

# BÁO CÁO ĐỒ ÁN CUỐI KỲ

Lớp: CS2225.CH1501

Môn: NHẬN DẠNG THỊ GIÁC VÀ ỨNG DỤNG

GV: PGS.TS Lê Đình Duy

Trường ĐH Công Nghệ Thông Tin, ĐHQG-HCM



# NHẬN DẠNG HÀNH ĐỘNG BẠO LỰC QUA VIDEO

CH1801024 - Phạm Đức Duy

CH1901005 - Đồng Đăng Khoa

CH1901025 - Võ Đình Nhã

**Github:** <https://github.com/nhavd/CS2225.CH1501>

**Youtube:** <https://youtu.be/mlStjbyaFzM>

# Tóm tắt

□ **Tên đề tài:** Nhận dạng hành động bạo lực qua video

□ **Tóm tắt về đề án**

- Ứng dụng: Kết quả bài toán được ứng dụng để truy xuất/ đánh giá các đoạn video có hành động bạo lực hay không hoặc cảnh báo hành vi bạo lực qua camera giám sát (surveillance camera).

- Sử dụng mô hình Deep Learning – Convolutional Neural Network (CNN) kết hợp phương pháp Long Short Term Memory (LSTM). Một mạng CNN đưa các input video (frame) và các output đặc trưng vào LSTM để học các đặc trưng global temporal, sau đó phân loại các đặc trưng bằng các mạng kết nối đầy đủ (fully connected layer). Network này không chỉ implement được bởi các pre-trained mode trong ImageNet database, mà nó rất linh hoạt để thực hiện trên các video có thời lượng bất định, và độ chính xác có thể lên đến 98,5% mặc dù xử lý hình ảnh theo thời gian thực.

# Tóm tắt

## □ Kết quả đạt được

Đánh giá được độ chính xác của giải thuật khi train với bộ dữ liệu video.

Hành động bạo lực trong video được phát hiện nhanh chóng

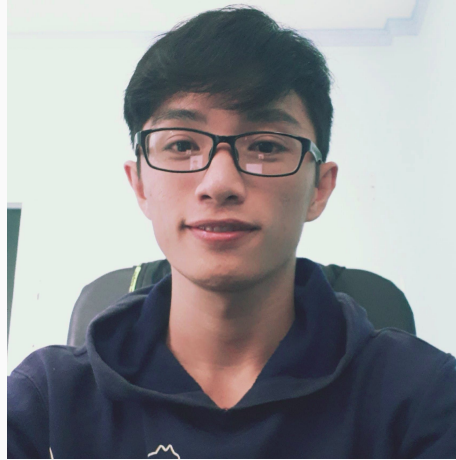


# Tóm tắt

## □ Ảnh của các thành viên của nhóm



CH1801024



CH1901005



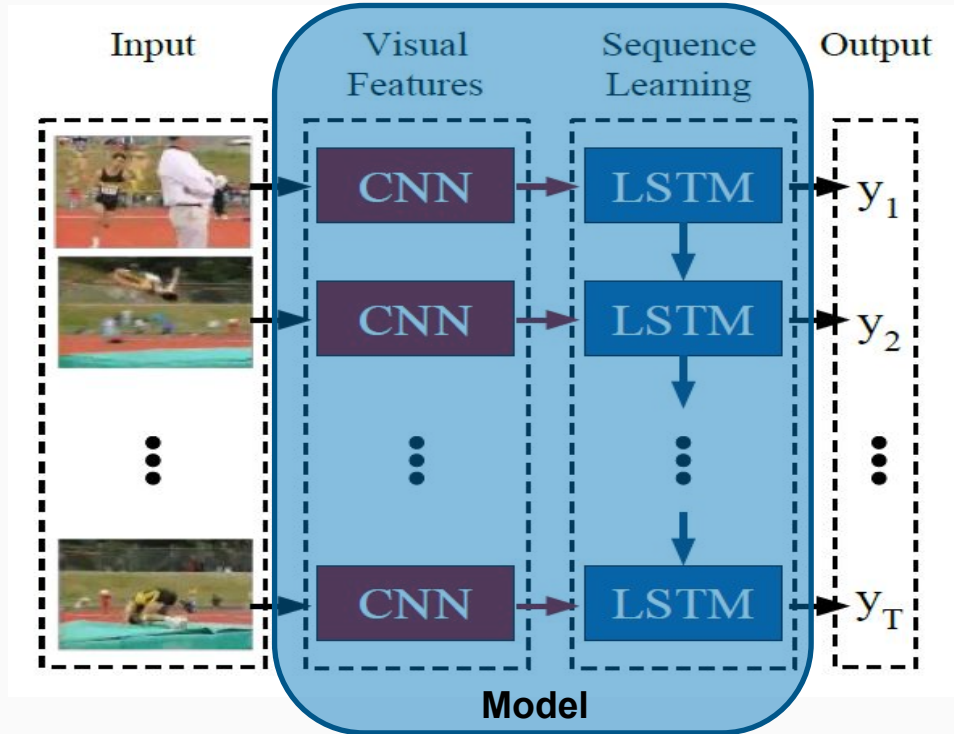
CH1901025

# Mô tả bài toán

- Input: Đoạn video
- Output: Nhãn thông báo video có chứa hành động bạo lực hay không
- Hướng tiếp cận: Phương pháp giải bài toán chỉ giới hạn ở việc nhận diện hành động bạo lực. Trong đó khi ảnh/video được upload lên thì các khung hình sẽ được trích xuất, đồng thời thuật toán trong mô hình huấn luyện sẽ quét qua tất cả các hình này và nhận diện xem có hành động bạo lực hay không?
- Giới hạn:
  - Số lượng đối tượng trong 1 khung ảnh  $< 10$ , ít chồng lấn
  - Tập dữ liệu huấn luyện còn hạn chế về số lượng

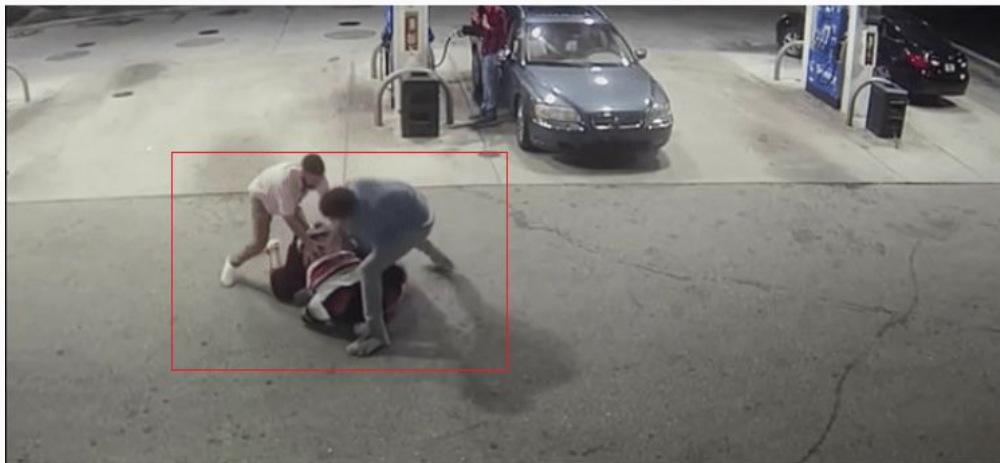
# Mô tả bài toán

- Minh hoạ



# Loại bài toán ML

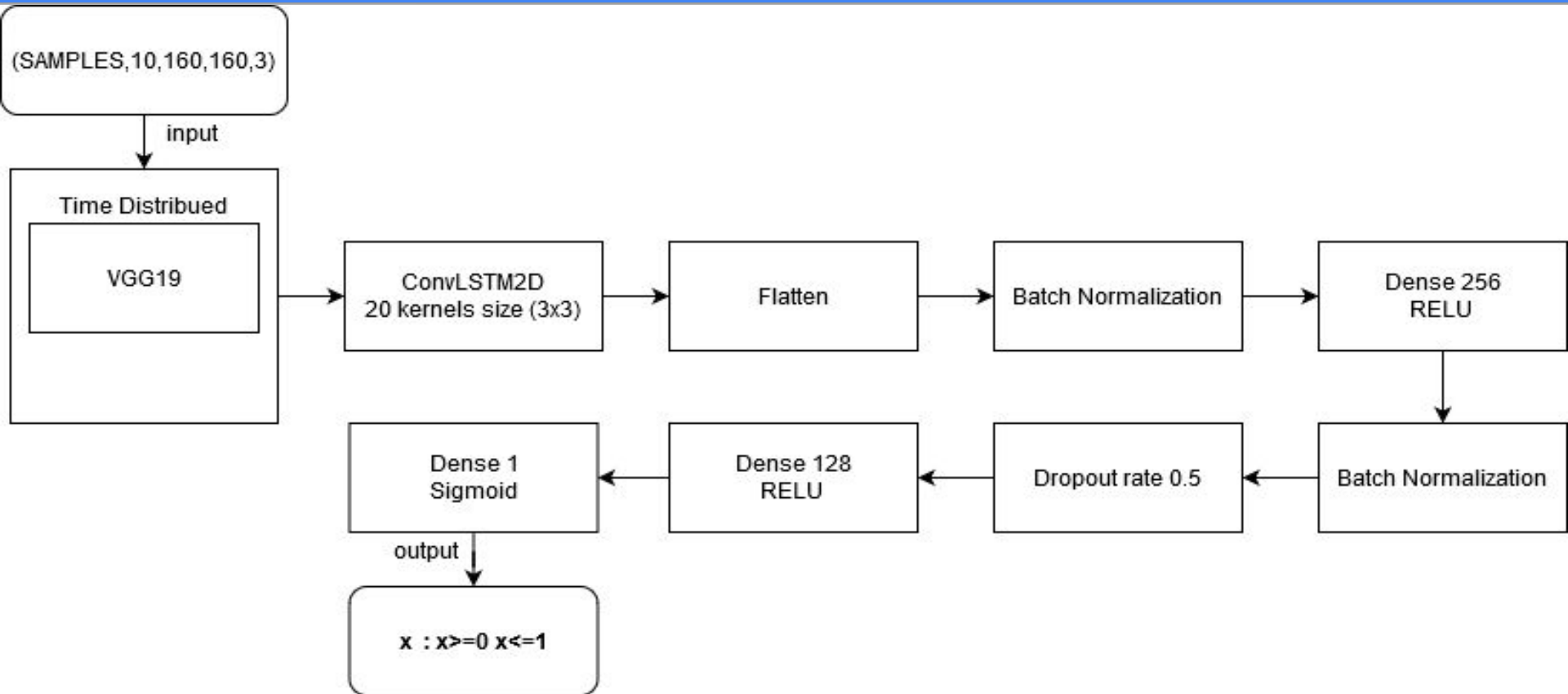
Bằng mô hình đã training, chúng ta có thể phân loại đâu là video có chứa hành động bạo lực, đâu là video bình thường và vị trí của chúng trong video.



## Object Detection



# Mô hình CNN - LSTM



# Dữ liệu

- Tổng số mẫu: 1201 videos + 30 videos
  - Training: 951 videos (475 violence, 476 non violence)
  - Testing: 250 videos (125 violence, 125 non violence)
  - Validation: 30 other sources videos (10 violence, 20 non violence)
- Cách thu thập
  - Nguồn dữ liệu: [Hockey Fight Dataset](#) và Movies Fight Detection
  - Số loại nhãn: 2
    - none\_violence
    - violence

# Thực thi

## Các thư viện chính cần import

```
from tensorflow.keras.applications import VGG19, ResNet50
from tensorflow.keras.layers import LSTM, Dense, TimeDistributed, Flatten, Dropout, GlobalAveragePooling2D, GlobalAveragePooling1D, Input
from tensorflow.keras.layers import Reshape, BatchNormalization, ConvLSTM2D
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam, RMSprop, SGD
```

## Sử dụng 2 mạng CNN: VGG19, RESNET

```
def CNN_LSTM(model_arch, opt, retrain, init_weights = 'imagenet'):

    input_shapes=(NUMBER_OF_FRAMES, IMAGE_SIZE, IMAGE_SIZE, 3)
    #np.random.seed(1234)
    if model_arch is 'Vgg19':
        base_model=VGG19(include_top=False, weights= init_weights, input_shape=(IMAGE_SIZE, IMAGE_SIZE, 3))
    elif model_arch is 'Resnet':
        base_model = ResNet50(include_top=False, weights=init_weights, input_shape=(IMAGE_SIZE, IMAGE_SIZE, 3))
```

# Thực thi

## Sử dụng các Optimization method cho LSTMs

```
if opt is 'Adam':  
    opti = Adam(lr=0.00001)  
elif opt is 'Sgd':  
    opti = SGD(lr=0.00001)  
elif opt is 'RMSprop':  
    opti = RMSprop(lr=0.00001)
```

## Mô hình kết hợp

```
vgg_adam_retrain = CNN_LSTM('Vgg19', 'Adam', True)  
vgg_adam_retrain_history = vgg_adam_retrain.fit(fvideo, flabels, batch_size=10, epochs=20, validation_data=(tvideo, tlabels))
```

Model `diff_vgg_rmsprop_retrain.save("./drive/MyDrive/Colab Notebooks/model/ViolenceDetection_RootDataset.h5")`

## Kết quả nhận dạng

```
def Violence_detection(videoname):  
    vid, img = extract_frames(videoname)  
  
    violence_res = ViolenceDetection(vid)  
    if violence_res > 0.6:  
        print("Violence Detected")  
    else:  
        print("No Violence Detected")
```

```
Violence_detection('fi001.mp4')  
  
Violence Detected
```

# Kết quả

```
def Violence_detection(videoname):  
    vid, img = extract_frames(videoname)  
  
    violence_res = ViolenceDetection(vid)  
    if violence_res > 0.6:  
        print("Violence Detected")  
    else:  
        print("No Violence Detected")
```

```
Violence_detection('fi001.mp4')
```

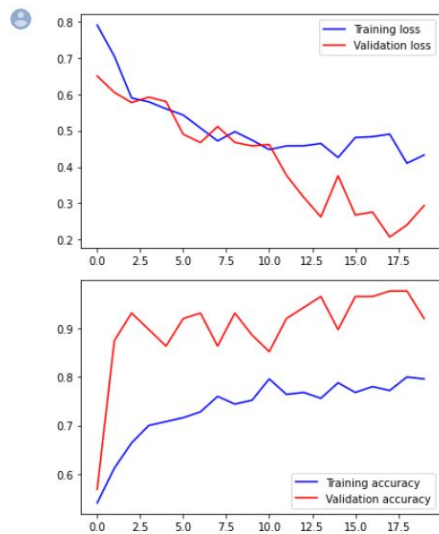
```
Violence Detected
```



# Phương pháp đánh giá

Sử dụng lần lượt các mạng CNN và optimization method với nhau

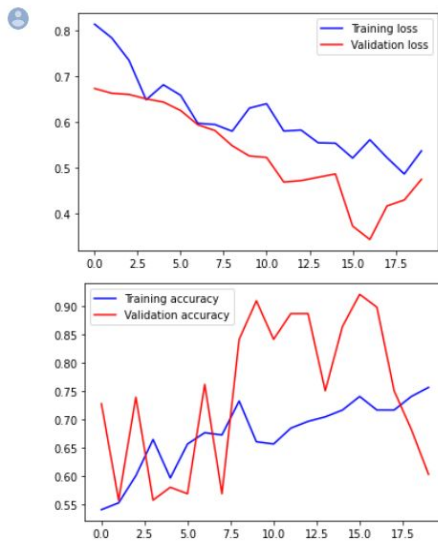
plot\_results(vgg\_adam\_retrain\_history)



```
[ ] val_results(vgg_adam_retrain)
```

```
count: 91
len(hockey_labels): 100
Count2: 10
len(movies_labels): 10
```

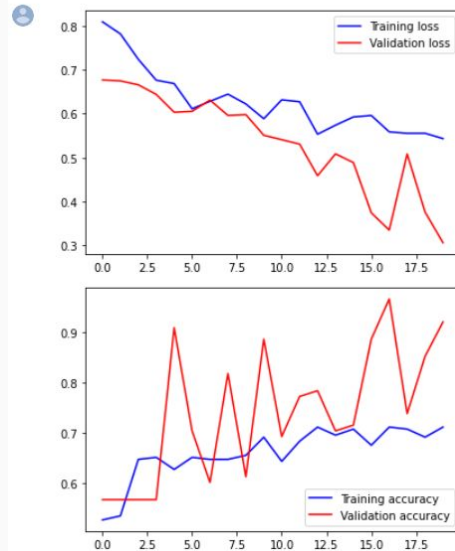
plot\_results(resnet\_adam\_static\_history)



```
[ ] val_results(resnet_adam_static)
```

```
count: 100
len(hockey_labels): 100
Count2: 3
len(movies_labels): 10
```

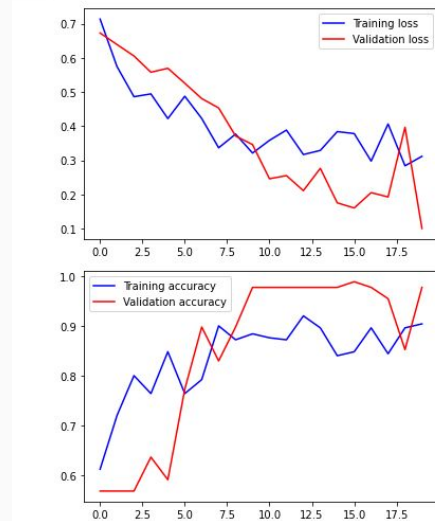
plot\_results(resnet\_rmsprop\_static\_history)



```
[ ] val_results(resnet_rmsprop_static)
```

```
count: 88
len(hockey_labels): 100
Count2: 7
len(movies_labels): 10
```

plot\_results(diff\_vgg\_rmsprop\_retrain\_history)



```
[ ] val_results(diff_vgg_rmsprop_retrain)
```

```
count: 98
len(hockey_labels): 100
Count2: 10
len(movies_labels): 10
```