

# 브랜드 Q스킬

상처 진단과 응급 처치를 위한 화상  
이미지 분류 AI

우영웅 왕국 조

우영웅, 김민규, 이재경, 김창민, 김주찬



## 목차

- 팀 소개
- 문제 인식
- 기술 구현
- 기대효과

# 우영웅 왕국 조

우영웅

기획, 총괄, 인공지능 개발

이재경

기획, 데이터 수집 및 분류

김민규

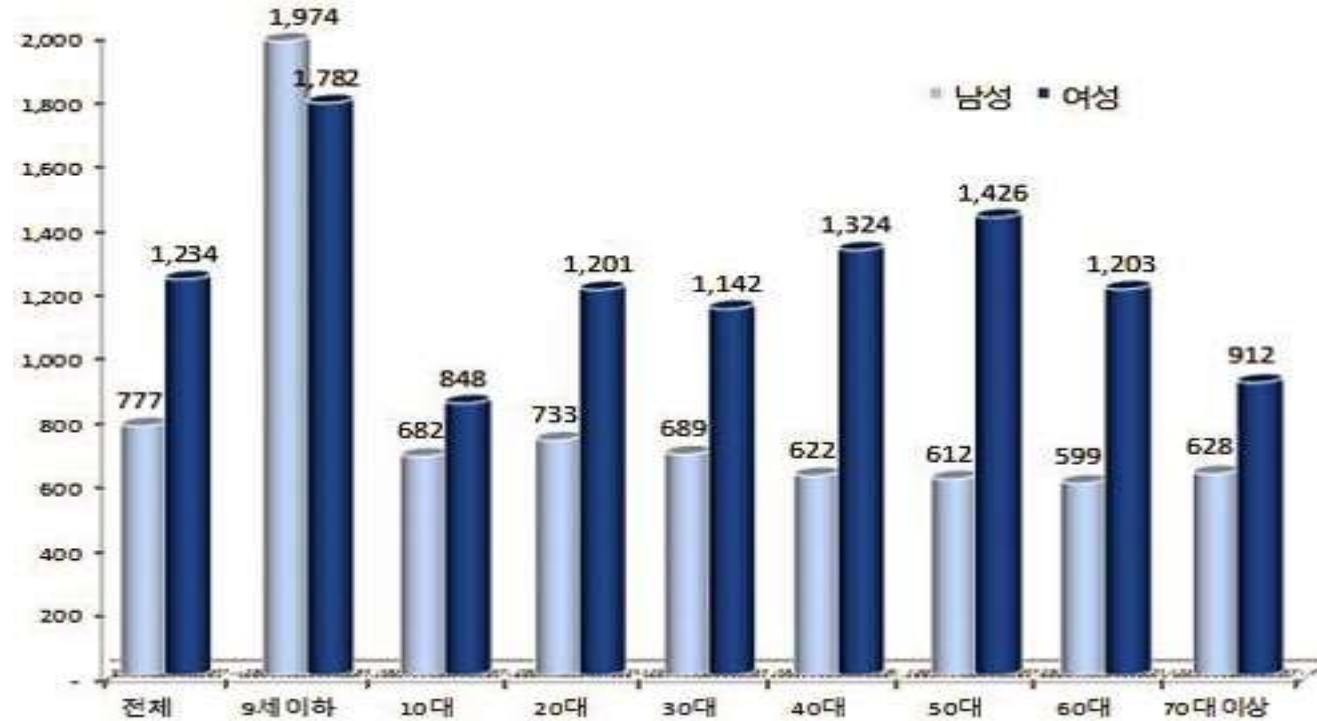
김창민

잡일

김주찬

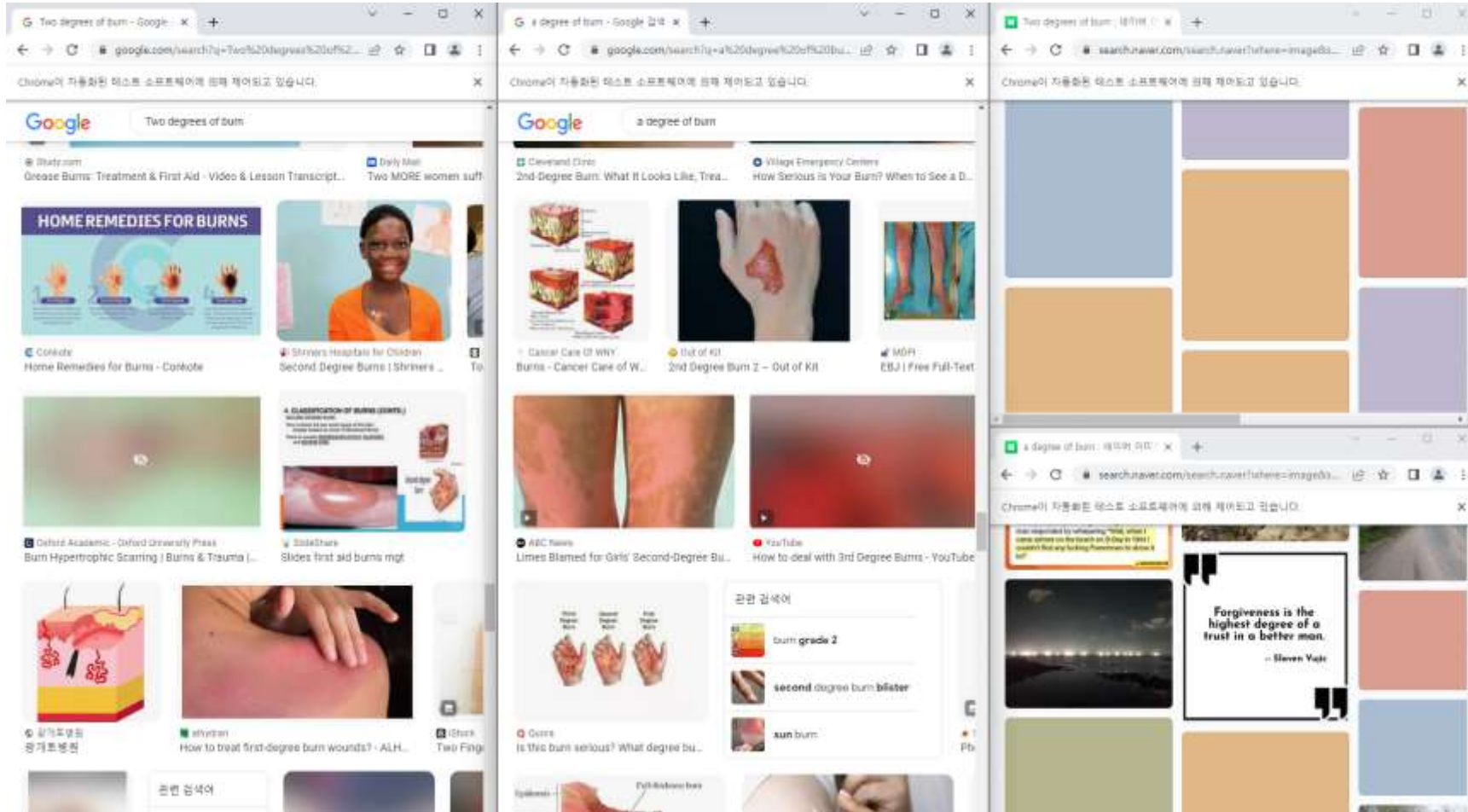


# 문제 인식



연령별 화상 피해자 나이 대

# 기술 구현 - 데이터 수집



```
def size(name):
    from PIL import Image
    import os

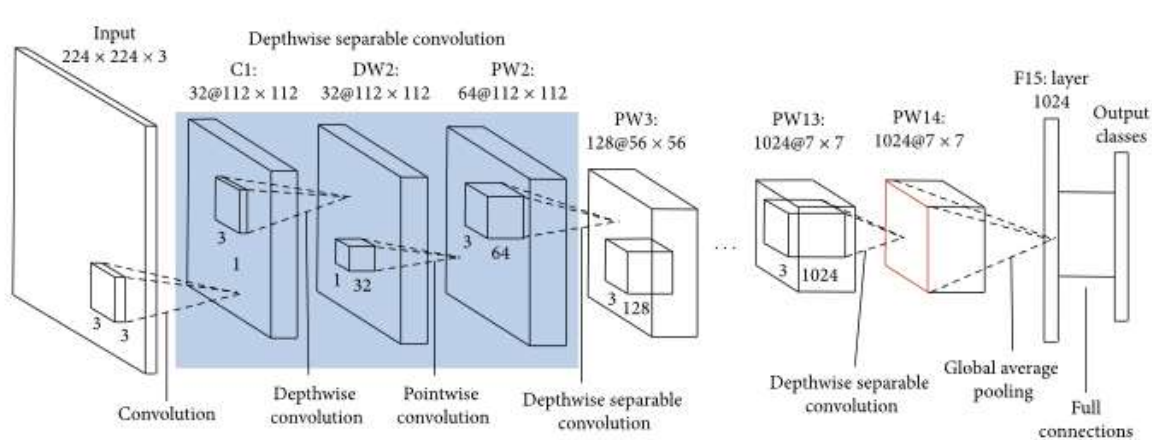
    def delete_small_images(directory):
        files_to_delete = []
        for filename in os.listdir(directory):
            file_path = os.path.join(directory, filename)
            if os.path.isfile(file_path):
                try:
                    with Image.open(file_path) as img:
                        width, height = img.size
                        if width <= 32 and height <= 32:
                            files_to_delete.append(file_path)
                except:
                    pass

        for file_path in files_to_delete:
            try:
                os.remove(file_path)
                print(f"Deleted {file_path}")
            except Exception as e:
                print(f"Error occurred while deleting {file_path}: {e}")

    # 이미지가 저장된 디렉토리 경로 지정
    image_directory = r"C:\Users\pon kj\Desktop\AutoCrawler-master\download\\" + name
    delete_small_images(image_directory)
```

AutoCrawler을 통한 데이터 수집 및 데이터 정리

# 기술 구현 – 이미지 분류 AI



```
base_model = MobileNet(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

model = Sequential()
model.add(base_model)
model.add(GlobalAveragePooling2D())
model.add(Dense(64, activation='relu'))
model.add(Dense(128, activation='selu'))
model.add(Dense(classes, activation='softmax'))
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

32/32 [=====] - 0s 14ms/step - loss: 0.2825 - accuracy: 0.9192

Evaluation loss: 0.2825028896331787

Evaluation accuracy: 0.9191616773605347

# 기술 구현 - 이미지 분류 AI

```
try {
    Model model = Model.newInstance(context);

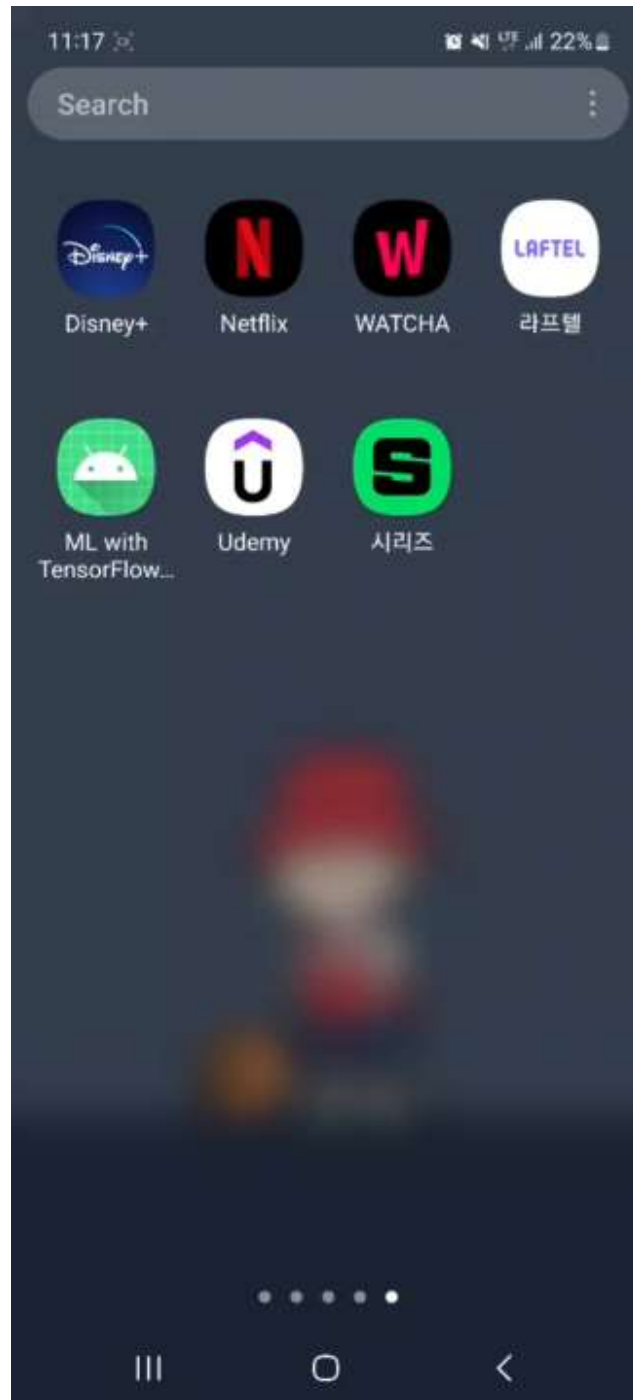
    // Creates inputs for reference.
    TensorBuffer inputFeature0 = TensorBuffer.createFixedSize(new int[]{1, 224, 224, 3}, DataType.FLOAT32);
    inputFeature0.loadBuffer(byteBuffer);

    // Runs model inference and gets result.
    Model.Outputs outputs = model.process(inputFeature0);
    TensorBuffer outputFeature0 = outputs.getOutputFeature0AsTensorBuffer();

    // Releases model resources if no longer used.
    model.close();
} catch (IOException e) {
    // TODO Handle the exception
}
```

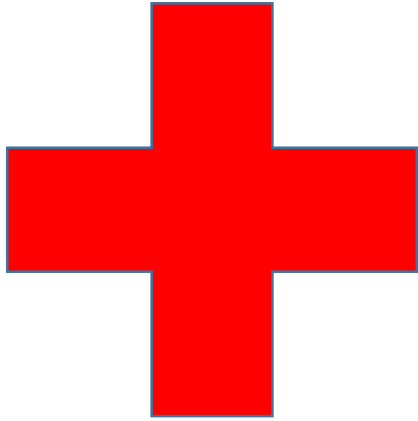
Android Studio를 사용

# 기술 구현

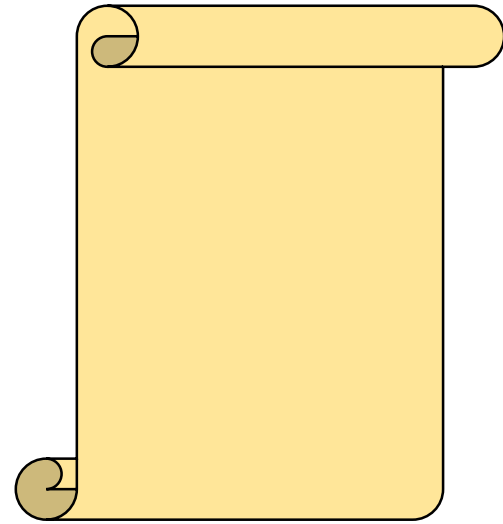




기대효과



조기 진단



치료 지침 제공

감사합니다