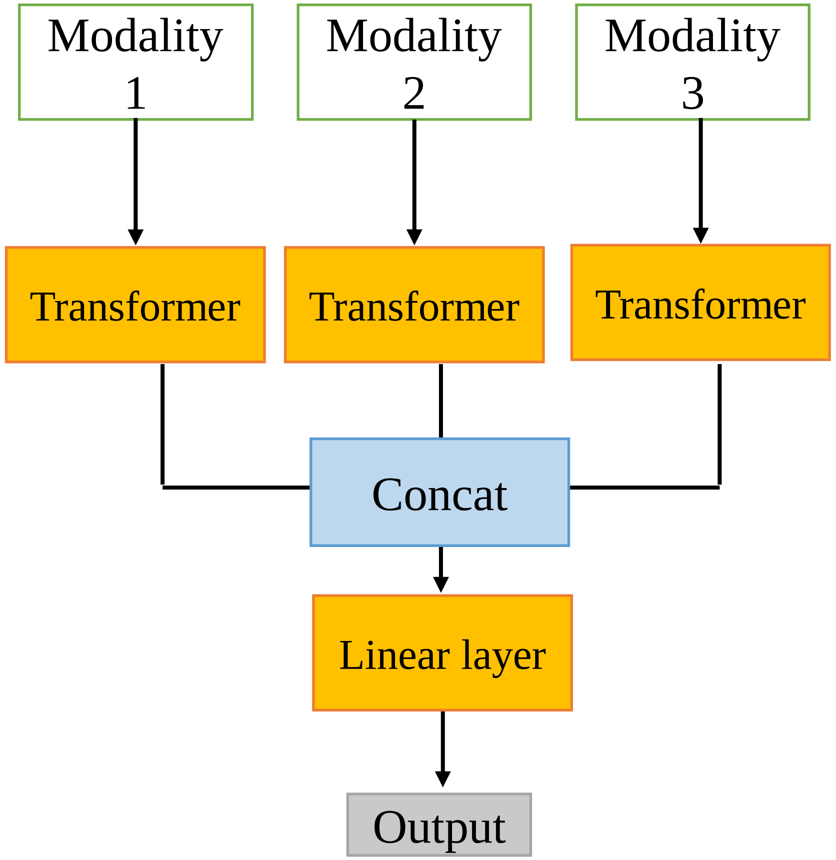
- Combine the data by removing the correlation between two sensors.

2)

- Uses PCA, canonical correlation and independent component analysis

Proposal: *Multimodal Learning*

- Late Fusion



- 개별 모델을 통과한 후에 데이터를 합치는 방식으로, 가장 많이 사용되는 방식임

Experiment Result

- Accuracy (Index : NASDAQ)
  - 비교모델 중 가장 좋은 정확성을 보인 모델과 Late fusion의 정확성의 차이가 큰 순서로 정렬

Data	Proposed -Early fusion Transformer	Proposed -Late fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]	비교모델 중 최대 Accuracy – Late fusion accuracy
PEP	0.514583	0.53854	0.4875	0.50313	0.48906	0.03541
MRK	0.50625	0.53021	0.49375	0.49375	0.49844	0.03177
MA	0.528125	0.54271	0.5125	0.49219	0.50156	0.03021
BAC	0.542708	0.54375	0.51563	0.50781	0.48438	0.02812
RDS-B	0.527083	0.54063	0.4875	0.49688	0.51719	0.02344
SPY	0.55625	0.54271	0.47813	0.50313	0.52344	0.01927
DIS	0.529167	0.52813	0.4875	0.51094	0.5125	0.01563
BHP	0.485417	0.51563	0.47813	0.5	0.48125	0.01563
MMM	0.519792	0.52708	0.47969	0.5125	0.46875	0.01458
NVS	0.51875	0.52604	0.49375	0.50781	0.5125	0.01354
PFE	0.492708	0.51979	0.48594	0.47813	0.50625	0.01354
BRK-B	0.490625	0.50833	0.47188	0.5	0.49219	0.00833
BA	0.5375	0.52813	0.50625	0.51563	0.52188	0.00625
PTR	0.528125	0.53021	0.52188	0.48438	0.525	0.00521
CVX	0.507292	0.50833	0.48125	0.49688	0.50313	0.0052
AAPL	0.544792	0.525	0.52031	0.47656	0.48281	0.00469
MO	0.544792	0.55625	0.48906	0.5125	0.55469	0.00156
TOT	0.515625	0.52292	0.48438	0.52031	0.52188	0.00104

Experiment Result

- Accuracy (Index : NASDAQ)
  - 비교모델 중 가장 좋은 정확성을 보인 모델과 Late fusion의 정확성의 차이가 큰 순서로 정렬

Data	Proposed -Early fusion Transformer	Proposed -Late fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]	비교모델 중 최대 Accuracy – Late fusion accuracy
GE	0.526042	0.52188	0.51563	0.52188	0.50469	0
JPM	0.514583	0.50521	0.50156	0.50313	0.50625	-0.00104
EXC	0.529167	0.50625	0.49531	0.46875	0.50781	-0.00156
SO	0.522917	0.51563	0.5	0.51719	0.51406	-0.00156
UNH	0.525	0.5125	0.46406	0.51563	0.51406	-0.00313
CMCSA	0.516667	0.51563	0.49063	0.4875	0.52031	-0.00468
TM	0.511458	0.49375	0.48125	0.5	0.47188	-0.00625
WMT	0.504167	0.50833	0.47656	0.51563	0.5	-0.0073
D	0.517708	0.51979	0.51094	0.50313	0.52813	-0.00834
NGG	0.538542	0.52708	0.5375	0.51563	0.51406	-0.01042
KO	0.5125	0.5125	0.47344	0.47969	0.52344	-0.01094
UPS	0.519792	0.49792	0.51094	0.49844	0.47813	-0.01302
JNJ	0.526042	0.51146	0.525	0.48281	0.49219	-0.01354
HD	0.514583	0.51458	0.50469	0.52813	0.52031	-0.01355
WFC	0.523958	0.50833	0.52188	0.47656	0.4875	-0.01355
DCM	0.534375	0.51146	0.52656	0.48438	0.48906	-0.0151
AMZN	0.520833	0.51458	0.5125	0.53125	0.49688	-0.01667
DUK	0.517708	0.53438	0.48281	0.47344	0.55469	-0.02031

Experiment Result

- Accuracy (Index : NASDAQ)
  - 비교모델 중 가장 좋은 정확성을 보인 모델과 Late fusion의 정확성의 차이가 큰 순서로 정렬

Data	Proposed -Early fusion Transformer	Proposed -Late fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM -AM [8]	비교모델 중 최대 Accuracy – Late fusion accuracy
ORCL	0.511458	0.50729	0.50938	0.50469	0.52969	-0.0224
CHL	0.479167	0.50104	0.52344	0.50781	0.50938	-0.0224
DOW	0.51875	0.47708	0.5	0.48281	0.46406	-0.02292
SYT	0.485417	0.48854	0.5125	0.50469	0.49688	-0.02396
GOOGL	0.516667	0.50417	0.52969	0.50938	0.52813	-0.02552
INTC	0.525	0.50417	0.51094	0.48594	0.53125	-0.02708
VZ	0.551042	0.49896	0.46406	0.51406	0.52813	-0.02917
PG	0.516667	0.49375	0.48438	0.525	0.47344	-0.03125
RIO	0.501042	0.49167	0.52344	0.51719	0.49375	-0.03177
VALE	0.513542	0.48125	0.49688	0.49219	0.51875	-0.0375
MSFT	0.501042	0.49479	0.53438	0.49531	0.5125	-0.03959
XOM	0.46875	0.46979	0.44844	0.48906	0.51563	-0.04584
T	0.514583	0.4875	0.5375	0.50938	0.50469	-0.05
NTT	0.509375	0.51563	0.56719	0.51563	0.49063	-0.05156

Experiment Result

- Accuracy (Index : NASDAQ)
  - 비교모델 중 가장 좋은 정확성을 보인 모델과 Early fusion의 정확성의 차이가 큰 순서로 정렬

Data	Proposed -Early fusion Transformer	Proposed -Late fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]	비교모델 중 최대 Accuracy – Early fusion accuracy
SPY	0.55625	0.54271	0.47813	0.50313	0.52344	0.03281
BAC	0.542708	0.54375	0.51563	0.50781	0.48438	0.027078
AAPL	0.544792	0.525	0.52031	0.47656	0.48281	0.024482
VZ	0.551042	0.49896	0.46406	0.51406	0.52813	0.022912
EXC	0.529167	0.50625	0.49531	0.46875	0.50781	0.021357
DOW	0.51875	0.47708	0.5	0.48281	0.46406	0.01875
DIS	0.529167	0.52813	0.4875	0.51094	0.5125	0.016667
MA	0.528125	0.54271	0.5125	0.49219	0.50156	0.015625
BA	0.5375	0.52813	0.50625	0.51563	0.52188	0.01562
TM	0.511458	0.49375	0.48125	0.5	0.47188	0.011458
PEP	0.514583	0.53854	0.4875	0.50313	0.48906	0.011453
RDS-B	0.527083	0.54063	0.4875	0.49688	0.51719	0.009893
UNH	0.525	0.5125	0.46406	0.51563	0.51406	0.00937
UPS	0.519792	0.49792	0.51094	0.49844	0.47813	0.008852
JPM	0.514583	0.50521	0.50156	0.50313	0.50625	0.008333
DCM	0.534375	0.51146	0.52656	0.48438	0.48906	0.007815
MRK	0.50625	0.53021	0.49375	0.49375	0.49844	0.00781
MMM	0.519792	0.52708	0.47969	0.5125	0.46875	0.007292

Experiment Result

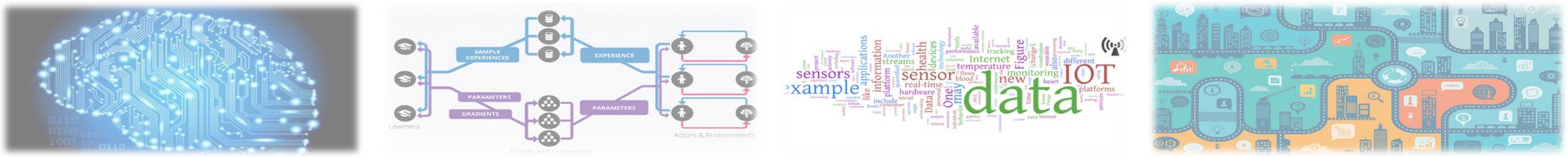
- Accuracy (Index : NASDAQ)
  - 비교모델 중 가장 좋은 정확성을 보인 모델과 Early fusion의 정확성의 차이가 큰 순서로 정렬

Data	Proposed -Early fusion Transformer	Proposed -Late fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM AM [8]	비교모델 중 최대 Accuracy – Early fusion accuracy
NVS	0.51875	0.52604	0.49375	0.50781	0.5125	0.00625
SO	0.522917	0.51563	0.5	0.51719	0.51406	0.005727
CVX	0.507292	0.50833	0.48125	0.49688	0.50313	0.004162
GE	0.526042	0.52188	0.51563	0.52188	0.50469	0.004162
PTR	0.528125	0.53021	0.52188	0.48438	0.525	0.003125
WFC	0.523958	0.50833	0.52188	0.47656	0.4875	0.002078
NGG	0.538542	0.52708	0.5375	0.51563	0.51406	0.001042
JNJ	0.526042	0.51146	0.525	0.48281	0.49219	0.001042
CMCSA	0.516667	0.51563	0.49063	0.4875	0.52031	-0.00364
VALE	0.513542	0.48125	0.49688	0.49219	0.51875	-0.00521
INTC	0.525	0.50417	0.51094	0.48594	0.53125	-0.00625
TOT	0.515625	0.52292	0.48438	0.52031	0.52188	-0.00626
PG	0.516667	0.49375	0.48438	0.525	0.47344	-0.00833
BRK-B	0.490625	0.50833	0.47188	0.5	0.49219	-0.00938
MO	0.544792	0.55625	0.48906	0.5125	0.55469	-0.0099
AMZN	0.520833	0.51458	0.5125	0.53125	0.49688	-0.01042
D	0.517708	0.51979	0.51094	0.50313	0.52813	-0.01042
KO	0.5125	0.5125	0.47344	0.47969	0.52344	-0.01094

Experiment Result

- Accuracy (Index : NASDAQ)
  - 비교모델 중 가장 좋은 정확성을 보인 모델과 Early fusion의 정확성의 차이가 큰 순서로 정렬

Data	Proposed -Early fusion Transformer	Proposed -Late fusion Transformer	DTML [2]	AdvLSTM [7]	CNN BiLSTM -AM [8]	비교모델 중 최대 Accuracy – Early fusion accuracy
WMT	0.504167	0.50833	0.47656	0.51563	0.5	-0.01146
GOOGL	0.516667	0.50417	0.52969	0.50938	0.52813	-0.01302
PFE	0.492708	0.51979	0.48594	0.47813	0.50625	-0.01354
HD	0.514583	0.51458	0.50469	0.52813	0.52031	-0.01355
BHP	0.485417	0.51563	0.47813	0.5	0.48125	-0.01458
ORCL	0.511458	0.50729	0.50938	0.50469	0.52969	-0.01823
RIO	0.501042	0.49167	0.52344	0.51719	0.49375	-0.0224
T	0.514583	0.4875	0.5375	0.50938	0.50469	-0.02292
SYT	0.485417	0.48854	0.5125	0.50469	0.49688	-0.02708
MSFT	0.501042	0.49479	0.53438	0.49531	0.5125	-0.03334
DUK	0.517708	0.53438	0.48281	0.47344	0.55469	-0.03698
CHL	0.479167	0.50104	0.52344	0.50781	0.50938	-0.04427
XOM	0.46875	0.46979	0.44844	0.48906	0.51563	-0.04688
NTT	0.509375	0.51563	0.56719	0.51563	0.49063	-0.05781



Hypernym Discovery

윤건일

2022. 09. 21

Overall Progress

Problem

- Hypernym Discovery
- Low Accuracy (MAP, MRR etc.)

Related Works

- CRIM
- RMM

Experiments

- SemEval2018 task9 dataset
- UMBC corpus
- 1500 train, 1500 test data
- Metric
  - MAP, MRR, P@n

Post-experimental analysis

- MRR improvement (2.32->4.12)
- MAP(1.21->1.87)

Analysis

- word2vec embeddings not working properly for phrase

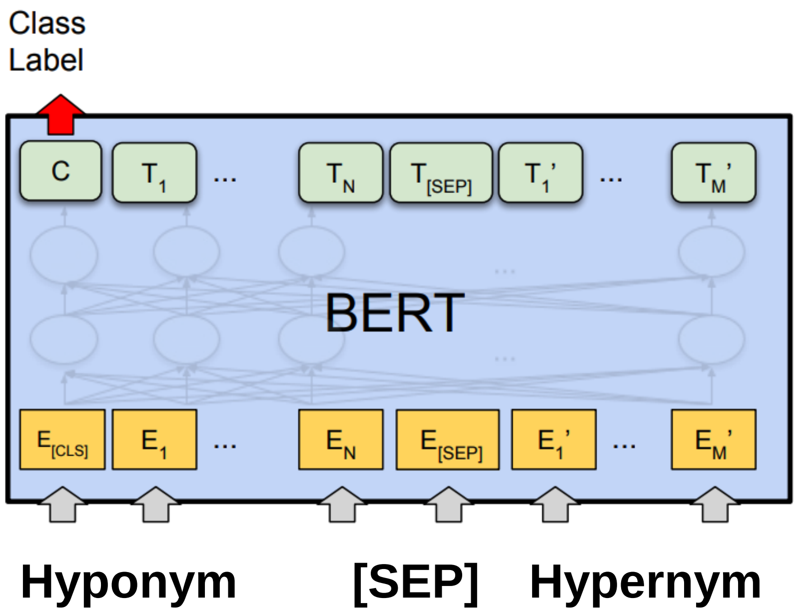
Conclusion

- Accuracy improvement
- Using contextual embeddings

Proposal

- Contextual Embeddings

Contextual Embeddings for hypernym relation



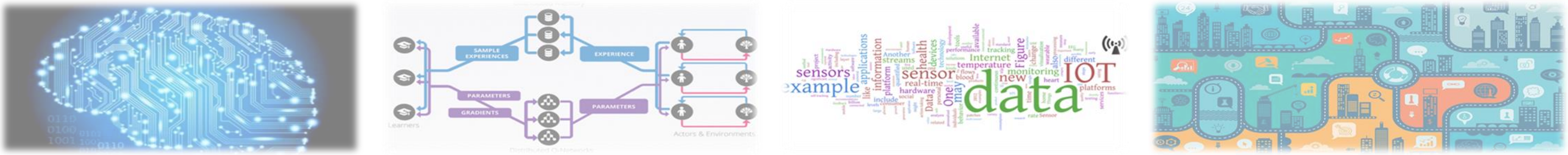
- Using **Language Model**(Like **BERT, RoBERTa** ..) for contextual embedding.
- Input hyponym and hypernym with **[SEP]** token.

For subtask 1A – English general

	MRR	MAP	P@1	P@3	P@5
CRIM	2.3276	1.2140	1.5333	0.9444	0.9199
RMM	0.3561	0.2250	0.2666	0.1555	0.1566
Proposed	**4.1244**	**1.8716**	**2.0000**	**1.7555**	**1.8811**

Considerations

- Experiments for Subtask 2A, 2B (Medical, Music)
- Compare with other language models(RoBERTa ..)
- Pretraining for hypernym discovery (using hearst patterns) – new contribution
    - Input : (previous sentence) [SEP] (Hearst patterns)
    - Masking hypernym or hyponym



Restoration of Homoglyph Attack on Spam Using BERT Masked Language Model

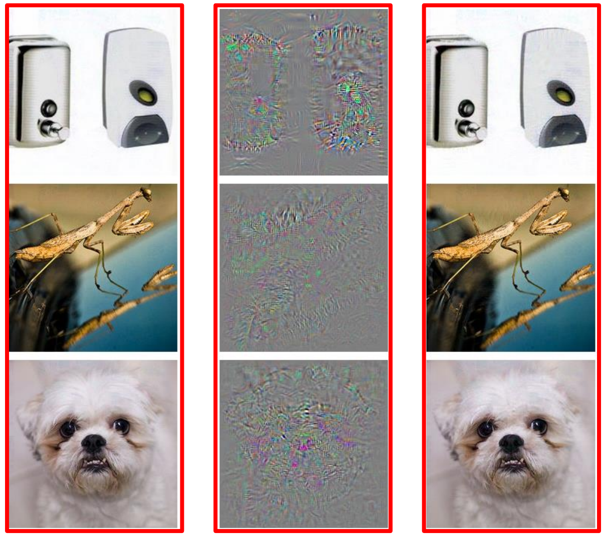
Hanyong Lee

2022. 09. 21

NLP Attack

Adversarial examples in **Computer Vision**

①　②　③



① correctly predicted samples
② image difference between ① and ③
③ images predicted as "ostrich"

Adversarial examples in **Natural Language Processing**

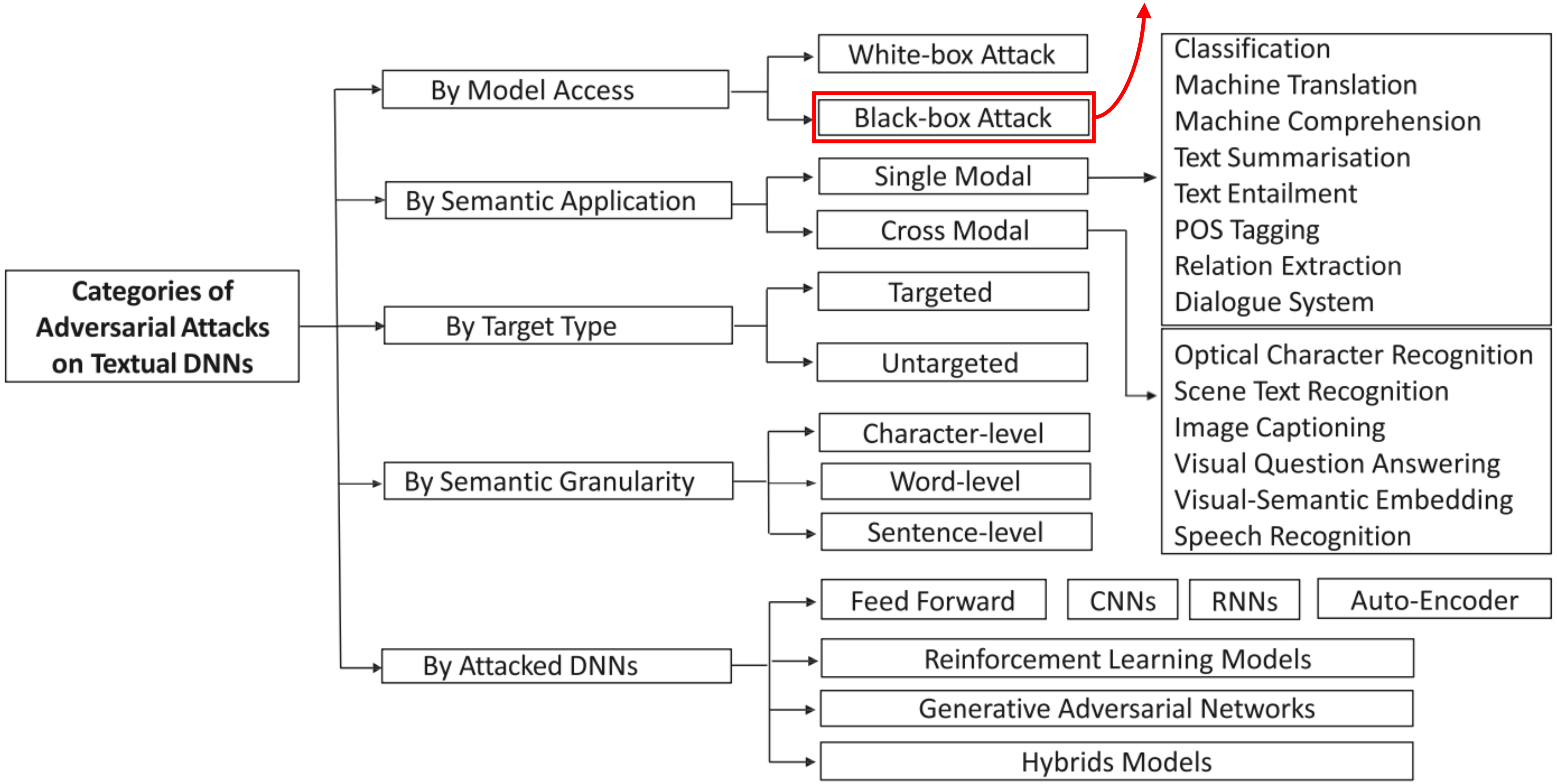
?

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199.*

NLP Attack

Adversarial examples in **Natural Language Processing**

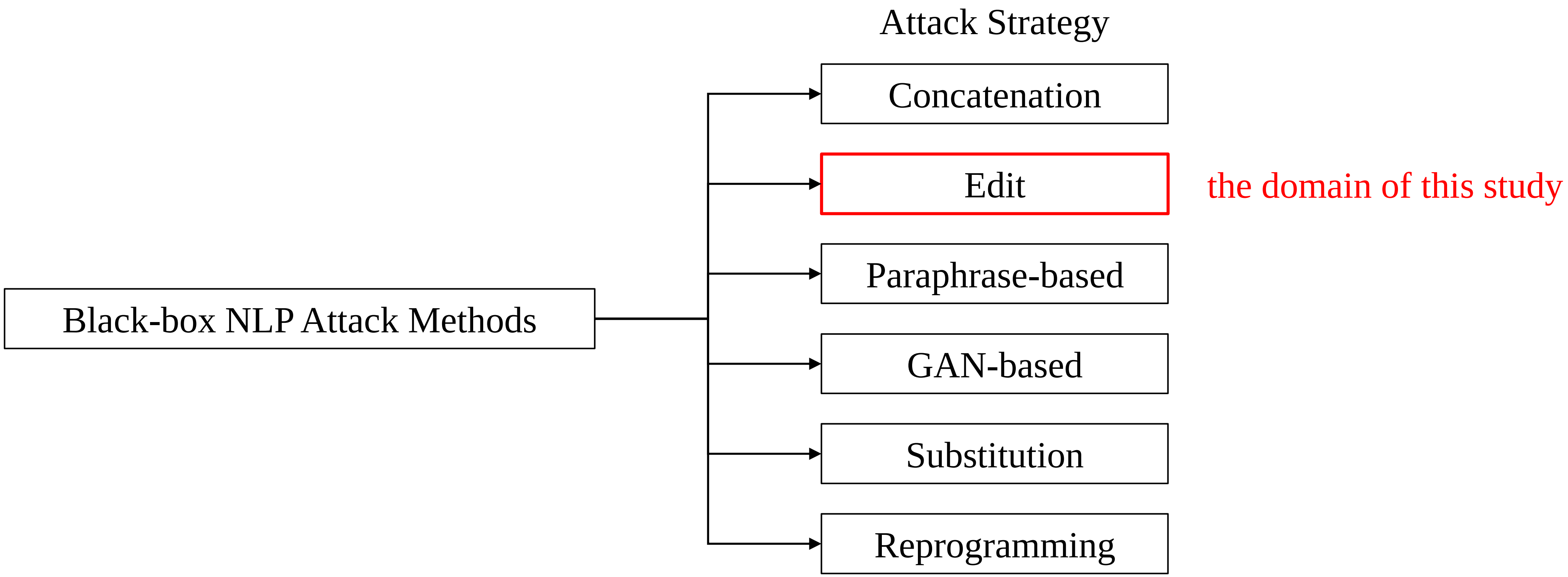
the domain of this study



Zhang, W. E., Sheng, Q. Z., Alhazmi, A., & Li, C. (2020). Adversarial attacks on deep-learning models in natural language processing: A survey. ACM Transactions on Intelligent Systems and Technology (TIST), 11(3), 1-41.

NLP Attack

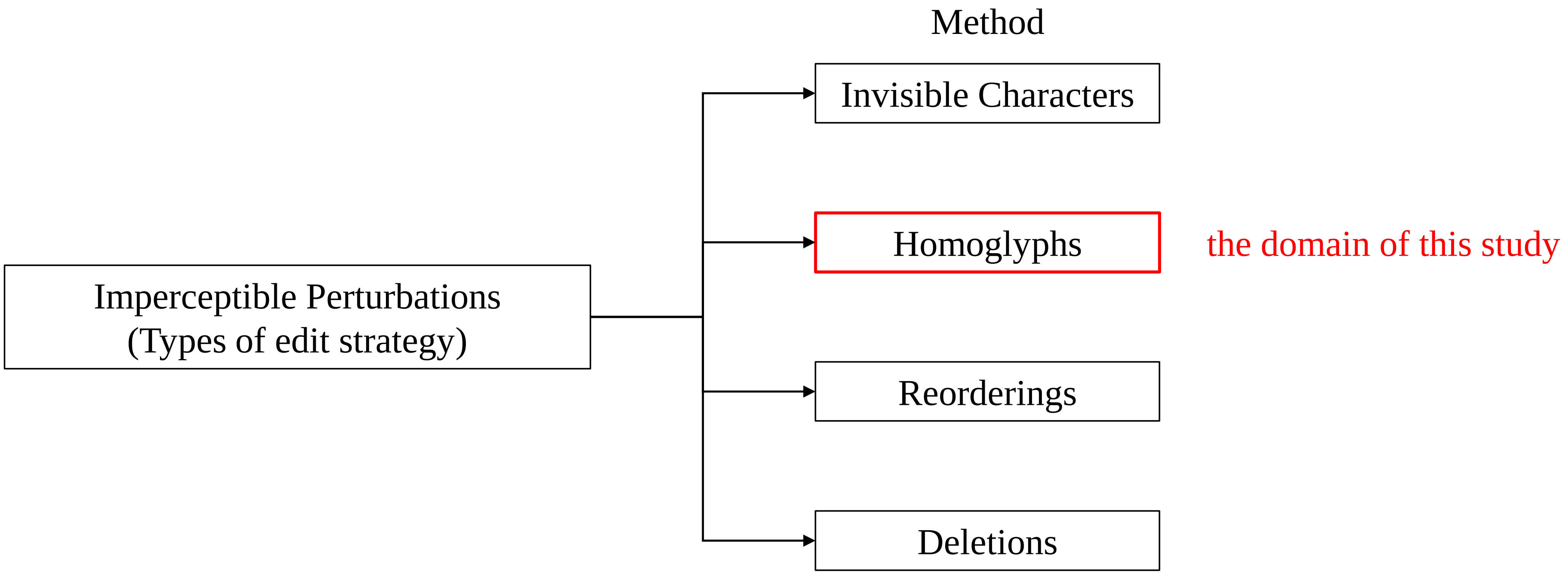
Black-box NLP Attack Methods



Zhang, W. E., Sheng, Q. Z., Alhazmi, A., & Li, C. (2020). Adversarial attacks on deep-learning models in natural language processing: A survey. ACM Transactions on Intelligent Systems and Technology (TIST), 11(3), 1-41.

Homoglyph Attack

Imperceptible Perturbations

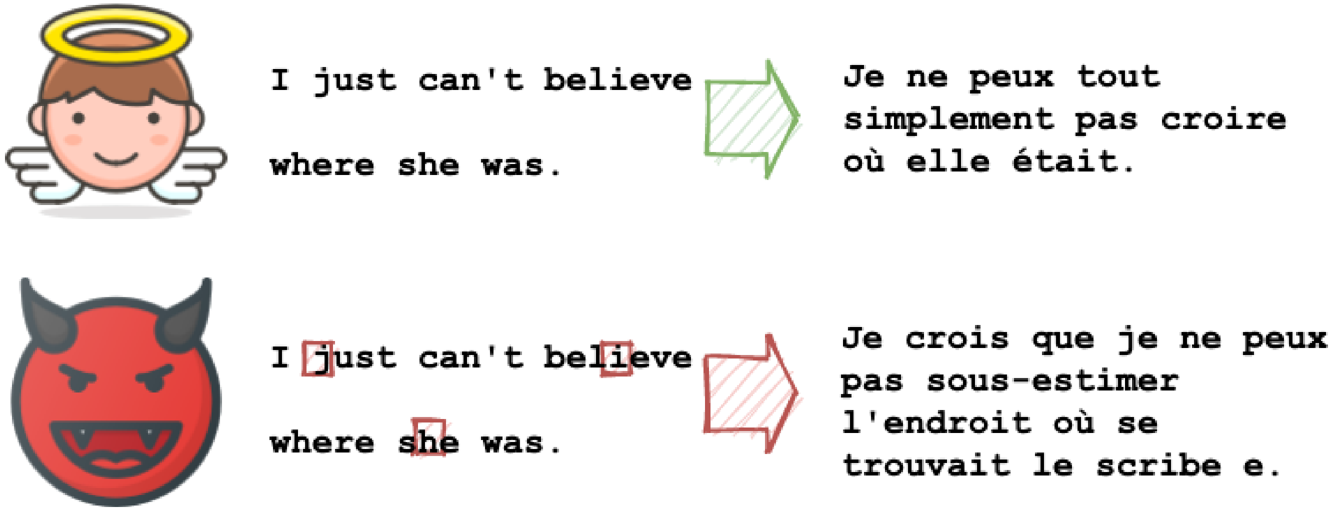


Boucher, N., Shumailov, I., Anderson, R., & Papernot, N. (2022, May). Bad characters: Imperceptible nlp attacks. In 2022 *IEEE Symposium on Security and Privacy (SP)* (pp. 1987-2004). IEEE.

Related Work

Homoglyph Attack

Example of an attack using homoglyph to confuse English-Franch translation task.



wrong result

“j” is replaced with U+3F3(“j”), “i” with U+456(“i”), and “h” with U+4BB(“h”).

Boucher, N., Shumailov, I., Anderson, R., & Papernot, N. (2022, May). Bad characters: Imperceptible nlp attacks. In 2022 *IEEE Symposium on Security and Privacy (SP)* (pp. 1987-2004). IEEE.

Related Work

Defense Methods of Homoglyph Attack

Translate table for confusable characters

⋮

```
1D4C5 ; 0070 ;  MA    # ( 𝓅 → p ) MATHEMATICAL SCRIPT SMALL P → LATIN SMALL LETTER P  #
1D4F9 ; 0070 ;  MA    # ( 𝓹 → p ) MATHEMATICAL BOLD SCRIPT SMALL P → LATIN SMALL LETTER P     #
1D52D ; 0070 ;  MA    # ( 𝔭 → p ) MATHEMATICAL FRAKTUR SMALL P → LATIN SMALL LETTER P #
1D561 ; 0070 ;  MA    # ( 𝕡 → p ) MATHEMATICAL DOUBLE-STRUCK SMALL P → LATIN SMALL LETTER P    #
```

⋮

It doesn’t contain some pairs.

- μ → u
- $ → S
- 5 → S
- …

Unicode Consortium 2022, *Unicode Utilities: Confusables.* accessed 10 September 2022, <https://www.unicode.org/Public/security/15.0.0/confusables.txt>.

Defense Methods of Homoglyph Attack

SimChar database

H. Suzuki et al. used the following formula to calculate the similarity of two bitmap images of characters.

pixel of image I_1, I_2 at coordinate (i, j)

$$\Delta = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |I_1(i,j) - I_2(i,j)|$$

character difference

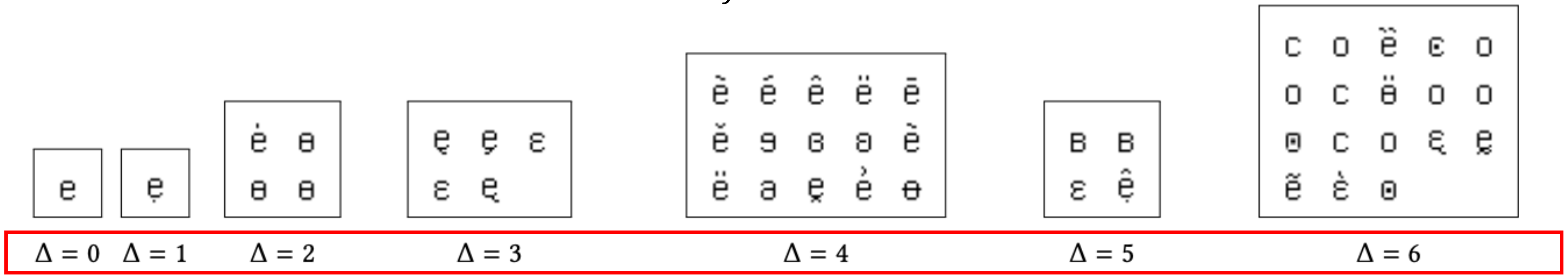


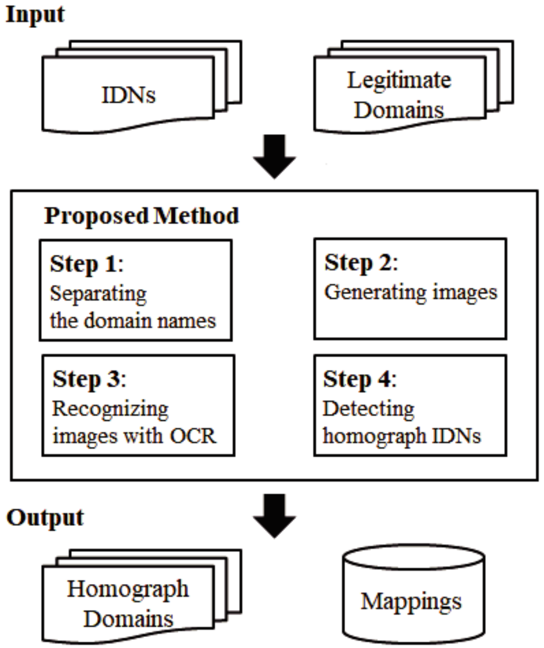
Figure 6: Basic Latin lowercase letter 'e' and characters under different values of the threshold Δ. In this work, characters with Δ ≤ 4 are detected as homoglyphs.

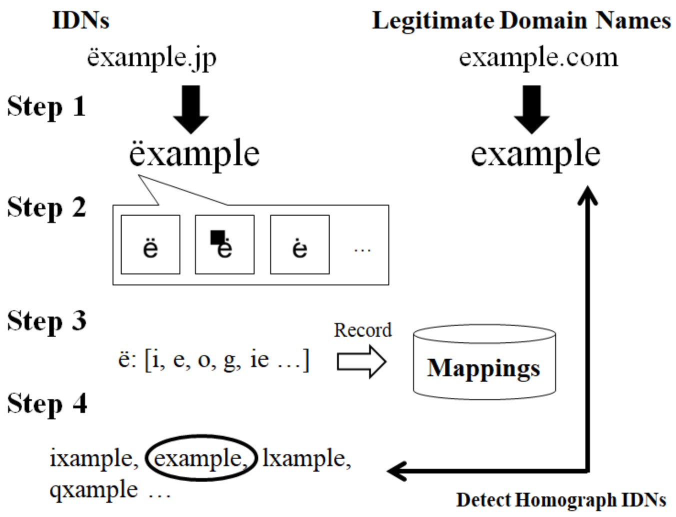
Suzuki, H., Chiba, D., Yoneya, Y., Mori, T., & Goto, S. (2019, October). ShamFinder: An automated framework for detecting IDN homographs. *In Proceedings of the Internet Measurement Conference* (pp. 449-462).

Defense Methods of Homoglyph Attack

OCR-based method

Sawabe et al. used the OCR-based Method to detect Homoglyph Attack.





Sawabe, Y., Chiba, D., Akiyama, M., & Goto, S. (2019). Detection method of homograph internationalized domain names with OCR. *Journal of Information Processing, 27*, 536-544.

Related Work

Defense Methods of Homoglyph Attack

Spellchecker-based method

Imam, N. H et al. used a spellchecker-based method to restore the homoglyph attacks.

Table 2
Some of the adversarial text examples used in experiments.

Original	Flipping	Swapping	Deletion
Busy	Bu$y	Bsuy	Bsy
Caller	C@1ler	Claler	Cller
Reply	Rep1y	Rpely	Rply
winner	w!nner	wniner	wnner

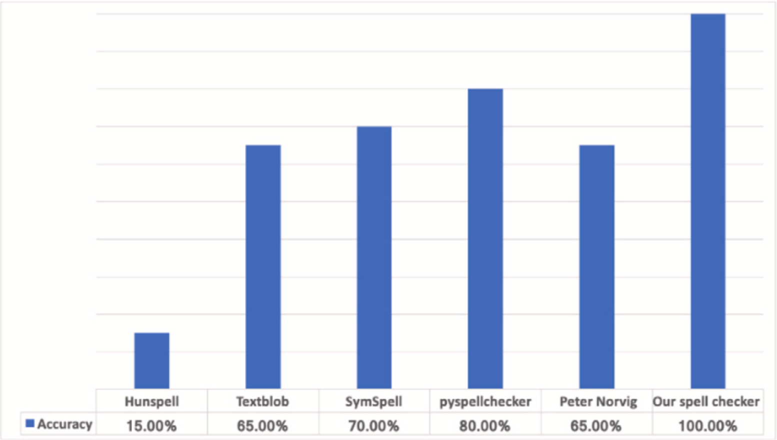
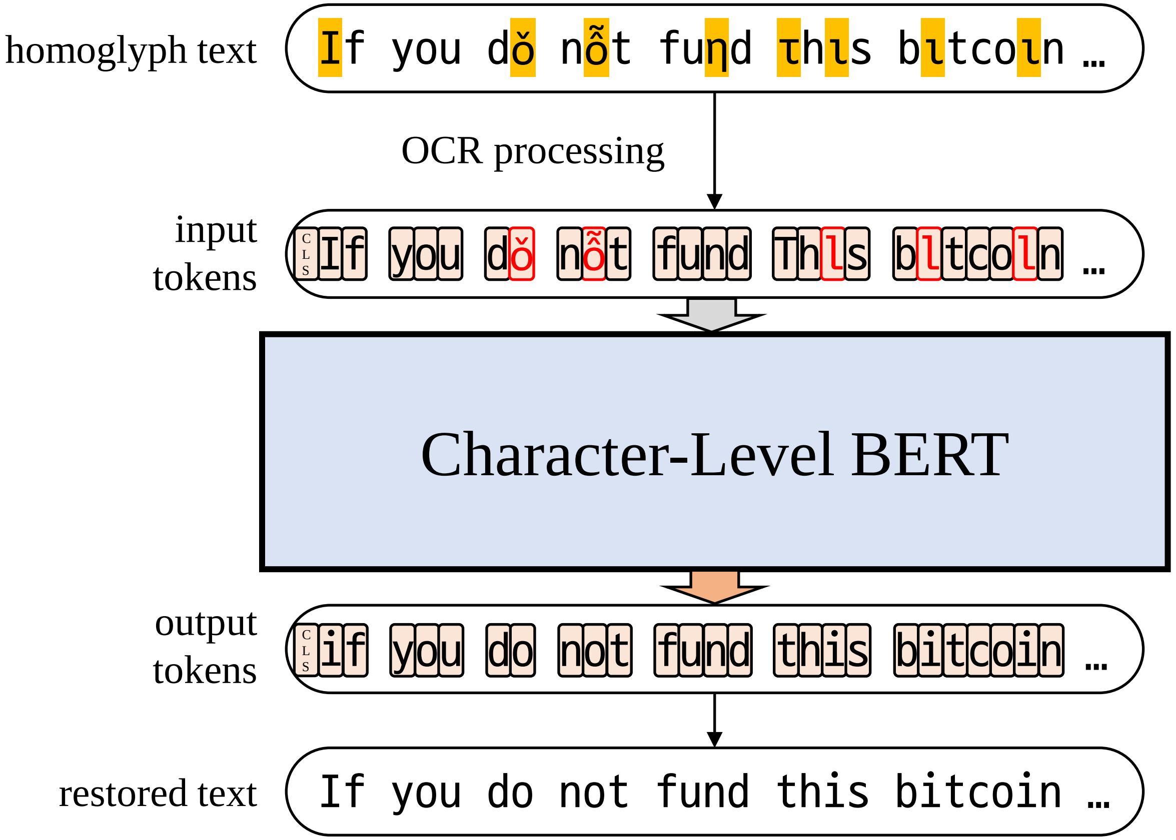


Fig. 9. Flipping.

Imam, N. H., Vassilakis, V. G., & Kolovos, D. (2022). OCR post-correction for detecting adversarial text images. *Journal of Information Security and Applications*, *66*, 103170.

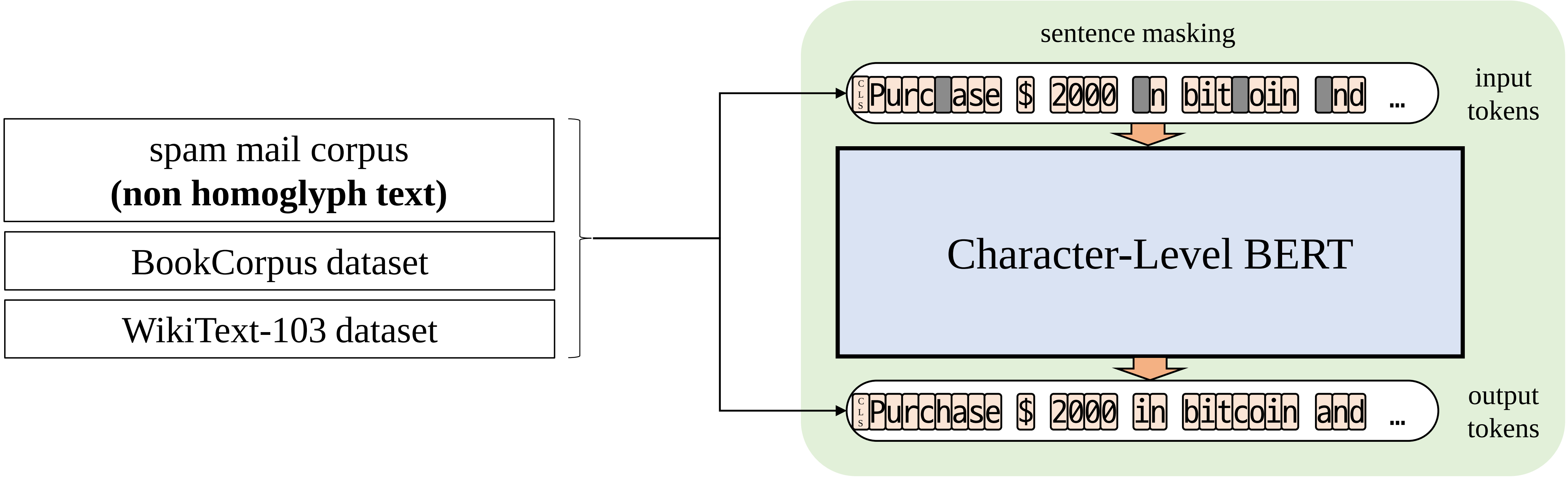
Restoring Process Overview



Experimental Settings

Pre-training

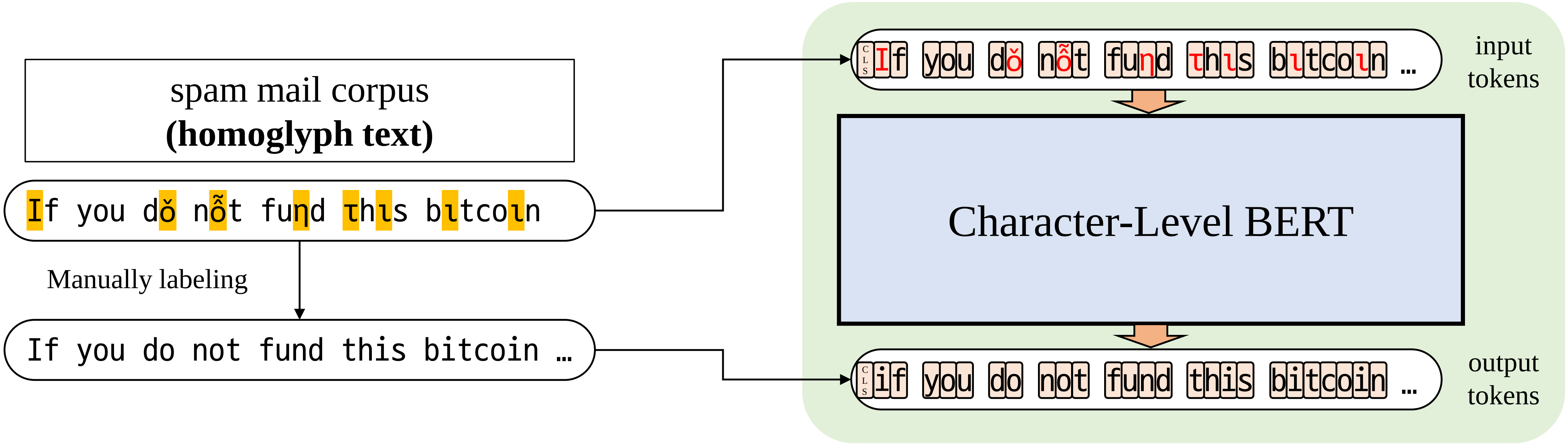
Epochs num: 25 Epochs



Pre-training

Finetuning

Evaluated with a blind test (Train/Test set split ratio = 8:2 / Iteration = 50 times)



Finetuning

Finetuned Model Evaluation

Algorithm	Word Accuracy
Proposed	**0.9959 ± 0.0008**
Normal BERT	0.6713 ± 0.0029 ▼
OCR	0.6518 ± 0.0028 ▼
SimChar DB	0.5512 ± 0.0046 ▼
Spell Checker	0.9087 ± 0.0011 ▼

Word level accuracy and T-test with the accuracy of the proposed method
▼ means that the algorithm has a lower performance than the proposed algorithm based on t-test. ▼: $p < 0.01$

Contribution of the Study

1.Achieve higher performance than traditional methods such as OCR, SimChar DataBase, and spell checker.

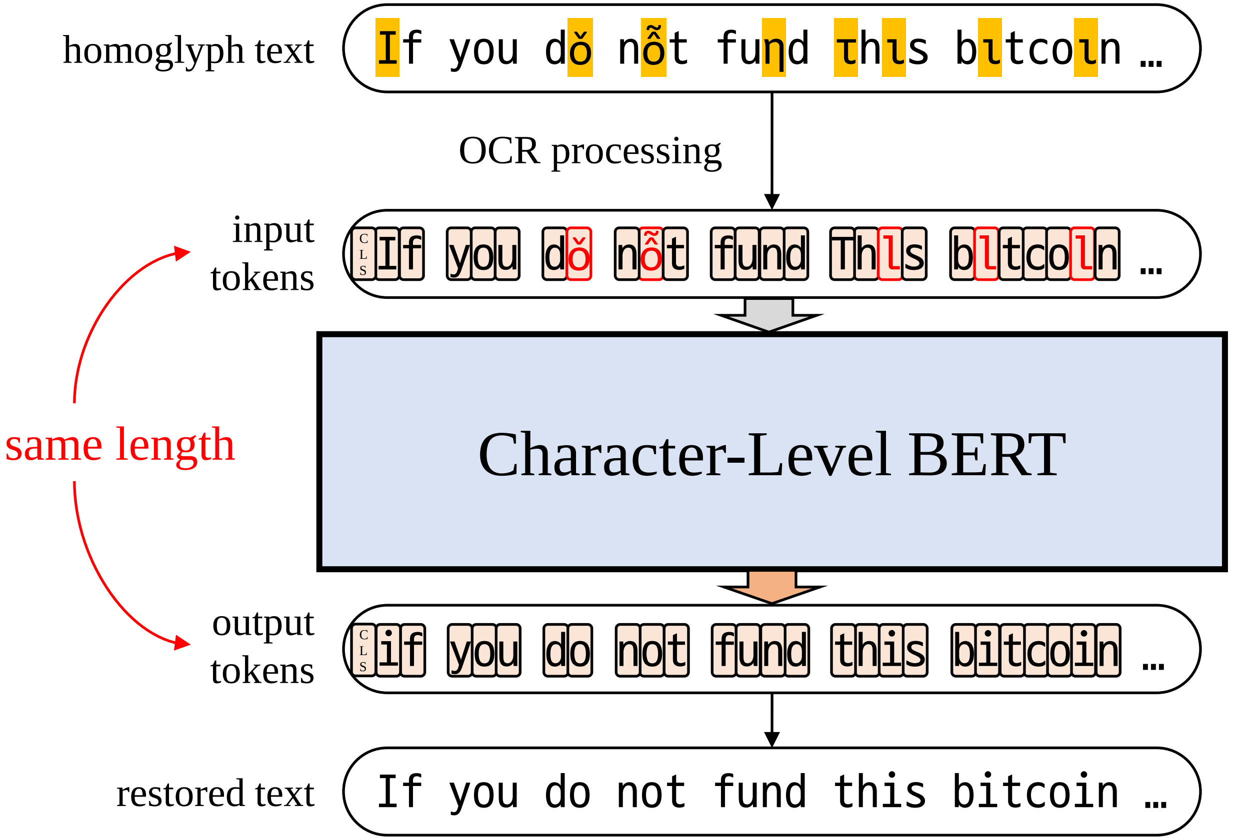
2.Propose an automated homoglyph restoration framework by using deep learning.

Limitation of Proposed Model

The proposed model is a masked language model, limited in that the input/output sentences should be the **same sequence length**.

- It can be used as a homoglyph of the alphabet ‘m’ by concatenating the letters ‘r’ and ‘n’. But the presented model does not solve this case.

The proposed model can be applied as a **sequence-to-sequence** model in future studies.



References

1. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. *a rXiv preprint arXiv:1312.6199.*
2. Zhang, W. E., Sheng, Q. Z., Alhazmi, A., & Li, C. (2020). Adversarial attacks on deep-learning models in natural language processing: A surv ey. *ACM Transactions on Intelligent Systems and Technology (TIST), 11(3),* 1-41.
3. Boucher, N., Shumailov, I., Anderson, R., & Papernot, N. (2022, May). Bad characters: Imperceptible nlp attacks. In 2022 *IEEE Symposium o n Security and Privacy (SP)* (pp. 1987-2004). IEEE.
4. Unicode Consortium 2022, *Unicode Utilities: Confusables*. accessed 10 September 2022, <https://www.unicode.org/Public/security/15.0.0/co nfusables.txt>.
5. Suzuki, H., Chiba, D., Yoneya, Y., Mori, T., & Goto, S. (2019, October). ShamFinder: An automated framework for detecting IDN homograph s. *In Proceedings of the Internet Measurement Conference* (pp. 449-462).
6. Sawabe, Y., Chiba, D., Akiyama, M., & Goto, S. (2019). Detection method of homograph internationalized domain names with OCR. *Journal of Information Processing*, *27*, 536-544.
7. Imam, N. H., Vassilakis, V. G., & Kolovos, D. (2022). OCR post-correction for detecting adversarial text images. *Journal of Information Secu rity and Applications*, *66*, 103170.
8. Norvig, P. 2016, *How to write a spelling corrector*. accessed 10 September 2022, <https://norvig.com/spell-correct.html>.
9. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv prep rint arXiv:1810.04805*.

A. Proposed Model Information

Character-level BERT

- Number of parameters: 86,720,369
- Max length of input tokens: 512
- Vocab size: 881

B. Pre-trained Model Evaluation

Evaluated with the **whole test sentences**

Character level BERT is only pretrained with the **WikiText-103, BookCorpus & spam mail datasets**.

Algorithm	Word Accuracy
Proposed (not finetuned)	**0.9516 ± 0.1289**
Normal BERT	0.6713 ± 0.2679 ▼
OCR	0.6519 ± 0.2976 ▼
SimChar DB	0.5506 ± 0.3627 ▼
Spell Checker	0.9090 ± 0.1620 ▼

Word level accuracy and T-test with the accuracy of the proposed method
▼ means that the algorithm has a lower performance than the proposed algorithm based on t-test.▼: $p < 0.01$

C. Test Output Examples of Pretrained Model

Example 1

- input sentence Ꮮ Neeđ y0ᑌᖇ TOTɑι ατTeNTιOɴ fOʀ Ꭲhe c0MιNg ᎢweNτʏ-fOᑌᖇ hᖇᏚ,
- output sentence : i need your total attention for the coming twenty - four hrs,

Example 2

- input sentence : ɑftêг thät, i hἀvë ṡtἀгtèd tгȧcĸi̇ṅģ ɣȯᴜг i̇ṅtéгпët ȧctívítìêṡ.
- output sentence : after that, i have started tracking yo ur internet activities.

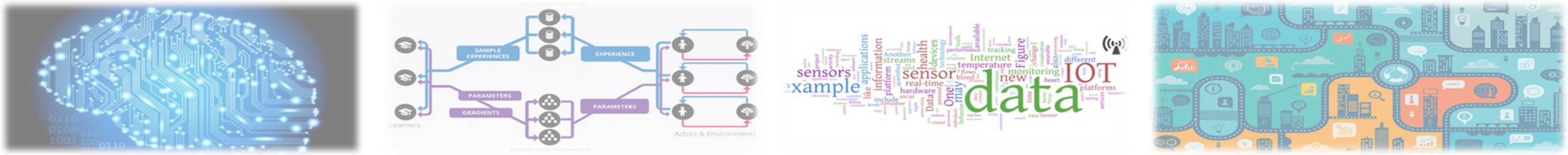
C. Test Output Examples of Pretrained Model

Example 3

- input sentence : do ṅỏt tгɣ tỏ coṅtáct the ῥolićê àɴd ỏtңêг sëcurityʏ servi̇ces.
- output sentence : don't try to contact the police and other security services.

Example 4

- input sentence : yøuř åccøuňt wås åttåckěd.
- output sentence : yogr account was attacked.



개인 연구

HYE-JIN BAEK

2022. 09. 21

진행 사항

- 논문(Tabular data/Missing value/Data Imputation)
    1. Tabular data:Deep learning is not all you need(Information Fusion, 2022)
    2. A Novel Algorithm for Missing Data Imputation on Machine Learning(IEEE, 2019)

1. Tabular data:Deep learning is not all you need

AutoML

- Tabular dataset에 새로운 deep model과 xgboost를 비교하여 더 나은 성능을 띈 모델 추천 목적
- Experiment result >

Table 2
Test results on tabular datasets. Presenting the performance for each model. MSE is presented for the YearPrediction and Rossman datasets, and cross-entropy loss (with 100X factor) is presented for the other datasets. The papers that used these datasets are indicated below the table. The values are the averages of four training runs (lower value is better), along with the standard error of the mean (SEM)

Model Name	Rossman	CoverType	Higgs	Gas	Eye	Gesture
XGBoost	490.18 ± 1.19	3.13 ± 0.09	21.62 ± 0.33	2.18 ± 0.20	**56.07**±0.65	80.64 ± 0.80
NODE	488.59 ± 1.24	4.15 ± 0.13	21.19 ± 0.69	2.17 ± 0.18	68.35 ± 0.66	92.12 ± 0.82
DNF-Net	503.83 ± 1.41	3.96 ± 0.11	23.68 ± 0.83	**1.44** ±0.09	68.38 ± 0.65	86.98 ± 0.74
TabNet	**485.12**±1.93	3.01 ± 0.08	**21.14**±0.20	1.92 ± 0.14	67.13 ± 0.69	96.42 ± 0.87
1D-CNN	493.81 ± 2.23	3.51 ± 0.13	22.33 ± 0.73	1.79 ± 0.19	67.9 ± 0.64	97.89 ± 0.82
Simple Ensemble	488.57 ± 2.14	3.19 ± 0.18	22.46 ± 0.38	2.36 ± 0.13	58.72 ± 0.67	89.45 ± 0.89
Deep Ensemble w/o XGBoost	489.94 ± 2.09	3.52 ± 0.10	22.41 ± 0.54	1.98 ± 0.13	69.28 ± 0.62	93.50 ± 0.75
Deep Ensemble w XGBoost	485.33 ± 1.29	**2.99** ±0.08	22.34 ± 0.81	1.69 ± 0.10	59.43 ± 0.60	**78.93** ±0.73
		TabNet			DNF-Net	

Model Name	YearPrediction	MSLR	Epsilon	Shrutime	Blastchar
XGBoost	77.98 ± 0.11	55.43±2e−2	11.12±3e−2	13.82 ± 0.19	20.39 ± 0.21
NODE	76.39 ± 0.13	55.72±3e−2	**10.39**±1e−2	14.61 ± 0.10	21.40 ± 0.25
DNF-Net	81.21 ± 0.18	56.83±3e−2	12.23±4e−2	16.8 ± 0.09	27.91 ± 0.17
TabNet	83.19 ± 0.19	56.04±1e−2	11.92±3e−2	14.94±, 0.13	23.72 ± 0.19
1D-CNN	78.94 ± 0.14	55.97±4e−2	11.08±6e−2	15.31 ± 0.16	24.68 ± 0.22
Simple Ensemble	78.01 ± 0.17	55.46±4e−2	11.07±4e−2	13.61±, 0.14	21.18 ± 0.17
Deep Ensemble w/o XGBoost	78.99 ± 0.11	55.59±3e−2	10.95±1e−2	14.69 ± 0.11	24.25 ± 0.22
Deep Ensemble w XGBoost	**76.19** ±0.21	**55.38**±1e−2	11.18±1e−2	**13.10**±0.15	**20.18**±0.16
	NODE			New datasets	

Table 3
Average relative performance deterioration for each model on its unseen datasets (lower value is better).

Name	Average relative Performance (%)
XGBoost	3.34
NODE	14.21
DNF-Net	11.96
TabNet	10.51
1D-CNN	7.56
Simple Ensemble	3.15
Deep Ensemble w/o XGBoost	6.91
Deep Ensemble w XGBoost	**2.32**

- tabular data에 대해 deep model + XGBoost 앙상블 모델에서 성능이 좋음.
- 논문의 저자는, 다양한 데이터셋에서 성능 확인 필요

2. A Novel Algorithm for Missing Data Imputation on Machine Learning

- Medical dataset에 적합한 missXGBoost(missForest+xgb) imputation method 제시

- Experiment result >

TABLE-II: TEST CLASSIFIER ACCURACY WITH XGBOOST CLASSIFIER.

Datasets	Proposed Method + XGBoost		
	RMSE (%)	Accuracy (%)	Variance (%)
Cleveland	1.01	53.73	8.4
Diabetes	0.19	78.22	2.89
Dermatology	0.18	96.45	2.72
Hepatitis	0.11	82.45	8.6
Mammographic	0.13	82.5	3.7
Wisconsin	0.16	95.71	4.2

TABLE-III: TEST CLASSIFIER WITH ITERATIVE IMPUTATION AND XGBOOST.

Datasets	Iterative Method + XGBoost		
	RMSE (%)	Accuracy (%)	Variance (%)
Cleveland	0.99	54.7	7.94
Diabetes	0.19	75.7	5.59
Dermatology	0.19	96.45	2.73
Hepatitis	0.15	81.66	11.24
Mammographic	0.13	82.7	4.2
Wisconsin	0.16	95.8	3.9

TABLE-IV: TEST CLASSIFIER WITH MISSFOREST IMPUTATION AND XGBOOST

Datasets	MissForest Method + XGBoost		
	RMSE (%)	Accuracy (%)	Variance (%)
Cleveland	1.01	54.71	7.94
Diabetes	0.19	77.45	6.04
Dermatology	0.19	96.45	2.72
Hepatitis	0.11	85.00	8.6
Mammographic	0.13	82.30	3.77
Wisconsin	0.16	95.70	3.78

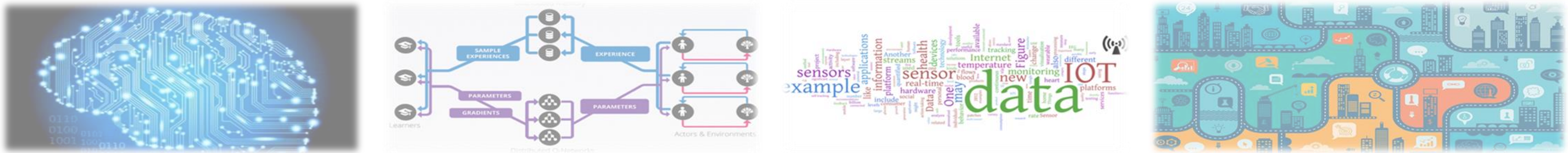
→ 논문에서 제안한 MissForest Method + XGBoost 에서 좋은 성능 결과가 나타남

→ 하지만, medical dataset에서만 적합한 method

► 최종 : 2개의 논문 연구를 진행한 결과, 다른 tabular dataset을 이용하여 다양한 Imputation method를 적용한 실험 진행 필요

진행 예정

- Tabular data/Missing value/Data Imputation 관련된 추가적인 논문 연구 진행 예정
- 실험
  1. 사용 데이터 : UCI(Breast-cancer) dataset
  2. 전체 데이터셋 중 Missing value 0%, 20%, 40%, 60%, 80%의 비율로 데이터셋 생성
  3. XGBoost/deep model 등 5개 선정 후, missing value imputation 진행하여 성능 확인 예정



T AO Object Detection

Go-Eun Kim

2022. 09. 21

Overall Progress

AutoML

Task

- TAO Object detection

Related Work / Survey (Counterparts)

- YOLO v1-7
- Faster R-CNN

Experiments

- TAO Dataset
  1189 samples
- Metric
  PR curve
  mAP

Proposal

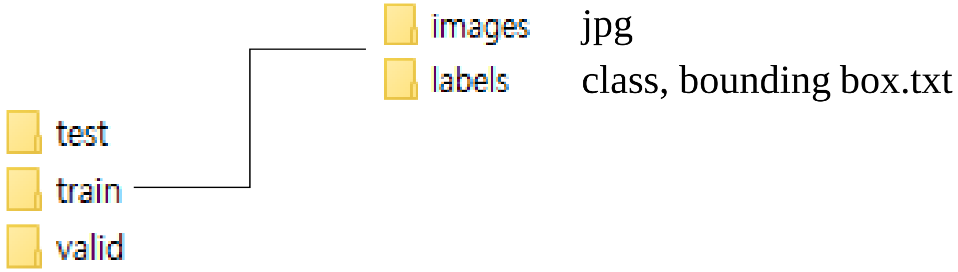
- Finetuning
- Augmenation
- hyper parameter
- Model

Introduction : Dataset

TAO **t**hyroid **a**ssociated **o**rbitopathy (or thyroid-associated ophthalmopathy)

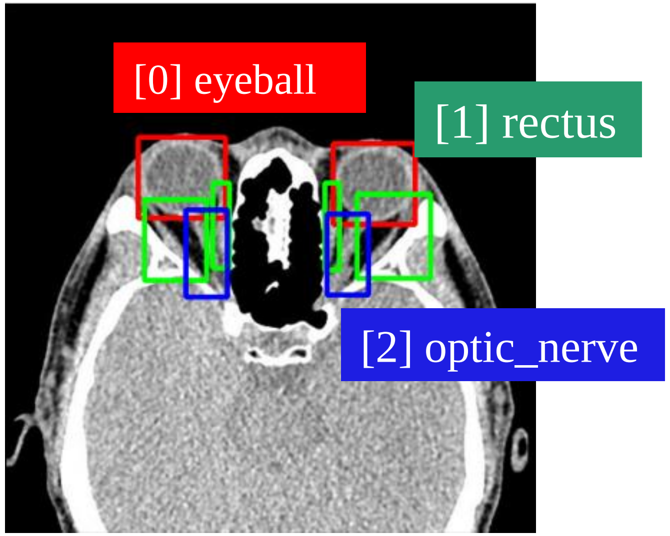
✓ 일명 그레이브스 병이라고 불리는 갑상선 관련 안와병증(TAO)

✓ 안와의 주변 조직, 갑상선, 드물게 전전공 피부나 손가락에 영향을 줄 수 있는 질병으로 자가면역과정의 일부.



< forder structure >

< case 1: CT axial >

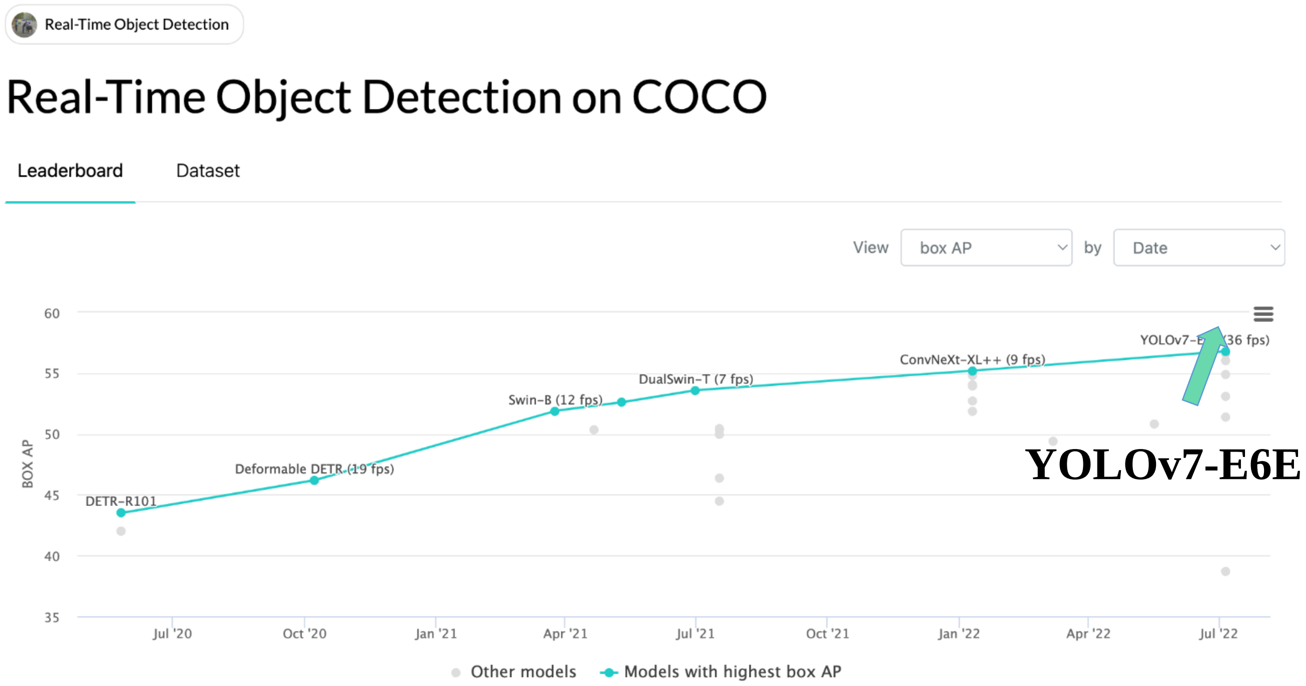


< case 1: CT axial with label >

Introduction

Paper : https://arxiv.org/pdf/2207.02696.pdf
GitHub : https://github.com/WongKinYiu/yolov7

YOLOv7: Trainble bag-of-freebies sets new state-of-the-art for real-time object detectors



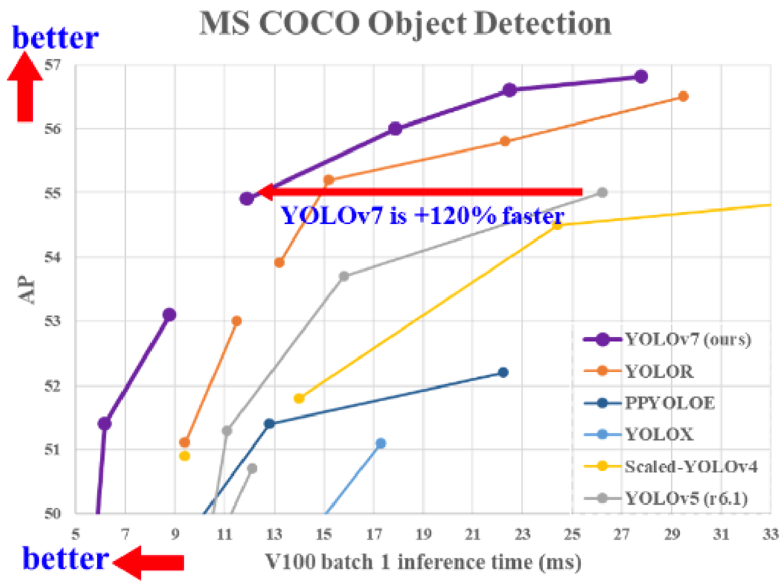


Figure 1: Comparison with other real-time object detectors, our proposed methods achieve state-of-the-arts performance.

Introduction : Object Detection model

YOLOv7: Trainble bag-of-freebies sets new state-of-the-art for real-time object detectors

5FPS to 160 FPS 56.8% AP 50% reduction in cost

Abstract

- ✓ 1. real-time object detection 이면서 inference cost를 증가시키지 않고도 정확도를 향상시킬 수 있는 trainable bag-of-freebies 방법을 제안
- ✓ 2. re-parameterized module이 original module을 대체하는 방법과 different output layers에 대한 dynamic label assignment strategy 방법 + 여기서 발생하는 어려움을 해결하기 위한 방법 제안
- ✓ 3. parameter 계산을 효과적으로 활용할 수 있는 real time object detector를 위한 "extend" 및 "compound scaling" 제안
- ✓ SOTA 방법 보다 parameter 수를 40%, 계산량을 50% 감소시킬 수 있으며 더 빠른 inference time과 더 높은 accurcy를 달성

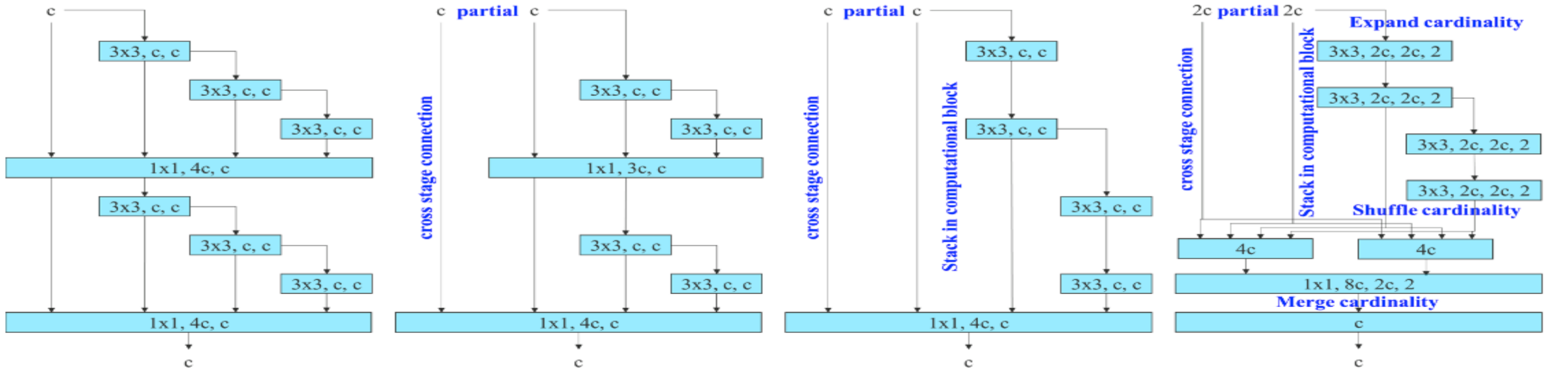
Architecture

Extended efficient layer aggregation network

✓ gradient path length와 computainal block의 stack 수와 관계 없이 stable state에 도달.

✓ 더 많은 computainal block들이 쌓이게 되면 이러한 상태가 파괴될 수 있으며 parameter utilization rate가 감소.

✓ expand, shuffle, merge cardinality를 사용하여 original gradient path를 파괴하지 않고 nework의 learning ability를 지속적으로 향상.

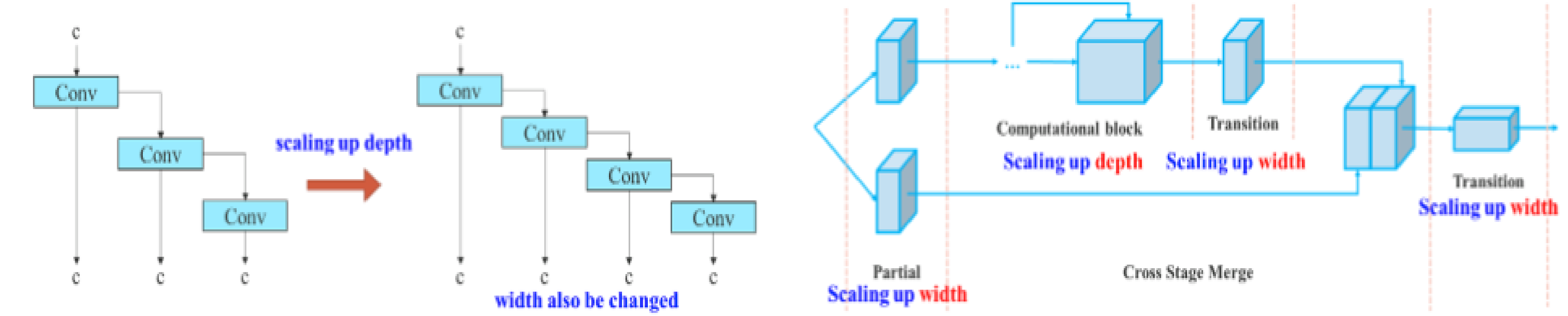


(a) VoVNet [39] (b) CSPVoVNet [79] (c) ELAN [1] (d) E-ELAN

Architecture

Model scaling for concatenation-based models

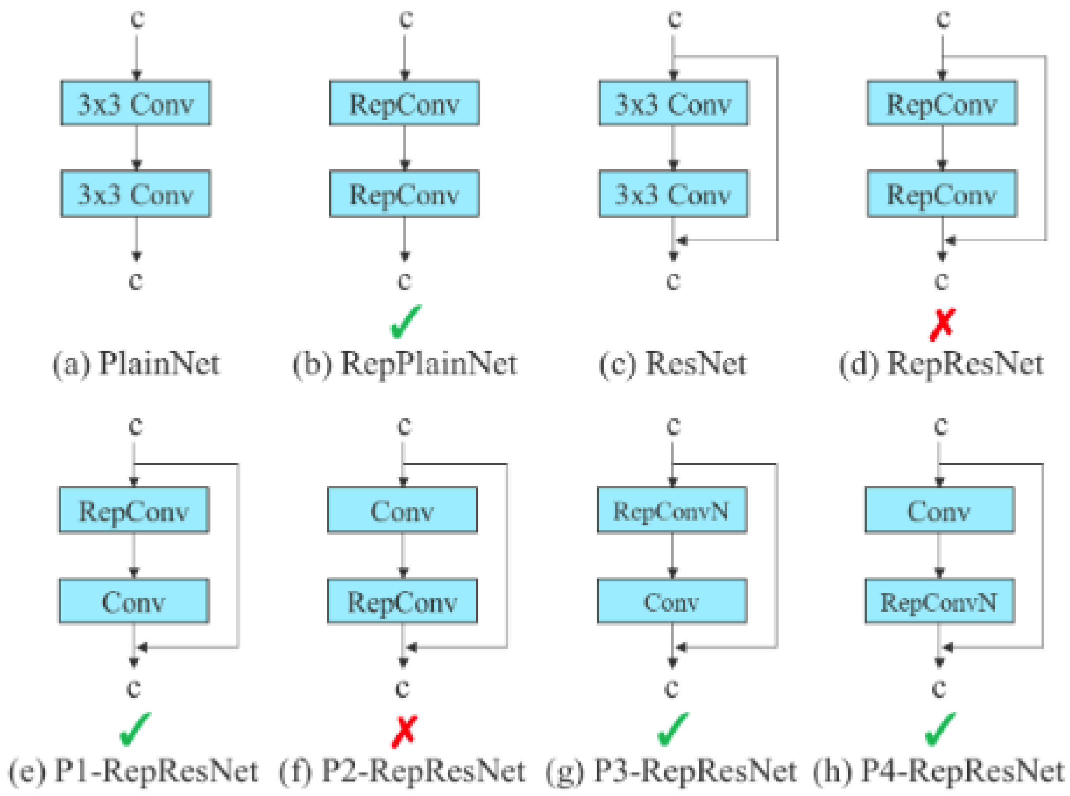
- ✓ concatenation-based model을 위한 compound model scaling method
- ✓ computational block의 depth factor를 조정할 때도 해당 block의 출력 채널 변경도 계산
- ✓ transition layer에 동일한 변화량으로 width factor scaling를 수행



(a) concatenation-based model (b) scaled-up concatenation-based model (c) compound scaling up depth and width for concatenation-based model

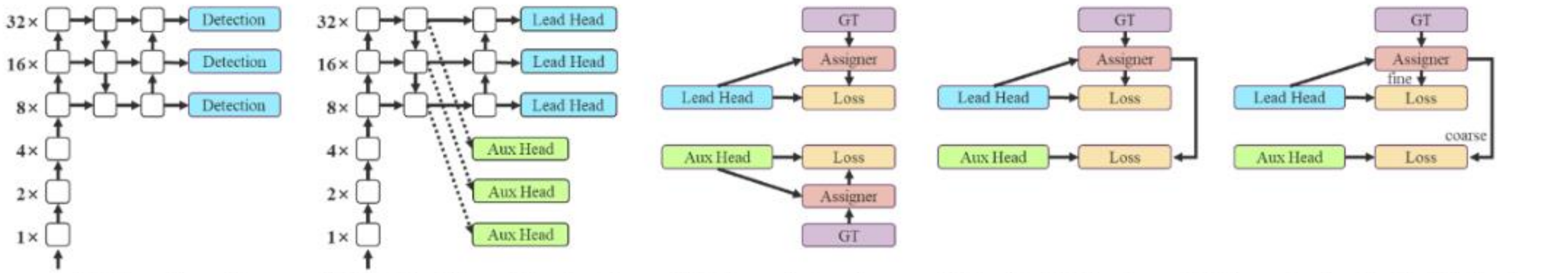
Trainable bag-of-freebies

Planned re-parameterized convolution



Trainable bag-of-freebies

Coarse for auxiliary and fine for lead loss



(a) Normal model (b) Model with auxiliary head (c) Independent assigner (d) Lead guided assigner (e) Coarse-to-fine lead guided assigner

Trainable bag-of-freebies

Coarse for auxiliary and fine for lead loss

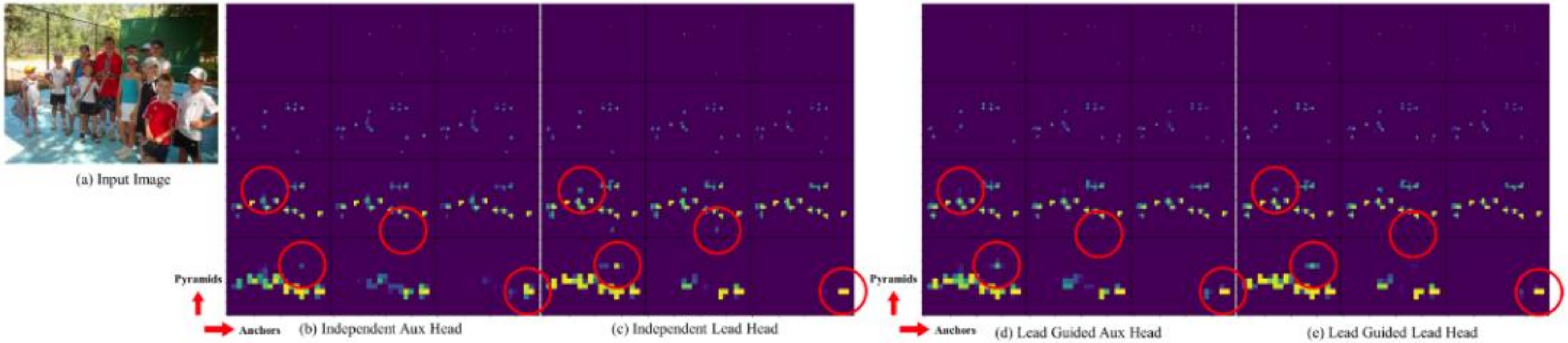


Figure 8: Objectness map predicted by different methods at auxiliary head and lead head.

Experiment

Table 1: Comparison of baseline object detectors.

Model	#Param.	FLOPs	Size	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}
YOLOv4 [3]	64.4M	142.8G	640	49.7%	68.2%	54.3%	32.9%	54.8%	63.7%
YOLOR-u5 (r6.1) [81]	46.5M	109.1G	640	50.2%	68.7%	54.6%	33.2%	55.5%	63.7%
YOLOv4-CSP [79]	52.9M	120.4G	640	50.3%	68.6%	54.9%	34.2%	55.6%	65.1%
YOLOR-CSP [81]	52.9M	120.4G	640	50.8%	69.5%	55.3%	33.7%	56.0%	65.4%
YOLOv7	36.9M	104.7G	640	**51.2%**	**69.7%**	**55.5%**	**35.2%**	**56.0%**	**66.7%**
improvement	-43%	-15%	-	+0.4	+0.2	+0.2	+1.5	=	+1.3
YOLOR-CSP-X [81]	96.9M	226.8G	640	52.7%	**71.3%**	57.4%	36.3%	57.5%	68.3%
YOLOv7-X	71.3M	189.9G	640	**52.9%**	71.1%	**57.5%**	**36.9%**	**57.7%**	**68.6%**
improvement	-36%	-19%	-	+0.2	-0.2	+0.1	+0.6	+0.2	+0.3
YOLOv4-tiny [79]	6.1	6.9	416	24.9%	42.1%	25.7%	8.7%	28.4%	39.2%
YOLOv7-tiny	6.2	5.8	416	**35.2%**	**52.8%**	**37.3%**	**15.7%**	**38.0%**	**53.4%**
improvement	+2%	-19%	-	+10.3	+10.7	+11.6	+7.0	+9.6	+14.2
YOLOv4-tiny-3l [79]	8.7	5.2	320	30.8%	47.3%	32.2%	**10.9%**	31.9%	51.5%
YOLOv7-tiny	6.2	3.5	320	**30.8%**	**47.3%**	**32.2%**	10.0%	**31.9%**	**52.2%**
improvement	-39%	-49%	-	=	=	=	-0.9	=	+0.7
YOLOR-E6 [81]	115.8M	683.2G	1280	55.7%	73.2%	60.7%	40.1%	**60.4%**	69.2%
YOLOv7-E6	97.2M	515.2G	1280	**55.9%**	**73.5%**	**61.1%**	**40.6%**	60.3%	**70.0%**
improvement	-19%	-33%	-	+0.2	+0.3	+0.4	+0.5	-0.1	+0.8
YOLOR-D6 [81]	151.7M	935.6G	1280	56.1%	73.9%	61.2%	**42.4%**	60.5%	69.9%
YOLOv7-D6	154.7M	806.8G	1280	56.3%	73.8%	61.4%	41.3%	60.6%	70.1%
YOLOv7-E6E	151.7M	843.2G	1280	**56.8%**	**74.4%**	**62.1%**	40.8%	**62.1%**	**70.6%**
improvement	=	-11%	-	+0.7	+0.5	+0.9	-1.6	+1.6	+0.7

Experiment

Table 2: Comparison of state-of-the-art real-time object detectors.

Model	#Param.	FLOPs	Size	FPS	AP^{test} / AP^{val}	AP^{test}_{50}	AP^{test}_{75}	AP^{test}_{S}	AP^{test}_{M}	AP^{test}_{L}
YOLOX-S [21]	9.0M	26.8G	640	102	40.5% / 40.5%	-	-	-	-	-
YOLOX-M [21]	25.3M	73.8G	640	81	47.2% / 46.9%	-	-	-	-	-
YOLOX-L [21]	54.2M	155.6G	640	69	50.1% / 49.7%	-	-	-	-	-
YOLOX-X [21]	99.1M	281.9G	640	58	51.5% / 51.1%	-	-	-	-	-
PPYOLOE-S [85]	7.9M	17.4G	640	208	43.1% / 42.7%	60.5%	46.6%	23.2%	46.4%	56.9%
PPYOLOE-M [85]	23.4M	49.9G	640	123	48.9% / 48.6%	66.5%	53.0%	28.6%	52.9%	63.8%
PPYOLOE-L [85]	52.2M	110.1G	640	78	51.4% / 50.9%	68.9%	55.6%	31.4%	55.3%	66.1%
PPYOLOE-X [85]	98.4M	206.6G	640	45	52.2% / 51.9%	69.9%	56.5%	33.3%	56.3%	66.4%
YOLOv5-N (r6.1) [23]	1.9M	4.5G	640	159	- / 28.0%	-	-	-	-	-
YOLOv5-S (r6.1) [23]	7.2M	16.5G	640	156	- / 37.4%	-	-	-	-	-
YOLOv5-M (r6.1) [23]	21.2M	49.0G	640	122	- / 45.4%	-	-	-	-	-
YOLOv5-L (r6.1) [23]	46.5M	109.1G	640	99	- / 49.0%	-	-	-	-	-
YOLOv5-X (r6.1) [23]	86.7M	205.7G	640	83	- / 50.7%	-	-	-	-	-
YOLOR-CSP [81]	52.9M	120.4G	640	106	51.1% / 50.8%	69.6%	55.7%	31.7%	55.3%	64.7%
YOLOR-CSP-X [81]	96.9M	226.8G	640	87	53.0% / 52.7%	71.4%	57.9%	33.7%	57.1%	66.8%
YOLOv7-tiny-SiLU	6.2M	13.8G	640	286	38.7% / 38.7%	56.7%	41.7%	18.8%	42.4%	51.9%
YOLOv7	36.9M	104.7G	640	161	51.4% / 51.2%	69.7%	55.9%	31.8%	55.5%	65.0%
YOLOv7-X	71.3M	189.9G	640	114	53.1% / 52.9%	71.2%	57.8%	33.8%	57.1%	67.4%
YOLOv5-N6 (r6.1) [23]	3.2M	18.4G	1280	123	- / 36.0%	-	-	-	-	-
YOLOv5-S6 (r6.1) [23]	12.6M	67.2G	1280	122	- / 44.8%	-	-	-	-	-
YOLOv5-M6 (r6.1) [23]	35.7M	200.0G	1280	90	- / 51.3%	-	-	-	-	-
YOLOv5-L6 (r6.1) [23]	76.8M	445.6G	1280	63	- / 53.7%	-	-	-	-	-
YOLOv5-X6 (r6.1) [23]	140.7M	839.2G	1280	38	- / 55.0%	-	-	-	-	-
YOLOR-P6 [81]	37.2M	325.6G	1280	76	53.9% / 53.5%	71.4%	58.9%	36.1%	57.7%	65.6%
YOLOR-W6 [81]	79.8G	453.2G	1280	66	55.2% / 54.8%	72.7%	60.5%	37.7%	59.1%	67.1%
YOLOR-E6 [81]	115.8M	683.2G	1280	45	55.8% / 55.7%	73.4%	61.1%	38.4%	59.7%	67.7%
YOLOR-D6 [81]	151.7M	935.6G	1280	34	56.5% / 56.1%	74.1%	61.9%	38.9%	60.4%	68.7%
YOLOv7-W6	70.4M	360.0G	1280	84	54.9% / 54.6%	72.6%	60.1%	37.3%	58.7%	67.1%
YOLOv7-E6	97.2M	515.2G	1280	56	56.0% / 55.9%	73.5%	61.2%	38.0%	59.9%	68.4%
YOLOv7-D6	154.7M	806.8G	1280	44	56.6% / 56.3%	74.0%	61.8%	38.8%	60.1%	69.5%
YOLOv7-E6E	151.7M	843.2G	1280	36	56.8% / 56.8%	74.4%	62.1%	39.3%	60.5%	69.0%

[1] Our FLOPs is calaculated by rectangle input resolution like 640 × 640 or 1280 × 1280.
[2] Our inference time is estimated by using letterbox resize input image to make its long side equals to 640 or 1280.

Experiment

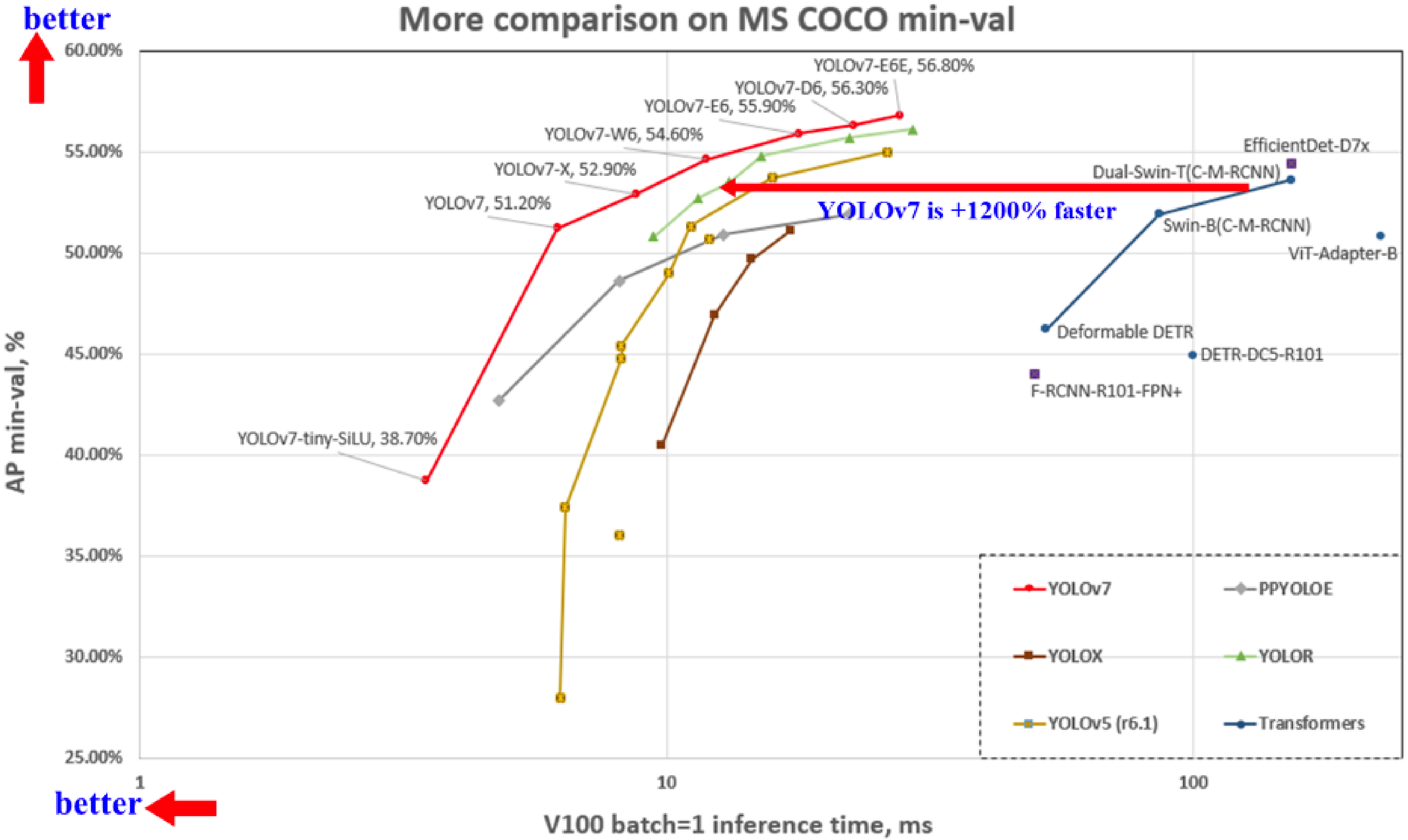
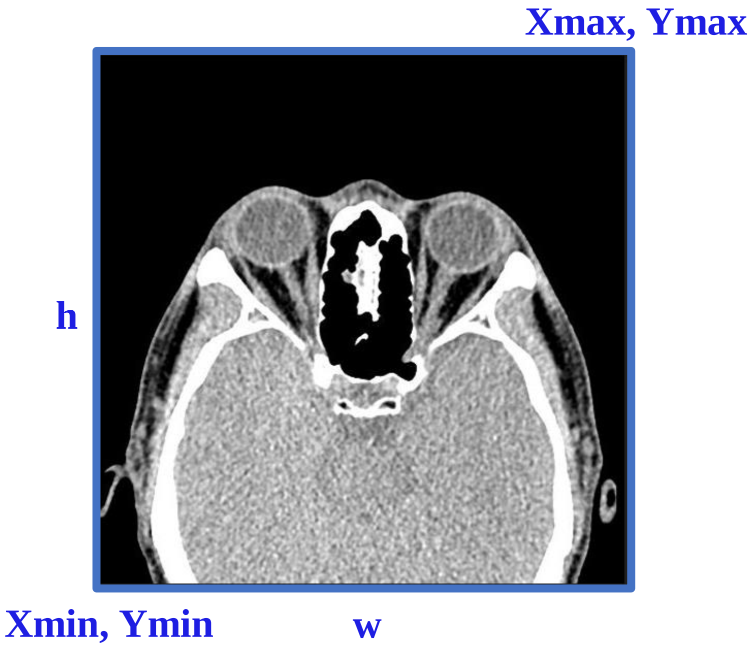


Figure 9: Comparison with other object detectors.

Train Information

Preprocessing

- ✓ Data format : jpg
- ✓ Input Size : 512 x512 ➡ 640 x 640
- ✓ Annotation Normalization



Normalization Formula

- ✓ Normalized(Xmin) = (Xmin+w/2)/Image_Width
- ✓ Normalized(Ymin) = (Ymin+h/2)/Image_Height
- ✓ Normalized(w) = w/Image_Width
- ✓ Normalized(h) = h/Image_Height

label index, N(Xmin), N(Ymin), N(w), N(h)

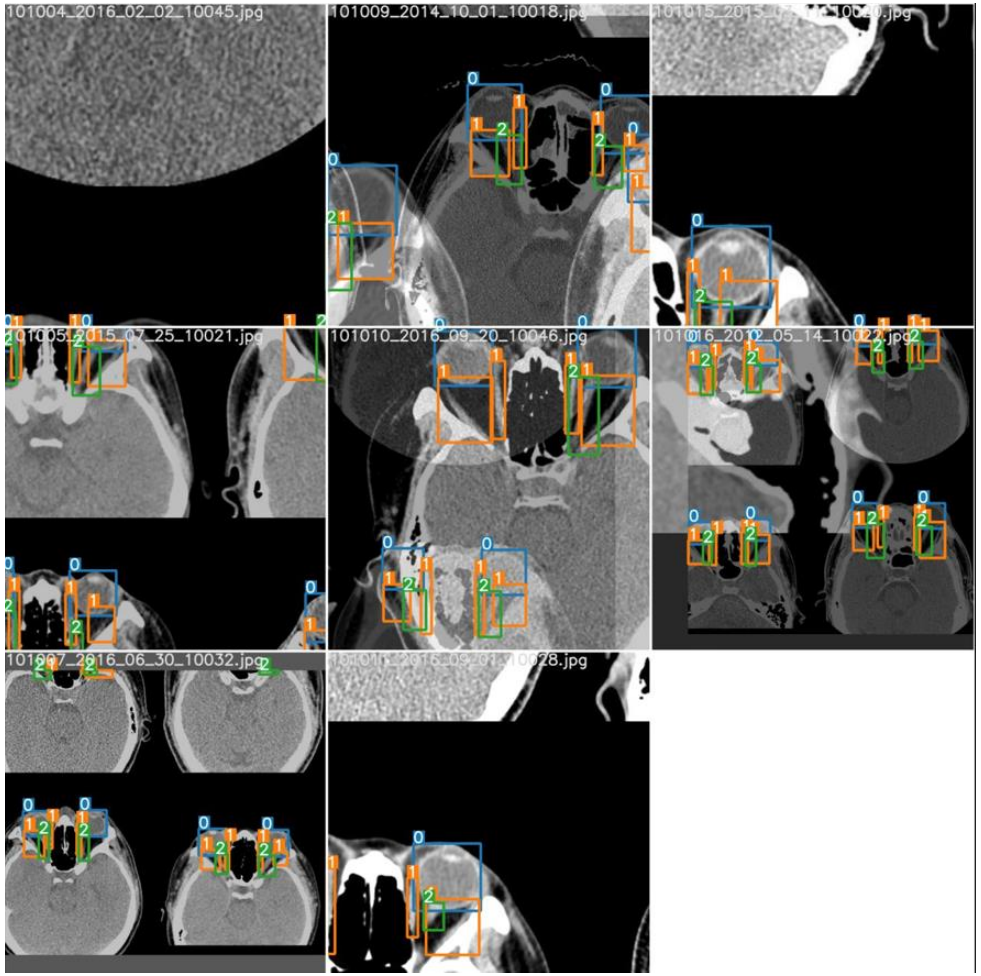
```
0 0.33203125 0.330078125 0.1640625 0.15234375
0 0.6943359375 0.341796875 0.154296875 0.15234375
1 0.40625 0.419921875 0.03125 0.16015625
1 0.615234375 0.421875 0.02734375 0.1640625
1 0.3203125 0.447265625 0.1171875 0.15234375
1 0.7314453125 0.4404296875 0.138671875 0.158203125
2 0.37890625 0.47265625 0.078125 0.1640625
2 0.64453125 0.474609375 0.078125 0.15234375
```

Train Information

Augmentation

- ✓ hsv_h : 0.015 image HSV-Hue augmentation (fraction)
- ✓ hsv_s : 0.7 image HSV-Saturation augmentation (fraction)
- ✓ hsv_v: 0.4 image HSV-Value augmentation (fraction)
- ✓ translate : 0.2 image translation (+/- fraction)
- ✓ scale : 0.9 image scale (+/- gain)
- ✓ fliplr : 0.5 image flip left-right (probability)
- ✓ mosaic : 1.0 image mixup (probability)
- ✓ mixup : 0.15 image mixup (probability)
- ✓ paste_in : 0.15 segment copy-paste (probability)

< training batch images >



AutoML

Experiment Setting

✓ Datasets (TAO)

Number of Samples: 1,189

train: 832

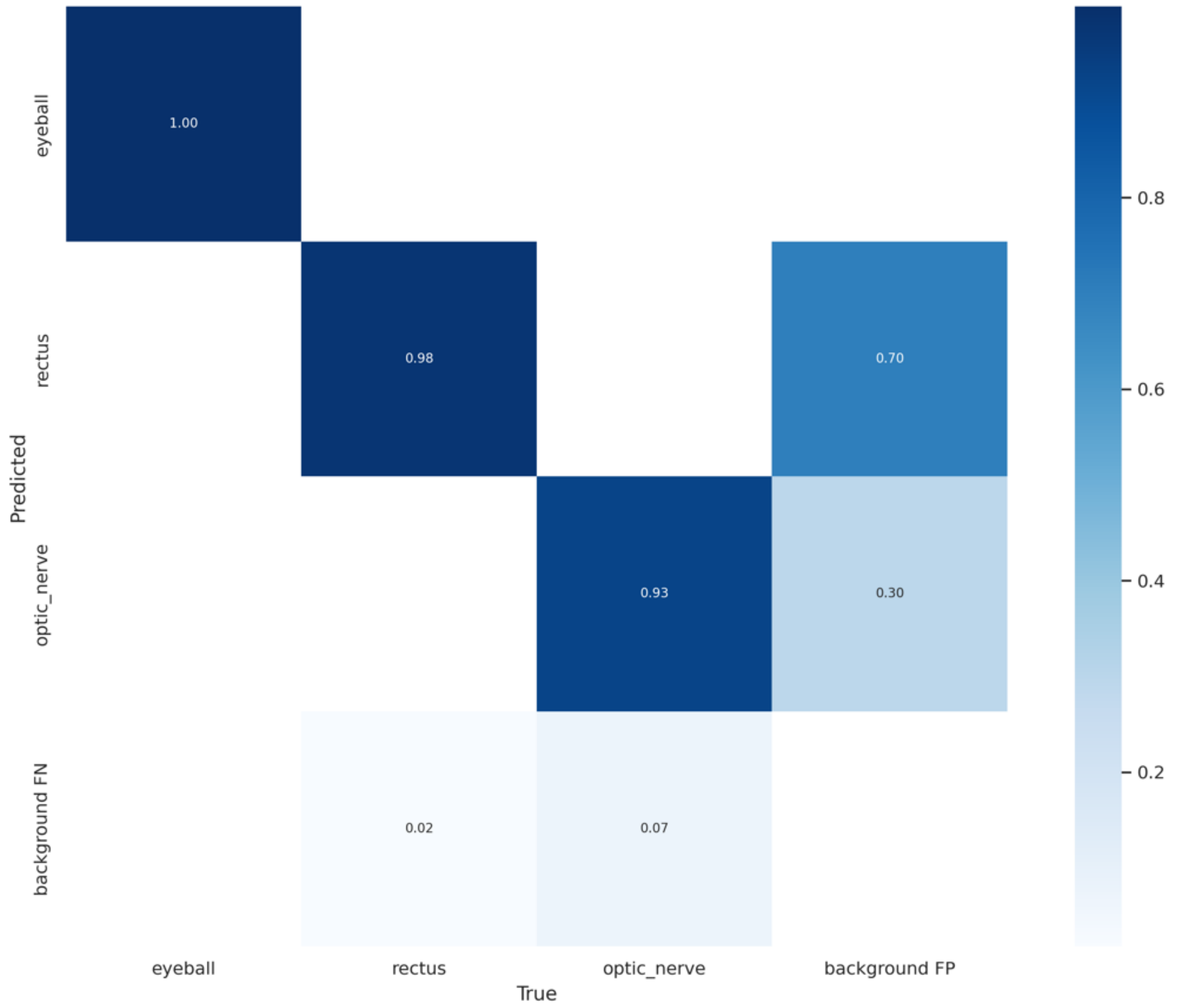
valid: 178

test: 179

✓ Hyper Parameter Setting

Pretrained weight	Model	Loss function	Optimizer	Epochs	Batch_size
Imagenet	YOLOv7	binary CE	SGD	50	8

Experimental Results

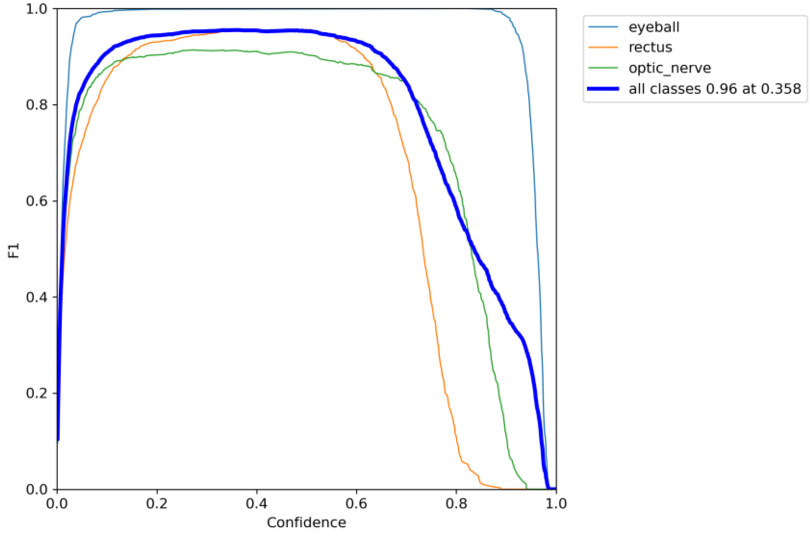


Confusion matrix

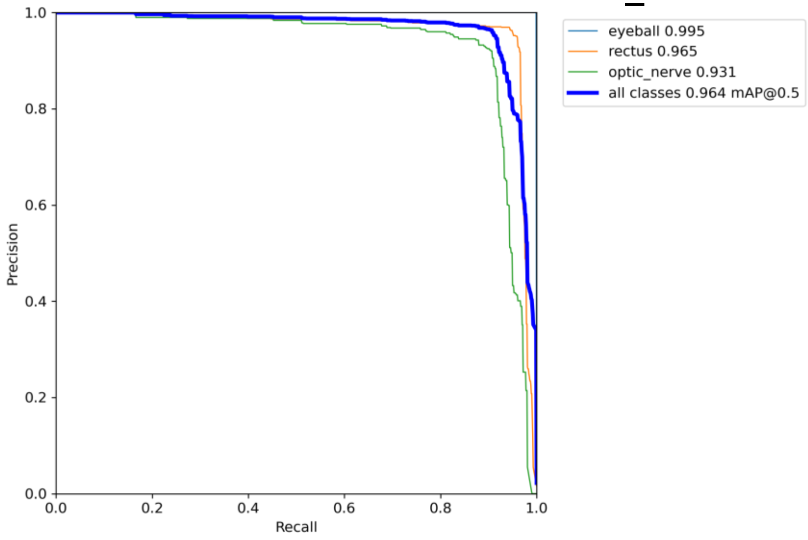
Experimental Results

AutoML

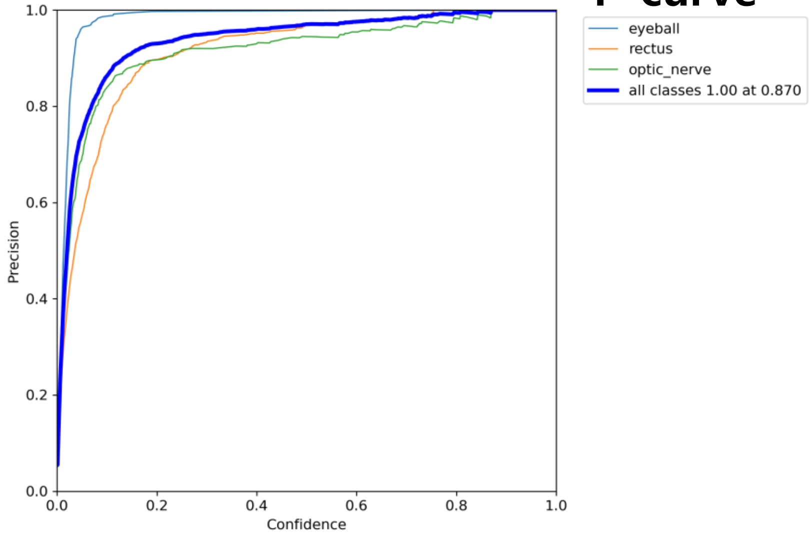
F1_curve



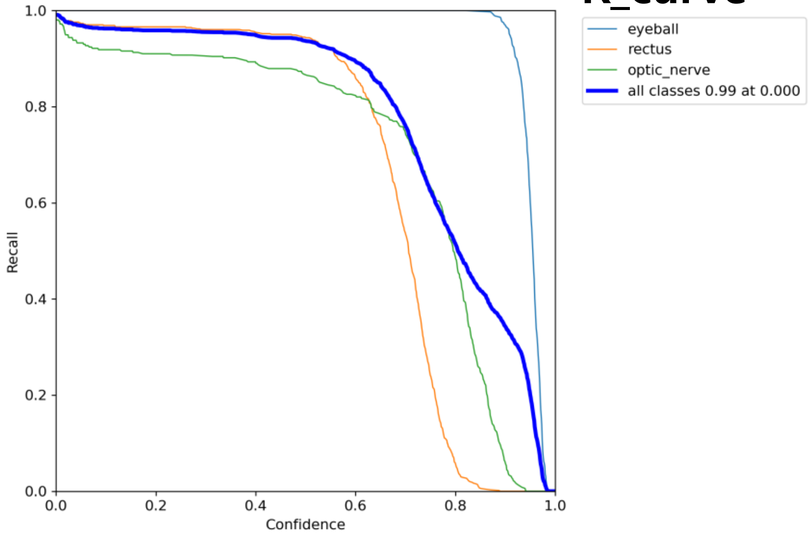
PR_curve



P-curve

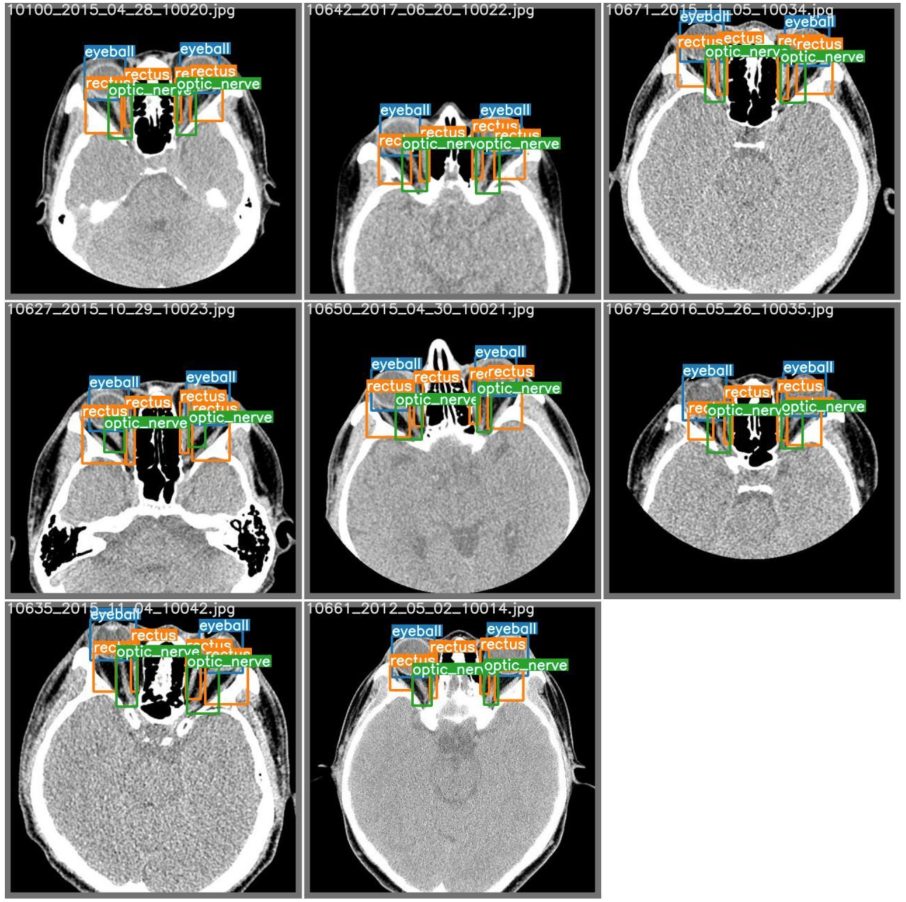


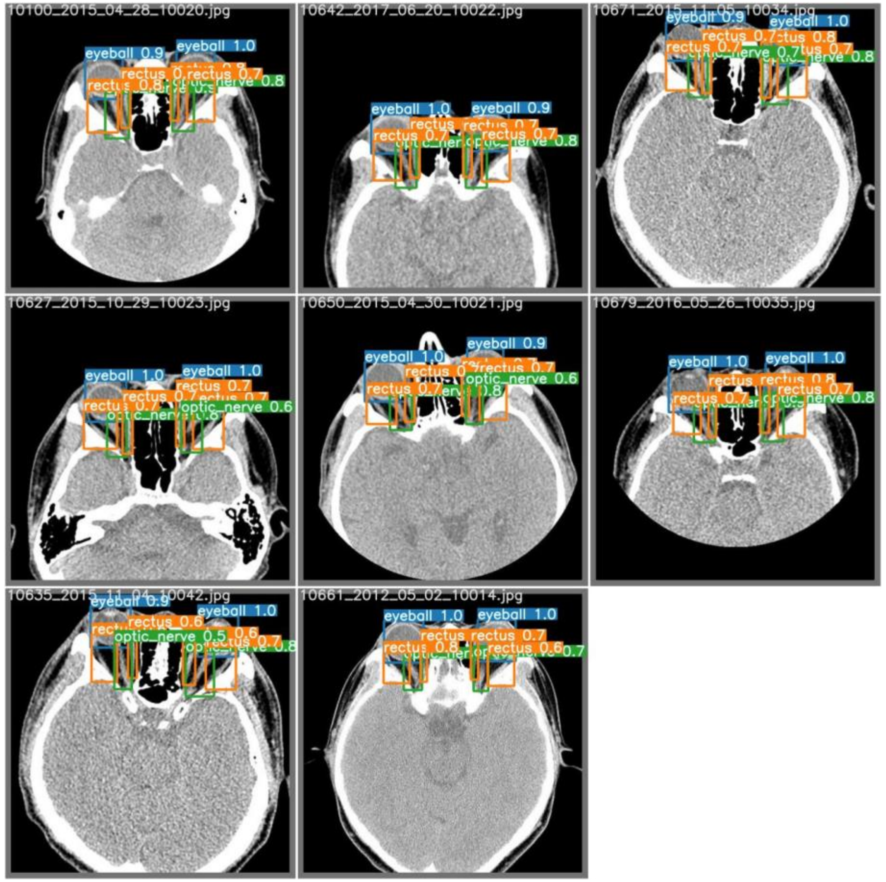
R_curve



Experimental Results

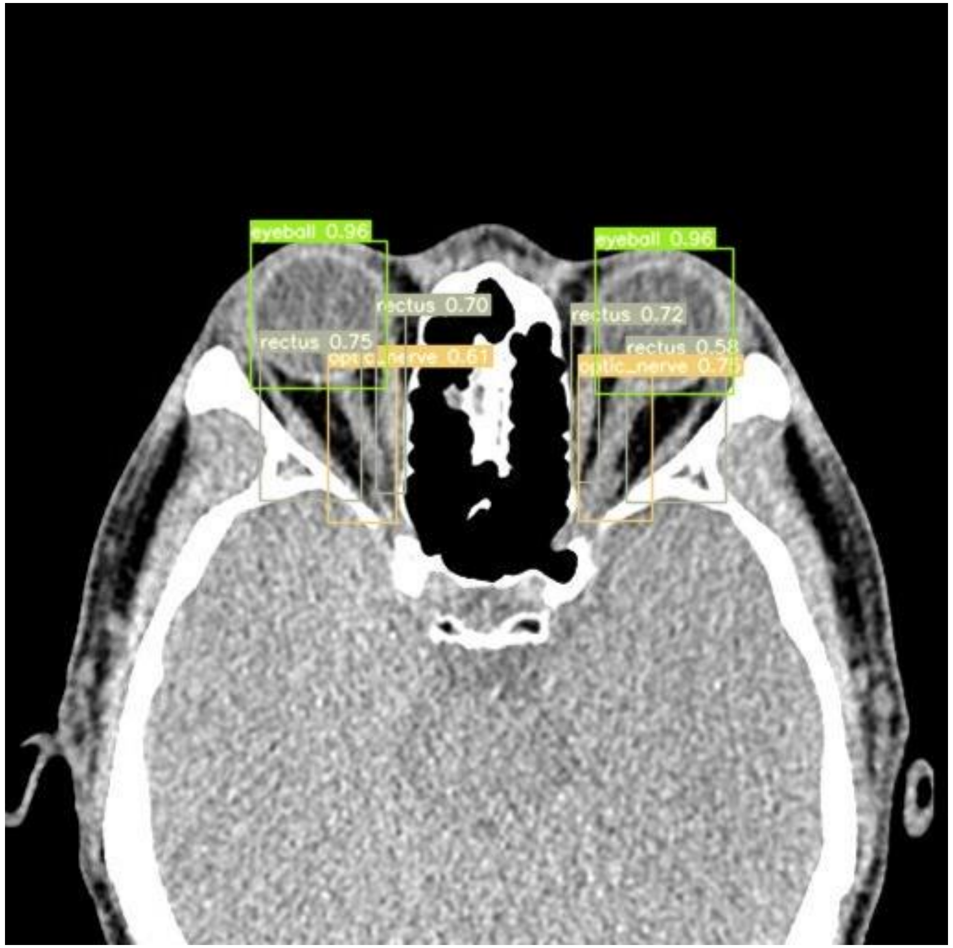
labels & pred





Experimental Results

Detect



Proposal

- ✓ Augmentation
- ✓ Fine tuning
- ✓ Optimizer (Hyper parameter)
- ✓ Model

Model	Test Size	AP^{test}	AP_{50}^{test}	AP_{75}^{test}	batch 1 fps	batch 32 average time
YOLOv7	640	**51.4%**	**69.7%**	**55.9%**	161 *fps*	2.8 *ms*
YOLOv7-X	640	**53.1%**	**71.2%**	**57.8%**	114 *fps*	4.3 *ms*
YOLOv7-W6	1280	**54.9%**	**72.6%**	**60.1%**	84 *fps*	7.6 *ms*
YOLOv7-E6	1280	**56.0%**	**73.5%**	**61.2%**	56 *fps*	12.3 *ms*
YOLOv7-D6	1280	**56.6%**	**74.0%**	**61.8%**	44 *fps*	15.0 *ms*
YOLOv7-E6E	1280	**56.8%**	**74.4%**	**62.1%**	36 *fps*	18.7 *ms*