

```
In [1]: import tensorflow as tf
import numpy as np
from tensorflow.keras.preprocessing.image import load_img, img_to_array

# 저장된 모델 로드
model = tf.keras.models.load_model('weather_ImageNet1205_ver2.h5')
```

WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.

```
In [2]: # 테스트 이미지 로드 및 전처리
def preprocess_image(image_path, target_size=(128, 128)):
    # 이미지 로드
    image = load_img(image_path, target_size=target_size) # 지정한 크기로 이미지 로드
    # 이미지를 배열로 변환
    image = img_to_array(image)
    # 배치 차원 추가 (모델은 배치 형태의 입력을 요구)
    image = np.expand_dims(image, axis=0)
    # 모델 학습 시 사용한 사전처리 함수 적용
    image = tf.keras.applications.resnet50.preprocess_input(image)
    return image
```

```
In [67]: # 클래스 이름 (폴더 이름과 동일)
class_names = ['clear', 'cloudy', 'rain', 'sunrise'] # 실제 클래스 이름에 맞게 수정

# 이미지 경로
image_path = '/Users/User/Experiment_6/Model_test_data/test016.jpg' # 테스트할 이미지

# sunrise_ , clear_

# 이미지 전처리
preprocessed_image = preprocess_image(image_path)

# 예측 수행
predictions = model.predict(preprocessed_image)

# 예측 결과 해석
predicted_class = class_names[np.argmax(predictions)]
confidence = np.max(predictions)

print(f"Predicted Class: {predicted_class} with confidence {confidence:.2f}")
```

1/1 ————— 0s 30ms/step
Predicted Class: rain with confidence 0.71

```
In [69]: import matplotlib.pyplot as plt
from tensorflow.keras.utils import load_img

# 테스트 이미지 로드 (시각화를 위해 로드)
test_image = load_img(image_path, target_size=(128, 128))

# 이미지 시각화
plt.imshow(test_image)
plt.title(f"Prediction: {predicted_class} ({confidence:.2f})")
```

```
plt.axis('off')  
plt.show()
```

Prediction: rain (0.71)



In []: