유방암의 임파선 전이 예측

14기 - KRSJ

권진욱 류제욱 송일수 전대광

고용노동부 KOREATECH 직업능력심사평가원

목차

01. 프로젝트 개요

- 프로젝트 주제
  유방암 병리 슬라이드 영상과 임상 항목을 통한 유방암의 임파선 전이 여부 이진 분류 예측

- 프로젝트 개요
  tabular data, image data에서 각각 특징을 추출하고 합쳐, 하나의 딥러닝 모델 사용을 통한 이진 분류

- 활용 장비 및 재료(개발 환경 등)
  Python을 기반으로 Pytorch, sklearn, cv2 등의 라이브러리 사용

- 기대 효과
  임파선은 암의 전이에 치명적인 역할을 하므로, 림프절 전이 여부에 따라 치료와 예후가 크게 좌우된다.
  따라서 림프절 전이 여부와 전이 단계를 파악하는 것이 암의 치료와 진단에 매우 중요하므로 이에 기여한다.

02. 프로젝트팀 구성 및 역할

이름	역할	담당 업무	공동업무
권진욱	팀장	▶ Image data 전처리 및 EDA	▶ 전이여부 판별 모델링
류제욱	팀원	▶ Tabular data 특징 추출 탐색	
송일수	팀원	▶ Image data 특징 추출 탐색	
전대광	팀원	▶ Tabular data 전처리 및 EDA	

03. 프로젝트 수행 절차 및 방법

구분	기간	활동	비고
사전 기획 및 데이터 전처리	▶ 11/17(목) ~ 11/24(금)	▶ 프로젝트 기획 및 주제 선정 ▶ Baseline Model 분석 ▶ 데이터 전처리	
모델링	▶ 11/28(월) ~ 12/2(금)	▶ 다방면의 데이터 전처리 사용 및 테스트 ▶ HP 튜닝	
대회 종료	▶ 12/5(월)	▶ 최종 제출	
보완점 체크 및 성능 향상	▶ 12/5(월) ~ 12/9(금)	▶ 개선 방향 논의 ▶ 개선 모델 테스트	

04. 프로젝트 수행 결과

결과 제시 ① 데이터셋 소개 및 설명

학습 및 테스트 데이터 소개 (Train/test set)

· 유방암 환자의 전이여부 검사 결과 항목과 환자 개인정보, 병리학 이미지 경로 등이 입력된 28개 컬럼의 데이터

· 28개 컬럼 : ID, img_path, mask_path(Just train set), 나이, 수술연월일, 진단명, 암의위치, 암의개수, 암의장경, NG, HG, HG_score1, HG_score2, HG_score3, DCIS_or_LCIS 여부, DCIS_or_LCIS_type, T_category, ER, ER_Allred_score, PR, PR_Allred_score, KI-67_LI_percent, HER2, HER2_IHC, HER2_SISH, HER2_SISH_ratio, BRCA_mutation, N_categort(Just train set)

· Rows : 1000

	ID	img_path	mask_path	나이	수술연월일	진단명	암의위치	암의개수	암의장경	NG	...	ER_Allred_score	PR	PR_Allred_score	KI-67_LI_percent	HER2	HER2_IHC	HER2_
0	BC_01_0001	./train_imgs/BC_01_0001.png	-	63	2015-10-23	1	2	1	19.0	2.0	...	8.0	1.0	6.0	12.0	0.0	1.0	
1	BC_01_0002	./train_imgs/BC_01_0002.png	-	51	2015-10-28	1	1	1	22.0	3.0	...	NaN	0.0	NaN	70.0	0.0	0.0	
2	BC_01_0003	./train_imgs/BC_01_0003.png	-	37	2015-10-29	1	2	1	NaN	2.0	...	7.0	1.0	4.0	7.0	0.0	1.0	
3	BC_01_0004	./train_imgs/BC_01_0004.png	-	54	2016-03-08	1	2	1	0.0	3.0	...	NaN	0.0	NaN	1.0	1.0	3.0	
4	BC_01_0005	./train_imgs/BC_01_0005.png	-	57	2015-10-30	1	2	1	8.0	2.0	...	8.0	0.0	NaN	8.0	1.0	2.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
995	BC_01_3464	./train_imgs/BC_01_3464.png	-	65	2006-12-22	1	2	1	25.0	1.0	...	NaN	0.0	NaN	NaN	0.0	0.0	
996	BC_01_3482	./train_imgs/BC_01_3482.png	-	48	2006-11-17	1	1	1	7.0	1.0	...	NaN	1.0	NaN	NaN	0.0	0.0	
997	BC_01_3485	./train_imgs/BC_01_3485.png	-	64	2006-11-10	1	2	1	15.0	1.0	...	NaN	1.0	NaN	NaN	0.0	0.0	
998	BC_01_3502	./train_imgs/BC_01_3502.png	-	50	2006-09-22	1	1	1	7.0	1.0	...	NaN	0.0	NaN	NaN	0.0	0.0	
999	BC_01_3518	./train_imgs/BC_01_3518.png	-	76	2010-09-01	1	2	1	20.0	3.0	...	4.0	0.0	2.0	NaN	0.0	1.0	

1000 rows × 28 columns

	ID	img_path	나이	수술연월일	진단명	암의위치	암의개수	암의장경	NG	HG	...	ER	ER_Allred_score	PR	PR_Allred_score	KI-67_LI_percent	HER2	HER2_IHC	HER2_S
0	BC_01_0011	./test_imgs/BC_01_0011.png	55	2015-11-17	2	2	1	23.0	2.0	2.0	...	1.0	8.0	0.0	2.0	5.00	0.0	1.0	
1	BC_01_0220	./test_imgs/BC_01_0220.png	43	2020-06-09	4	2	1	13.0	3.0	2.0	...	1.0	4.0	1.0	8.0	8.67	0.0	0.0	
2	BC_01_0233	./test_imgs/BC_01_0233.png	76	2020-05-14	1	1	1	NaN	NaN	NaN	...	1.0	6.0	1.0	6.0	NaN	NaN	2.0	
3	BC_01_0258	./test_imgs/BC_01_0258.png	58	2020-05-20	1	2	1	1.3	2.0	2.0	...	1.0	7.0	0.0	NaN	21.17	1.0	3.0	
4	BC_01_0260	./test_imgs/BC_01_0260.png	56	2020-05-20	1	2	2	15.0	3.0	3.0	...	1.0	8.0	1.0	3.0	20.57	1.0	3.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
245	BC_01_3328	./test_imgs/BC_01_3328.png	61	2009-10-30	1	1	1	18.0	2.0	2.0	...	1.0	5.0	1.0	3.0	NaN	0.0	0.0	
246	BC_01_3404	./test_imgs/BC_01_3404.png	42	2009-05-19	1	2	1	20.0	1.0	2.0	...	1.0	5.0	1.0	5.0	NaN	0.0	1.0	
247	BC_01_3418	./test_imgs/BC_01_3418.png	37	2009-04-24	1	1	1	17.0	1.0	1.0	...	1.0	5.0	1.0	5.0	NaN	0.0	0.0	
248	BC_01_3438	./test_imgs/BC_01_3438.png	37	2009-02-06	1	1	1	7.0	1.0	1.0	...	1.0	5.0	1.0	5.0	NaN	0.0	0.0	
249	BC_01_3446	./test_imgs/BC_01_3446.png	84	2009-02-06	1	1	1	3.0	1.0	1.0	...	0.0	NaN	0.0	NaN	NaN	0.0	0.0	

250 rows × 26 columns

04. 프로젝트 수행 결과

결과 제시 ① 데이터셋 소개 및 설명

학습 및 테스트 이미지데이터 소개 (Train/test set)

· 유방암 환자의 병리 슬라이드 이미지 사용

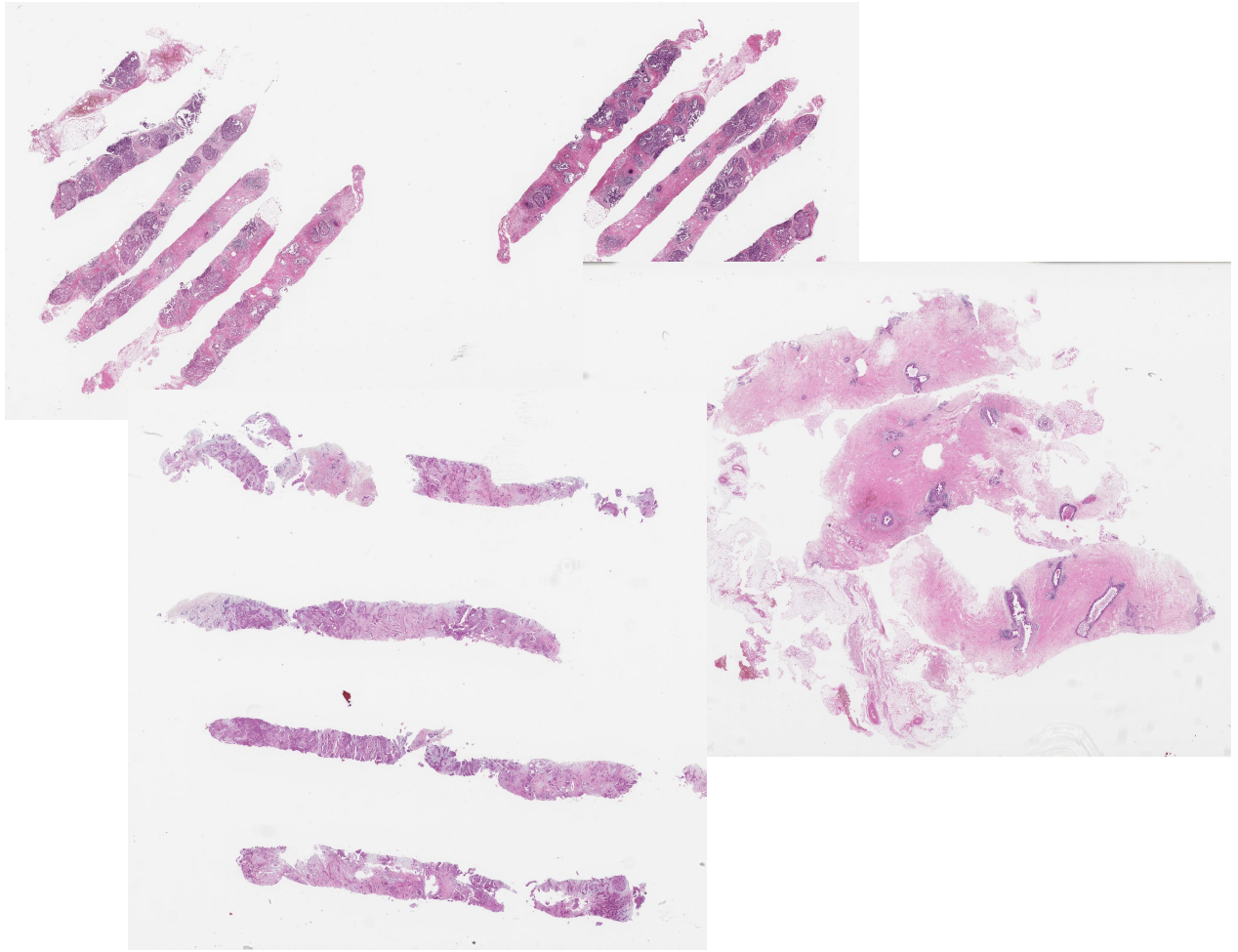
· 이미지 크기 : height max 8299 / width max 3991

```
train_width = train_path['width'].max()
train_height = train_path['height'].max()

print(train_width, train_height)
```

```
8299 3991
```

```
test_width = test_path['width'].max()
test_height = test_path['height'].max()

print(test_width, test_height)
```

```
8134 3965
```



04. 프로젝트 수행 결과

결과 제시 ① 데이터셋 소개 및 설명

데이터셋 설명

** 숫자형 데이터 : 나이, 암의장경, ER_Allred_score, PR_Allred_score, KI-67_LI_percent, HER2_SISH_ratio

· 나이 : 환자 나이

· 암의 장경 : 암의 Size(mm), 여러 개의 암일 경우 가장 큰 종양의 장경

· ER_Allred_score / PR_Allred_score : Allred 점수를 통해 양성 또는 음성으로 분류하였으며, 비율점수(범위 0~5점)와 강도점수(범위 0~3점)의 합으로 Allred 점수를 계산.

· KI-67_LI_percent : 유방암 조직의 세포증식 수치를 나타내는 척도(높을수록 증식 세포수가 높기 때문에 증식속도가 빨라질 가능성이 높음. 정수 범위 0 ~ 100)

· HER2_SISH_ratio : HER2_SISH 검사의 결과를 소수점 첫째자리까지 숫자로 기입

04. 프로젝트 수행 결과

결과 제시 ① 데이터셋 소개 및 설명

데이터셋 설명

** 범주형 데이터 : 진단명, 암의 위치, 암의개수, NG, HG, HG_score_1, HG_score_2, HG_score_3, DCIS_or_LCIS_여부, DCIS_or_LCIS_type, T_category, ER, PR, HER2, HER2_IHC, HER2_SISH, BRCA_mutation, N_category

· 진단명 : Ductal(젖관암종) / Lobular(소엽성 상피내암) / Mucinous(점액암) / other(기타) 4개 범주 분류

· 암의위치 : Right(우측) / Left(좌측) / Both(둘다) 3개 범주 분류

· 암의개수 : Single(한개) / Multiple(두개이상)

· NG(Nuclear grade) : 핵의 모양에 대한 분화도. NG1/NG2/NG3 3개 범주 분류. 숫자가 커질수록 정상에서 멀어짐 분화도가 나쁘면 성장이 빠르고 공격적인 암세포임.

· HG(Histologic grade) : 조직 분화도. HG1/HG2/HG3/not grade 4개 범주 분류. HG_score들의 합산 점수 산출 결과이며, 분화도가 좋을수록 합산 점수가 낮다.

· HG_score_1/2/3 : Tubule formation(세뇨관으로 형성되는 암세포 비율) / Nuclear Pleomorphism(정상세포의 핵과의 차이점) / Mitotic Rate(암세포의 증식 또는 분열 속도), 점수가 낮을수록 정상에 가까움.

04. 프로젝트 수행 결과

결과 제시 ① 데이터셋 소개 및 설명

데이터셋 설명

** 범주형 데이터 : 진단명, 암의 위치, 암의개수, NG, HG, HG_score_1, HG_score_2, HG_score_3, DCIS_or_LCIS_여부, DCIS_or_LCIS_type, T_category, ER, PR, HER2, HER2_IHC, HER2_SISH, BRCA_mutation, N_category

· DCIS_or_LCIS_여부 : 제자리 암종의 유무. 암의 확장 유무. no DCIS/LCIS | DCIS/LCIS present EIC | DCIS/LCIS presern EIC 3개 범주 분류

· DCIS_or_LCIS_type : 제자리 암종내 괴사 유무. non-comedo/comedo 2개 범주 분류

· T_category : 종양 침범도. TIs:상피내암종/T1:고유층, 점막근층, 또는 점막하층 침범/T2:고유근층 침범/T3:내장 복막 등 인접 구조 침범 없이 결합조직 침범/T4:종양이 복막 또는 인접 구조를 침범

· ER(Estrogen receptor) : 에스트로겐 수용체의 발현 여부. Allred_score 에 의해 결정. 호르몬 수용체가 유방암의 성장을 돕기 때문에 수용체 판별이 필요함.

· PR(Progesteron receptor) : 프로게스테론 수용체의 발현 여부. Allred_score 에 의해 결정.

· HER2 : HER2 사람상피세포성장인자수용체로, 정상 유방세포에서도 발견되는 단백질이지만 원암유전자로서 유전자가 증폭되거나 HER2 단백질이 과발현 되면 암세포 증식이 촉진된다.

04. 프로젝트 수행 결과

결과 제시 ① 데이터셋 소개 및 설명

데이터셋 설명

** 범주형 데이터 : 진단명, 암의 위치, 암의개수, NG, HG, HG_score_1, HG_score_2, HG_score_3, DCIS_or_LCIS_여부, DCIS_or_LCIS_type, T_category, ER, PR, HER2, HER2_IHC, HER2_SISH, BRCA_mutation, N_category

· HER2_IHC : HER2 단백질 양성 판별을 위한 검사. Negative/Equivocal/Positive 3개 카테고리 분류

· HER2_SISH : 로슈의 허셉틴 치료를 요하는 환자군을 선별하는 유전자 검사. 양성일 경우 허셉틴 치료를 진행. 2개 카테고리 분류

· BRCA_mutation : 유방암 유전자 변이 여부. BRCA1/BRCA2 모두 손상된 DNA를 복구하는데 도움을 주는 단백질 생산 유전자. 이 유전자에서 돌연변이가 발생하여 유전 확률이 50% 이며 유방암 위험도가 높아진다. No BRCA1,BRCA2/BRCA1 mutation/BRCA2 mutation/검사 안한 경우

· N_category : 임파선(림프절) 전이 여부. 0: 미발생 / 1: 발생

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

데이터셋 정보

```
train.info()
✓ 0.1s
Output exceeds the size limit. Open the full output d
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 28 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   ID                 1000 non-null   object
 1   img_path           1000 non-null   object
 2   mask_path          1000 non-null   object
 3   나이                 1000 non-null   int64
 4   수술연월일              1000 non-null   object
 5   진단명                1000 non-null   int64
 6   암의 위치              1000 non-null   int64
 7   암의 개수              1000 non-null   int64
 8   암의 장경              931 non-null    float64
 9   NG                 949 non-null    float64
 10  HG                 914 non-null    float64
 11  HG_score_1         908 non-null    float64
 12  HG_score_2         908 non-null    float64
 13  HG_score_3         911 non-null    float64
 14  DCIS_or_LCIS_여부    1000 non-null   int64
 15  DCIS_or_LCIS_type  126 non-null    float64
 16  T_category         996 non-null    float64
 17  ER                 999 non-null    float64
 18  ER_Allred_score    703 non-null    float64
 19  PR                 999 non-null    float64
...
 26  BRCA_mutation      55 non-null     float64
 27  N_category         1000 non-null   int64
dtypes: float64(18), int64(6), object(4)
memory usage: 218.9+ KB
```

<학습 데이터 정보>

```
test.info()
✓ 0.7s
Output exceeds the size limit. Open the full output c
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 250 entries, 0 to 249
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   ID                 250 non-null    object
 1   img_path           250 non-null    object
 2   나이                 250 non-null    int64
 3   수술연월일              250 non-null    object
 4   진단명                250 non-null    int64
 5   암의 위치              250 non-null    int64
 6   암의 개수              250 non-null    int64
 7   암의 장경              237 non-null    float64
 8   NG                 235 non-null    float64
 9   HG                 234 non-null    float64
 10  HG_score_1         232 non-null    float64
 11  HG_score_2         232 non-null    float64
 12  HG_score_3         232 non-null    float64
 13  DCIS_or_LCIS_여부    250 non-null    int64
 14  DCIS_or_LCIS_type  31 non-null     float64
 15  T_category         249 non-null    float64
 16  ER                 250 non-null    float64
 17  ER_Allred_score    175 non-null    float64
 18  PR                 250 non-null    float64
 19  PR_Allred_score    133 non-null    float64
...
 24  HER2_SISH_ratio    45 non-null     float64
 25  BRCA_mutation      12 non-null     float64
dtypes: float64(18), int64(5), object(3)
memory usage: 50.9+ KB
```

<테스트 데이터 정보>

- 컬럼 및 인덱스 범위 확인.
- 컬럼별 결측치 갯수 확인.
- 컬럼별 데이터 타입 확인.

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

중복데이터 제거 및 학습에 사용하지 않을 컬럼 삭제

```
# drop_duplicates
dataframe.drop_duplicates(inplace=True)

# drop non-training columns
dataframe.drop(['DCIS_or_LCIS_type', 'HER2_SISH', 'HER2_SISH_ratio'], axis = 1, inplace=True)
```

· DCIS_or_LCIS_type : 결측치 대체에 참고할 컬럼이 없다고 판단하여 삭제 결정.

```
train['HER2_SISH'].isnull().sum(), train['HER2_SISH_ratio'].isnull().sum(), train['HER2_IHC'].isnull().sum()
✓ 0.3s
(753, 825, 24)
```

· HER2_SISH/HER2_SISH_ratio : 결측치가 매우 많고, HER2 단백질 양성 판별 검사에서 우선적으로 진행되는 HER2_IHC 컬럼의 결측치가 적기 때문에 해당 두 컬럼 삭제 결정.

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

결측치 대체

```
# fillna BRCA_mutation, 3 => '3' is new category that means 'not examine'
dataframe['BRCA_mutation'].fillna(3.0, inplace=True)
```

· BRCA_mutation : 검사 시행 안한 경우가 빈칸으로 설정되어있어 3이라는 새로운 범주를 부여

```
# 'T_category' column fill in mode
dataframe['T_category'].fillna(dataframe['T_category'].mode()[0], inplace=True)
```

```
train['T_category'].isnull().sum(), test['T_category'].isnull().sum()
```

✓ 0.3s

```
(4, 1)
```

· T_category : 종양의 침범 정도를 5개 범주로 구분하면서, 결측치가 적으므로 최빈도 값으로 결측치 대체

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

결측치 대체

```python
# data processing about 'HG' columns
for i in range(len(train_df)):
    # if every values about 'HG' is null that fill in grade '4'
    if (np.isnan(train_df.loc[i, 'HG'])) & (np.isnan(train_df.loc[i, 'HG_score_
        train_df.loc[i, 'HG'] = 4.0
        train_df.loc[i, 'HG_score_1'] = 4.0
        train_df.loc[i, 'HG_score_2'] = 4.0
        train_df.loc[i, 'HG_score_3'] = 4.0
    # if every values about 'HG_score' except 'HG' is null that fill in conditi
    elif (not np.isnan(train_df.loc[i, 'HG'])) & (np.isnan(train_df.loc[i, 'HG_
        # if 'HG' is 2.0 that fill in 2.0
        if train_df.loc[i, 'HG'] == 2.0:
            train_df.loc[i, 'HG_score_1'] = 2.0
            train_df.loc[i, 'HG_score_2'] = 2.0
            train_df.loc[i, 'HG_score_3'] = 2.0
        # if 'HG' is 3.0 that fill in 3.0
        elif train_df.loc[i, 'HG'] == 3.0:
            train_df.loc[i, 'HG_score_1'] = 3.0
            train_df.loc[i, 'HG_score_2'] = 3.0
            train_df.loc[i, 'HG_score_3'] = 3.0
        else:
            train_df.loc[i, 'HG_score_1'] = 1.0
            train_df.loc[i, 'HG_score_2'] = 1.0
            train_df.loc[i, 'HG_score_3'] = 1.0
    # other 'HG' cases drop index at the end of EDA
```

```
train['HG'].isnull().sum(), train['HG_score_1'].isnull().sum()
✓ 0.4s
(86, 92)

train['HG_score_2'].isnull().sum(), train['HG_score_3'].isnull().sum()
✓ 0.7s
(92, 89)
```

· 관련 컬럼 모두가 결측치일 경우, 4(not grade)

· HG 제외한 모든 컬럼이 결측치일 경우,

- HG가 2.0이면, 나머지 컬럼 2로 대체
- HG가 3.0이면, 나머지 컬럼 3으로 대체
- 나머지 경우 1로 대체

· 위 조건을 제외한 경우는 EDA 마지막 부분에서 삭제

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

결측치 대체

```
train['NG'].isnull().sum(), test['NG'].isnull().sum()
```

✓ 0.5s

```
(51, 15)
```

```
# data processing about 'NG' columns
for f in range(len(dataframe)):
    if (np.isnan(dataframe.loc[f, 'NG'])) & (dataframe.loc[f, 'HG'] == 1.0) | (dataframe.loc[f, 'HG'] == 4.0):
        dataframe.loc[f, 'NG'] = 1.0
    elif (np.isnan(dataframe.loc[f, 'NG'])) & (dataframe.loc[f, 'HG'] == 2.0):
        dataframe.loc[f, 'NG'] = 2.0
    elif (np.isnan(dataframe.loc[f, 'NG'])) & (dataframe.loc[f, 'HG'] == 3.0):
        dataframe.loc[f, 'NG'] = 3.0
```

· NG : 결측값이 적고, 핵 모양에 대한 분화도를 나타내기 때문에 조직분화도를 나타내는 HG를 참조하여 결측치 대체

- HG가 1이거나 4인 경우, 1로 대체
- HG가 2인 경우, 2로 대체
- HG가 3인 경우, 3으로 대체

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

결측치 대체

```
train['ER'].isnull().sum(), train['PR'].isnull().sum()
```

34] ✓ 0.2s

```
(1, 1)
```

· 에스트로겐 수용체, 프로게스테론 수용체 발현 여부 판별 컬럼 결측치 수가 적음.

```
# if 'ER' or 'PR' is null that fill in condition
for g in range(len(dataframe)):
    if (np.isnan(dataframe.loc[g, 'ER'])) | (np.isnan(dataframe.loc[g, 'PR'])):
        dataframe.drop(g, inplace=True, axis = 0)
dataframe.reset_index(drop=True, inplace=True)
```

· 둘 중 하나라도 결측값이 있다면 해당 인덱스는 삭제 하는 것으로 결정.

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

결측치 대체

```
train['ER_Allred_score'].isnull().sum(), train['PR_Allred_score'].isnull().sum()
```

✓ 0.6s

```
(297, 453)
```

· 각 Allred_score 점수를 바탕으로 ER과 PR이 결정되는데 ER과 PR의 결측값이 매우 적었으므로, ER과 PR을 기준으로 해당 컬럼 결측치를 대체

```
for j in range(len(dataframe)):
    if (dataframe.loc[j, 'ER'] == 0.0) & (np.isnan(dataframe.loc[j, 'ER_Allred_score'])):
        dataframe.loc[j, 'ER_Allred_score'] = 0.0
    elif (dataframe.loc[j, 'ER'] == 1.0) & (np.isnan(dataframe.loc[j, 'ER_Allred_score'])):
        dataframe.loc[j, 'ER_Allred_score'] = dataframe['ER'].mean()

for d in range(len(dataframe)):
    if (dataframe.loc[d, 'PR'] == 0.0) & (np.isnan(dataframe.loc[d, 'PR_Allred_score'])):
        dataframe.loc[d, 'PR_Allred_score'] = 0.0
    elif (dataframe.loc[d, 'PR'] == 1.0) & (np.isnan(dataframe.loc[d, 'PR_Allred_score'])):
        dataframe.loc[d, 'PR_Allred_score'] = dataframe['PR'].mean()
```

· 각 ER/PR이 0이면서 Allred_score가 결측치이면 0으로 대체

· 각 ER/PR이 1이면서 Allred_score가 결측치이면 각 ER/PR의 평균값으로 대체

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

결측치 대체

```
train['KI-67_LI_percent'].isnull().sum(), test['KI-67_LI_percent'].isnull().sum()
```

✓ 0.4s

```
(235, 52)
```

```
# 'KI-67' column fill in mean
dataframe['KI-67_LI_percent'].fillna(dataframe['KI-67_LI_percent'].mean(), inplace=True)
```

· KI-67_LI_percent : 결측치가 적고 세포증식을 수치화한 자료이므로, 평균값으로 대체

```
train['암의 장경'].isnull().sum(), test['암의 장경'].isnull().sum()
```

✓ 0.1s

```
(69, 13)
```

```
# '암의 장경' fill in mean
dataframe['암의 장경'].fillna(dataframe['암의 장경'].mean(), inplace=True)
```

· 암의 장경 : 결측치가 적고 암의 크기를 나타내는 값이므로, 평균값으로 대체

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

결측치 대체

```
train['HER2'].isnull().sum(), train['HER2_IHC'].isnull().sum(), test['HER2'].isnull().sum(), test['HER2_IHC'].isnull().sum()
```

✓ 0.3s

```
(14, 24, 6, 3)
```

```
# 'HER2_IHC' fill in contion
for h in range(len(dataframe)):
    if np.isnan(dataframe.loc[h, 'HER2_IHC']):
        dataframe.loc[h, 'HER2_IHC'] = dataframe.loc[h, 'HER2']
    elif (dataframe.loc[h, 'HER2_IHC'] < 2) & (dataframe.loc[h, 'HER2'] == 1):
        dataframe.loc[h, 'HER2_IHC'] = 2.0
    elif (dataframe.loc[h, 'HER2_IHC'] > 1) & (dataframe.loc[h, 'HER2'] == 0):
        dataframe.loc[h, 'HER2_IHC'] = 1.0
    elif np.isnan(dataframe.loc[h, 'HER2']):
        if dataframe.loc[h, 'HER2_IHC'] >= 2:
            dataframe.loc[h, 'HER2'] = 1.0
        elif dataframe.loc[h, 'HER2_IHC'] < 2:
            dataframe.loc[h, 'HER2'] = 0.0
```

· HER2는 HER2_IHC와 HER2_SISH를 모두 고려한 결과이지만, HER2_SISH를 결정하는 ratio와 HER2_ratio 모두 결측값이 75~85%이기 때문에 HER2와 HER2_IHC는 두 컬럼간에 상호의존성을 갖는 것으로 판단하고 결측치 대체 시행.

· HER2_IHC가 결측치일 경우, HER2_IHC 값은 HER2값을 따른다.

- HER2_IHC가 2보다 작고 HER2가 1이면, HER2_IHC는 2로 대체
- HER2_IHC가 1보다 크고 HER2가 0이면, HER2_IHC는 1로 대체
- HER2가 결측치일 경우 HER2_IHC가 2보다 크거나 같으면 HER2는 1, HER2_IHC가 2보다 작으면 HER2는 0으로 대체

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

결측치 대체

```
# drop another 'NaN' values
dataframe.dropna(inplace = True)
dataframe.reset_index(drop=True, inplace=True)
```

· 이외 결측치인 인덱스 행들에 대하여 삭제 진행.

· 데이터프레임 인덱스 리셋.

```
# EDA
def data_preprocessing(dataframe):
```

```
train_set = data_preprocessing(train)
test_df = data_preprocessing(test)
```

✓ 0.9s

· 전처리 과정 재사용을 위한 전체 함수화 진행

· 전처리 후 훈련데이터 인덱스 : 1000 → 996

전처리 후 테스트데이터 인덱스 : 250 → 250

```
train_set.isnull().count()
```

✓ 0.8s

ID	996
img_path	996
mask_path	996
나이	996
수술연월일	996
진단명	996
암의 위치	996
암의 개수	996
암의 장경	996
NG	996
HG	996
HG_score_1	996
HG_score_2	996
HG_score_3	996
DCIS_or_LCIS_여부	996
T_category	996
ER	996
ER_Allred_score	996
PR	996
PR_Allred_score	996
KI-67_LI_percent	996
HER2	996
HER2_IHC	996
BRCA_mutation	996
N_category	996

dtype: int64

```
test_df.isnull().count()
```

✓ 0.8s

ID	250
img_path	250
나이	250
수술연월일	250
진단명	250
암의 위치	250
암의 개수	250
암의 장경	250
NG	250
HG	250
HG_score_1	250
HG_score_2	250
HG_score_3	250
DCIS_or_LCIS_여부	250
T_category	250
ER	250
ER_Allred_score	250
PR	250
PR_Allred_score	250
KI-67_LI_percent	250
HER2	250
HER2_IHC	250
BRCA_mutation	250

dtype: int64

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

이미지 데이터 전처리

```
train_width = train_path['width'].max()
train_height = train_path['height'].max()

print(train_width, train_height)
```

```
8299 3991
```

```
test_width = test_path['width'].max()
test_height = test_path['height'].max()

print(test_width, test_height)
```

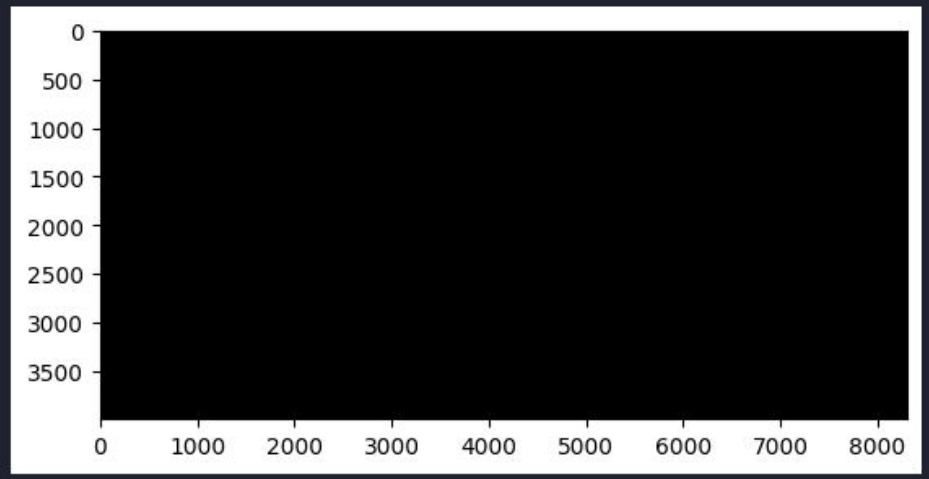
```
8134 3965
```

· 학습 및 테스트 이미지 width 최댓값 : 8299

· 학습 및 테스트 이미지 height 최댓값 : 3991

```
pannel = np.zeros((pannel_height, pannel_width, 3), np.uint8)
plt.imshow(pannel)
```

```
<matplotlib.image.AxesImage at 0x15765ec10>
```

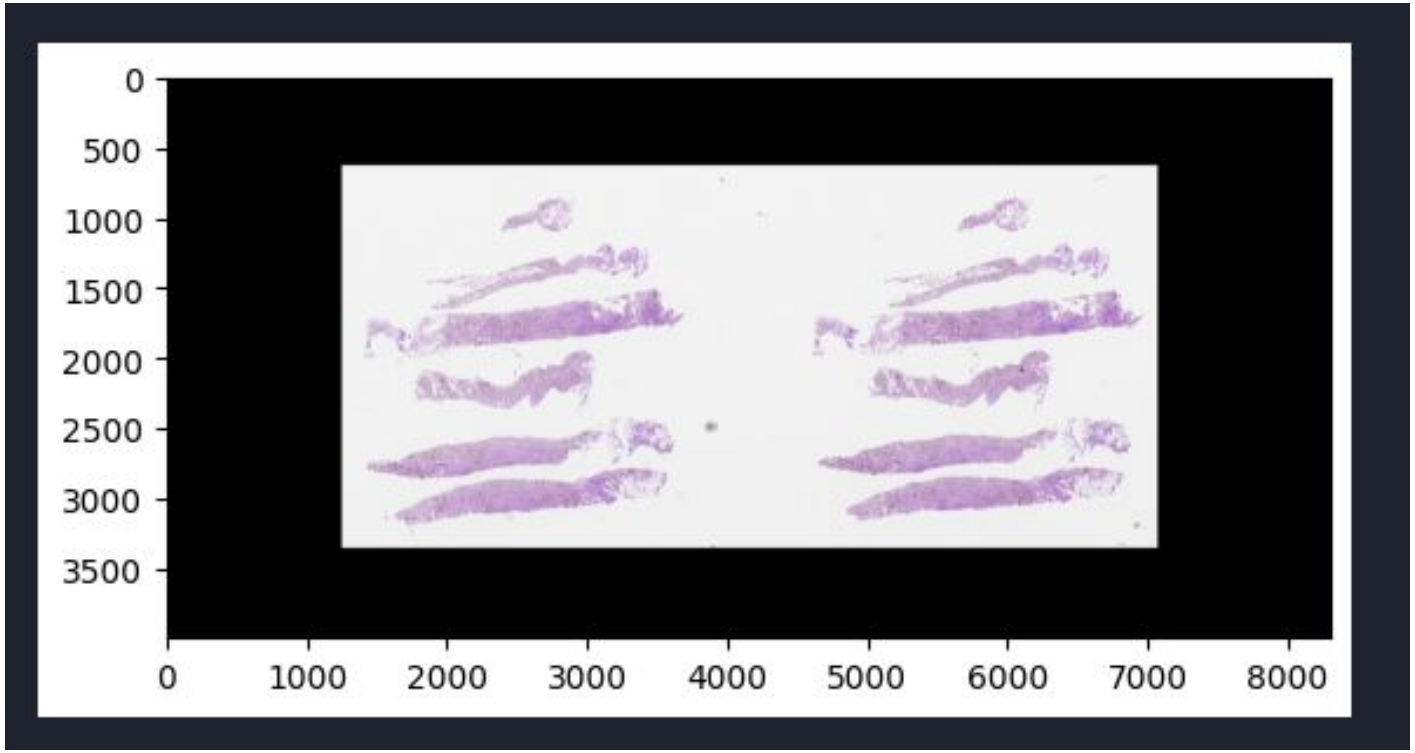


· width, height 최댓값에 맞춰 패딩 생성

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

이미지 데이터 전처리



· 원본이미지 패딩 전처리 예시

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

이미지 데이터 전처리

```
%%time
def img_trans(dataframe):

    for i in range(len(dataframe)):

        pannel = np.zeros((8299, 3965, 3), np.uint8)
        p_height, p_width, p_channel = pannel.shape

        w_start = math.ceil( (p_width - dataframe['width'].iloc[i] )/ 2)
        w_end = math.ceil( (p_width - dataframe['width'].iloc[i] )/ 2)  + dataframe['width'].iloc[i]
        h_start = math.ceil( (p_height - dataframe['height'].iloc[i] )/ 2)
        h_end = math.ceil( (p_height - dataframe['height'].iloc[i] )/ 2) + dataframe['height'].iloc[i]


        #overlap 할 이미지 불러온다
        image = cv2.imread(dataframe.loc[i].img_path, cv2.IMREAD_ANYCOLOR)

        #이미지 오버랩 시켜준다
        pannel[h_start:h_end, w_start:w_end] = image

        #최종 pannel 사본 저장한다.
        if 'test_imgs' in dataframe.loc[i].img_path:
            tmp_path = dataframe.loc[i].img_path.replace('test_imgs', 'p_test_imgs')
        elif 'train_imgs' in dataframe.loc[i].img_path:
            tmp_path = dataframe.loc[i].img_path.replace('train_imgs', 'p_train_imgs')

        cv2.imwrite(tmp_path, pannel)
```

· 이미지 데이터 전처리 재사용을 위한 함수화 진행

· 학습 이미지 수 1000개, 테스트 이미지 수 250개로 수가 많기 때문에 학습에는 사전에 전처리작업을 실시하고 저장한 이미지를 사용하여 진행 하였음.

04. 프로젝트 수행 결과

결과 제시 ② 탐색적 분석 및 전처리

커스텀 데이터셋 - 이미지

```
def __getitem__(self, index):
    img_path = self.medical_df['img_path'].iloc[index]
    image = cv2.imread(img_path)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

· 이미지를 불러올 때, BGR 채널을 RGB로 컨버트 시킨다.

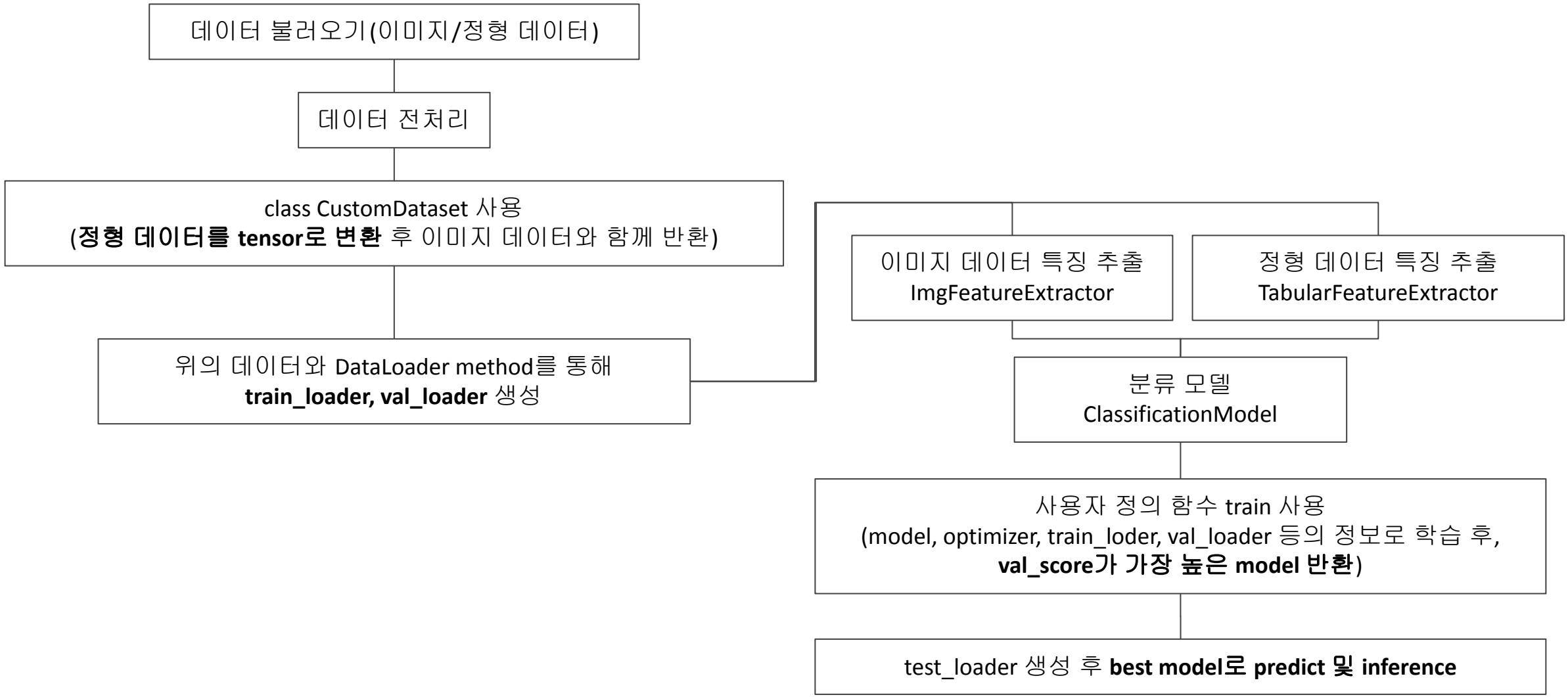
```
train_transforms = A.Compose([
                        A.HorizontalFlip(),
                        A.VerticalFlip(),
                        A.Rotate(limit=90, border_mode=cv2.BORDER_CONSTANT,p=0.3),
                        A.Resize(CFG['IMG_SIZE'],CFG['IMG_SIZE']),
                        A.Normalize(mean=(0.485, 0.456, 0.406), std=(0.229, 0.224, 0.225), max_pixel_value=255.0, always_apply=False,
                        ToTensorV2()
                        ])

test_transforms = A.Compose([
                        A.Resize(CFG['IMG_SIZE'],CFG['IMG_SIZE']),
                        A.Normalize(mean=(0.485, 0.456, 0.406), std=(0.229, 0.224, 0.225), max_pixel_value=255.0, always_apply=False,
                        ToTensorV2()
```

· 커스텀 데이터셋 클래스 내부 파라미터인 transform에서 이미지 증강 실시

- 수직/수평 전환, 90도 회전, 이미지 512 리사이즈
- efficientnet 사전학습에 사용된 이미지넷의 transform normalize 적용

04. 프로젝트 수행 결과

결과 제시 ③ 모델 개요



04. 프로젝트 수행 결과

결과 제시 ④ 모델 개선

	[Baseline]_Multi-Modal	*shufflenetV2_X2_추가학습(x)*	*[Baseline]_EDA_csv*	*04_ELU_IMG_*	*[Baseline]_EDA_RGB_10ep_Dense*
Submission Score	0.74149	0.73832	0.73862	0.73447	0.71864
Epoch	5	5	5	20	10
Learning rate	1e-4	1e-4	1e-4	1e-4	1e-4
Batch Size	16	16	16	10	16
Tabular preprocessing	O	O	O	O	O
Image preprocessing	X(원본 사용)	X	X	O (resize 1024*1024)	X
Transfer Learning Model	efficientnet_b0	shufflenetV2_X2	efficientnet_b0	efficientnet_b0	efficientnet_b0
Tabular feature extraction	linear layer 1024	linear layer 1024	linear layer 512	linear layer 2048	first linear layer 64
Image feature extraction	nn.linear(1000, 512)	nn.linear(1000, 512)	nn.linear(1000, 512)	nn.linear(1000, 512)	nn.linear(1000, 512)
Classification	in_features = 1536	in_features = 1536	in_features = 1024	in_features = 2560	in_features = 1024

04. 프로젝트 수행 결과

결과 제시 ④ 모델 구조

이미지 데이터 특징 추출 모델

Baseline

```python
class ImgFeatureExtractor(nn.Module):
    def __init__(self):
        super(ImgFeatureExtractor, self).__init__()
        self.backbone = models.efficientnet_b0(pretrained=True)
        self.embedding = nn.Linear(1000,512)

    def forward(self, x):
        x = self.backbone(x)
        x = self.embedding(x)
        return x
```

· backbone으로 사전 학습된 이미지 분류 모델과 가중치를 사용 (e.g. EfficientNet)

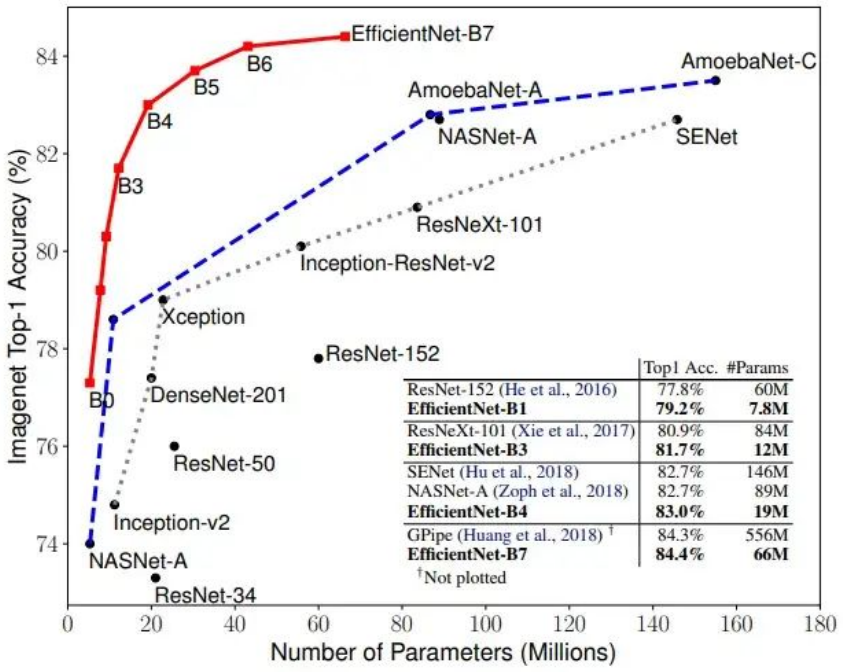
· 사전학습 모델에서 out_feature 수가 1000으로 학습되어 차원 수를 좀 더 줄이기 위해 embedding층을 추가하여 downsampling 진행 (1000 > 512)

· embedding 층은 분산된 데이터의 차원을 축소 시키기 위하여 사용

04. 프로젝트 수행 결과

결과 제시 ④ 모델 구조

이미지 데이터 특징 추출 모델 - 전이 학습



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.2%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.7%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**83.0%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

Weight	Acc@1	Acc@5	Params
AlexNet_Weights.IMAGENET1K_V1	56.522	79.066	61.1M
ConvNeXt_Base_Weights.IMAGENET1K_V1	84.062	96.87	88.6M
ConvNeXt_Large_Weights.IMAGENET1K_V1	84.414	96.976	197.8M
ConvNeXt_Small_Weights.IMAGENET1K_V1	83.616	96.65	50.2M
ConvNeXt_Tiny_Weights.IMAGENET1K_V1	82.52	96.146	28.6M
DenseNet121_Weights.IMAGENET1K_V1	74.434	91.972	8.0M
DenseNet161_Weights.IMAGENET1K_V1	77.138	93.56	28.7M
DenseNet169_Weights.IMAGENET1K_V1	75.6	92.806	14.1M
DenseNet201_Weights.IMAGENET1K_V1	76.896	93.37	20.0M
EfficientNet_B0_Weights.IMAGENET1K_V1	77.692	93.532	5.3M
EfficientNet_B1_Weights.IMAGENET1K_V1	78.642	94.186	7.8M
EfficientNet_B1_Weights.IMAGENET1K_V2	79.838	94.934	7.8M
EfficientNet_B2_Weights.IMAGENET1K_V1	80.608	95.31	9.1M
EfficientNet_B3_Weights.IMAGENET1K_V1	82.008	96.054	12.2M
EfficientNet_B4_Weights.IMAGENET1K_V1	83.384	96.594	19.3M
RegNet_X_1_6GF_Weights.IMAGENET1K_V1	77.04	93.44	9.2M
RegNet_X_1_6GF_Weights.IMAGENET1K_V2	79.668	94.922	9.2M
ShuffleNet_V2_X1_5_Weights.IMAGENET1K_V1	72.996	91.086	3.5M
ShuffleNet_V2_X2_0_Weights.IMAGENET1K_V1	76.23	93.006	7.4M

· ImageNet-1K 데이터셋 기준, baseline 모델에 사용되었던 EfficientNet이 비교적 모델 사이즈가 작음에도 불구하고 높은 정확도를 가짐

· 추가 학습 시간을 고려하여 ShuffleNet, RegNetX16gf 등 학습 파라미터 수가 10M 이하인 모델들을 시도

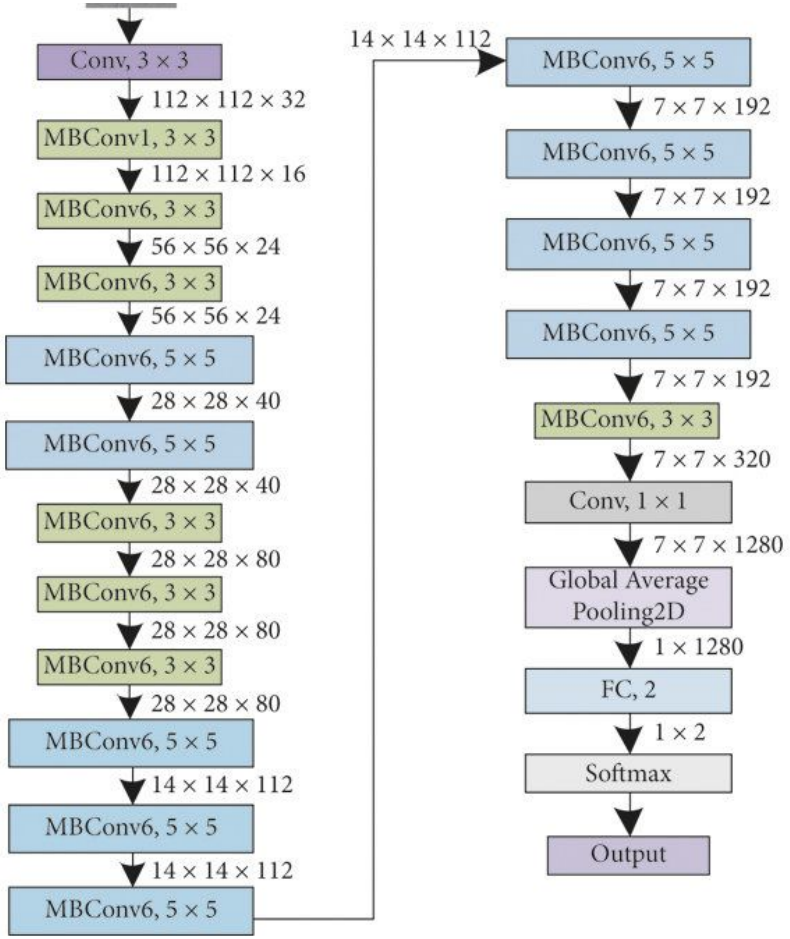
· 해당 프로젝트의 1000개의 암 이미지 데이터에 관하여 추가학습의 영향을 비교하기 위해 추가학습을 진행하지 않고,
더 많은 모델들(e.g. EfficientNetB7)을 시도했으나, 모델의 성능이 떨어짐

· baseline에 사용된 EfficientNetb0에 추가 학습을 시킨 모델이 제일 높은 성능을 보였음

04. 프로젝트 수행 결과

결과 제시 ④ 모델 구조

이미지 데이터 특징 추출 모델 - 전이 학습



·**MBConv**라는 다수의 층을 묶어서 모듈화하여 블럭(MB)으로 구성

· **Compound scaling**을 통해 모델의 **depth**와 **width**를 늘려 특징을 더 잘 잡아내는 개념을 사용

· 활성화 함수로 sigmoid에 입력값(*x*)를 다시 곱해주는 **Swish 함수** 사용

· MB의 채널이 적은 얇은층 사이에 채널 수가 많은 층을 배치하고 얇은층들을 **skip connection**으로 연결한 **Inverted Residual Block** 개념 사용

· **Squeeze and excitation(SE)** 층에서 **각 channel에게 가중치를 부여**하여 **사용될 channel 선택**

04. 프로젝트 수행 결과

결과 제시 ④ 모델 구조

정형 데이터 특징 추출 모델

Baseline

```
class TabularFeatureExtractor(nn.Module):
    def __init__(self):
        super(TabularFeatureExtractor, self).__init__()
        self.embedding = nn.Sequential(
            nn.Linear(in_features=20, out_features=128),
            nn.BatchNorm1d(128),
            nn.LeakyReLU(),
            nn.Linear(in_features=128, out_features=256),
            nn.BatchNorm1d(256),
            nn.LeakyReLU(),
            nn.Linear(in_features=256, out_features=512),
            nn.BatchNorm1d(512),
            nn.LeakyReLU(),
            nn.Linear(in_features=512, out_features=512)
        )

    def forward(self, x):
        x = self.embedding(x)
        return x
```

· CNN 이미지 특징 추출 모델의 데이터를 같이 사용하여 분류를 진행하기 위해 신경망 모델 사용

· EDA 및 전처리된 정형 데이터를 입력받아 더 자세한 특징을 추출하기 위해 upsampling 진행

- feature: **20** > 128 > 256 > **512**

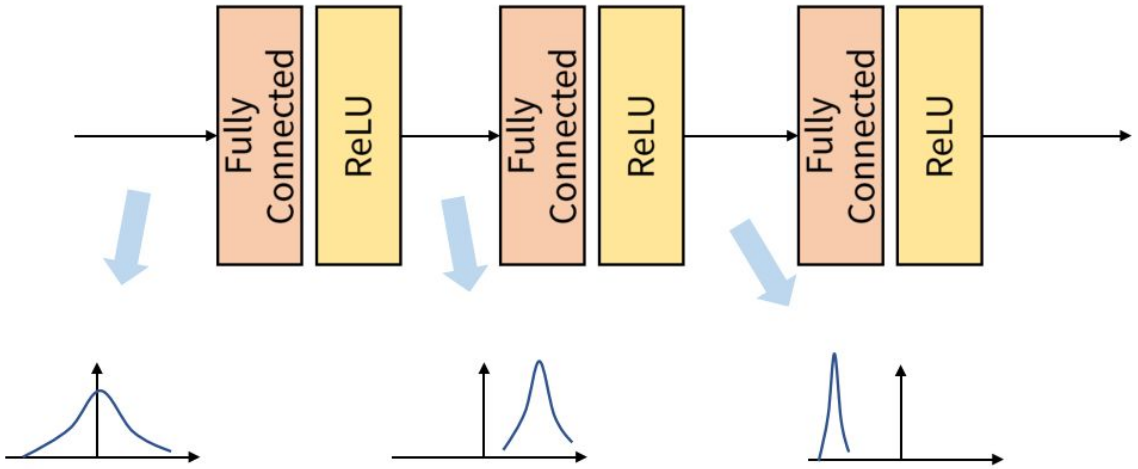
· 3가지 층으로 구성:

1. Linear/Fully-connected - 연산
2. Batch Normalization - 정규화
3. Activation(e.g.ReLU) - 활성화

04. 프로젝트 수행 결과

결과 제시 ④ 모델 구조

정형 데이터 특징 추출 모델 - Batch Normalization



· 각 계층에서 feature를 입력받아 fully-connected층의 연산을 거쳐 활성화 함수를 적용

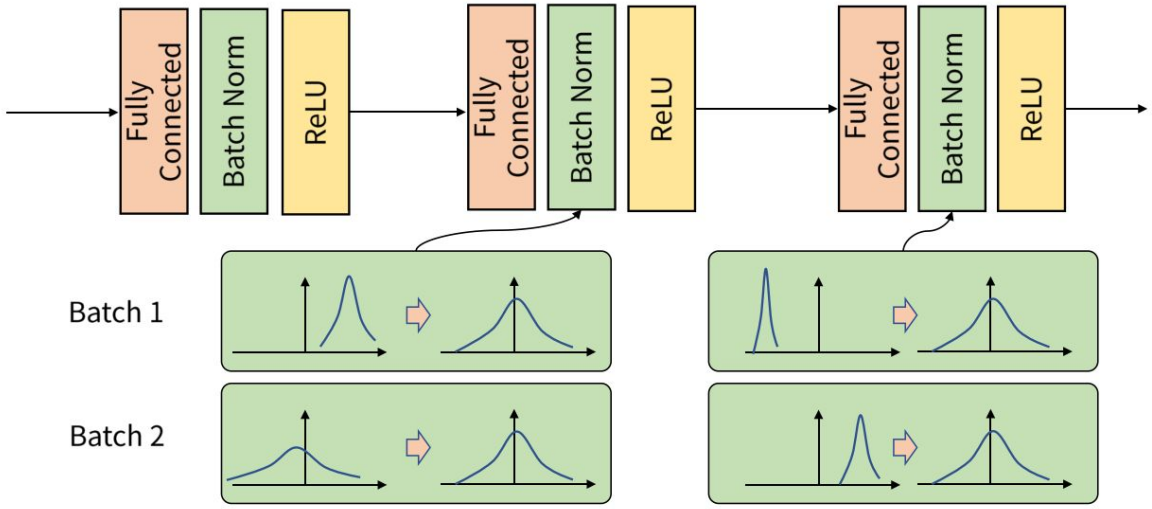
· 학습 과정에서 연산(FC)을 거친 뒤 활성화 함수 적용 결과, 계층 별로 입력 데이터/feature의 분포가 달라짐으로 인해 내부 **공변량 변화 (internal covariate shift)**가 일어남

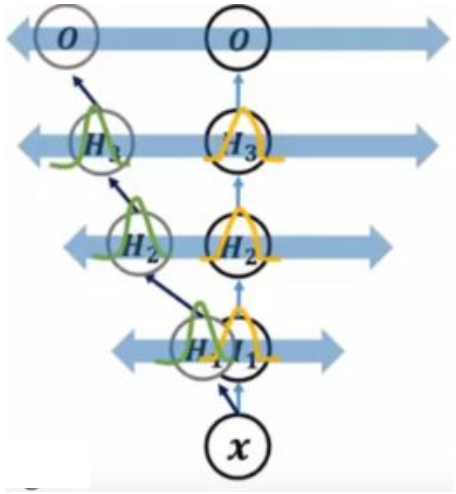
· **연산 전/후에 데이터 간 분포가 상이**하여 불특정다수의 노드가 비활성화되어 모델의 성능을 저하

04. 프로젝트 수행 결과

결과 제시 ④ 모델 구조

정형 데이터 특징 추출 모델 - Batch Normalization





· Batch 단위로 학습을 하면 추가적으로 **batch 단위별로 데이터의 분포가 상이**

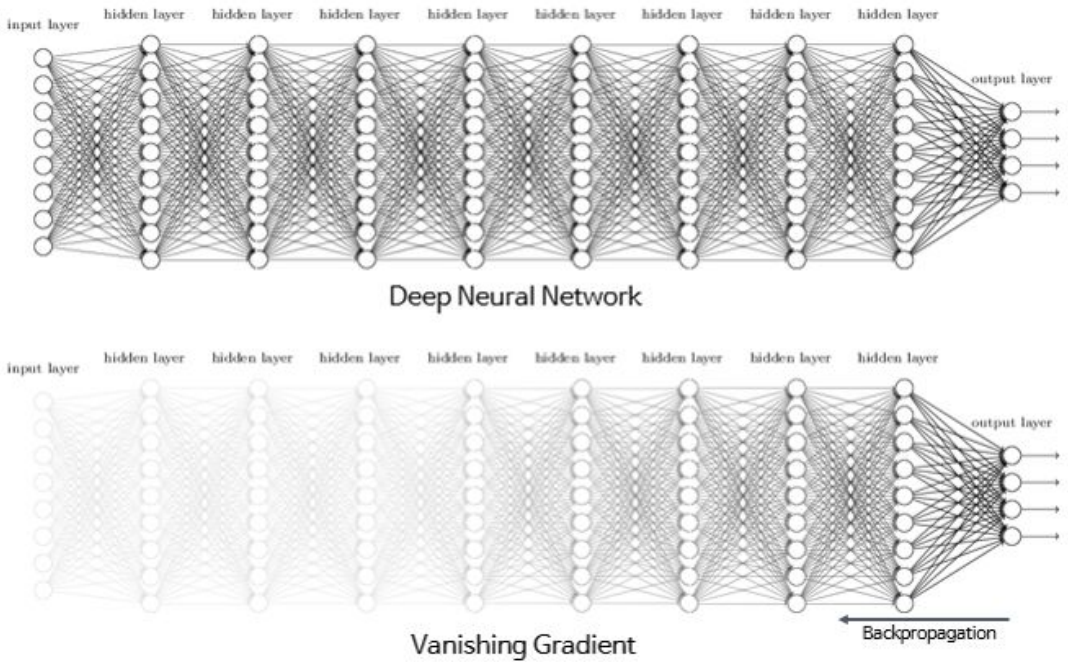
· 입력 데이터의 분포가 점점 달라지면서 층을 통과할수록 **output의 값이 실제값과 멀어짐** -> 정확도 저하

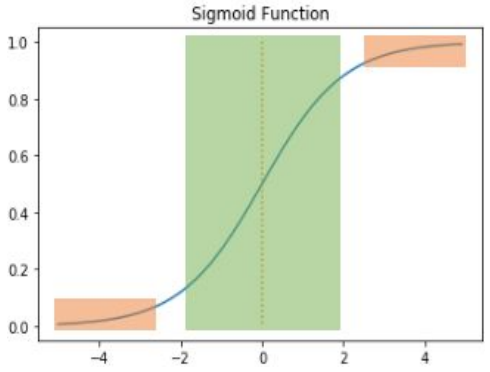
· 따라서 batch normalization층을 통해 각 **배치별로 평균과 분산을 조정하여** 가우시안/정규 분포로 **정규화**

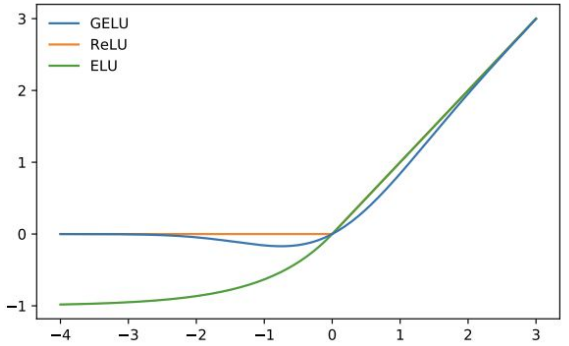
04. 프로젝트 수행 결과

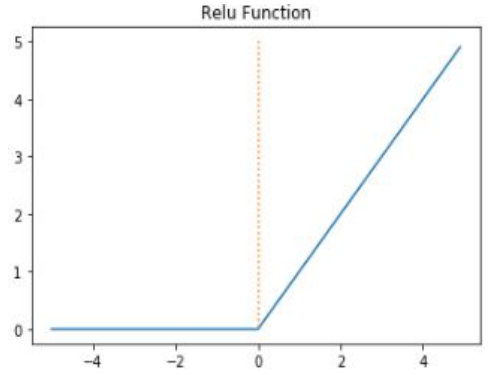
결과 제시 ④ 모델 구조

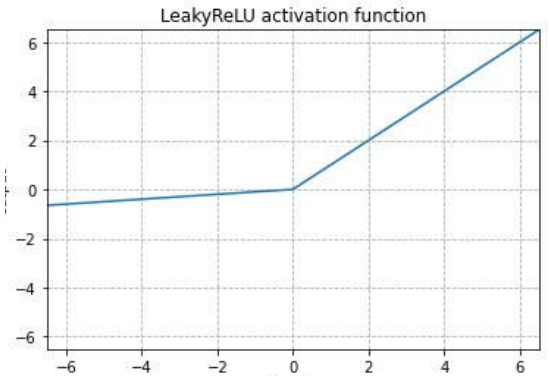
정형 데이터 특징 추출 모델 - Activation











· **Sigmoid** 또는 **Tanh**과 같은 함수는 은닉층에서 사용하면 역전파 과정에서 기울기가 0에 가까운 값이 누적해서 곱해지게 되어 앞단 노드들의 **기울기 소실** 발생

· 따라서 은닉층에 **ReLU** 함수 사용 - 하지만 음수 영역에서 기울기가 0이 되어 노드가 완전히 비활성화되는 **dying ReLU** 발생

· 이를 보완하기 위해 새로운 ReLU 계열의 활성화 함수들이 연구되었음(e.g. **ELU**, **GELU**, **SELU,** etc.)

04. 프로젝트 수행 결과

결과 제시 ④ 모델 구조

정형 데이터 특징 추출 모델 - 변경점

Baseline

```python
class TabularFeatureExtractor(nn.Module):
    def __init__(self):
        super(TabularFeatureExtractor, self).__init__()
        self.embedding = nn.Sequential(
            nn.Linear(in_features=20, out_features=128),
            nn.BatchNorm1d(128),
            nn.LeakyReLU(),
            nn.Linear(in_features=128, out_features=256),
            nn.BatchNorm1d(256),
            nn.LeakyReLU(),
            nn.Linear(in_features=256, out_features=512),
            nn.BatchNorm1d(512),
            nn.LeakyReLU(),
            nn.Linear(in_features=512, out_features=512)
        )

    def forward(self, x):
        x = self.embedding(x)
        return x
```

```python
class TabularFeatureExtractor(nn.Module):
    def __init__(self):
        super(TabularFeatureExtractor, self).__init__()
        self.embedding = nn.Sequential(
            nn.Linear(in_features=20, out_features=128),
            nn.BatchNorm1d(128),
            nn.ELU(),
            nn.Linear(in_features=128, out_features=256),
            nn.BatchNorm1d(256),
            nn.ELU(),
            nn.Linear(in_features=256, out_features=512),
            nn.BatchNorm1d(512),
            nn.ELU(),
            nn.Linear(in_features=512, out_features=1024),
            nn.BatchNorm1d(1024),
            nn.ELU(),
            nn.Linear(in_features=1024, out_features=1024)
        )

    def forward(self, x):
        x = self.embedding(x)
        return x
```

· 다양한 ReLU계열의 활성화 함수를 시도(e.g.GELU, ELU, etc.)

· 모델층의 수를 조정하며 모델에서 학습하는 특징의 수를 조절

· FC, BatchNorm, Activation층을 다 같이 추가 또는 제거

04. 프로젝트 수행 결과

결과 제시 ④ 모델 구조

활성화함수 결과(Val Score) 비교

Activation	Last Epoch	Max Achieved	Average
LeakyReLU[baseline]	77.892	77.892	74.228
LeakyReLU(reduced)	75.758	76.540	75.286
ELU	75.706	**80.390**	75.958
ELU(added)	76.768	79.884	**77.997**
GELU	77.249	77.679	74.743
GELU(reduced)	75.485	77.422	75.806
GELU(added)	78.304	78.435	76.896
SELU(added)	72.826	74.960	73.193

· val score 기준, 활성 함수 비교 결과 ELU가 max 점수와 평균 점수가 가장 높아 ELU 함수 사용

· 신경망 모델의 층을 늘렸을 때 max 점수는 큰 차이가 없었지만 평균 점수는 증가함

04. 프로젝트 수행 결과

결과 제시 ④ 모델 구조

분류 모델

```
class ClassificationModel(nn.Module):
    def __init__(self):
        super(ClassificationModel, self).__init__()
        self.img_feature_extractor = ImgFeatureExtractor()
        self.tabular_feature_extractor = TabularFeatureExtractor()
        self.classifier = nn.Sequential(
            nn.Linear(in_features=1536, out_features=1),
            nn.Sigmoid(),
        )

    def forward(self, img, tabular):
        img_feature = self.img_feature_extractor(img)
        tabular_feature = self.tabular_feature_extractor(tabular)
        feature = torch.cat([img_feature, tabular_feature], dim=-1)
        output = self.classifier(feature)
        return output
```

· 이미지 특징 추출 모델과 정형 데이터 특징 추출 모델을 통해 나온 feature들을 합친 신경망 분류 모델

· concatenation을 사용하여
이미지 특징 추출 모델(out_feature=**512**) +
정형 데이터 특징 추출 모델(out_feature=**1024**) = **1536**

· 이진분류 결과값(암 임파선 전이 여부)을 구하기 위해 **1536** → **1**개의 feature로 축소하여 sigmoid함수를 통해 분류

05. 자체 평가 의견

- **아쉬웠던 부분**

- 제한된 리소스로 인해 용량이 큰 모델들을 사용하여 추가학습을 시도해보지 못해 아쉽다.(Out Of Memory)
- 이미지 특징 추출 부분에서 WSI(Whole Slide Imaging)와 MIL(Multi Instance Learning)을 구현해 보지 못해 아쉽다.
- 조직 병리학 이미지에 대한 지식(의료 도메인)이 부족하여 이미지 프로세싱에서의 어려움과 모델링에서 다양한 커스텀을 시도하지 못해 아쉽다.
- 이전에 다뤄보았던 tensorflow가 아닌, 새로운 프레임워크 pytorch로 구성하다보니 부족한 점이 많아 구성에 어려움이 많았다.
- 다른 참여 팀들과 달리 늦게 경진대회에 참여하게 되어 시간이 조급했고, 연구결과를 더욱 세부적으로 기록하며 점진적으로 구성했어야 하지만 미흡한 부분이 있어 아쉬웠다.
- 각 Score도 팀 단위로 공유되어야 한다는 사실을 깨달았다.
- 아직까지는 원하는 것을 자유자재로 구현할 역량이 없다는 사실이 아쉬웠다.

05. 자체 평가 의견

- **좋았던 부분**

- 두가지 타입의 학습 데이터 특징 추출에 사용된 신경망 모델을 합쳐 분류를 하는 Multi-Modal 모델을 알 수 있었다.
- 이미지 프로세싱에 다양한 처리 방법, 이미지 특징 추출에 다양한 전이학습 모델을 적용해볼 수 있었다.
- 사전학습모델을 사용한다고 하더라도, 사전학습모델까지 역전파 학습 시키는 것이 시간만 더 늘어나고 성능이 잘 안나올 수도 있다는 사실을 깨달았다.
- 팀으로 작업하면서도 각자가 맡은 부분에 대한 진행, 모델링은 모두가 진행하면서 부족한 부분을 깨달아가는 과정이었다고 생각한다.
- 현재 역량을 알아볼 수 있는 좋은 기회였던 것 같다.
- 다음에도 대회에 참여한다면 기록과 정리, 결과 등 모든 부분에 있어서 나아질 것이라 확신한다.