

2024년 연구데이터 분석활용 경진대회

프로젝트명	딥러닝 기반 반려동물 피부질환 자가 진단 서비스		
팀명	둘둘즈	팀원	김항민, 박세연, 황성아

I. 모델 개발 환경

OS	Ubuntu 18.04																																										
GPU	<div>Thu Sep 12 16:14:09 2024</div> <div><table><tr><td colspan="2">NVIDIA-SMI 535.183.06</td><td colspan="2">Driver Version: 535.183.06</td><td colspan="2">CUDA Version: 12.2</td></tr><tr><td>GPU</td><td>Name</td><td>Persistence-M</td><td>Bus-Id</td><td>Disp.A</td><td>Volatile Uncorr. ECC</td></tr><tr><td>Fan</td><td>Temp</td><td>Pwr:Usage/Cap</td><td>Memory-Usage</td><td>GPU-Util</td><td>Compute M.</td></tr><tr><td colspan="6">MIG M.</td></tr><tr><td>0</td><td>Tesla V100-PCIE-32GB</td><td>Off</td><td>00000000:3B:00.0 Off</td><td></td><td>0</td></tr><tr><td>N/A</td><td>29C P0</td><td>34W / 250W</td><td>26420MiB / 32768MiB</td><td>0%</td><td>Default</td></tr><tr><td colspan="6">N/A</td></tr></table></div> <div>Tesla V100-PCIE-32GB</div> <div>Driver version: 535.183.06</div> <div>CUDA version: 12.2</div>	NVIDIA-SMI 535.183.06		Driver Version: 535.183.06		CUDA Version: 12.2		GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC	Fan	Temp	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.	MIG M.						0	Tesla V100-PCIE-32GB	Off	00000000:3B:00.0 Off		0	N/A	29C P0	34W / 250W	26420MiB / 32768MiB	0%	Default	N/A					
NVIDIA-SMI 535.183.06		Driver Version: 535.183.06		CUDA Version: 12.2																																							
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC																																						
Fan	Temp	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.																																						
MIG M.																																											
0	Tesla V100-PCIE-32GB	Off	00000000:3B:00.0 Off		0																																						
N/A	29C P0	34W / 250W	26420MiB / 32768MiB	0%	Default																																						
N/A																																											
Requirements	<div>torch==1.10.1</div> <div>torchaudio==0.10.1</div> <div>torchvision==0.11.2</div> <div>scikit-learn==1.3.2</div> <div>timm</div> <div>pandas</div> <div>requests</div>																																										

- DataON CANVAS에서 제공하는 DSuper-1pod-GPU를 사용
- 데이터와 모델 및 코드의 통일성을 위하여 모든 개발 과정은 seyeon0627 계정에서 진행
- **conda activate duldulz** 명령어를 통해 duldulz 프라이빗 가상환경 사용 (python==3.8)
- 모든 코드의 실행은 DataON CANVAS 분석환경의 **workspace** 폴더 내에서 진행
- 설치가 필요한 의존성 라이브러리는 **pip install -r MyFiles/requirements.txt** 명령어를 통해 일괄 설치

II. 파일 저장 구조

1. 디렉토리 구조

```

My Files/      # 프로젝트 루트 디렉토리
├── data/      # 데이터가 저장된 폴더
│   ├── 05동물질병.accdb          # 원본 질병 정보 데이터
│   ├── anomalydetection/        # 이상치 탐지 테스트 데이터
│   ├── csv_output/              # 전처리 후 질병 정보 데이터
│   ├── TS02/반려견/피부/일반카메라/유증상/ # 원본 반려견 학습 데이터
│   ├── TS02/반려묘/피부/일반카메라/유증상/ # 원본 반려묘 학습 데이터
│   ├── VL01/반려견/피부/일반카메라/유증상/ # 원본 반려견 평가 데이터
│   ├── VL01/반려묘/피부/일반카메라/유증상/ # 원본 반려묘 평가 데이터
│   ├── train_new/               # 전처리 후 학습 데이터
│   └── test_new/                # 전처리 후 평가 데이터
├── LLMapi/    # 질병 매칭 태스크 관련 폴더
│   ├── classification.py        # 질병 매칭(분류) 코드
│   ├── convertcsv.py            # 전처리 코드1 - 파일 유형 변환
│   └── preprocessing.py         # 전처리 코드2 - 질병 필터링
├── modeling/  # 이미지 분류(이상치 탐지, 질병 분류) 태스크 관련 폴더
│   ├── anomaly_detection_VAE.py # 이상치 탐지 학습 코드
│   ├── anomaly_detection_VAE_test.py # 이상치 탐지 평가 코드
│   ├── preprocessing.py         # 전처리 코드
│   ├── train_cat_convnextv2.py   # 반려묘 질병 분류 학습 코드
│   ├── train_dog_convnextv2.py   # 반려견 질병 분류 학습 코드
│   ├── train_cat_inceptionv4.py  # 반려묘 질병 분류 학습 코드
│   ├── train_dog_inceptionv4.py  # 반려견 질병 분류 학습 코드
│   └── train_evaluation.py        # 이상치 탐지, 질병 분류 모델 성능 평가 코드
├── models/    # 결과 모델이 저장된 폴더
│   ├── 50000_AnomalyDetection_inception_v4__model.pth # Inception 기반 이상치 탐지 모델
│   ├── classification_cat_convnextv2_base_model.pth  # ConvNeXt v2 기반 반려묘 질병 분류 모델
│   ├── classification_dog_convnextv2_base_model.pth  # ConvNeXt v2 기반 반려견 질병 분류 모델
│   ├── classification_cat_inception_v4_model.pth     # Inception 기반 반려묘 질병 분류 모델
│   ├── classification_dog_inception_v4_model.pth     # Inception 기반 반려견 질병 분류 모델
│   └── vae_model.pth                                 # VAE 기반 이상치 탐지 모델
├── results/   # 모델 평가 결과 및 성능이 저장된 폴더
│   ├── Mistral-7B_cat_classified_diseases.csv        # 반려묘 질병 매칭 결과
│   ├── Mistral-7B_dog_classified_diseases.csv        # 반려견 질병 매칭 결과
│   ├── classification_cat_convnextv2_evaluation.txt   # ConvNeXt v2 기반 반려묘 질병 분류 모델 성능
│   ├── classification_dog_convnextv2_evaluation.txt   # ConvNeXt v2 기반 반려견 질병 분류 모델 성능
│   ├── classification_cat_inceptionV4_evaluation.txt  # Inception 기반 반려묘 질병 분류 모델 성능
│   ├── classification_dog_inceptionV4_evaluation.txt  # Inception 기반 반려견 질병 분류 모델 성능
│   └── vae_anomaly_detection_results.txt              # VAE 기반 이상치 탐지 모델 결과
├── system/    # 시스템 실행 코드가 저장된 폴더
│   ├── static/uploads/          # 시스템 테스트를 위한 샘플값이 저장된 폴더
│   ├── app.py                    # window 기반 시스템 코드
│   └── app_mac.py                # mac 기반 시스템 코드
└── requirements.txt              # 패키지 목록 파일

```

Ⅲ. 모델 실행 방법

1. 모델 실행을 위한 정보

- 모든 파일은 DataON CANVAS 분석환경의 workspace/MyFiles/ 안에 존재

1-1. data

- data/ 폴더는 모델의 학습 및 평가를 위한 데이터가 들어있는 폴더로, 크게 AI-hub에서 가져온 반려묘 및 반려견의 피부질환 이미지 데이터(TS02, VL01)와 DataON에 등록된 동물질병 관련 정보 관련 데이터(05동물질병.accdb), 그리고 이상치 탐지가 제대로 되었는지를 테스트하기 위한 이미지 데이터(anomalydetection)로 나뉨
- 이때 이미지 데이터의 경우 학습에 필요한 학습 데이터와 모델의 성능 평가에 필요한 평가 데이터로 나뉘며, 원본 데이터 및 이를 다운샘플링한 전처리 후 데이터(train_new, test_new)가 모두 존재
- 동물질병 정보 데이터 또한 원본 데이터와 기존 accdb에서 csv로 파일 유형을 변경하고 필터링하는 등 전처리 후 데이터(csv_output)가 존재
- 이상치 탐지 테스트 데이터는 반려견과 반려묘가 아닌 다른 사물 혹은 동물의 이미지로 구성

1-2. LLMapi

- LLMapi/ 폴더는 질병 매칭 태스크 관련 폴더로, DataON에 등록된 세부적인 동물질병 데이터를 AI-hub에서 가져온 피부질환 종류에 매칭시키기 위한 태스크를 진행
- convertcsv.py는 원본 동물질병 정보 데이터(data/05동물질병.accdb)의 파일 유형을 csv로 변경하는 코드로, 결과값은 data/csv_output/ 안에 저장됨
- preprocessing.py는 csv파일로 변경된 데이터를 가지고 반려견과 반려묘 피부질환을 분류하고 이에 대한 결과값(dog_diseases.csv, cat_diseases.csv)을 생성. 이후 각각의 피부 관련 질병을 필터링한 결과값(dog_skin_diseases.csv, cat_skin_diseases.csv)를 생성하는 코드로, 이때 결과값은 모두 data/csv_output/ 폴더 안에 저장됨
- classification.py는 huggingface의 Mistral-7B-Instruct-v0.3 LLM 모델을 사용하여 DataON의 세부적인 동물질병 정보를 토대로 이의 동물질병명을 AI-hub의 반려견 및 반려묘의 피부질환 종류와 매칭시키는 코드이며, 이의 결과값은 results/ 폴더 안에 Mistral-7B_cat_classified_diseases.csv, Mistral-7B_dog_classified_diseases.csv로 저장됨

1-3. modeling

- modeling/ 폴더는 이상치 탐지 태스크와 질병 분류(매칭) 태스크가 포함된 이미지 분류 태스크 관련 폴더
- preprocessing.py는 데이터 불균형 문제를 해결하기 위하여 data/ 폴더 안의 반려묘 및 반려견의 피부질환 이미지 데이터(TS02, VL01)의 카테고리별 개수를 언더샘플링 기법을 통해 그 개수를 맞추어 저장하는 코드이며, 결과값은 data/ 폴더의 train_new, test_new 폴더에 저장
- anomaly_detection_VAE.py는 전처리 후 학습 이미지 데이터(train_new)를 기반으로 inceptionv4와 VAE 모델을 사용하여 반려견 혹은 반려묘가 아닌 사진을 이상치로 분류하는 모델을 생성하는 코드이며, 결과 모델은 models/ 폴더에 vae_model.pth로 저장
- anomaly_detection_VAE_test.py는 학습한 모델을 이용하여 새로운 이미지를 인풋값으로 주었을 때, 모델이 제대로 작동하는지 테스트하는 코드이며, 프로젝트에서는 샘플 데이터로 data/anomalydetection/ 폴더 안의 사람, 자동차, 호랑이 등의 이미지를 이용하여 테스트를 진행

- train_cat_convnextv2.py와 train_dog_convnextv2.py는 ConvNeXt V2 모델을 기반으로 반려묘와 반려견의 피부질환 이미지를 보고 어떤 피부질환인지를 판별하는 질병 분류 모델을 학습하는 코드이며, 학습 완료된 모델은 models/ 폴더의 classification_cat_convnextv2_base_model.pth, classification_dog_convnextv2_base_model.pth에 저장
- train_cat_inceptionv4.py와 train_dog_inceptionv4.py는 Inception v4를 기반으로 반려묘와 반려견의 피부질환 질병 분류 모델 학습 코드이며, 학습이 완료된 모델은 models/ 폴더의 classification_cat_inception_v4_model.pth와 classification_dog_inception_v4_model.pth에 저장
- train_evaluation.py는 학습 후 저장된 이상치 탐지 및 질병 분류 모델의 성능을 평가하는 코드이며, 이때 성능은 정확도, 재현율, F1-스코어, 민감도로 계산하고 관련 성능 결과값은 results/ 폴더에 텍스트 파일로 저장됨

1-4. models

- models/ 폴더는 기본 모델이상치 탐지 및 질병 분류 태스크 학습 후 결과 모델이 저장된 폴더
- 50000_AnomalyDetection_inception_v4_model.pth는 modeling/ 폴더의 modeling_anomaly_detection_VAE.py에서 학습된 반려묘와 반려견을 분류 모델로 이상탐지를 위해 이미지로부터 특징을 추출하는 모델임
- vae_model.pth는 modeling/ 폴더의 anomaly_detection_VAE.py 결과값으로, VAE 기반 이상치 탐지 태스크를 위해 학습시킨 모델이며, 이는 anomaly_detection_VAE_test.py에서 테스트하는 모델에 해당됨
- classification_cat_convnextv2_base_model.pth와 classification_dog_convnextv2_base_model.pth는 modeling/ 폴더의 train_cat_convnextv2.py와 train_dog_convnextv2.py의 결과값으로, ConvNeXt V2 모델을 기반으로 학습시킨 질병 분류 모델임
- classification_cat_inception_v4_model.pth와 classification_dog_inception_v4_model.pth는 modeling/ 폴더의 train_cat_inceptionv4.py와 train_dog_inceptionv4.py의 결과값으로, Inception을 기반으로 학습시킨 질병 분류 모델임

1-5. results

- results/ 폴더는 모델의 성능과 태스크를 진행한 후 얻게 된 결과값이 저장된 폴더
- Mistral-7B_cat_classified_diseases.csv와 Mistral-7B_dog_classified_diseases.csv는 LLMapi/classification.py의 결과값으로 DataON의 세부 동물질환을 AI-hub의 피부질환 카테고리 와 매칭시킨 결과를 나타냄
- vae_anomaly_detection_results.txt 는 anomaly_detection_VAE_test.py에서 생성된 결과값으로, 반려묘와 반려견이 아닌 사람, 호랑이, 자동차 사진의 이상탐지 결과가 정리된 파일임
- classification_cat_convnextv2_evaluation.txt, classification_dog_convnextv2_evaluation.txt, classification_cat_inceptionV4_evaluation.txt, classification_dog_inceptionV4_evaluation.txt 는 train_evaluation.py에서 생성된 결과값으로, 반려묘와 반려견의 피부 이미지를 토대로 피부질환의 종류를 판별해주는 ConvNeXt V2와 Inception V4 기반으로 학습된 모델의 성능이 정리된 파일임

1-6. system

- system/ 폴더는 생성한 모델 등을 이용하여 본 프로젝트의 시스템을 구현한 폴더

- app.py를 실행함으로써 프로젝트의 플로우에 맞춘 시스템이 동작됨
- 시스템은 다음과 같은 플로우로 동작: 반려견(강아지)인지 혹은 반려묘(고양이)인지를 설정 > 분석할 이미지 파일(반려견 혹은 반려묘의 피부 이미지)의 경로를 입력 > 이미지가 반려견 혹은 반려묘에 해당되는지 여부를 이상치 탐지로 판별 > 이미지가 반려견 혹은 반려묘의 어떤 피부질환에 해당되는지를 판별 > 해당 피부질환과 관련된 세부 질병 정보 제공
- 이때 app_mac.py는 window가 아닌 mac을 위한 시스템 실행 코드임
- static/uploads/ 안의 파일은 시스템 테스트를 위한 샘플값에 해당

1-7. requirements.txt

- requirements.txt 파일은 본 프로젝트의 시스템 및 코드를 실행하는 데에 필요한 의존성 라이브러리 목록에 해당하며, 이를 pip install 명령어를 통해 일괄 설치 가능

2. 모델 실행 방법

2-1. 기본 설정

- 모든 코드의 실행은 DataON 분석환경의 터미널을 통해 진행하며, 모든 데이터, 모델, 코드가 존재하는 seyeon0627 계정에서 진행
- 터미널을 실행하고 디렉토리가 workspace로 되어있는지 확인
- **conda activate duldulz** 명령어를 실행해 duldulz 가상환경으로 진입
- **pip install -r MyFiles/requirements.txt** 명령어를 통해 필요한 라이브러리 설치

2-2. 시스템 실행 방법

- 시스템 실행을 통하여 프로젝트의 플로우에 맞춘 시스템을 동작할 수 있음
- **python "MyFiles/system/app.py"**를 통해 시스템을 실행
- 시스템 실행 결과 예시는 다음과 같음 - 시연 영상으로 자세한 실행 결과 제공함

```
반려동물 피부질환 자가 진단 시스템
1. 강아지
2. 고양이
강아지 (1) 또는 고양이 (2)를 선택하세요 (프로그램 종료를 원하면 'exit'를 입력하세요): 1
이미지 파일 경로를 입력하세요: MyFiles/system/static/uploads/dog_test1.jpeg
이상 탐지를 수행하고 있습니다...
정상 이미지입니다.
강아지 피부질환 분류 결과: 클래스 2 (농포_여드름 (Pustules_Acne))
세부 질병 보기를 원하시나요? (y/n): y

[세부 질병 목록]
- 개의 비늘증 (Canine ichthyosis)
- 개의 호산성 모낭염과 절창증 (Canine eosinophilic folliculitis and furunculosis)
- 혈관육종, 피부 (Hemangiosarcoma, skin)
- Cuterebra, 쇠파리류충피부종의 감염 (Cuterebra, dermatobia, infection in dogs and cats)
- Dirofilaria repens 육아종 (Dirofilaria repens granuloma in dogs)
- Klippel-Trenaunay 증후군 (Klippel-trenaunay syndrome in a border collie)
- Melaleuca 오일, 티트리 오일의 개와 고양이에서의 독성 (Melaleuca oil, tea tree oil, toxicity in dogs and cats)
- Orthopoxvirus에 감염된 개, 우두, 천연두 (Orthopoxvirus infection in dogs, cowpox, pox)
- Phenol 독성 (Phenol toxicity in dogs and cats)
- Shar pei 피부 증후군 (Shar pei dermatologic syndrome in dogs)
- coccidia와 관련된 개에서 피부 결절증 (Skin nodules in dogs associated with coccidia)
- german shepherds의 가족 피부 혈관병증 (Familial cutaneous vasculopathy of german shepherds)
- microfilaria와 관련된 피부염 (Dermatitis associated with microfilaria in dogs)
- springer spaniels에서의 태선모양 (Psoriasisiform) 피부염 (Psoriasisiform lichenoid dermatitis in springer spaniels)
- sulfonamides에 대한 특이 반응 (Idiosyncratic reactions to sulfonamides in dogs)
- 가성원형탈모증과 유사한 협부벽모낭염 (Isthmus mural folliculitis resembling pseudopelade of dogs)
- 가슴샘, 성장 호르몬 이상 (Thymic and growth hormone abnormalitis in weimaraners)
- 간선충피부염 (Pelodera, rhabditic dermatitis)
- 감각과민, 자기 핥기, 씹기와 핥기 (Hypoesthesia, self-licking, self-chewing and scratching in dogs and cats)
```

- 2-1과 2-2를 통하여 본 프로젝트의 모든 워크플로우 및 작동 과정을 이해하고 경험해볼 수 있으며, 2-3부터는 시스템 동작 과정이 아닌 그 이전의 데이터 전처리 및 모델 생성 과정에 해당되는 코드로 기능별로 파일을 실행할 때에 필요한 실행 방법이기때문에 2-3의 실행은 선택 사항임

2-3. 기능별 파일 실행 방법

- 2-1과 2-2에서 시스템의 작동 과정을 경험했다면, 2-3은 데이터 전처리, 모델 학습 및 평가에 관련된 실행 방법에 해당

2-3-1. 이미지 데이터 전처리

- 이미지 데이터 전처리를 위해 `python "MyFiles/modeling/preprocessing.py"` 명령어를 실행

2-3-2. 이미지 이상치 탐지 모델 학습

- 이미지가 반려견(강아지) 혹은 반려묘(고양이)에서 벗어난 다른 이미지인지 여부를 판단하는 이상치 탐지 모델을 학습은 `python "MyFiles/modeling/anomaly_detection_VAE.py"` 명령어로 실행

2-3-3. 이미지 이상치 탐지 모델 테스트

- 학습된 모델을 테스트 하기 위해서는 MyFiles/modeling/anomaly_detection_VAE_test.py의 line 99에 `new_image_paths = glob.glob('[테스트할 이미지 경로]')`에서 테스트할 이미지 경로를 입력
- 이후 터미널에서 `python "MyFiles/modeling/anomaly_detection_VAE_test.py"` 명령어로 실행

2-3-4. 이미지 질병 카테고리 분류 모델 학습

- 이미지가 반려견(강아지) 혹은 반려묘(고양이)의 어떤 질병에 해당하는지를 판단하는 질병 카테고리 분류 모델을 학습하기 위해 다음의 명령어 중 선택하여 실행
- 반려묘에 해당하는 이미지들을 ConvNeXt V2 기반의 모델로 학습시키기 위해서는 `python "MyFiles/modeling/train_cat_convnextv2.py"` 명령어로 실행
- 반려견에 해당하는 이미지들을 ConvNeXt V2 기반의 모델로 학습시키기 위해서는 `python "MyFiles/modeling/train_dog_convnextv2.py"` 명령어로 실행
- 반려묘에 해당하는 이미지들을 Inception 기반의 모델로 학습시키기 위해서는 `python "MyFiles/modeling/train_cat_inceptionv4.py"` 명령어로 실행
- 반려견에 해당하는 이미지들을 Inception 기반의 모델로 학습시키기 위해서는 `python "MyFiles/modeling/train_dog_inceptionv4.py"` 명령어로 실행

2-3-5. 이미지 이상치 탐지 및 질병 카테고리 분류 모델 성능 평가

- 학습된 이상치 탐지 및 질병 카테고리 분류 모델의 성능을 평가하기 위한 단계
- MyFiles/modeling/train_evaluation.py에서 `test_dir`에 테스트 데이터의 경로를 설정
- `num_classes` 변수에 분류할 클래스 수를 설정
- `model = create_model('[모델명]', pretrained=False, num_classes=num_classes)`에서 모델명 설정
- `output_file` 변수에 저장할 경로 및 파일명을 제시
- 이후 터미널에서 `python "MyFiles/modeling/train_evaluation.py"` 명령어로 실행

2-3-6. 질병 카테고리 매칭을 위한 데이터 전처리

- DataON의 세부 질병과 AI-hub의 질병 카테고리 매칭 태스크를 위한 파일 유형 변환 관련 전처리 코드 실행을 위해 `python "MyFiles/LLMapi/convertcsv.py"` 명령어를 통해 진행
- 다음으로 `python "MyFiles/LLMapi/preprocessing.py"` 명령어를 통해 질병 필터링 관련 전처리 코드 실행

2-3-7. 질병 카테고리 매칭

- DataON의 세부 질병과 AI-hub의 질병 카테고리를 매칭을 위해 명령어로 실행

IV. 연락처

1. 팀원 연락처 정보

김항민	khm0106@yonsei.ac.kr
	010-5754-3799
박세연	seyeon@yonsei.ac.kr
	010-8687-2246
황성아	sungahwang@yonsei.ac.kr
	010-2372-2497