

연구윤리 및 연구지도

(11주차 보고서)

경북대학교 전자공학부
2016113566 김남영

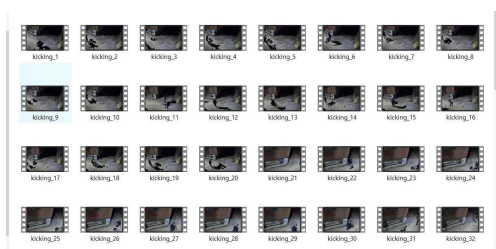
'LSTM을 활용한 CCTV 영상 속 범죄 상황 예측 모델' 구현하기

결론적으로 에러가 발생해서 완성하지는 못한 상태이지만 일주일 동안 어떻게 진행해왔는지 보고서로 작성하려 합니다.

1. 우선 AI HUB(<https://aihub.or.kr/aidata/139>)에서 이상행동 CCTV 영상 데이터 셋을 다운 받았습니다. 해당 데이터 셋에는 폭행, 싸움, 실신, 주취행동 등 12가지의 이상행동이 포함되어 있습니다. 하지만 12가지의 상황에 대해 한 번에 생각하긴 힘들 것 같아 우선 폭행에 대해서만 인식을 시켜보려 시도했습니다.



데이터 셋을 다운로드 받으면 위와 같이 약 5분짜리 영상과 영상에 대한 정보가 들어있는 xml 파일이 함께 다운로드 됩니다. 그리고 xml 파일을 열어보면 assault 내에도 kicking, pulling, punching 등 다양하게 행동이 분류되어 있다는 것을 알 수 있습니다. 처음에는 assault 영상을 잘 처리해서 입력으로 넣으면 되겠다고 생각했지만, punching, kicking 등 다양한 행동이 포함되어 있는 5분짜리 영상을 학습시키면 인식을 너무 떨어질 것 같다는 생각했습니다.



(punching만 따로 추출)

그래서 다음과 같이 xml 파일에 명시되어 있는 프레임 정보를 통해 punching과 kicking 영상을 추출하여 저장하였습니다. 다른 action도 있었지만 punching과 kicking에 포함되어 있거나 데이터의 개수가 너무 적거나 행동이 명확하지 않아 punching과 kicking만을 인식시키는 것을 목표로 했습니다.

```
main.py ✕
1 import keras
2 from keras import applications
3 from keras.preprocessing.image import ImageDataGenerator
4 from keras import optimizers
5 from keras.models import Sequential, Model
6 from keras.layers import *
7 from keras.callbacks import ModelCheckpoint, LearningRateScheduler, TensorBoard, EarlyStopping
8
9 import os
10 import cv2
11 import numpy as np
12 from sklearn.model_selection import train_test_split
13 from sklearn.metrics import *
14
15 data_dir = "./VideoFile/"
16 seq_len = 16
17
18 classes = ["punching", "kicking"]
19
20 def frames_extraction(video_path):
21     frames_list = []
22
23     vidObj = cv2.VideoCapture(video_path)
24     count = 1
25
26     while count <= seq_len:
27
28         success, image = vidObj.read()
29         if success:
30             if(count%16==0):
31                 print("ㄷ = ")
32                 image = cv2.resize(image, (224, 224))
33                 frames_list.append(image)
34                 count += 1
35             else:
36                 print("Defected frame")
37                 break
38
39     vidObj.release()
40     return frames_list
41
42 def create_data(input_dir):
43     X = []
44     Y = []
45
46     classes_list = os.listdir(input_dir)
47
48     for c in classes_list:
49         print(c)
50         files_list = os.listdir(os.path.join(input_dir, c))
51         for f in files_list:
52             frames = frames_extraction(os.path.join(os.path.join(input_dir, c), f))
53             if len(frames) == seq_len:
54                 X.append(frames)
```

```

56         y = [0] * len(classes)
57         y[classes.index(c)] = 1
58         Y.append(y)
59
60
61     X = np.asarray(X)
62     Y = np.asarray(Y)
63
64     return X, Y
65
66 X, Y = create_data("./VideoFile/")
67
68 X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, shuffle=True, random_state=0)
69
70 model = Sequential()
71
72 model.add(TimeDistributed(Conv2D(64, 5, padding='same', input_shape=(seq_len, 224, 224, 3))))
73 model.add(Activation('relu'))
74 model.add(MaxPooling2D(pool_size=3))
75 model.add(Dropout(0.2))
76
77 model.add(TimeDistributed(Conv2D(64, 5, padding='same')))
78 model.add(Activation('relu'))
79 model.add(MaxPooling2D(pool_size=3))
80 model.add(Dropout(0.2))
81
82 model.add(TimeDistributed(Flatten()))
83
84 model.add(LSTM(32))
85
86 model.add(Dense(32, activation="relu"))
87 model.add(Dropout(0.3))
88 model.add(Dense(2, activation="softmax"))
89
90 #model.summary()
91
92 opt = keras.optimizers.SGD(lr=0.001)
93 model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=["accuracy"])
94
95 #earlystop = EarlyStopping(patience=7)
96 #callbacks = [earlystop]
97
98 history = model.fit(x=X_train, y=y_train, epochs=40, batch_size=8, shuffle=True, validation_split=0.2)
99 model.summary()
100 y_pred = model.predict(X_test)
101
102 y_pred = np.argmax(y_pred, axis=1)
103 y_test = np.argmax(y_test, axis=1)
104
105 print(classification_report(y_test, y_pred))

```

제가 다양한 자료를 참고하며 구성해본 코드입니다. 처음에는 단순히 LSTM으로만 구성을 하려 했으나, CNN을 통해 먼저 특징 맵을 추출한 뒤 특징 맵을 LSTM의 입력으로 넣으면 video classification의 더 성능이 좋다는 자료를 보게 되어 CNN-LSTM 구조로 모델을 구성하였습니다.

처음 frame_extraction과 create_data를 통해 ./VideoFile/ 에 저장되어 있는 punching, kicking 영상을 프레임 단위로 추출한 뒤, 각 이미지를 keras로 구성한 CNN-LSTM 모델의 입력으로 넣는 방식입니다.

```

TypeError: Value passed to parameter 'input' has DataType uint8 not in list of allowed values: (float16, bfloat16, float32)

```

처음에는 메모리 초과 에러가 났었는데 지금 현재는 위와 같은 에러가 발생하고 있습니다. 아마 input shape를 고려하지 않아서 생긴 문제인 것 같습니다.

고찰

1. 이전에 예제를 풀어볼 때는 그냥 load_mnist 같은 명령어를 통해 알아서 데이터 셋이 정리가 되어 신경 쓸 부분이 없었는데, 실제로 프로젝트를 진행하다보니 데이터를 전 처리하는 과정이 가장 힘들다고 느꼈습니다.
2. keras를 통해 모델을 구성했는데 해당 함수들의 사용법을 제대로 이해하지 못해서 error가 발생한 것 같습니다.

3. 처음에는 전체 과정을 단순하게 바라봤는데 실제로 데이터를 구성하고 코드를 작성하다보니 생각보다 신경 쓸 부분이 너무 많다는 것을 알게 되었습니다....