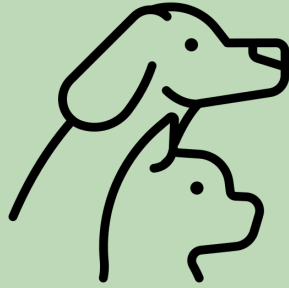


For your pets wellness,

Walness

4조 장진형 민지현 박재영 이서은 정겨운 최성림



Contents

01 Walness 기획 의도

02 시스템 아키텍처

03 데이터

04 모델링

- 1) 모델의 구성
- 2) 학습 방법
- 3) 모델 성능 평가 및 선택
- 4) 하이퍼파라미터 튜닝
- 5) 최적화 모델로 학습

05 웹페이지

06 SWOT 분석 및 향후 과제

07 기대효과

08 일정 관리 및 후기

References

Q & A





Team Walness



장진형



민지현



박재영



이서은



정겨운



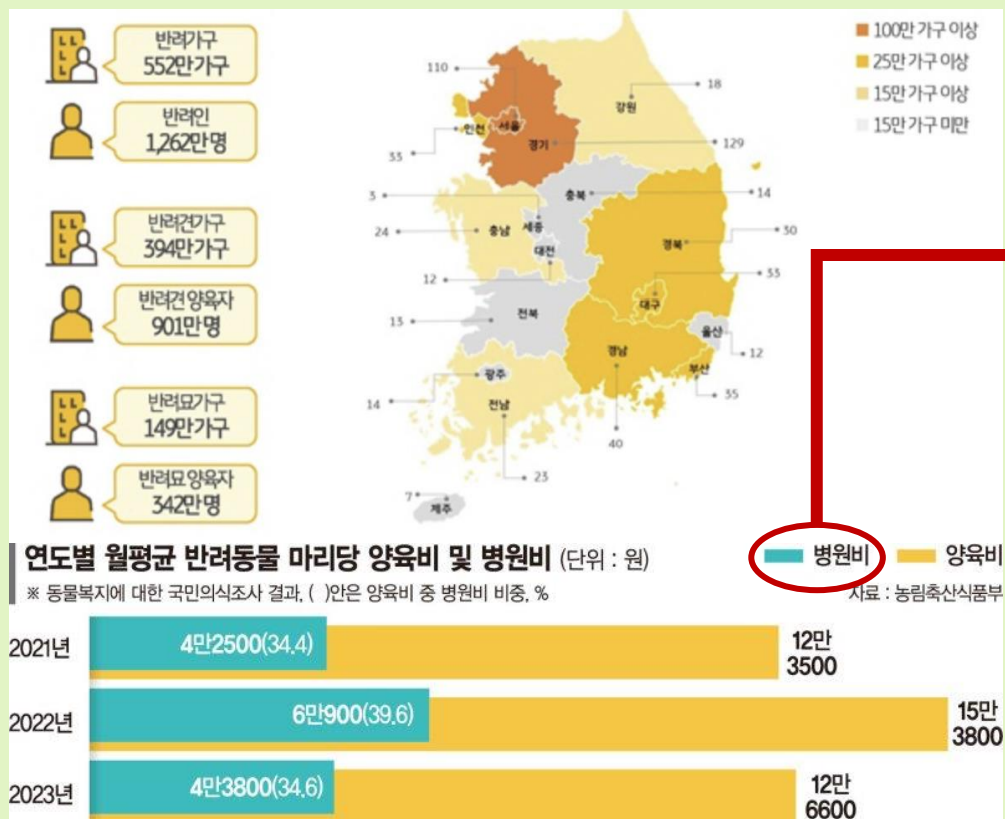
최성림

01 Walness 기획 의도



국내 반려동물 양육 가구 552만, 반려인 1262만명

2023년 한국 반려동물보고서 발간. 2020년 말 대비 양육가구 2.8% 증가

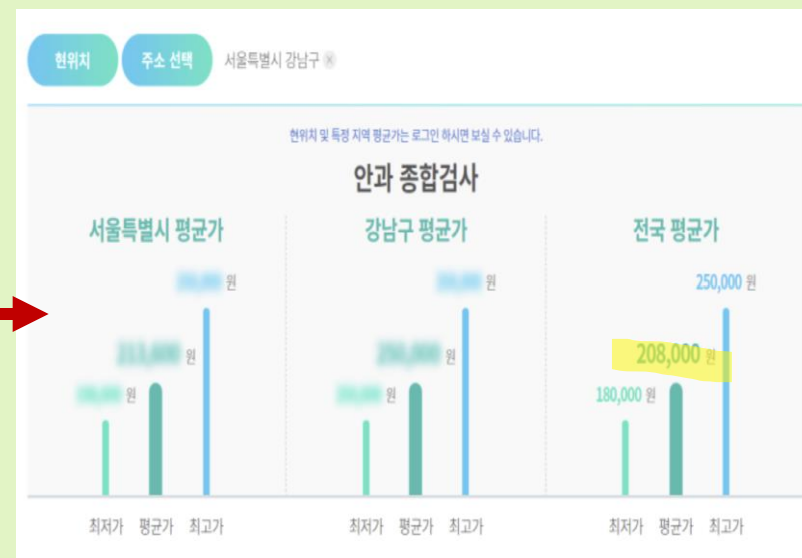


[반려동물 건강이야기] 강아지가 눈을 게슴츠레 뜨거나 못 뜬다면? 안과질환 신호!

신성우 화성 병점 블루메어동물병원 대표원장 | 정리: 김보람 기자 (rambo502@k-health.com) | 승인 2021.06.19 00:00 | 댓글 0



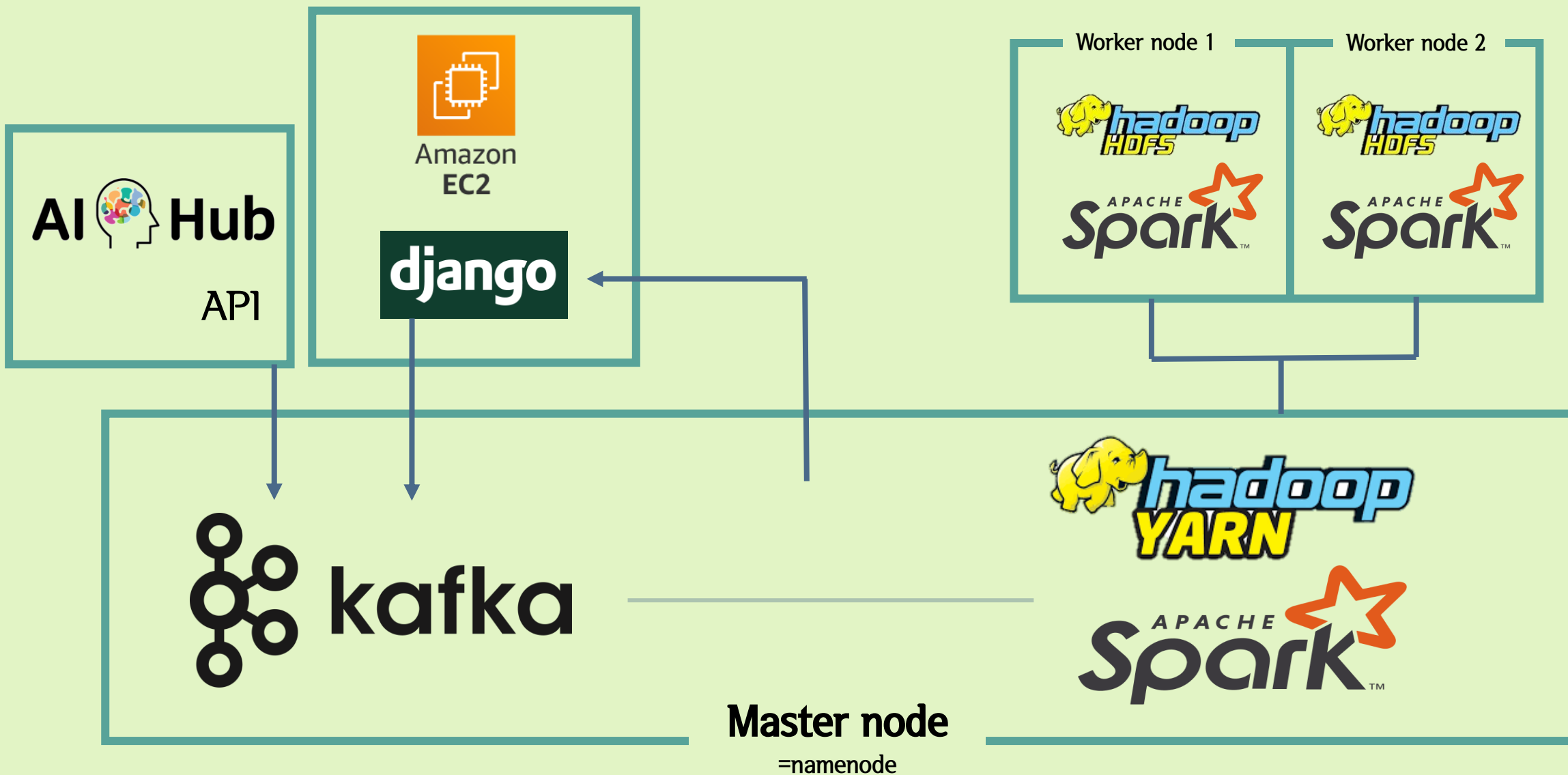
[의료상식] 노화로 인한 반려견 안구질환, 방치 시 실명할 수도



증가하는 반려동물 양육가구와 부담되는 동물병원비용
 그중에서도 안과질환은 검사 비용만으로도 부담

Walness : 증가하는 반려인들의 부담을
 덜어줄 안과질환 pre-진단 서비스

02 시스템 아키텍처



02 시스템 아키텍처



```
public class ImageProducer {
    Run | Debug
    public static void main(String[] args)
    {
        Properties props = new Properties();
        Producer<String, byte[]> producer=null;
        try {
            props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
            props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, "org.apache.kafka.common.serialization.StringSerializer");
            props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, "org.apache.kafka.common.serialization.ByteArraySerializer");

            producer= new KafkaProducer<>(props);
            String topic = "image-topic";
            File dir = new File("~/myimages");
            File[] files = dir.listFiles();

            while (true) {
                if (files != null) {
                    for (File file : files) {
                        if (file.isFile() && file.getName().endsWith(".jpg")) {
                            byte[] imageData = Files.readAllBytes(file.toPath());
                            ProducerRecord<String, byte[]> record = new ProducerRecord<>(topic, null, imageData);
                            producer.send(record);
                            System.out.println("Sent image: " + file.getName());
                        }
                    }
                }
                Thread.sleep(1500);
            }
        } catch (Exception e) {
            System.out.println(e.toString());
        } finally {
            if (producer != null) {
                producer.close();
            }
        }
    }
}
```

Kafka 프로듀서

```
def resize_image(data):
    image = Image.open(io.BytesIO(data))
    image = image.resize((224, 224))
    img_byte_arr = io.BytesIO()
    image.save(img_byte_arr, format='JPEG')
    return img_byte_arr.getvalue()

resize_udf = udf(resize_image, BinaryType())

spark = SparkSession.builder \
    .appName("Image Stream Processor") \
    .getOrCreate()

df = spark \
    .readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", "localhost:9092") \
    .option("subscribe", "image-topic") \
    .load()

images_df = df.select(resize_udf(col("value")).alias("image_data"))

query = images_df.writeStream \
    .format("binaryFile") \
    .option("path", "hdfs://localhost:8020/myexamples") \
    .option("checkpointLocation", "hdfs://localhost:8020/checkpoint") \
    .start()

query.awaitTermination()
```

Spark 스트림

03 데이터



데이터 영역	영상이미지	데이터 유형	이미지
데이터 형식	Jpg, png, jpeg	데이터 출처	동물병원
라벨링 유형	바운딩박스, 세그멘테이션 (이미지)	라벨링 형식	json
데이터 활용 서비스	안구질환 서비스	데이터 구축년 도 / 데이터 구축량	2021년/안구질환 데이 터 300,000장



AI 허브 반려동물 안구질환 데이터

사용도구

Django==5.0.1
numpy==1.26.3
pandas==2.2.0
matplotlib==3.8.2
seaborn==0.13.1
scikit-learn==1.4.0
beautifulsoup4==4.12.3
mysql-connector-python==8.3.0
tensorflow==2.15.0.post1
fastai==2.7.14
torch==2.2.1
torchvision==0.17.1
torchaudio==2.2.1
opencv-python==4.8.0.76
nbformat==5.10.4
plotly==5.20.0



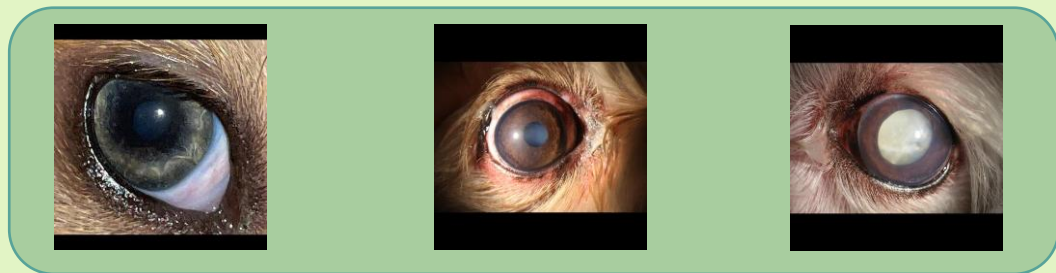
데이터 영역	영상이미지
데이터 형식	jpg
라벨링 유형	바운딩박스, 세그멘테이션(이)
데이터 활용 서비스	안구질환 서비스

질환		"안검염", "안검종양", "안검내반증", "유루증", "색소침착성각막염", "핵경화", "궤양성각막질환", "비궤양성각막질환", "백내장", "결막염", "각막궤양", "각막부골편", "유리체변성", "비궤양성각막염"
질환 정도	수의사1 라벨링	"null", "무", "유", "상", "하", "초기", "비성숙", "성숙"
	수의사2 라벨링	
	확정 라벨링	



질환 목록에서
[반려견들이
가장 흔하게 걸리는 질병]
선정

03 데이터 - 전처리



400 x 400



224 x 224

1. 이미지 Resizing

Model에서 요구하는 224*224 사이즈

[] # 데이터 증강을 위한 ImageDataGenerator 객체 생성

```
train_datagen = ImageDataGenerator(  
    rotation_range=20,      # 랜덤하게 이미지를 회전할 각도 범위  
    width_shift_range=0.1,  # 랜덤하게 이미지를 수평으로 이동할 범위  
    height_shift_range=0.1, # 랜덤하게 이미지를 수직으로 이동할 범위  
    shear_range=0.2,       # 랜덤하게 전단 변환을 적용할 각도 범위  
    zoom_range=0.2,        # 랜덤하게 이미지를 확대 또는 축소할 범위  
    horizontal_flip=True,   # 랜덤하게 이미지를 수평으로 반전  
    fill_mode='nearest'     # 이미지 변환 후 생성되는 빈 공간을 어떻게 채울지 설정  
)
```

```
val_test_datagen = ImageDataGenerator()
```

2. 데이터 증강

학습 성능을 향상시키기 위함

03 데이터 - 전처리



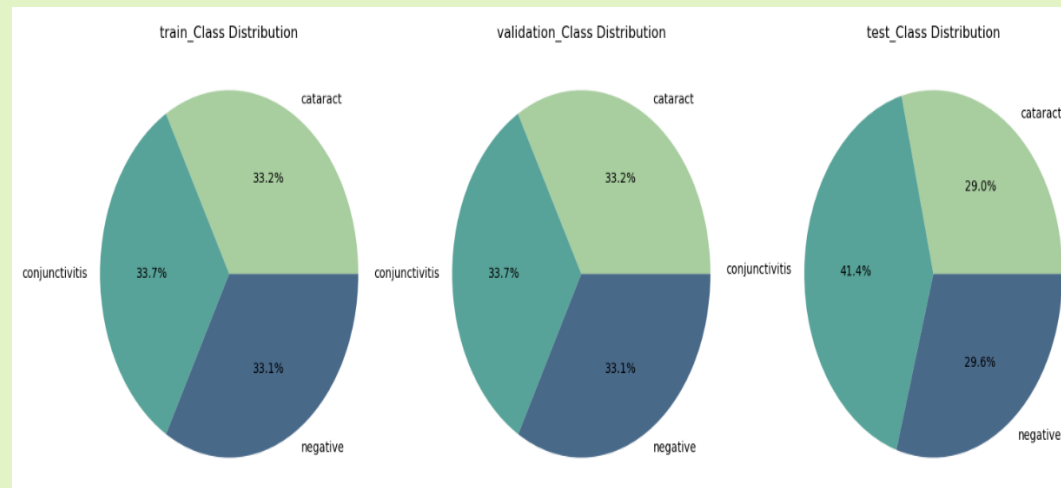
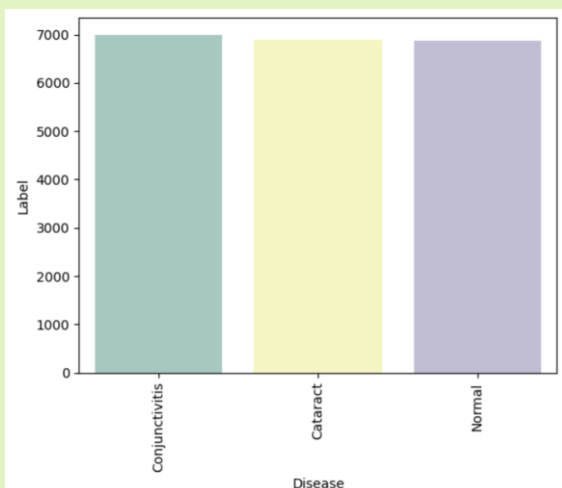
3. 클래스별 데이터 확인

```
[4] import os

def data_cnt(folder_path):
    return len(os.listdir(folder_path))

print(f"-- 백내장 파일 개수: {data_cnt('/content/drive/MyDrive/Colab Notebooks/final/data/train/Cataract')}개")
print(f"-- 결막염 파일 개수: {data_cnt('/content/drive/MyDrive/Colab Notebooks/final/data/train/Conjunctivitis')}개")
print(f"-- 정상 파일 개수 : {data_cnt('/content/drive/MyDrive/Colab Notebooks/final/data/train/Normal')}개")

-- 백내장 파일 개수: 6885개
-- 결막염 파일 개수: 7000개
-- 정상 파일 개수 : 7000개
```



데이터상 각 클래스
(백내장, 결막염, 정상)

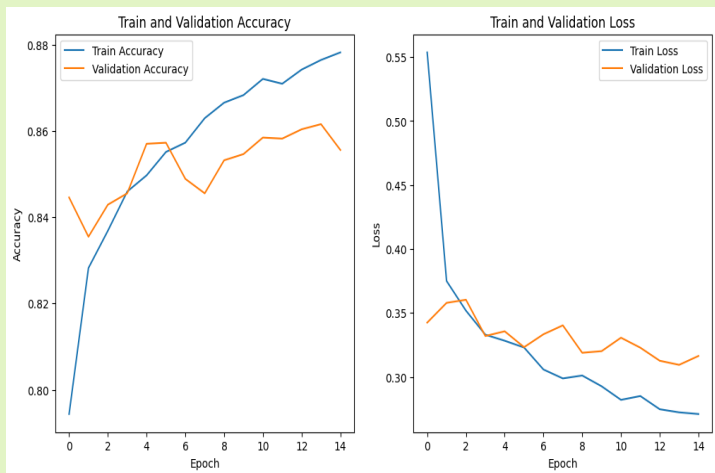
균형



1) 모델 성능 평가 및 선택

이미지 분류에 많이 이용되는 세가지의 모델을 동일데이터로 시도 후, 성능이 가장 좋은 모델 선택

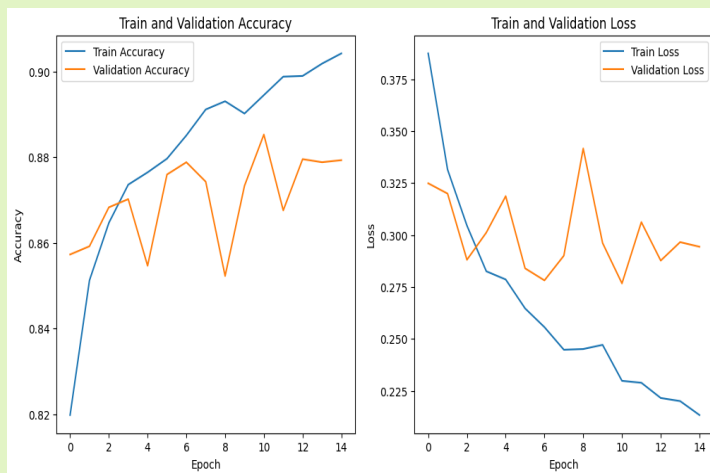
1. VGG16



- Test acc: 0.8595
Test loss: 0.30881

- 학습한 모델 중 가장 성능 좋지 않았음

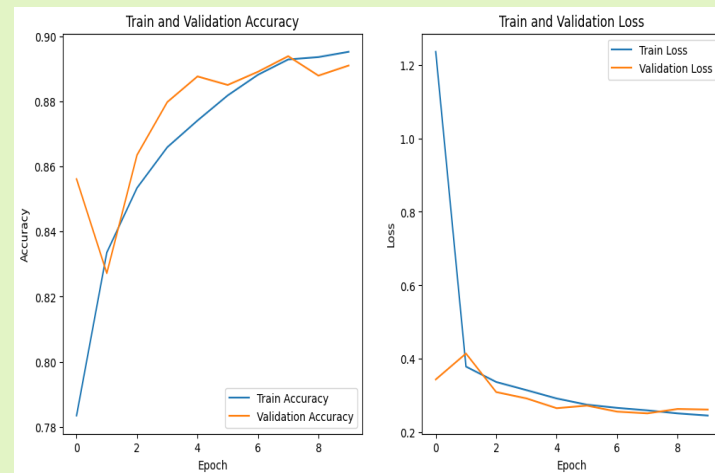
2. EfficientNet



- Test acc: 0.8724
Test loss: 0.30549

- 여러 번의 시도 중, 과적합 문제 발생 경우도 확인

3. Resnet50



- Test acc: 0.8908
Test loss: 0.25485

**Test acc이 가장 높고,
loss가 가장 낮은 것을 확인**



1) 모델 성능 평가 및 선택 - VGG16

1. VGG16

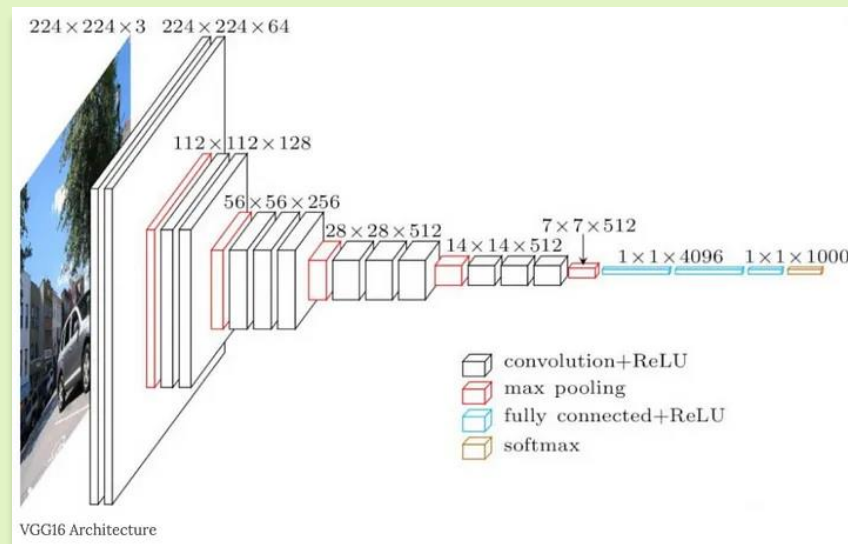
- 16개의 레이어로 구성된 모델
- Convolution Layer & Max Pooling Layer 의 반복으로 구성됨.

2. VGG16 의 한계점

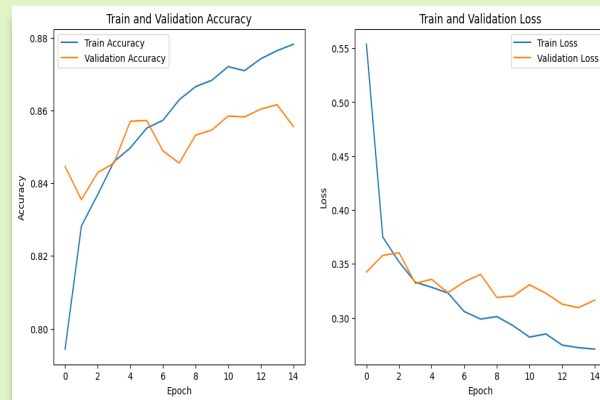
- 단순한 구성
 - Convolution Layer & Max Pooling Layer 만으로 구성됨.
- 적절한 regularization이 없음.
- 오차 역전파 도중에 gradient descent 발생.
- 위와 같은 이유로 모델이 간단하고 원시적인 만큼 다른 모델에 비해 성능이 떨어짐.

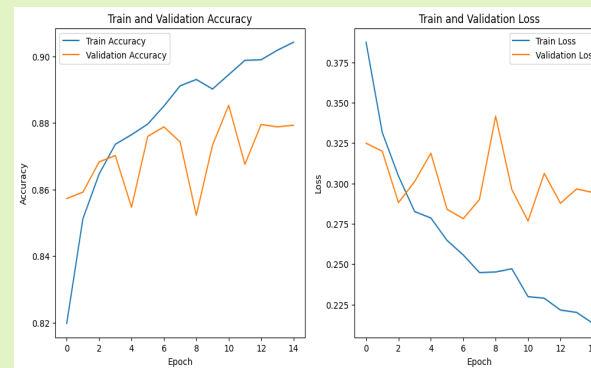
3. VGG16 의 한계점 극복

- VGG16이 추출한 방대한 양의 특징들을 손실없이 max pooling 하기 위해,
14 x 14 x 512인 마지막 convolution layer에서 MaxPooling 레이어를 거치지 않고,
바로 GlobalMaxPooling 레이어를 통과하도록 수정함.
- 더불어서 요약 추출된 특징들을 잘 분류할 수 있도록, 출력층 전에 Dense Layer를 추가함.



VGG16 Architecture





04 모델링



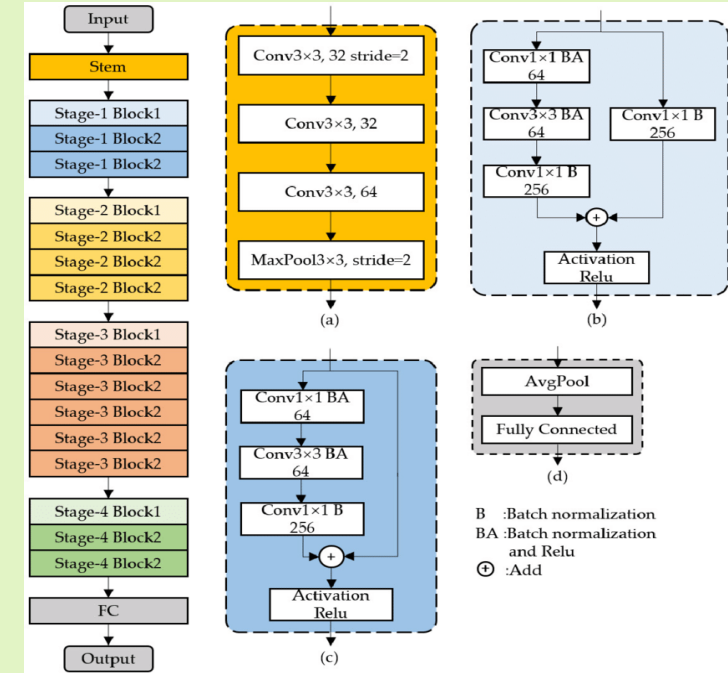
1) 모델 성능 평가 및 선택 - Resnet50

1. Resnet50

- 50개의 레이어로 구성된 모델
- VGG16 기반의 모델로 Convolution Layer & Max Pooling Layer 의 반복으로 구성됨.
- VGG16에서 문제였던 gradient vanishing을 해결함.
- 특징 추출 레이어만 볼 경우, 크게 4개의 Stage로 구성됨.
- 각 스테이지는 여러 개의 블록으로 구성됨.
- 블록내에서는 convolution layer와 batch normalization layer, relu activation layer를 통과를 반복한 후 나오는 결과값에 이전 스테이지의 output을 더하여 새로운 output을 출력함

2. Resnet50 의 한계점

- 다소 무거운 모델



ResNet50 Architecture

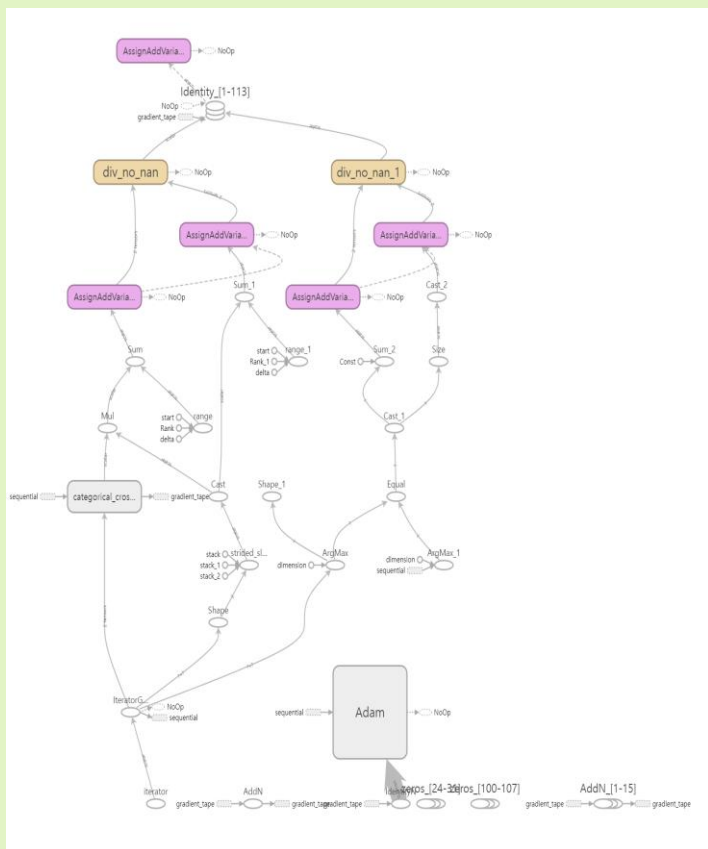


04 모델링



1) 모델의 구성

Resnet50 모델 레이어



데이터 분할

```
[ ] # train:val:test = 6:2:2
train, val_test = train_test_split(data, test_size=0.4, random_state=42)
val, test = train_test_split(val_test, test_size=0.5, random_state=42)
```

```
[ ] train_gen = train_datagen.flow_from_dataframe(train,
x_col = 'Filepath',
y_col = 'Label',
target_size = (224,224),
class_mode = 'categorical',
batch_size = 32,
shuffle =True,
seed=42)

valid_gen = val_test_datagen.flow_from_dataframe(val,
dataframe = val,
x_col = 'Filepath',
y_col = 'Label',
target_size = (224,224),
class_mode = 'categorical',
batch_size = 32,
shuffle =False,
seed=42)

test_gen = val_test_datagen.flow_from_dataframe(test,
dataframe = test,
x_col = 'Filepath',
y_col = 'Label',
target_size = (224,224),
class_mode = 'categorical',
batch_size = 32,
shuffle =False)
```

데이터 증강 ImageDataGenerator

```
[ ] # 데이터 증강을 위한 ImageDataGenerator 객체 생성
train_datagen = ImageDataGenerator(
    rotation_range=20, # 랜덤하게 이미지를 회전할 각도 범위
    width_shift_range=0.1, # 랜덤하게 이미지를 수평으로 이동할 범위
    height_shift_range=0.1, # 랜덤하게 이미지를 수직으로 이동할 범위
    shear_range=0.2, # 랜덤하게 전단 변환을 적용할 각도 범위
    zoom_range=0.2, # 랜덤하게 이미지를 확대 또는 축소할 범위
    horizontal_flip=True, # 랜덤하게 이미지를 수평으로 반전
    fill_mode='nearest' # 이미지 변환 후 생성되는 빈 공간을 어떻게 채울지 설정
)

val_test_datagen = ImageDataGenerator()
```

딥러닝 모델 학습

```
[ ] from tensorflow.keras.applications import ResNet50
from tensorflow.keras.models import Model

base_model = ResNet50(include_top=False, weights='imagenet', input_shape=(224, 224, 3), pooling='avg')
for layer in base_model.layers:
    if isinstance(layer, tf.keras.layers.Conv2D):
        layer.trainable = False

model = Sequential([
    base_model,
    Dense(2048, activation='relu'),
    Dense(3, activation='softmax')
])

model.summary()
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

early_stopping_cb = EarlyStopping(patience=3, monitor='val_loss', restore_best_weights=True)
history = model.fit(
    train_gen,
    epochs=15,
    batch_size=8,
    validation_data=valid_gen,
    callbacks=[early_stopping_cb]
)
```




2) 학습 방법

Con2D 층은 그대로 가져와서 전이학습 구현

딥러닝 모델 학습

```
[ ] from tensorflow.keras.applications import ResNet50
    from tensorflow.keras.models import Model

    base_model = ResNet50(include_top=False, weights='imagenet', input_shape=(224, 224, 3), pooling='avg')
    for layer in base_model.layers:
        if isinstance(layer, tf.keras.layers.Conv2D):
            layer.trainable = False
    model = Sequential([
        base_model,
        Dense(2048, activation='relu'),
        Dense(3, activation='softmax')
    ])

    model.summary()
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

    early_stopping_cb = EarlyStopping(patience=3, monitor='val_loss', restore_best_weights=True)
    history = model.fit(
        train_gen,
        epochs=15,
        batch_size = 8,
        validation_data=valid_gen,
        callbacks=[early_stopping_cb]
    )
```

ResNet50 모델을 이용한 학습

conv5_block1_out (Activation)	(None, 7, 7, 2048)	0	['conv5_block1_add[0][0]']
conv5_block2_1_conv (Conv2D)	(None, 7, 7, 512)	1049088	['conv5_block1_out[0][0]']
conv5_block2_1_bn (Batch Normalization)	(None, 7, 7, 512)	2048	['conv5_block2_1_conv[0][0]']
conv5_block2_1_relu (Activation)	(None, 7, 7, 512)	0	['conv5_block2_1_bn[0][0]']
conv5_block2_2_conv (Conv2D)	(None, 7, 7, 512)	2359808	['conv5_block2_1_relu[0][0]']
conv5_block2_2_bn (Batch Normalization)	(None, 7, 7, 512)	2048	['conv5_block2_2_conv[0][0]']
conv5_block2_2_relu (Activation)	(None, 7, 7, 512)	0	['conv5_block2_2_bn[0][0]']
conv5_block2_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	['conv5_block2_2_relu[0][0]']
conv5_block2_3_bn (Batch Normalization)	(None, 7, 7, 2048)	8192	['conv5_block2_3_conv[0][0]']
conv5_block2_add (Add)	(None, 7, 7, 2048)	0	['conv5_block1_out[0][0]', 'conv5_block2_3_bn[0][0]']
conv5_block2_out (Activation)	(None, 7, 7, 2048)	0	['conv5_block2_add[0][0]']

ResNet50 모델을 이용한 학습

04 모델링



1) 모델 성능 평가 및 선택 - Resnet50

3. Resnet50의 성능 극대화

1. 전이학습

- 전체 전이
- Conv2D만 전이
- Conv2D 중 conv5는 비전이
- Conv2D 중 conv5와 conv4는 비전이

2. 다양한 Optimizer를 사용

- AdamW
- Nadam
- Adam

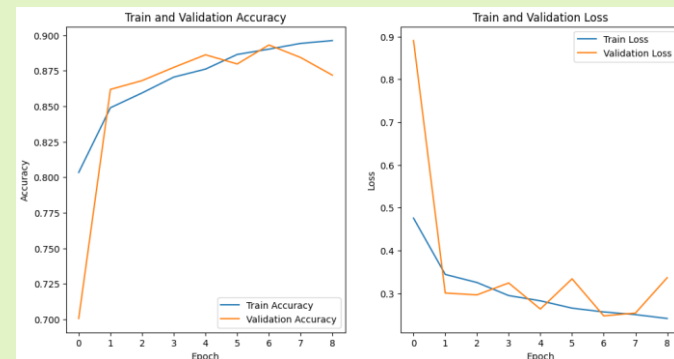
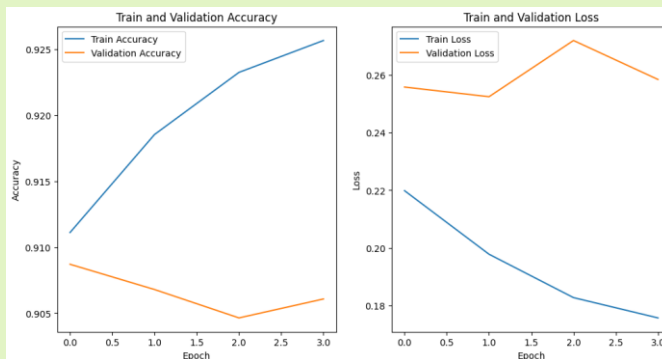
3. 다양한 Pooling을 시도

- No Pooling - Flatten
- GlobalMaxPooling
- GlobalAvgPooling

4. 분류하는 Dense layer의 unit 수를 조절

- 1024
- 2048
- 4096

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9





1) 모델 성능 평가 및 선택 - Resnet50

3. Resnet50의 성능 극대화

1. 전이학습

- 전체 전이
- Conv2D만 전이
- Conv2D 중 conv5는 비전이
- Conv2D 중 conv5와 conv4는 비전이

2. 다양한 Optimizer를 사용

- AdamW
- Nadam
- Adam

3. 다양한 Pooling을 시도

- No Pooling - Flatten
- GlobalMaxPooling
- GlobalAvgPooling

4. 분류하는 Dense layer의 unit 수를 조절

- 1024
- 2048
- 4096

```
resnet50 (Functional)      (None, 7, 7, 2048)      23587712
flatten_2 (Flatten)        (None, 100352)           0
```

```
conv5_block3_out (Activati (None, 7, 7, 2048)      0      ['conv5_block3_add[0][0]']
on)
max_pool (GlobalMaxPooling (None, 2048)           0      ['conv5_block3_out[0][0]']
2D)
```

```
conv5_block3_out (Activati (None, 7, 7, 2048)      0      ['conv5_block3_add[0][0]']
on)
avg_pool (GlobalAveragePoo (None, 2048)           0      ['conv5_block3_out[0][0]']
ling2D)
```



1) 모델 성능 평가 및 선택 - Resnet50

3. Resnet50의 성능 극대화

1. 전이학습

- 전체 전이
- Conv2D만 전이
- Conv2D 중 conv5는 비전이
- Conv2D 중 conv5와 conv4는 비전이

2. 다양한 Optimizer를 사용

- AdamW
- Nadam
- Adam

3. 다양한 Pooling을 시도

- No Pooling - Flatten
- GlobalMaxPooling
- GlobalAvgPooling

4. 분류하는 Dense layer의 unit 수를 조절

- 1024
- 2048
- 4096

tf.keras.optimizers.**AdamW**

AdamW optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments with an added method to decay weights per the techniques discussed in the paper, 'Decoupled Weight Decay Regularization' by [Loshchilov, Hutter et al., 2019](#).

According to [Kingma et al., 2014](#), the underlying Adam method is "computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameters".

tf.keras.optimizers.**Nadam**

Much like Adam is essentially RMSprop with momentum, Nadam is Adam with Nesterov momentum.



4) 하이퍼 파라미터 튜닝 : 최적화할 하이퍼 파라미터로 은닉층 뉴런 수, 옵티마이저 종류, 옵티마이저의 학습률 선택, val_acc 를 최대화하는 방식으로 튜닝

```
def create_model(trial):
    for layer in base_model.layers:
        if isinstance(layer, tf.keras.layers.Conv2D):
            layer.trainable = False

    model = Sequential()
    model.add(base_model)
    model.add(GlobalAveragePooling2D())

    # 하이퍼파라미터 튜닝
    num_hidden_units = trial.suggest_categorical('num_hidden_units', [1024, 2048, 4096]) # 은닉층 뉴런 수 조정
    model.add(Dense(num_hidden_units, activation='relu'))

    model.add(Dense(len(classes), activation='softmax'))

    # 3가지 옵티마이저에 따라 학습률 설정
    optimizer_names = ['adam', 'sgd', 'adamw']
    optimizer_name = trial.suggest_categorical('optimizer', optimizer_names)

    learning_rate = trial.suggest_loguniform('learning_rate', 1e-5, 1e-2)
    if optimizer_name == 'sgd':
        optimizer = SGD(learning_rate=learning_rate)
    elif optimizer_name == 'adamw':
        optimizer = AdamW(learning_rate=learning_rate)
    else:
        optimizer = Adam(learning_rate=learning_rate)

    model.compile(optimizer=optimizer,
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])

    return model
```

Optuna 튜닝 코드

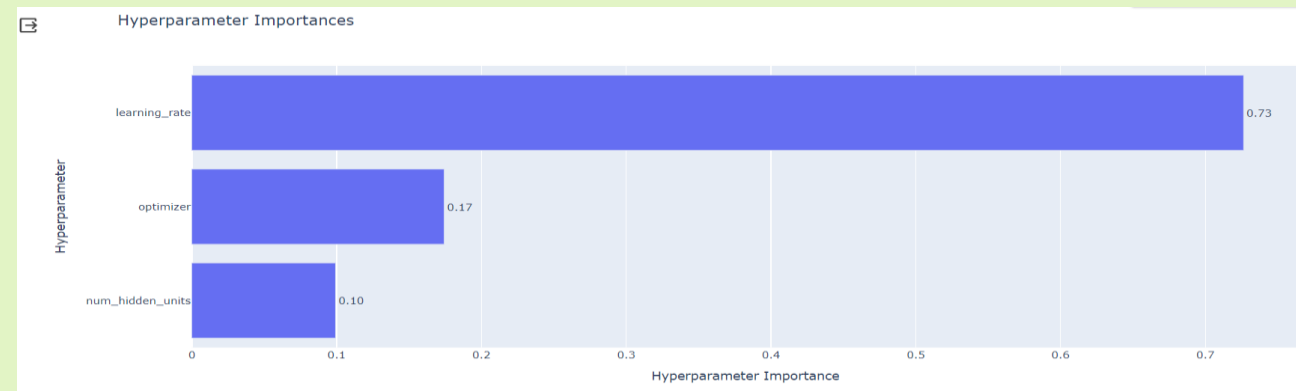
number	value	params_learning_rate	params_num_hidden_units	₩
0	0.861351	0.000581	1024	
1	0.899904	0.001299	4096	
2	0.902778	0.000330	1024	
3	0.893439	0.001433	4096	
4	0.893199	0.002344	4096	
5	0.904693	0.001101	1024	
6	0.840038	0.000011	2048	
7	0.901820	0.003980	1024	
8	0.903736	0.007428	4096	
9	0.835010	0.000011	1024	

number	params_optimizer	state
0	sgd	COMPLETE
1	adamw	COMPLETE
2	adamw	COMPLETE
3	adamw	COMPLETE
4	adam	COMPLETE
5	adamw	COMPLETE
6	sgd	COMPLETE
7	sgd	COMPLETE
8	sgd	COMPLETE
9	sgd	COMPLETE



여러 번 반복하여
가장 결과가 좋은
선택 하이퍼
파라미터 (은닉층
뉴런수, 옵티마이저 종류,
옵티마이저 학습률)
확인

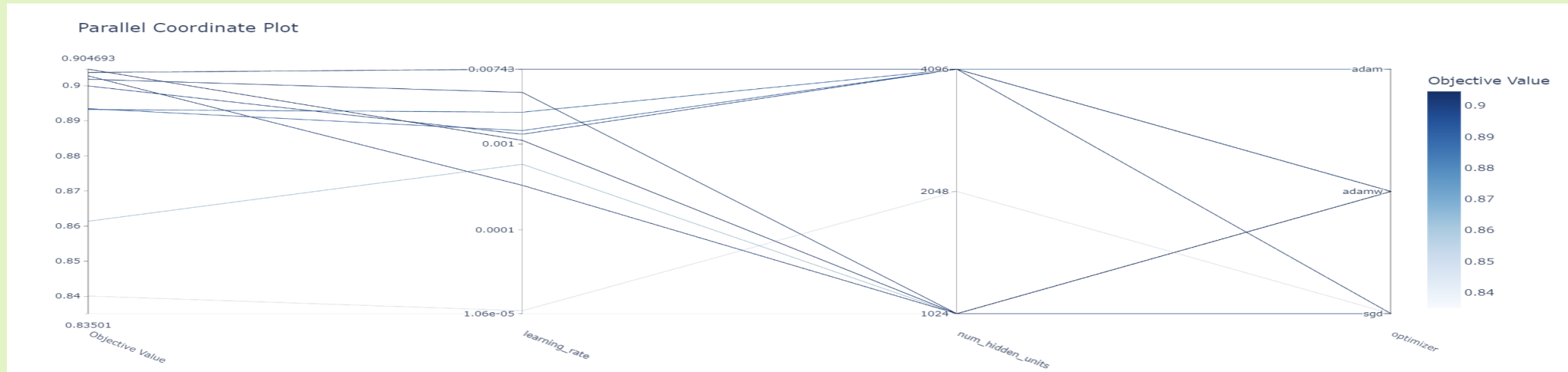
튜닝 결과



가장 중요한 하이퍼 파라미터



4) 하이퍼 파라미터 튜닝



```
[ ] print("Best params: ", study.best_params)
    print("Best value: ", study.best_value)
    print("Best Trial: ", study.best_trial)
    print("Trials: ", study.trials)
```

```
Best params: {'num_hidden_units': 1024, 'optimizer': 'adamw', 'learning_rate': 0.001100815494876212}
Best value: 0.9046934843063354
```

Best params

- 'num_hidden_units' : 1024
- 'optimizer' : adamw
- 'learning_rate' :
0.001100815494876212

Best Value :

0.9046934843063354

04 모델링



5) 최적화 모델로 학습

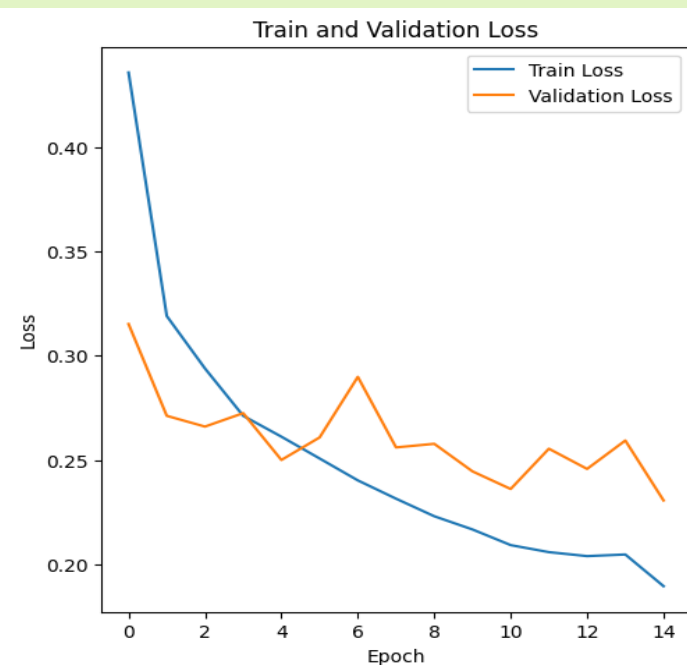
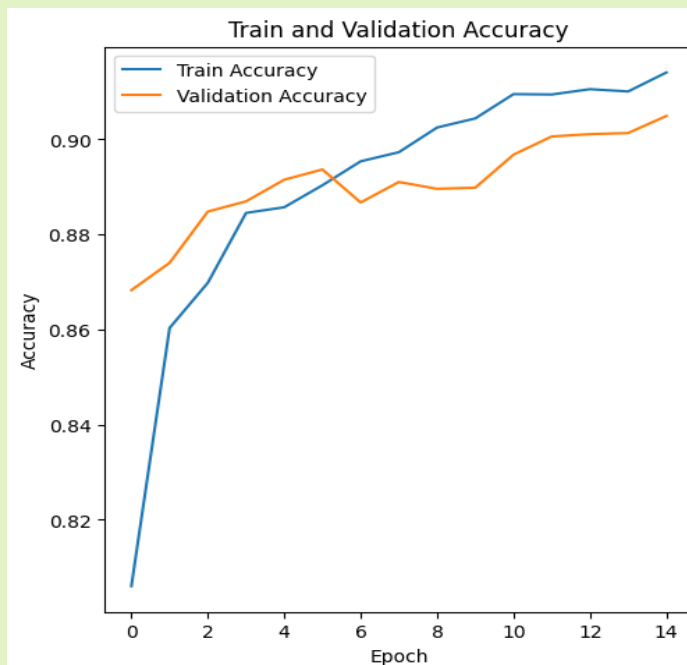
```
[ ] for layer in base_model.layers:
    if isinstance(layer, tf.keras.layers.Conv2D):
        layer.trainable = False

model = Sequential()
model.add(base_model)
model.add(GlobalAveragePooling2D())

# best_param 사용
num_hidden_units = 1024
optimizer = AdamW(learning_rate=0.001100815494876212)

model.add(Dense(num_hidden_units, activation='relu'))
model.add(Dense(len(classes), activation='softmax'))

model.compile(optimizer=optimizer,
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```



```
▶ # 테스트 데이터에서의 성능 평가
test_loss, test_accuracy = model.evaluate(test_generator)
print(f"Test Accuracy: {test_accuracy}")
print(f"Test Loss: {test_loss}")
```

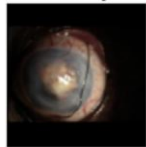
```
523/523 [=====] - 2029s 4s/step - loss: 0.2281 - accuracy: 0.9050
Test Accuracy: 0.9049556851387024
Test Loss: 0.22812870144844055
```

Test acc 변화 0.8908 ► **0.905 (1%p 증가)**

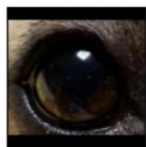
Test loss 변화 0.25485 ► **0.22813 (0.03 감소)**

결과 예상 - 모델 기반

True: Conjunctivitis
Predicted: Conjunctivitis



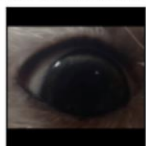
True: Normal
Predicted: Normal



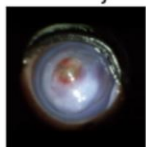
True: Normal
Predicted: Normal



True: Normal
Predicted: Normal



True: Conjunctivitis
Predicted: Conjunctivitis



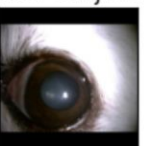
True: Normal
Predicted: Normal



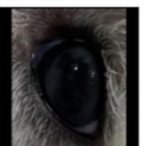
True: Cataract
Predicted: Cataract



True: Cataract
Predicted: Conjunctivitis



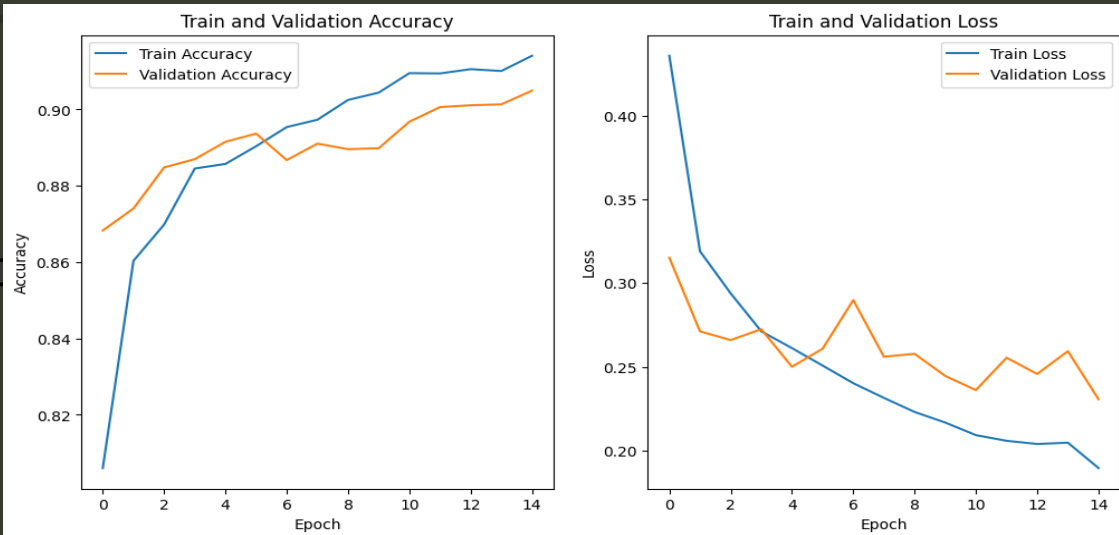
True: Normal
Predicted: Normal



True: Normal
Predicted: Normal

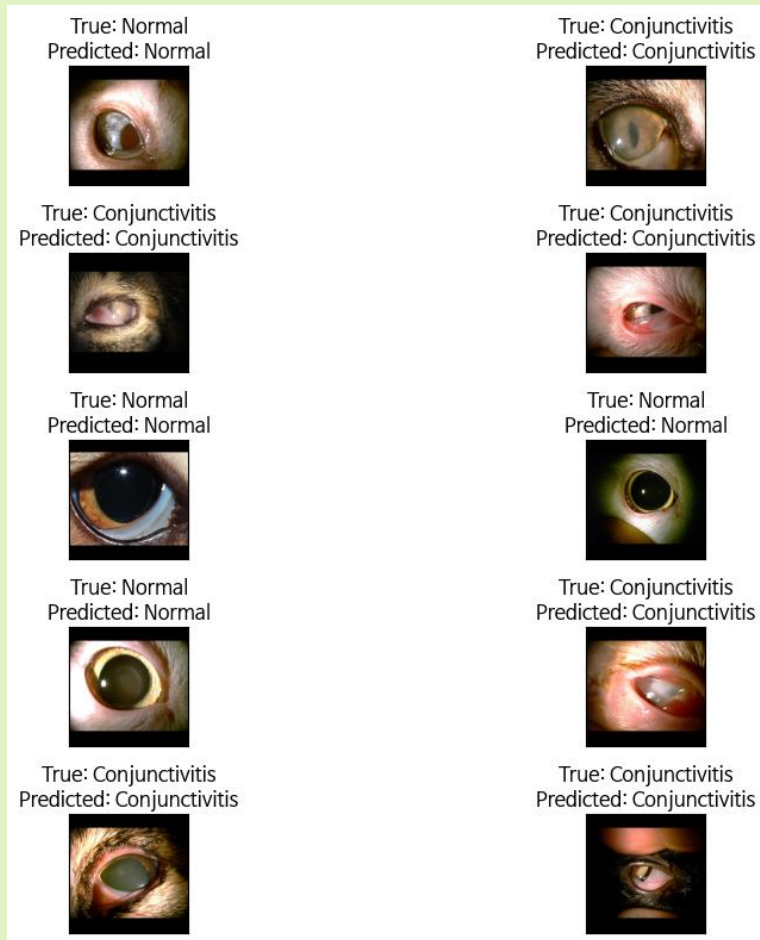


3. Resnet50



ResNet50 모델 기반으로 이미지 직접 넣어 test 결과,
백내장, 결막염, 정상 안구를 잘 판단하는 것 확인

04 모델링 - Grad CAM



테스트 데이터에서의 성능 평가

```
test_loss, test_accuracy = model.evaluate(test_generator)
print(f"Test Accuracy: {test_accuracy}")
print(f"Test Loss: {test_loss}")
```

```
157/157 [=====] - 774s 5s/step -
Test Accuracy: 0.9111288785934448
Test Loss: 0.2669685184955597
```

Test acc 0.9111

Test loss 0.26697

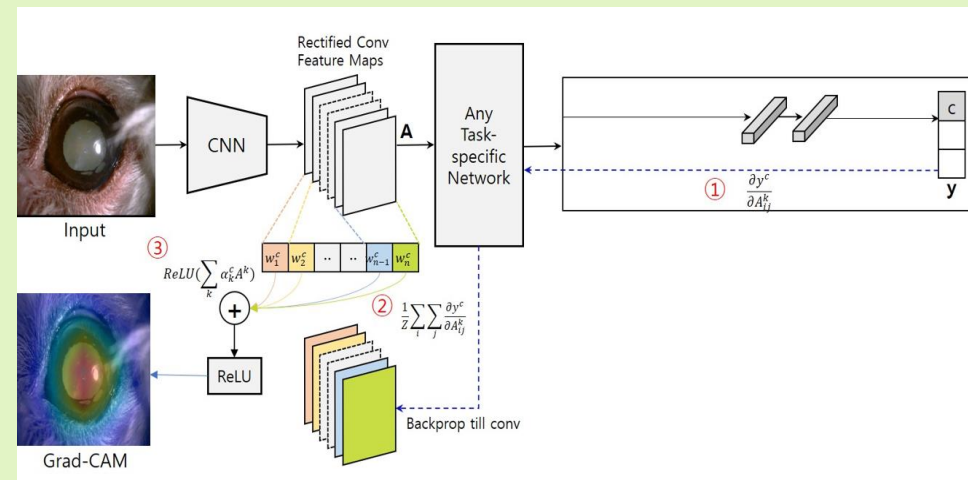
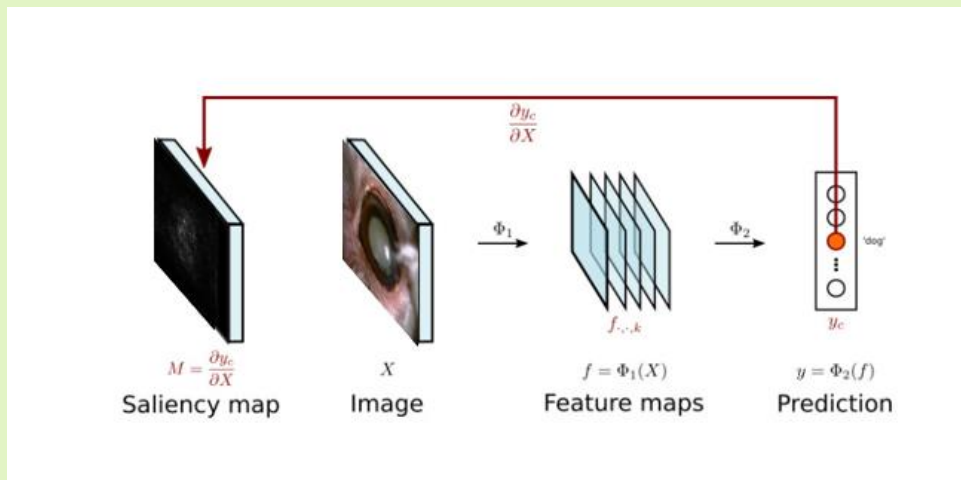
고양이 안구도 동일한 방식으로 진행 -> 잘 분류함

04 모델링 - Grad CAM

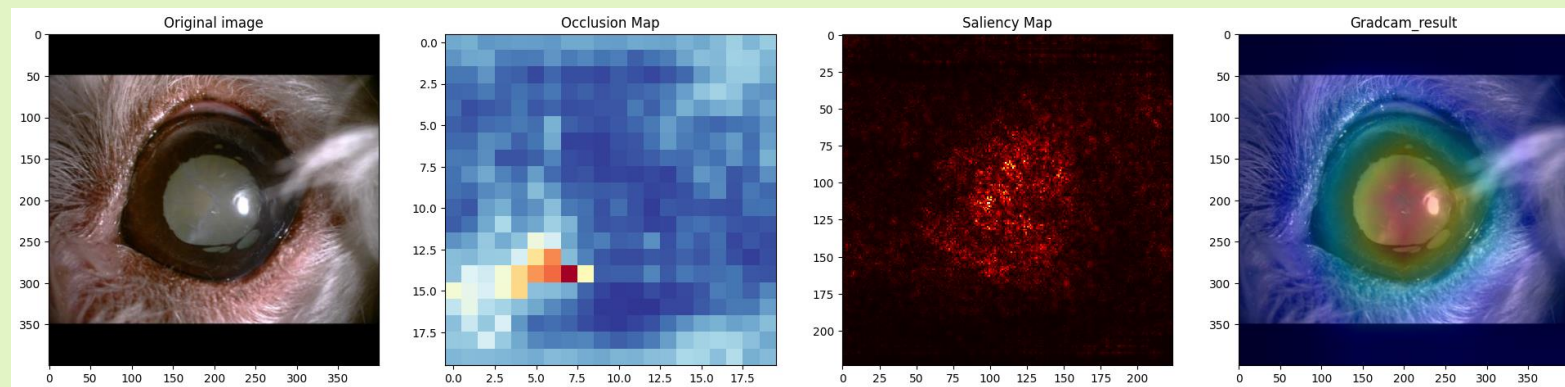


Grad CAM

: 딥러닝 신경망이 이미지의 어떤 부분을 근거로 분류 결정을 내렸는지를 시각화하여 보여줌. 어느 부분이 가장 중요했는지를 이해하기 위해 CNN 모델에서 Gradient를 활용하여 예측 결과에 대한 원인을 해석



실제 백내장 판정 강아지 안구 이미지에 대한 Grad CAM





[HOME](#) [왈니스를 소개합니다](#) [왈니스 동물 병원](#)

WALNESS

무심코 발견한 우리 아이 눈의 이상 징후, 왈니스로 한번 체크해 보세요 !

왈니스를 이용하기 위해 사진을 업로드해 주세요.

사진의 중앙에 우리 아이 눈이 올 수 있도록 위치하고 사진을 찍은 뒤, 업로드하면 완료!

DOG

강아지 or 고양이 선택

파일 선택

선택된 파일 없음

제출

웹사이트 초기 화면 및 서비스 이용 예시



결과 창 예시

HOME
왈니스를 소개합니다
왈니스 동물 병원

1.

2. **[결막염]** 이 의심됩니다.
신뢰도: 100.0%

3.

결막염이란?
고양이 결막염은 눈꺼풀의 얇은 막인 결막에 염증이 생기는 질병입니다.

원인은?
허피스, 칼리시 바이러스, 박테리아 등의 감염
알레르기
유전적 원인
환경 변화

증상은?
눈을 뜨는 것을 불편해함
흰자위 부분이 빨갛게 충혈됨
눈 주변이 부어있음
눈물, 눈곱이 많이 남

신속한 병원 방문 및 진료를 권장합니다.

4. 주위의 동물 병원 찾기

결과 창 구성


- 결과창은 업로드한 이미지와 Grad Cam 이미지로 구성
-> 업로드 이미지의 어떤 부분을 바탕으로 해당 질병이 의심되는지를 설명
- 신뢰도를 함께 표시
- 의심되는 질병과 원인 및 증상 함께 안내
- 동물병원 방문 요망 메시지와 함께 왈니스 웹사이트 내 동물병원 찾기 기능 이용 유도

05 웹페이지 - 이용 방법 시연



[HOME](#) [왈니스를 소개합니다](#) [왈니스 동물병원](#)


왈니스 왈니스

WALNESS 

무심코 발견한 우리 강아지 눈의 이상 징후, 왈니스로 한번 체크해보세요 !

왈니스를 이용하기 위해서는 사진을 업로드해주세요.
사진의 중앙에 우리아이 눈이 올 수 있도록 위치하고 사진을 찍은 뒤, 업로드하면 완료!

강아지



파일 선택

선택된 파일 없음

제출



HOME

알니스를 소개합니다

알니스 동물병원

지도

위성

구글 맵 연동

사용자 위치 기반으로

주위의 동물병원 표시

거리 순으로 보기 ↓

별점 순으로 보기 ↓

한리도우병원

★undefined (0.5km)

힐링동물병원

★4.8 (0.6km)

노산동물병원

★5 (0.6km)

중앙동물병원

★3.3 (0.7km)

(주)애니멀킹덤

★undefined (2.7km)

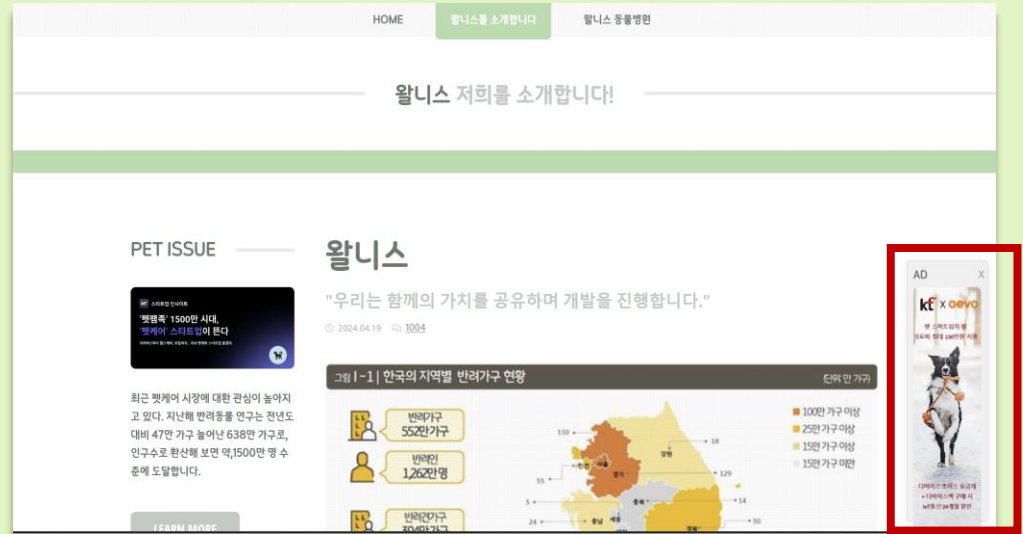
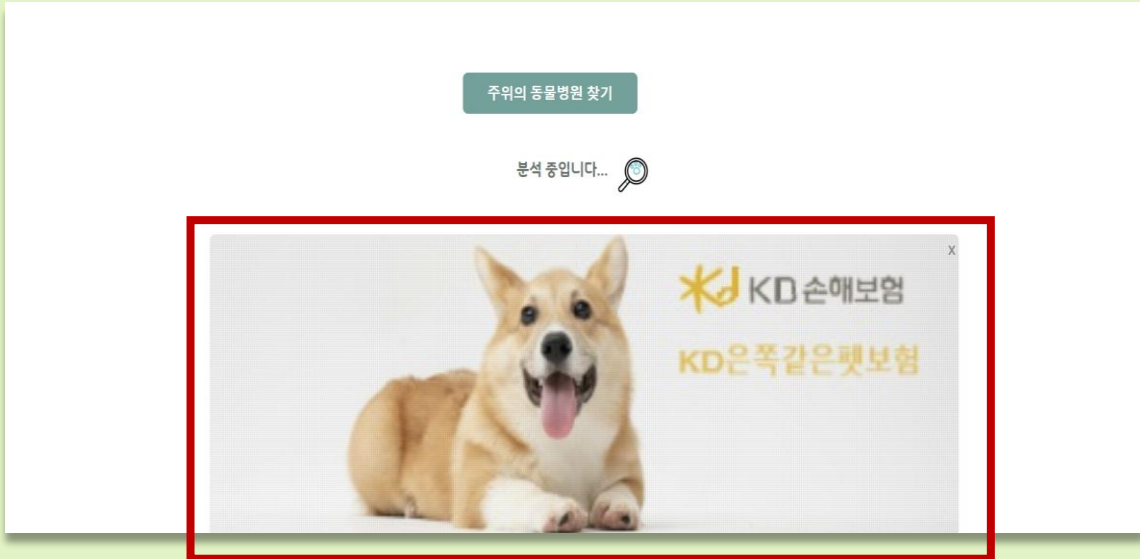
HOME



병원 추천
 니즈에 맞추어 주변의
 동물 병원을 거리 순
 혹은 별점 순으로 확인
 가능



광고 -> 광고 이미지 클릭시 각 연동된 펫보험, 펫 디바이스 정보 페이지로 이동

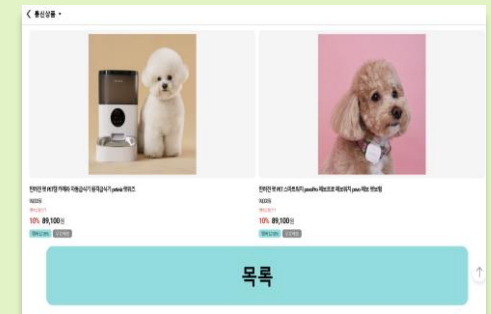


웹사이트 바로 노출 X
분석이 진행되는 동안
펫보험 광고를 노출하여
이용자의 주의를 끌고 클릭 유도



펫 보험

상대적으로 질병과는
연관성이 덜하나,
반려인 전반적으로 관심 가질
수 있는 펫 디바이스 광고는
서비스 소개페이지에 할당



펫 디바이스

06 SWOT 분석 - 출시 이전 서비스 내부 및 외부 환경 분석



- 정확도 약 90%에 달하는 모델의 성능
- 반려인들에게는 필수인 동물 병원 방문, 그러나 부담되는 동물 병원비 -> 동물 병원 방문 이전 이용하는 서비스로서 **병원 방문 수요만큼의 서비스 이용 수요 확보 가능**
- 사진을 찍고 업로드하기만 하면 되는 **쉬운 이용 방법**
- 광고를 통한, 반려동물 시장의 다양한 상품과의 제휴를 통해 수익성 확보



- 진단에 있어 한계 존재
ex) 특정 안과질환(백내장, 결막염)에 대해 확인 가능
-> 모델의 추가 및 발전으로 해결 가능
- 웹사이트 이용 방식으로 사진을 웹에 업로드해야하는 불편함
-> 모바일 버전 개발로 해결 가능

Opportunity



Threat



- 반려인구의 지속적인 증가 but 반려동물 비대면 진단 시장은 아직 크지 않음. 이에 대한 초기 수요를 충족할 수 있음.
- 성장하고 있는 펫보험 시장에 따른 광고수익 증대 기대할 수있음.

- 경쟁 및 선두 업체에서 제공하는 폭넓은 진단 서비스 (피부, 치아, 관절)
- 수익성이 반려동물 상품 시장의 흥망에 다소 의존될 수 있음.

06 SWOT 분석 - 향후 과제



- 진단에 있어 한계 존재
ex) 특정 안과질환(백내장, 결막염)에 대해 확인 가능
-> 모델의 추가 및 발전으로 해결 가능
- 웹사이트 이용 방식으로 사진을 웹에 업로드해야하는 불편함
-> 모바일 버전 개발로 해결 가능

Threat



- 경쟁 및 선두 업체에서 제공하는 폭넓은 진단 서비스 (피부, 치아, 관절)
- 수익성이 반려동물 상품 시장의 흥망에 다소 의존될 수 있음.

향후 과제

1. 시장에서의 경쟁성 확보를 위해 더욱 다양한 질병을 판별할 수 있는 서비스 제공 필요
2. 이용자가 안구 이미지를 직접 크롭하거나, 안구만을 찍어야 하는 번거로움을 덜 수 있도록, open CV를 이용하여 넓은 범위를 보여주는 이미지에서도 필요한 부분만 분석할 수 있도록 개발 필요
3. 모바일 버전 개발

기대 효과

- 보다 쉬운 반려동물 건강 관리
- 반려동물 양육비 중 약 35%에 해당 (2023년 기준) 하는
병원비에 대한 절감 효과
- 광고로서의 상품 및 서비스 제휴를 통해
반려동물 관련 데이터 축적 가능
 - ▶ 축적된 데이터를 기반으로
반려동물 관련 새로운 서비스 개발 가능
- 반려동물 관련 서비스 시장 활성화

08 일정 관리



GANTT Chart

WBS NUMBER	TASK TITLE	PHASE ONE															PHASE TWO															PHASE THREE														
		WEEK 1 (2.29~3.1)					WEEK 2 (3.4~3.8)					WEEK 3 (3.11~3.15)					WEEK 4 (3.18~3.22)					WEEK 5 (3.25~3.29)					WEEK 6 (4.1~4.5)					WEEK 7 (4.8~4.12)					WEEK 8 (4.15~4.19)									
		M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F	M	T	W	R	F										
1 Project Conception and Initiation (프로젝트 컨셉)																																														
1.1	주제 확정																																													
1.2	메인 데이터 확정																																													
1.3	팀명 정하기																																													
1.4	멘토님, 강사님 가이드 받기																																													
2 Project Definition and Planning (프로젝트 정의 및 계획)																																														
2.1	프로젝트 목표 정하기																																													
2.2	결과물 스케치 아웃																																													
2.3	개발 툴 확정																																													
2.4	개발 알고리즘 탐색																																													
2.5	추가 개발 사항 탐색																																													
2.6	초대용량 데이터 다운로드 방법																																													
3 Project Development (프로젝트 개발)																																														
3.1	데이터 전처리																																													
3.2	데이터 EDA																																													
3.3	다양한 알고리즘으로 모델 훈련																																													
3.3.1	GoogleNet																																													
3.3.2	VGGNet																																													
3.3.3	ResNet																																													
3.3.4	EfficientNet																																													
3.3.5	Transformer + AutoEncoder																																													
4 Project Deployment (프로젝트 서비스화)																																														
4.1	서버에 모델 올리기																																													
4.2	사용자가 웹페이지로 접속 및 이용																																													
4.3	구글지도 API 결합																																													
4.4	Ad 서비스 접속																																													
5.0	프로젝트 지연 혹은 기능 추가에 대비하는 기간																																													

프로젝트 지연 혹은 기능 추가에 대비하는 기간

08 일정 관리



Customized Scrum

지도에서 동물병원 검색이 안되는 현상 수정하기	장진형	민지현, 박재영	51	✓	4/4		
지도 검색된 동물병원 거리순 정렬	민지현	민지현	51	✓	4/4		
지도 검색된 동물병원 별점순 정렬	민지현	장진형	55	✓	4/4		
지도에 팝업 혹은 다른 방법으로 동물병원 정보 표시하기	민지현	민지현	59	✓	4/4		
지도에서 현재 위치를 실제 내 위치 정보를 표시하도록 바꾸기	박재영	박재영	67	✓	4/4		
지도에 현재 내 위치 표시하기 & 현재 클릭한 병원 컬러 강조	장진형	장진형	57	✓	4/4		
지도에 병원리스트가 초기화 안 되는 문제, 병원 리스트가 무한대로 뿔 붙여지는 문제, 병원리스트 더보기 하면 재정렬 안 되는 문제, 버튼 좌우로 정렬, curLocation 이function으로 들어와서 기존의 거리 계산 함수가 작동 안 하는 문제	장진형	장진형	63,65,69	✓	4/4		
펫보험 광고 시스템	ALL	민지현, 이서은		✓	4/2		
main 플로팅 배너 생성	민지현, 이서은	민지현	11	✓	4/2		
main 플로팅 배너 -> 가로 배너 수정	민지현, 이서은	민지현	82	✓	4/11	submit 버튼을 누른 후 광고가 뜨도록 수정	
left sidebar 플로팅 배너 생성	민지현, 이서은	이서은	27	✓	4/2		
utils.py 의 grad_cam 함수 완성하기	장진형	박재영, 최성림	92	✓	4/11	grad_cam 함수 추가	
utils.py 의 predict 함수 완성하기	장진형	민지현, 이서은	76	✓	4/11		
고양이 모델 추가	민지현	민지현	94	✓	4/17		
Milestone - Complete Frontend (프론트엔드)							98.70%
웹페이지 결과물 스케치 아웃	ALL	-	-	✓	-	추후 지속적인 업데이트 필요	
웹페이지 Django와 연결하기	ALL	최성림, 이서은	6, 8	✓	4/1		
웹페이지 정리	ALL	정겨운, 최성림	9	✓	4/2	dropdown 삭제, left sidebar 블로그 형식처럼 수정중	
왓슨 서비스 간략설명 이미지 편집	정겨운, 최성림	정겨운, 최성림	17	✓	4/3		
웹페이지 상 이미지 업로드 공간 편집	정겨운, 최성림	정겨운, 최성림	34	✓	4/4		
이미지 업로드 후 결과 디스플레이 구성	정겨운, 최성림	정겨운, 최성림	36	✓	4/5		
이미지 업로드 창 생성 및 구성	정겨운	정겨운		✓	4/5		
사용자가 첨부한 이미지를 받고 웹페이지에 띄우기	정겨운, 최성림	장진형	72, 74	✓	4/10		
각 질병에 따른 진단 결과 출력 완성하기	장진형	정겨운	78, 80	✓	4/10		
분석 중 안내 메시지 추가	민지현	민지현	86	✓	4/11		
페이지 위치 포커싱 추가	민지현	민지현	90	✓	4/11		
질병 정보 추가	이서은	이서은	101	✓	4/17		
일부 버튼 스타일 변경	박재영	박재영	104	✓	4/17		

08 버전 관리



Dev Containers v0.354.0
Microsoft microsoft.com | 23,648,919 | ★★★★★ (47)

Open any folder or repository inside a Docker container and take advantage of Visual Studio Code's full feature set.

[Disable](#) [Uninstall](#) [Switch to Pre-Release Version](#) ⚙️

This extension is enabled globally.

Visual Studio Code Dev Containers

The **Dev Containers** extension lets you use a [Docker container](#) as a full-featured development environment. Whether you deploy to containers or not, containers make a great development environment because you can:

- Develop with a consistent, easily reproducible toolchain on the same operating system you deploy to.
- Quickly swap between different, separate development environments and safely make updates without worrying about impacting your local machine.
- Make it easy for new team members / contributors to get up and running in a consistent development environment.

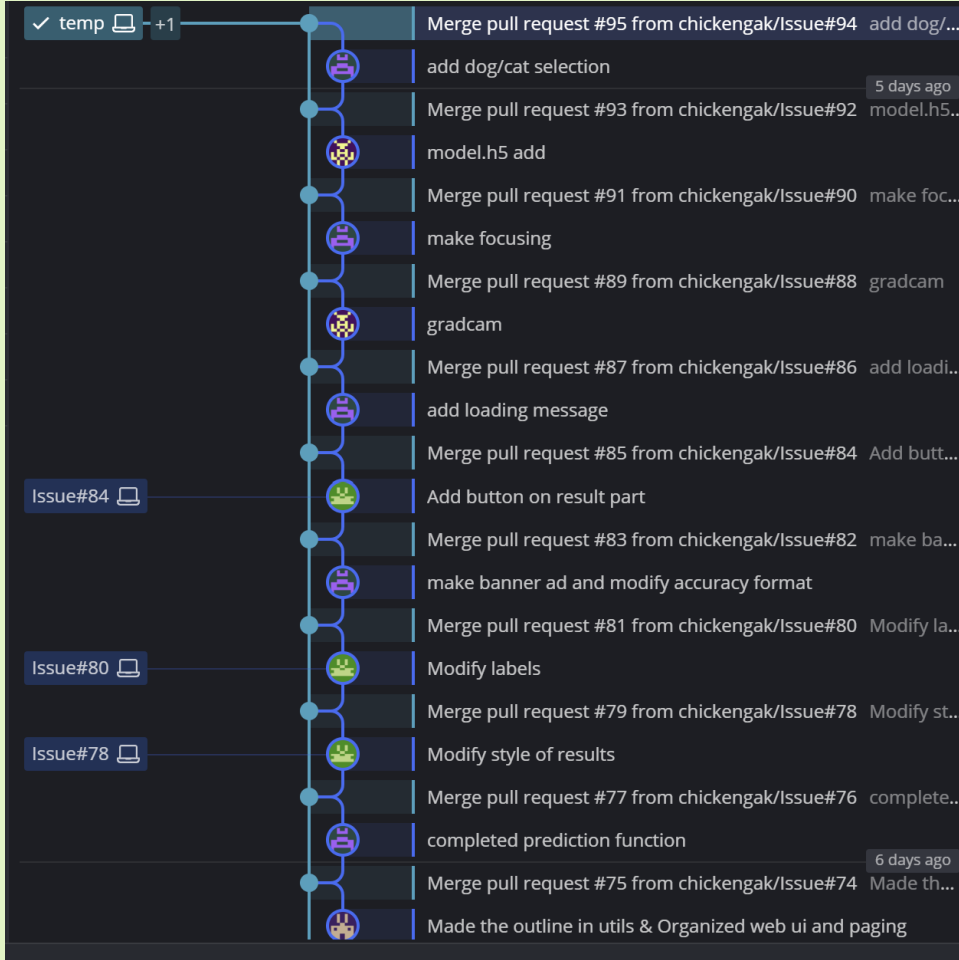
Categories
[Other](#)

Resources
[Marketplace](#)
[Issues](#)
[Repository](#)
[License](#)
[Microsoft](#)

Requirements.txt

- Django==5.0.1
- numpy==1.26.3
- pandas==2.2.0
- matplotlib==3.8.2
- seaborn==0.13.1
- scikit-learn==1.4.0
- beautifulsoup4==4.12.3
- mysql-connector-python==8.3.0
- tensorflow==2.15.0.post1
- fastai==2.7.14
- torch==2.2.1
- torchvision==0.17.1
- torchaudio==2.2.1
- opencv-python==4.8.0.76
- nbformat==5.10.4
- plotly==5.20.0

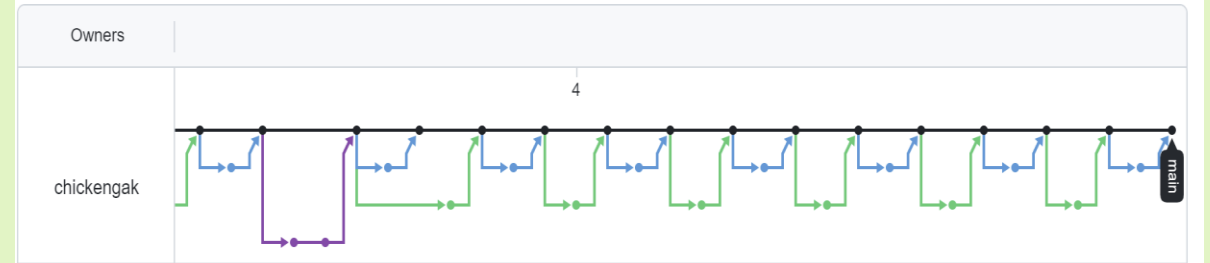
08 버전 관리



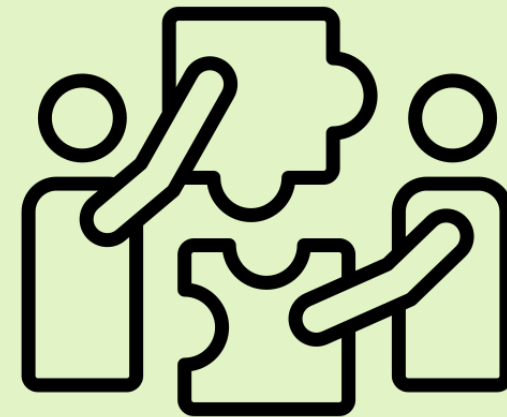
Gitkraken으로 실시간 확인

Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



팀원들의 협업을 git으로 관리



08 프로젝트 후기



장진형

능동적이고 적극적인 팀원들 덕분에 일정의 큰 차질없이 무사히 프로젝트를 마칠 수 있어서 감사드리고, 처음 겪어보는 이미지 데이터 프로젝트에 좌충우돌 사건사고가 많았는데, 꼭 다음에는 좋은 컴퓨팅으로 이미지데이터와 리벤지 매치를 뜨고 싶습니다.



민지현

반려인으로서 관심 있던 주제라 즐겁게 참여할 수 있었습니다. 모델 성능 향상을 목표로 딥러닝 모델의 구조를 살피고 여러 시도를 해볼 수 있어 좋았습니다. 또한 큰 규모의 이미지 데이터를 다룰 때 어떤 어려움이 있는지 몸소 확인할 수 있는 값진 시간이었습니다. 문제가 생기면 함께 해결해 주셨던 조장 및 조원분들께 감사하다는 말씀드립니다.



박재영

프로젝트 진행중에 많은 벽을 만나 여전히 자신이 부족하다는 것을 다시 한번 알게되었습니다. 또한 이번 프로젝트를 준비하면서 딥러닝은 생각할 것도 준비할 것도 대비할 것도 굉장히 많았다는 것을 배웠고 다음 프로젝트 부터는 만반의 준비를 갖추고 도전할 수 있도록 노력하겠습니다. 끝으로 부족한 저를 이끌어주신 팀원들에게 감사의 말씀을 전합니다.



이서은

모델을 만들고 학습시키는 과정에서 하이퍼 파라미터를 최적화하는 방법에 대해 학습할 수 있었고, 딥러닝 모델에 대한 전반적인 이해도를 높일 수 있었습니다. 의료 데이터임에도 불구하고 결과가 잘 나온 것 같아 만족스럽습니다. 조장님과 다른 조원분들이 도와주셔서 무사히, 즐겁게 프로젝트를 마무리할 수 있었습니다.



정겨운

단기간에 딥러닝에 대한 이해도를 높일 수 있는 정말 좋은 기회였습니다. 혼자서 공부하는 것과 협업을 통해 프로젝트를 진행하는 것에는 큰 차이가 있다는 것을 느낄 수 있었습니다. 파이널 프로젝트 진행하는 동안 협업 과정 자체가 매우 체계적이라 협업하는 과정이 즐거웠습니다. 어떤 사소한 문제라도 소통을 통해 함께 풀어나갈 수 있는 분위기를 조성해주신 팀원 분들과 프로젝트 진행 경험을 바탕으로 매끄러운 진행으로 팀을 이끌어 주신 팀장님께 감사합니다.



최성림

다양한 모델을 접해볼 수 있는 기회를 가질 수 있어서 좋았습니다. 모델을 설명할 수 있는 XAI라는 것도 찾아볼 수 있어서 좋았고 좋은 팀원을 만나서 일정에 문제가 없고 잘 진행된 프로젝트를 해볼 수 있어서 너무 좋았습니다 감사합니다.

Reference

- Kim, J. Y., Lee, H. E., Choi, Y. H., Lee, S. J., & Jeon, J. S. (2019). CNN-based diagnosis models for canine ulcerative keratitis. *Scientific reports*, 9(1), 14209.
- Kim, D., Kim, J., Yoo, J., & Choi, D. (2023). Image-Based Diagnosis Service for Pet Eye Diseases. *한국콘텐츠학회/ ICCC 논문집*, 311-312.
- Kim, J. Y., Han, M. G., Chun, J. H., Huh, E. A., & Lee, S. J. (2022). Developing a diagnosis model for dry eye disease in dogs using object detection. *Scientific Reports*, 12(1), 21351.
- Yang, Z., Mitra, A., Liu, W., Berlowitz, D., & Yu, H. (2023). TransformEHR: transformer-based encoder-decoder generative model to enhance prediction of disease outcomes using electronic health records. *Nature Communications*, 14(1), 7857.
- 김도우, 이주원, & 최도진. (2023). 이미지 기반 반려동물 안구질환 진단 모바일 애플리케이션. *한국정보과학회 학술발표논문집*, 1649-1651.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105-6114). PMLR.
- Qassim, H., Feinzimer, D., & Verma, A. (2017). Residual squeeze vgg16. *arXiv preprint arXiv:1705.03004*.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- PyTorch. (s.f.). *Quantization Aware Training (QAT) on VGG*.
- 이학범. (2023, 6월 7일). 국내 반려동물 양육 가구 552만, 반려인 1262만명. *데일리벳*.
- 이학범. (2023, 6월 7일). 동물병원 선택 기준 1위, 2년 만에 '접근성'→'가격'으로 바뀐다. *데일리벳*.
- 김예지. (2024, 2월 12일). 반려동물 양육비 중 30%가 진료비... 펫보험 가입은 1% 그쳐 [반려동물 800만시대, '펫보험' 국민보험 되려면 (上)]. *파이낸셜뉴스*.

Data

- 반려동물 안구 질환 데이터_AI HUB



Q&A



For you pets wellness, Walness

감사합니다

