**Pain Horse 인식하기**

**1. 프로젝트 개요**

말은 굉장히 섬세한 동물로써, 통증이 있어도 참고 잘 표현하지 않다가 심하게 진행되었을 때 증상이 나타나면 이미 늦은 경우가 많다고 한다. 말의 얼굴 표정을 보고 통증 여부를 파악할 수 있다면, 병에 걸린 말을 사전에 발견하고 관리, 치료할 수 있을 것이다. AI를 통해 이 같은 사전 진단이 가능한지 본 프로젝트를 통해 접근해 보고자 한다.

**2. 프로젝트 목적**

Pain Horse는 ‘눈(눈동자, 눈꺼풀)’, ‘귀’, ‘코’, ‘입’ 등에서 통증 증상이 나타날 수 있다고 알려져 있으나, 특징이 아주 명확하지는 않다. 본 프로젝트에서는 ‘눈’을 중점적으로 학습하여 Pain Horse를 구별하는 모델을 개발해보도록 하며, 다음과 같이 3단계로 모델을 개발하도록 한다.

- 1단계: Front-Side classification 모델 (말은 측면 얼굴이 표정을 분석하기 좋기 때문에 측면 얼굴만 골라내는 모델을 개발한다.)

- 2단계: Eye detection 모델 (눈을 통해 통증유무가 잘 나타난다고 알려져 있고, 갈기나 다른 부위는 혼선만 야기할 수 있기 때문에, 눈 영역만 찾아내는 모델을 개발한다.)

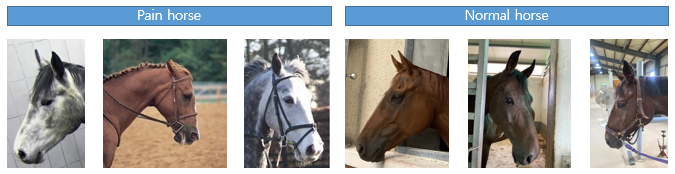
- 3단계: Pain-Normal classification모델 (인식한 ‘눈’ 이미지만 가지고 통증과 정상을 구별하는 모델을 만들어 본다.)

**3. 프로젝트 내용**

3-1. 데이터 수집

- 수의과대학원 연구실에서 제공한 데이터로 기본적인 학습 진행

- 통증 말 데이터가 부족하여 데이터에 편향이 생겨, google에서 ‘pain horse’로 검색하여 수집한 데이터를 추가로 활용함



[사진 1] Pain horse와 Normal horse의 얼굴 표정. (눈에서 아픈 표정을 읽을 수 있나요?)

- 정면 얼굴과 측면 사진은 수작업으로 분류함.

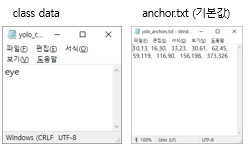
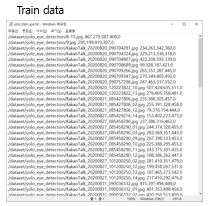


[사진 2] 정면 얼굴과 측면 얼굴 구분 (정면 얼굴보다 측면 얼굴에서 ‘눈’ 표정을 더 잘 읽을 수 있어서, 측면 얼굴만 걸러낼 수 있도록 학습데이터 구성)

- 눈 영역 인식하는 detection model개발을 위해 annotation진행



[사진 3] 별도 annotation tool을 이용하여 사진에서 눈영역 좌표 저장



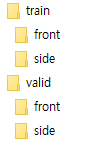
[사진 4] keras기반 yolov3용 학습데이터 포맷. class정의 데이터

3-2. Training

1) Front-Side Classification: Keras Resnet50

- Keras2.2.4기반 Resnet50 모델을 사용하였고, pre-trained model을 이용하여 학습의 효율성을 높임

- 위에서 준비한 Front-side 데이터셋을 다음과 같이 train : valid = 8 : 2로 구분하여 폴더에 저장함



- 학습모델에 데이터셋 경로를 설정하고, 모델 기본설정을 한 후 학습진행

2) Eye-detection: Keras Yolov3

**-** Detection model은 Keras2.2.4 기반 yolov3 모델을 사용함

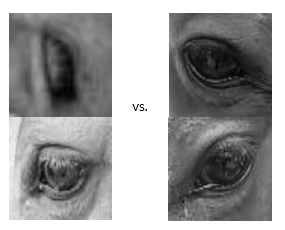
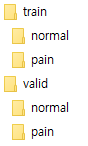
**-** Pre-trained 모델을 사용하여 학습의 효율성을 높임

**-** 기본 모델코드에 모델경로, annotation정보(학습데이터), class정보, yolo\_anchors.txt 경로를 설정하고 학습 진행

3) Pain-Normal Classification: Keras Resnet50

**-** Keras2.2.4기반 Resnet50 모델을 사용하였고, pre-trained model을 이용하여 학습의 효율성을 높임

**-** 위에서 준비한 Pain-Normal 데이터셋을 다음과 같이 train : valid = 8 : 2로 구분하여 폴더에 저장함



- 눈 데이터의 경우, Pain horse 데이터가 적어서 말 색깔 등이 학습에 영향을 줄 수도 있어서 위와 같이 gray-scale로 변환하여 학습을 진행함

3-3. 학습결과

1) Front-Side Classification Model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Predict | | | |
| Label | Front | Side | Sum | accuracy |
|  |  |  |  |  |
| Front | 81 | 2 | 83 | 97.6% |
| Side | 1 | 272 | 273 | 99.6% |
|  |  | Total | 356 | 99.2% |

- Validation 결과 모델 정확도 99.2%로 Front와 Side를 잘 구별할 수 있는 것으로 판단된다.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| (a) | (b) | (c) |

[사진 5] a. Front로 인식한 사진. (CAM(Class Activation Map)이미지가 얼굴 영역을 보고 판단한 것으로 나타남 – 결과 도출에 많은 영향을 준 영역이 Heatmap에서 밝게 나타남) b. Side로 인식한 사진. (CAM이미지가 한쪽 눈, 입 위치, 굴레(?) 등의 정보가 많이 반영되었음을 말해줌) c. Front중 Side로 인식한 사진 (한쪽 눈이 가려져있고, 얼굴 각도가 약간 돌아가 있어 Side로 봐도 무방함)

2) Eye Detection Model

- Eye Detection은 측면 사진 1,001장에서 100% 인식에 성공하였음.

|  |  |
| --- | --- |
|  |  |

[사진 6] 측면사진을 Input으로 넣으면 눈 영역만 인식, crop해서 저장.

3) Normal-Pain Eye Classification Model

**-** Validation 결과 모델 정확도는 74.6%로써, 눈을 통해 ‘정상말’ 과 ‘통증말’을 구별하는 것에 어느 정도 가능성을 보임

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Predict | | | |
| Label | Normal | Pain | sum | accuracy |
| Normal | 92 | 26 | 118 | 78.0% |
| Pain | 10 | 14 | 24 | 74.6% |
|  |  | Total | 142 | 74.6% |

- 단, 데이터 숫자가 적기 때문에 더 많은 데이터를 통해 확인해 볼 필요가 있음

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| (a) | | (b) | |

[사진 7] ‘정상말’ CAM 이미지: 눈 자체와 눈 주변의 차이를 보는 경향이 있음. b. ‘통증말’ CAM이미지: 눈 가 주름 등에서 특징을 찾는 경향이 있음 (단, 모든 사진이 이런 것은 아니어서 더 많은 데이터를 통해 검증 필요)

**4. 주요 기술**

- 언어 : Python

- 모델 : Resnet50, Yolov3

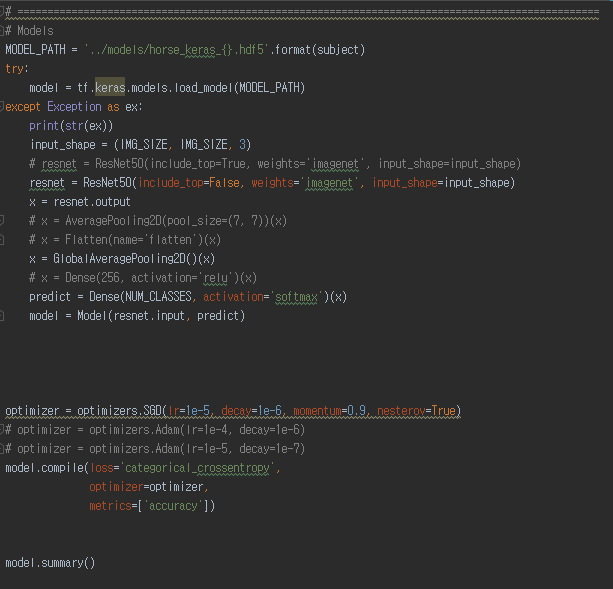
- 프레임워크 : Keras2.2.4

**5. 주요 소스코드**

- Front-Side classification 모델 환경설정 (Pain-Normal도 거의 유사)



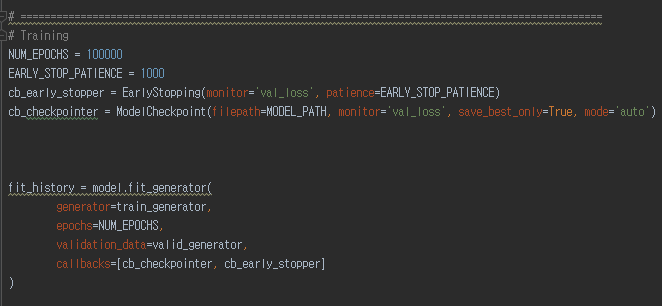
- 모델초기화 부분



- 데이터셋 설정 부분



- Training 실행 부분



- Detection 모델 데이터설정 및 실행 부분

