

미니프로젝트_03

게임 분류모델 제작

4조: 구본현, 김동현, 김성학

목차

1. 주제 선정 과정

2. 데이터 준비(크롤링)

3. 데이터 전처리

데이터 증식, 카테고리 재분류

4. 모델 제작 및 검증

CNN, VGG16, Resnet

5. 결과 및 개선할 점



0-1 조원 소개



김성학

국어국문창작학과



구본현

글로벌경영학과



김동현

화학공학과

0-2 프로젝트 일정

5/23

5/24

5/25

5/26

5/27

5/28

5/29

5/30

5/31

6/1

6/2

6/3

주제 선정 및 데이터 수집 준비

데이터 수집(크롤링)

모델 제작

모델 검증 및 보완

발표 준비

발표

1: 주제 선정 과정

-문제 제기

-프로젝트의 목표 설정



1-1 문제 제기

캐주얼 게임 매출 매년 증가

2023년 한국 모바일 게임 수익 순위 및 수익 성장 순위 TOP 10

수익 순위	수익 성장 순위
1  Lineage M NCSoft	1  Night Crows Wemade
2  Odin: Valhalla Rising Kakao Games	2  ArcheAge War Kakao Games
3  Night Crows Wemade	3  Whiteout Survival Century Games
4  Lineage W NCSoft	4  Honkai: Star Rail miHoYo
5  Lineage 2M NCSoft	5  Devil M 37Games
6  ArcheAge War Kakao Games	6  Seven Knights Idle Adventure Netmarble Games
7  EA SPORTS FC™ Mobile Soccer Electronic Arts & NEXON	7  EA SPORTS FC™ Mobile Soccer Electronic Arts & NEXON
8  EA SPORTS FC Online M NEXON	8  Wars of Pracia NEXON
9  Genshin Impact miHoYo	9  Ares: Rise of Guardians Kakao Games
10  Whiteout Survival Century Games	10  Royal Match Dream Games

2023년 한국 모바일 게임 수익 순위 및 수익 성장 순위. / 이미지=센서타워

-팬데믹 이후 주춤하던 글로벌 모바일 게임 시장이 증가하는 흐름을 보이고 있다.

-시기의 흐름에 따라 인기있는 게임들의 트렌드가 변화하고 있다.

〉 사용자의 성향의 변화가 계속해서 변하는 요즘 선호 게임 장르의 변화에 빠르게 맞춰주는 시스템이 필요하다.

[시사저널e=박금재 기자] 엔데믹 이후 주춤했던 글로벌 모바일 게임 시장이 다시 기지개를 켜고 있다. 다만 트렌드는 많이 변화한 것으로 보인다. 과거 다중접속역할수행게임(MMORPG)이 매출 순위 상위권을 대부분 차지했던 것과 달리 현재 모바일 게임 시장에선 보드 게임과 퍼즐 게임의 활약이 두드러진다.

1-2 프로젝트의 목표



목표: 이미지, 텍스트, 수치를 통해 16개의 게임 카테고리 분류

2: 데이터 준비

-데이터 수집 사이트 선택

-이미지 크롤링

-비이미지 크롤링

-크롤링 결과



2-1 데이터 수집 사이트 선정 sensor tower

스토어: App Store, Google Play | 날짜: 2024/05/31 | 국가/지역: 대한민국 | 카테고리: 모든 게임들 | 인앱 구매: 모두

순위	앱명	개발사	별점	다운로드	가격
1	스워드 버스터즈	Supercell	★★★★☆ (102,698)	무료 다운로드	인앱 구매 • 가격: 무료
2	캣 판타지 - 300불기 증명	Fundoll Global Limited	★★★★☆ (185)	무료 다운로드	인앱 구매 • 가격: 무료
3	레이븐2	Netmarble	★★★★☆ (1095)	무료 다운로드	인앱 구매 • 가격: 무료

유료 다운로드

앱명	개발사	별점	다운로드	가격
Minecraft	Mojang	★★★★☆ (5,032,786)	유료 다운로드	인앱 구매 • 가격: ₩3,000
A Dance Of Fire and Ice	7th Beat Games	★★★★☆ (28,203)	유료 다운로드	인앱 구매 • 가격: ₩2,200
수박게임	Aladdin X Inc.	★★★★☆ (680)	유료 다운로드	가격: ₩2,500

23	Death Crow : dc idle RPG 무지개토끼	★★★★☆ (1,514)	인앱 구매 • 가격: 무료
24	Tubi: Free Movies & Live TV Tubi TV	★★★★☆ (1,893,388)	가격: 무료
25	Max: Stream HBO, TV, & Movies WarnerMedia Global Digital Services, LLC	★★★★☆ (1,349,163)	인앱 구매 • 가격: 무료

Mini Metro Dinosaur Polo Club	★★★★☆ (68,153)	가격: \$0.99
Papa's Freezeria To Go! Flipline Studios	★★★★☆ (23,423)	가격: \$0.99
FL STUDIO MOBILE Image-Line	★★★★☆ (37,393)	인앱 구매 • 가격: \$14.99

① 추가 스토어 순위 및 분석을 알아보려면 [로그인](#) 하거나 [무료 계정을 생성](#)하세요.

장점: 각 카테고리별로 많은 자료 수집 가능.

단점: 로그인 필요.

2-1 데이터 수집 사이트 선정 모바일인덱스 GAME

마켓

All

Google Play

App Store

기간

2024.05.31

국가

대한민국

아시아

북미,유럽,기타



















순위 차트







매출

무료

유료

엑셀다운로드

순위	Google Play	App Store	One Store
1	<div></div> <div>라스트 워: 서바이벌</div> <div>FirstFun</div> <div></div>	<div></div> <div>라스트 워: 서바이벌</div> <div>FirstFun</div> <div></div>	<div></div> <div>EA SPORTS FC Online M</div> <div>NEXON Company</div> <div></div>
2	<div></div> <div>라니지M</div> <div>NCSOFT</div> <div></div>	<div></div> <div>레이븐2</div> <div>Netmarble Corporation</div> <div></div>	<div></div> <div>블루 아카이브</div> <div>NEXON Company</div> <div></div>
3	<div></div> <div>영조:워더링 웨이브</div> <div>HK KURO GAMES LIMITED</div> <div></div>	<div></div> <div>나 혼자만 레벨업:아라이즈</div> <div>Netmarble</div> <div></div>	<div></div> <div>버섯커 키우기 - 3000품기 증명</div> <div>Joy Nice Games</div> <div></div>
4	<div></div> <div>나 혼자만 레벨업:아라이즈</div> <div>Netmarble</div> <div></div>	<div></div> <div>FC 모바일</div> <div>NEXON Company</div> <div></div>	<div></div> <div>나이트 크로우</div> <div>Wemade Co., Ltd</div> <div></div>
5	<div></div> <div>WOS: 화이트아웃 서바이벌</div> <div>Century Games Pte. Ltd.</div> <div></div>	<div></div> <div>세븐나이트 키우기</div> <div>Netmarble</div> <div></div>	<div></div> <div>창세가전 모바일</div> <div>LineGames</div> <div></div>
6	<div></div> <div>라니지W</div> <div>NCSOFT</div> <div></div>	<div></div> <div>오단: 빌빌라 라이징</div> <div>Kakao Games Corp.</div> <div></div>	<div></div> <div>에이지 오브 제트: 타워 디펜스</div> <div>CamelStudio</div> <div></div>

95	<div></div> <div>모두의마블</div> <div>Netmarble</div>
96	<div></div> <div>Pokémon GO</div> <div>Niantic, Inc.</div>
97	<div></div> <div>히어로즈 테일즈</div> <div>37 Mobile Games</div>
98	<div></div> <div>작혼: 리치 마작</div> <div>Yostar Limited.</div>
99	<div></div> <div>명일방주</div> <div>Yostar Limited.</div>
100	<div></div> <div>삼국지 전략판</div> <div>Qookka Games</div>

- 장점: 크롤링에 유리하고 로그인이 필요없음.
- 단점: 자료가 상대적으로 부족해 추가적인 조치가 필요.

2-1 데이터 수집 사이트 선정 모바일인덱스 GAME

기간 국가

일간 < 2020.05.01 > 대한민국

2020년 5월

월	화	수	목	금	토	일
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

취소 적용

2020년부터
2024년까지
3개월 단위로
게임 데이터
최대 100개씩 수집

기간 국가

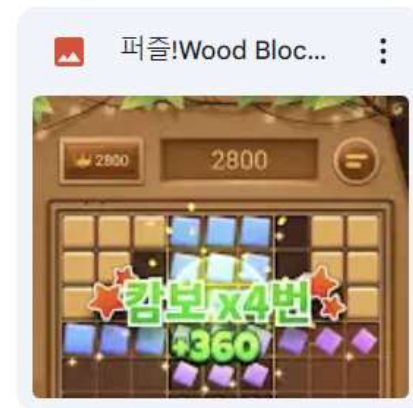
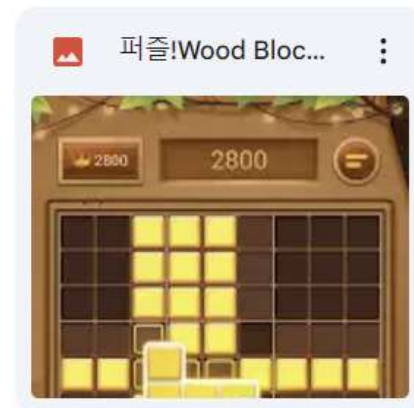
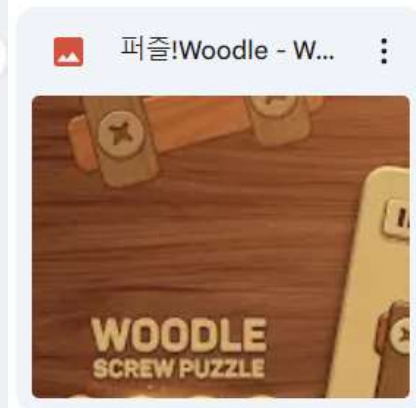
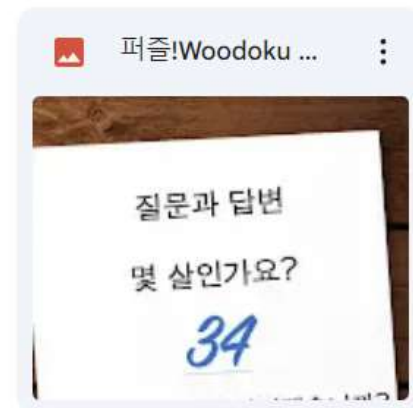
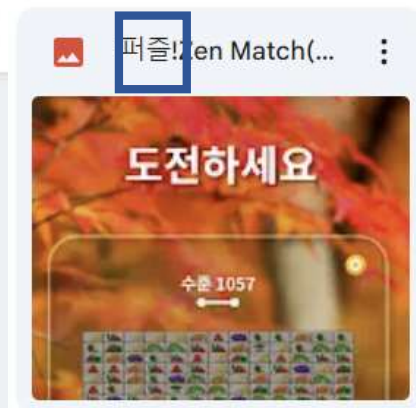
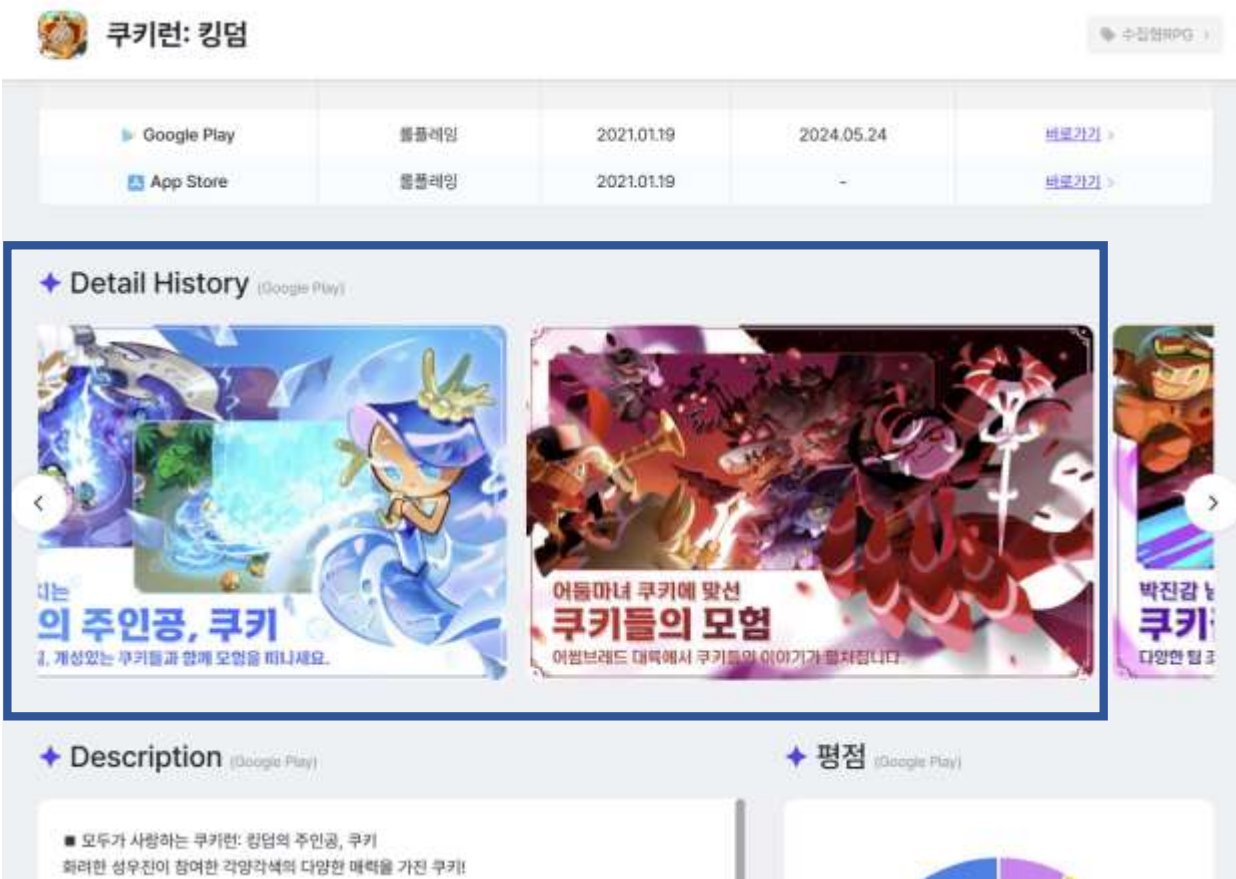
일간 < 2024.05.31 > 대한민국

2024년 5월

월	화	수	목	금	토	일
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

취소 적용

2-2 크롤링한 자료(이미지)



앱 내부 이미지 2개씩 확보 후 게임 카테고리 별로 분류

2-3 크롤링한 자료(텍스트, 수치)

```
title_lst.append(title) #게임 제목
appUrl_lst.append(appUrl) #게임 URL
comapny_lst.append(company) #게임 회사
gcategory_lst.append(gcategory) #대분류
dcategory_lst.append(dcategory) #소분류(태그)
description_lst.append(description) #설명
comment_rank_lst.append(comment_rank) #별점
comment_rank_num_lst.append(comment_rank_num) #별점 수
```

title	appUrl	company	gcategory	dcategory	description	comment	comment_rank_num
Minecraft	https://gar	Mojang	아케이드	샌드박스/프	Minecraft 는 블록으 로 만든 게임으로 상상할 수 있는 모든 것으로 변 신할 수 있습니다. 자원을 무 제한으로 사용할 수 있는 크리 에이티브 모드에서 플레이하 거나 서바 이별 모드 에서 도구	4 총	5,024,221 건

각 시기별 100개의 게임 데이터를 모아 데이터프레임으로
저장 후 훈련데이터로 활용.

3: 데이터 전처리

-이미지 증식

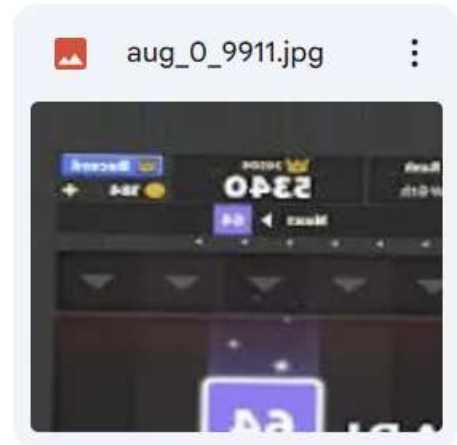
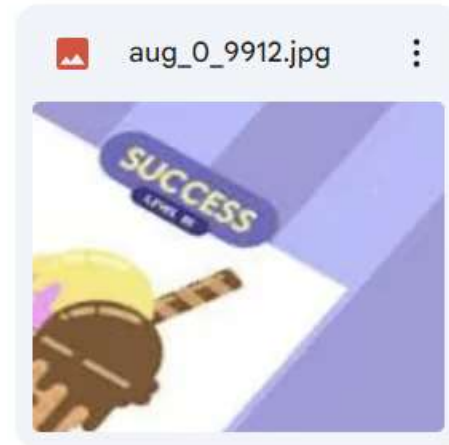
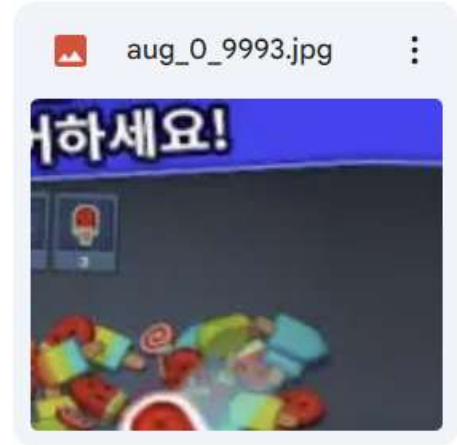
-카테고리 재분류



3-1 Image generator를 활용한 이미지 증강

```
# 데이터 증강을 위한 ImageDataGenerator 객체 생성
datagen = ImageDataGenerator(
    rotation_range=40,          # 이미지 회전 범위 (degrees)
    width_shift_range=0.2,      # 수평 이동 범위
    height_shift_range=0.2,     # 수직 이동 범위
    shear_range=0.2,           # 층밀리기 변환 범위
    zoom_range=0.2,            # 확대 축소 범위
    horizontal_flip=True,       # 수평 반전
    fill_mode='nearest'        # 빈 공간을 채우는 방식
)

# 이미지가 저장된 디렉토리 경로와 출력 디렉토리 경로
input_dir = '/content/퍼즐-20240529T081531Z-001/퍼즐'
output_dir = '/content/퍼즐-20240529T081531Z-001/퍼즐 테스트2'
```

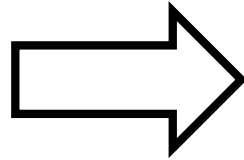


이미지를 변형한 복사본을 통해 자료를 늘렸다.

3-2 카테고리 재분류

BEFORE

퍼즐, 롤플레이팅, 액션
아케이드, 캐주얼 게임
시뮬레이션, 어드벤처, 보드, 전략
자동차 경주, 음악, 스포츠
교육, 퀴즈, 카드, 단어



AFTER

퍼즐, RPG(롤플레이팅 등), 전략
러닝(자동차 경주 등)
시뮬레이션, 퀴즈

한정된 데이터로 **16개 카테고리** 중 1개를 구별하는 것은 현실적으로 어렵다.

따라서 비슷한 카테고리끼리 묶어 클래스를 줄임.

4: 모델 제작 및 검증

-CNN

-Resnet

-VGG16



4-1. CNN

```
# 모델 구축
model = Sequential([
    Conv2D(8, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    #8 >> 16 >> 32
    MaxPooling2D(pool_size=(2, 2)),
    Dropout(0.5),
    Conv2D(16, (3, 3), activation='relu'), #16 >> 32 >> 64
    MaxPooling2D(pool_size=(2, 2)),
    Dropout(0.5),
    Conv2D(32, (3, 3), activation='relu'), #32 >> 64 >> 128
    MaxPooling2D(pool_size=(2, 2)),
    Dropout(0.5),
    Flatten(),
    Dense(128, activation='relu'), #128 >> 256 >> 512
    Dropout(0.5),
    Dense(7, activation='softmax')
])
```

```
=====  
Total params: 1190807 (4.54 MB)  
Trainable params: 1190807 (4.54 MB)  
Non-trainable params: 0 (0.00 Byte)
```

퍼셉트론 / 2

```
=====  
Total params: 4760615 (18.16 MB)  
Trainable params: 4760615 (18.16 MB)  
Non-trainable params: 0 (0.00 Byte)
```

기본 퍼셉트론

```
=====  
Total params: 19037255 (72.62 MB)  
Trainable params: 19037255 (72.62 MB)  
Non-trainable params: 0 (0.00 Byte)
```

퍼셉트론 * 2

퍼셉트론 값을 2배 높이고 반으로 나누는 식으로 조정하여 비교분석
RELU를 활용했으며 드롭아웃을 통해 과적합을 방지

4-1. CNN

	퍼셉트론 / 2	퍼셉트론(디폴트)	퍼셉트론 * 2
Image enerator 미사용	loss: 1.7381 accuracy: 0.3359 val_accuracy: 0.3229	loss: 1.9850 accuracy: 0.3672 val_accuracy: 0.4107	loss: 2.5471 accuracy: 0.4062 val_accuracy: 0.5878
Image generator 사용	loss: 1.7177 accuracy: 0.3149 val_accuracy: 0.3393	loss: 1.9329 accuracy: 0.3534 val_accuracy: 0.3750	loss: 2.9259 accuracy: 0.3125 val_accuracy: 0.3839

- 퍼셉트론이 늘어날 수록 정확도가 올라가는 한편 오차도 늘어나고 있다.
- 정확도가 가장 높으면서 상대적으로 오차가 적은 Image generator 사용하지 않고 퍼셉트론을 두 배로 높인 모델을 최적의 모델로 선정하였다.

4-2. Resnet

```
# 사전 학습된 ResNet50 모델 로드 (include_top=False로 최종 레이어 제외)
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(150, 150, 3))

# 사전 학습된 모델의 마지막 몇 개 레이어를 학습 가능하게 설정
for layer in base_model.layers[:143]: # ResNet50의 앞부분 레이어는 고정
    layer.trainable = False
for layer in base_model.layers[143:]:
    layer.trainable = True

# 새로운 레이어 추가
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(num_classes, activation='softmax')(x)
```

- PreTrained 모델인 Resnet50을 사용.

	퍼셉트론 / 2	퍼셉트론(디폴트)	퍼셉트론 * 2
Image generator 사용	loss: 1.4483 accuracy: 0.4775 val_accuracy: 0.3281	loss: 1.9825 accuracy: 0.3104 val_accuracy: 0.3063	loss: 1.4285 accuracy: 0.4752 val_accuracy: 0.3656

- 퍼셉트론을 2로 나눈 값과 2배로 곱한 모델이 비슷한 성능을 보이고 있음.
- 오차가 더 작은 후자가 최적의 모델로 보임

4-3. VGG16

```
# 사전 학습된 VGG16 모델 로드 (include_top=False로 최종 레이어 제외)
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(150, 150, 3))

# 새로운 레이어 추가
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(num_classes, activation='softmax')(x)
```

PreTrained 모델인 VGG16을 사용.

	퍼셉트론 / 2	퍼셉트론(디폴트)	퍼셉트론 * 2
Image generator 사용	loss: 1.3846 accuracy: 0.4952 val_accuracy: 0.4062	loss: 1.3972 accuracy: 0.4688 val_accuracy: 0.4028	loss: 1.3958 accuracy: 0.4639 val_accuracy: 0.3750

- 퍼셉트론 값을 2로 나눈 모델이 오차값이 가장 낮고 정확도가 높은 최적의 모델임.

5: 결과 및 개선할 점

-결과 및 총평

-개선할 점



5. 결과 및 총평

CNN	Resnet	VGG16
<pre>loss: 2.5471 accuracy: 0.4062 val_accuracy: 0.5878</pre>	<pre>loss: 1.4285 accuracy: 0.4752 val_accuracy: 0.3656</pre>	<pre>loss: 1.3846 accuracy: 0.4952 val_accuracy: 0.4062</pre>

결과적으로 정확도가 높으면서 오차가 작았던
VGG16모델이 가장 적합한 모델이라 할 수 있음.

5. 결과 및 총평

```
# 테스트할 이미지의 파일 경로 설정
image_path = '/content/rpg.png'

# 이미지 로딩 및 전처리
test_image = load_and_preprocess_image(image_path)

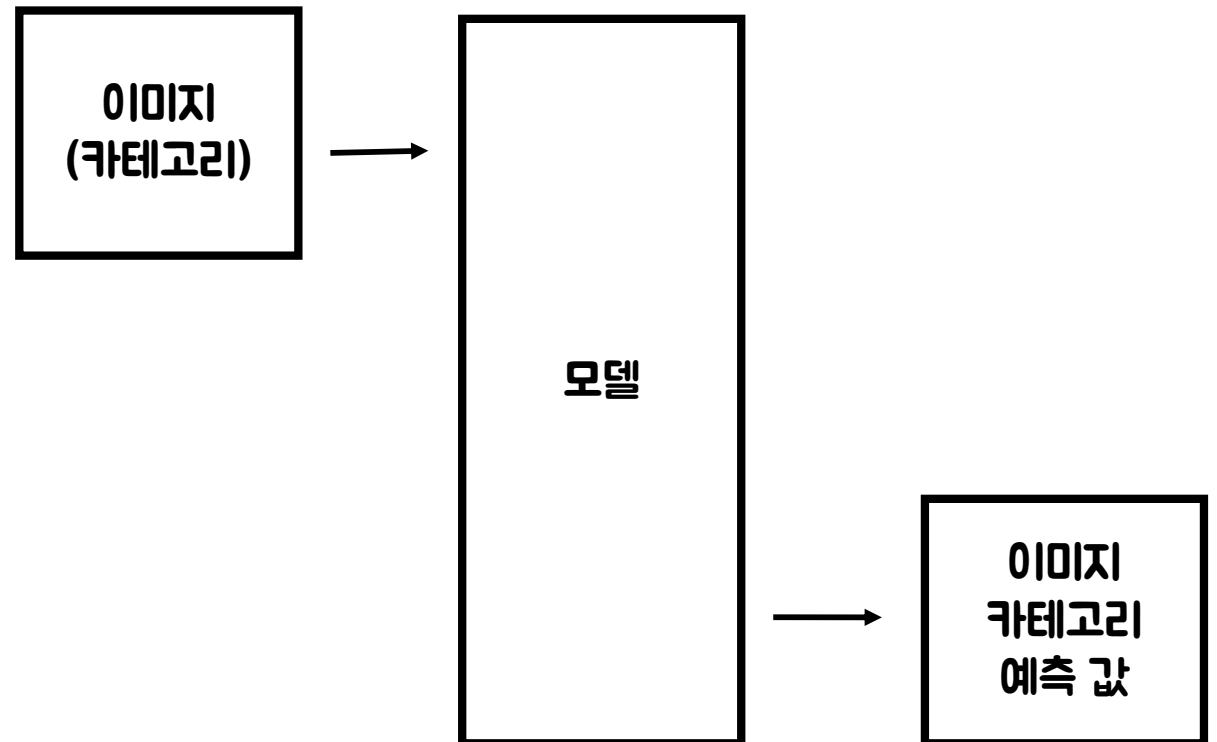
# 모델 로드
model = load_model('game_category_classifier.h5')

# 예측 수행
predictions = model.predict(test_image)
predicted_class = np.argmax(predictions)

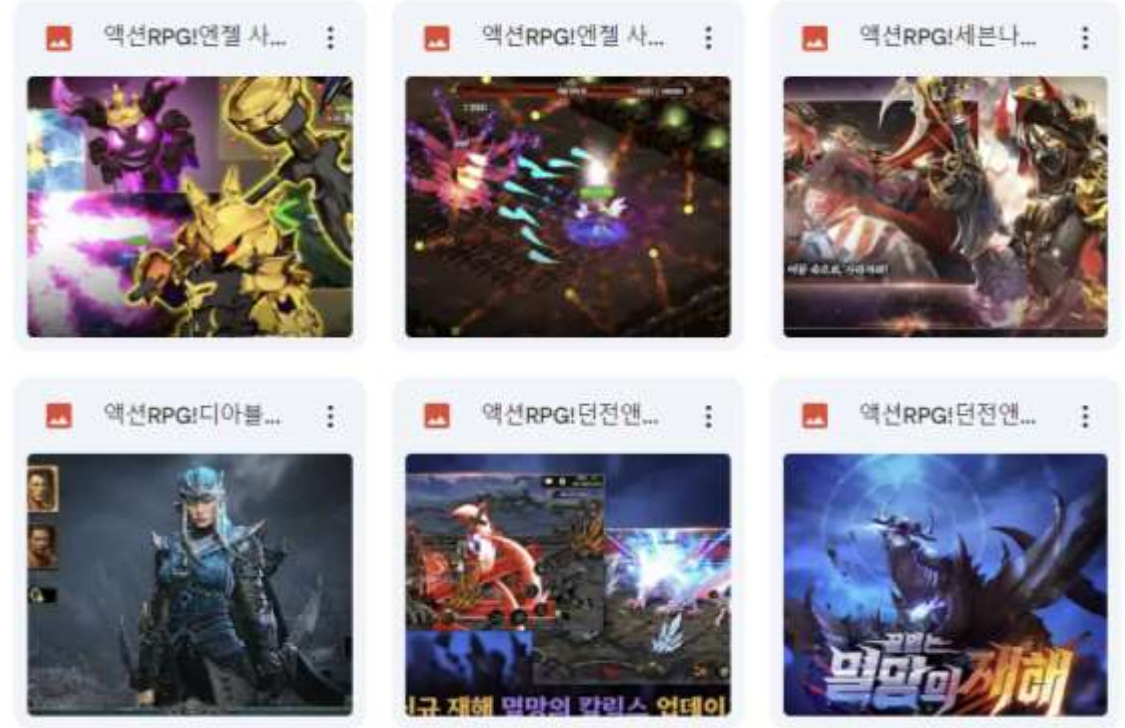
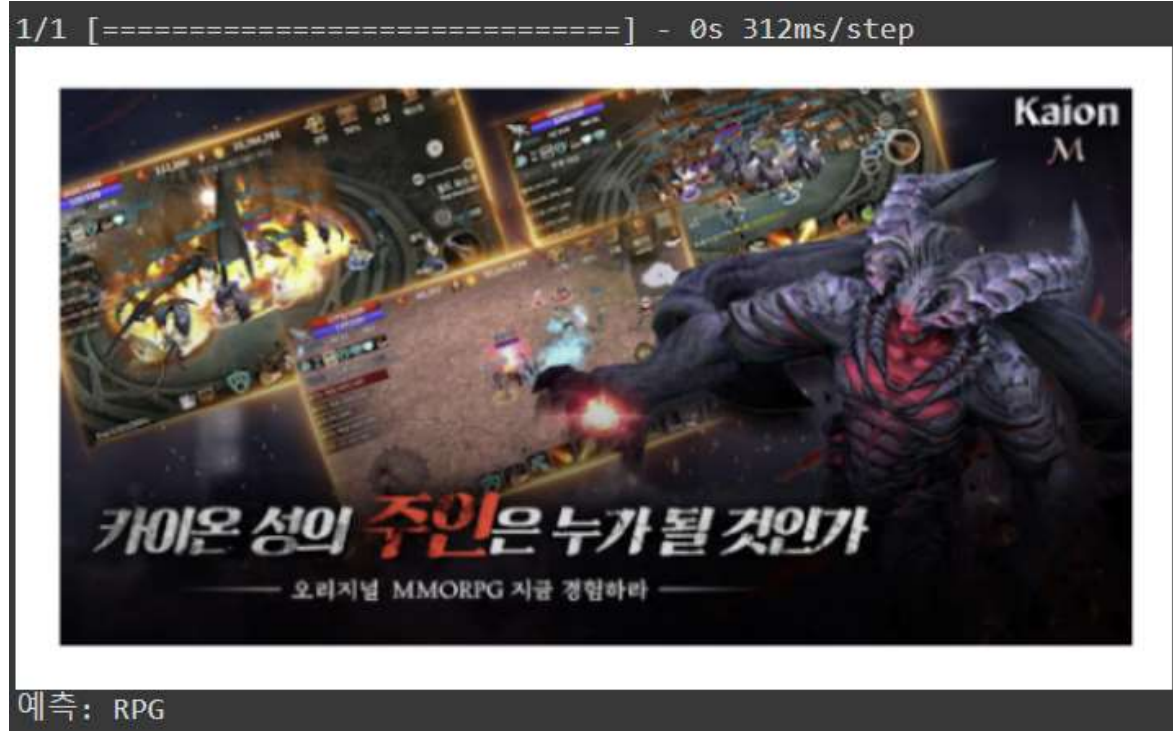
# 이미지 출력
img = load_img(image_path)
plt.imshow(img)
plt.axis('off') # 축 숨기기
plt.show()

# 예측 결과 출력
index = ['RPG', '러닝', '시뮬레이션', '전략', '퀴즈', '퍼즐']
print('예측: {}'.format(index[predicted_class]))
```

각 카테고리들의 학습 상황을 확인할 수 있는 프로그램을 별도로 마련했음.

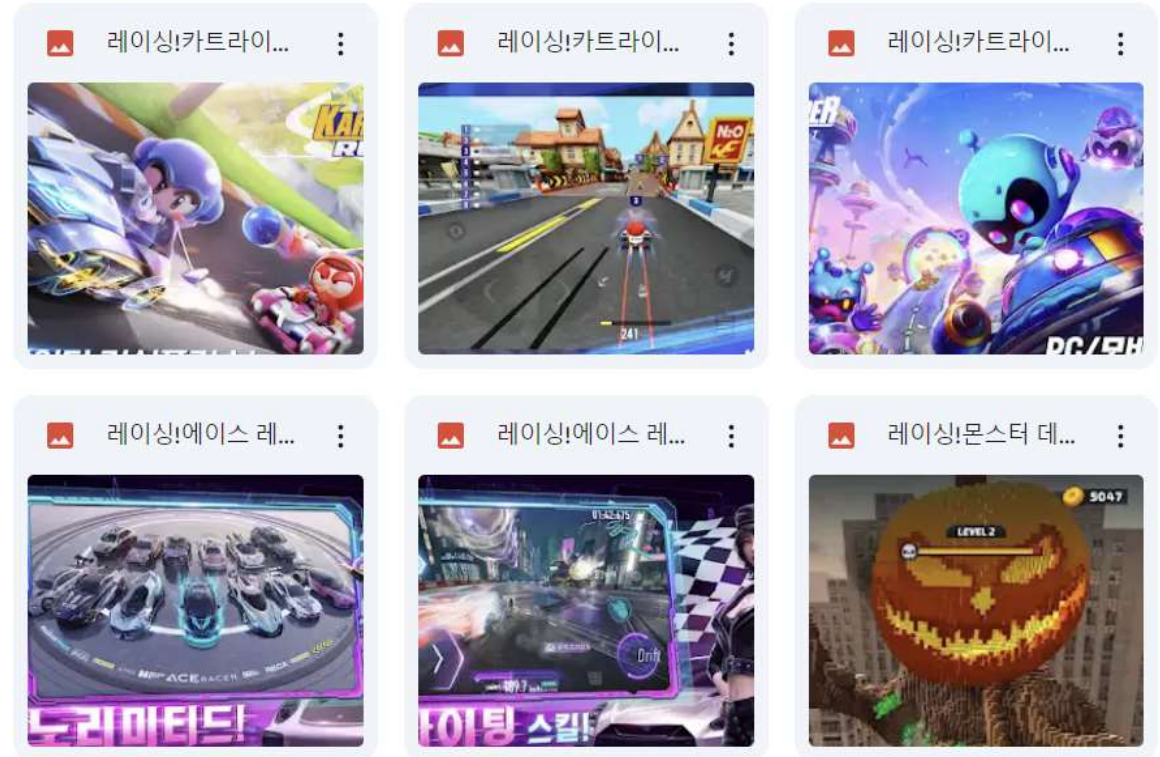


5. 결과 및 총평



퍼즐, 퀴즈, RPG의 경우 학습이 잘 되었음을 확인할 수 있었음.
이들의 경우 같은 카테고리 안의 이미지들 사이 유사성이 뚜렷한 것이 원인으로 보임.

5. 결과 및 총평

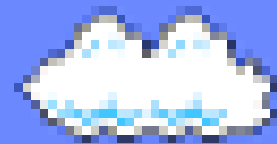


러닝, 시뮬레이션, 하이퍼캐주얼, 전략의 경우 학습이 잘 되었음을 확인할 수 있었음.
해당 카테고리의 이미지들 사이 유사성이 적은 것이 원인인것으로 보임.

5. 개선할 점

크롤링을 하는 사이트에서 확보할 수 있는 데이터의 수에 대한 한계가 많았다.
더 많은 데이터를 확보할 수 있는 사이트를 찾는데 더 많은 투자가 필요하다.

단순 이미지 만으로는 게임의 장르를 구별하는 데에는 한계가 있었다.
> 이미지를 벗어나 텍스트, 수치형 자료를 통해 더욱 개선시킬 가능성이 있다.



감사합니다!