ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

---ꕥ🕮ꕥ---

MÁY HỌC – MACHINE LEARNING

Đề tài:

TÓM TẮT VIDEO TRẬN ĐẤU BÓNG ĐÁ

Giảng viên hướng dẫn: TS. **Lê Đình Duy**
Ths. **Phạm Nguyễn Trường An**

Thành viên:

- **Nguyễn Đức Thắng - 19522206**
- **Nguyễn Xuân Minh – 19521848**
- **Nguyễn Tam Điệp - 19521360**

Lớp: **CS114.M11**

THÀNH PHỐ HỒ CHÍ MINH – NĂM 2022

Nội dung

I. Giải trình chỉnh sửa sau vấn đáp:

- Sau buổi báo cáo đồ án, nhóm chúng em tiến hành tìm thêm bộ dữ liệu để đánh giá kết quả của các mô hình mà nhóm đã đào tạo. Phần này sẽ được nhóm chúng em trình bày ở Kết quả thực nghiệm.

II. Tổng quan về đề tài

1. Ngữ cảnh ứng dụng

Do sự gia tăng phổ biến của các trang web phát video trực tuyến như youtube, các đài truyền hình, số lượng video về bóng đá nói riêng và về thể thao nói chung ngày càng tăng lên theo cấp số nhân. Tuy nhiên, nhiều người lại không có thời gian (hoặc không có nhu cầu) để xem chi tiết hết một trận bóng đá 90 phút hơn đó, do đó, nhóm muốn tìm cách để tóm tắt video dài thành một video ngắn hơn để tiết kiệm thời gian cũng như công sức của người xem, bên cạnh đó cũng không làm mất đi những chi tiết quan trọng.

2. Xác định bài toán

Bài toán đặt ra: tóm tắt diễn biến chính của một trận video bóng đá 90 phút hơn.

Trong đó, các diễn biến chính được tóm tắt sẽ gồm có:

- Các tình huống ghi bàn.
- Các tình huống suýt ghi bàn (nguy hiểm đến khung thành).
- Các tình huống đá phạt góc.
- Các tình huống phản công nhanh (tấn công dồn dập).

Input:

- Video một trận bóng đá muốn tóm tắt.
- Độ phân giải tối thiểu của video: 480p.

Output:

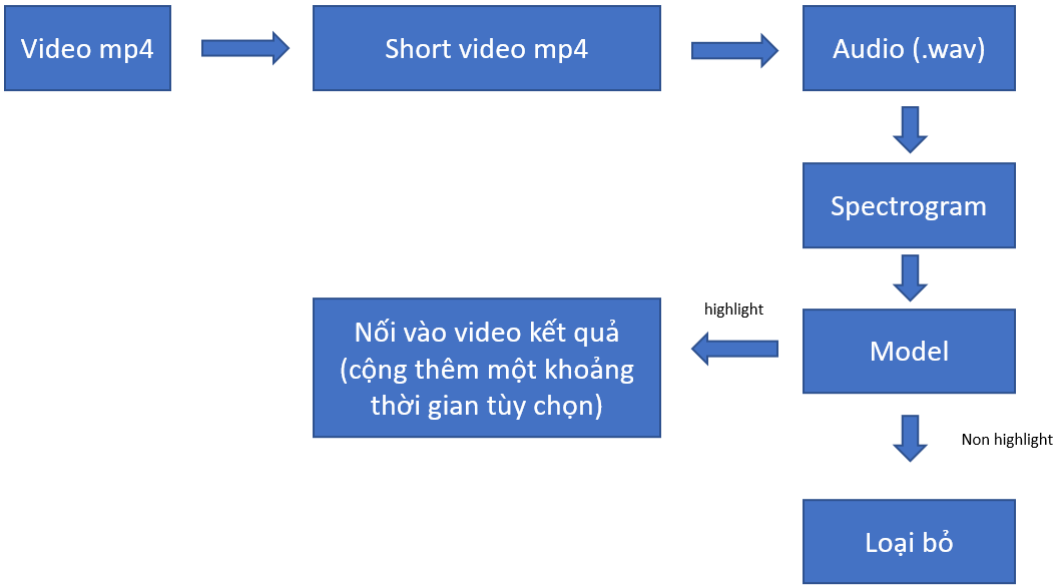
- Video input đã được tóm tắt (chỉ giữ lại những diễn biến chính của trận đấu).

3. Hướng tiếp cận cho bài toán

Một trong những phần quan trọng nhất của bài toán máy học là dữ liệu (data), Ở trong đồ án lần này, sau khi xây dựng một số lượng dữ liệu, nhóm chúng em tiến hành đào tạo (training) trên hai phương diện là audio và khung hình (frame) trên các mô hình chuyển tiếp (SVM, DenseNet) và mô hình tuần tự (LSTM).

Do đó, quá trình phân loại của nhóm như sau:

Quá trình tạo video tóm tắt:



III. Thu thập dữ liệu

1. Quá trình thu thập dữ liệu

Dữ liệu thô gồm các video nhóm bọn em thu thập thủ công từ youtube, từ các kênh của fifa, từ các kênh của các đội bóng, câu lạc bộ.

Sau khi có dữ liệu thô là các video toàn bộ trận đấu (full match), nhóm chúng em tiến hành chia ra xem và tách ra những đoạn (tầm 5-10s) được xem là diễn biến chính (như đã được đề cập ở trước). Nhưng trong khi tách thì bọn em nhận ra được là các đoạn mà bọn em quan tâm (diễn biến chính) của 1 video 90 phút hơn là rất ít. Ban đầu, chúng em tách được 1396 video non-highlight (không phải diễn biến chính) và 81 video highlight (diễn biến chính). Chính sự chênh lệch data giữa hai class quá lớn đã dẫn đến overfiting cho model. Vì vậy chúng em tiến hành thu thập thêm các video highlight ở youtube, sau đó định vị chính xác và cắt nhỏ ra từng video có độ dài 5-10s để bổ sung vào tập highlight, tránh bị chênh lệch dữ liệu dẫn đến việc đào tạo không đem lại hiệu quả cao.

2. Khó khăn của việc thu thập dữ liệu

Mặc dù các video bóng đá đã có sẵn rất nhiều trên internet, nhưng vẫn có một số khó khăn sau đây mà nhóm em đã gặp phải.

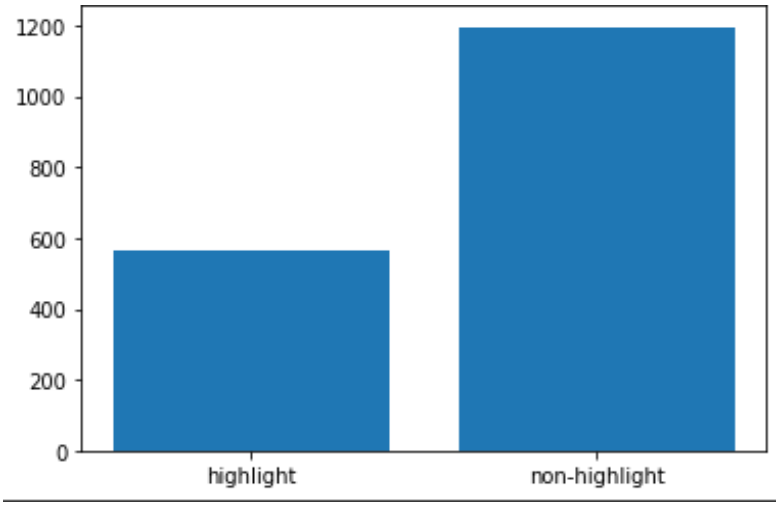
- Việc thu thập các video highlight tốn rất nhiều thời gian, phải xem những video dài chi tiết, sau đó mới có thể tìm ra được những đoạn nhỏ (Trung bình 1 video hơn 100p chỉ tìm được khoảng 2-3p highlight).
- Việc thu thập data highlight phải đa dạng ở nhiều góc độ.

- Các video thu thập phải chọn lọc kỹ lưỡng, tránh các video có gắn quảng cáo ở màn hình.

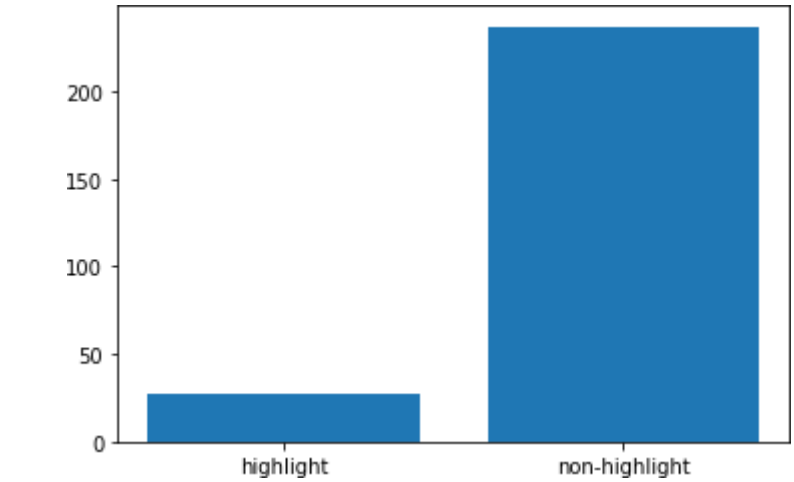
3. Tổng quan về bộ dữ liệu

Sau khi trải qua nhiều lần làm lại bộ dữ liệu (do không tương thích với mô hình) nhóm đã tạo dựng được cho mình 2 bộ dữ liệu tương ưng với 2 hướng xử lí của nhóm (theo audio và theo frame).

- Bộ đầu tiên gồm: 1196 video non-highlight, 563 video highlight (gồm các cảnh quan trọng), tất cả đều có độ dài cố định là 10s.
  - Bộ data train



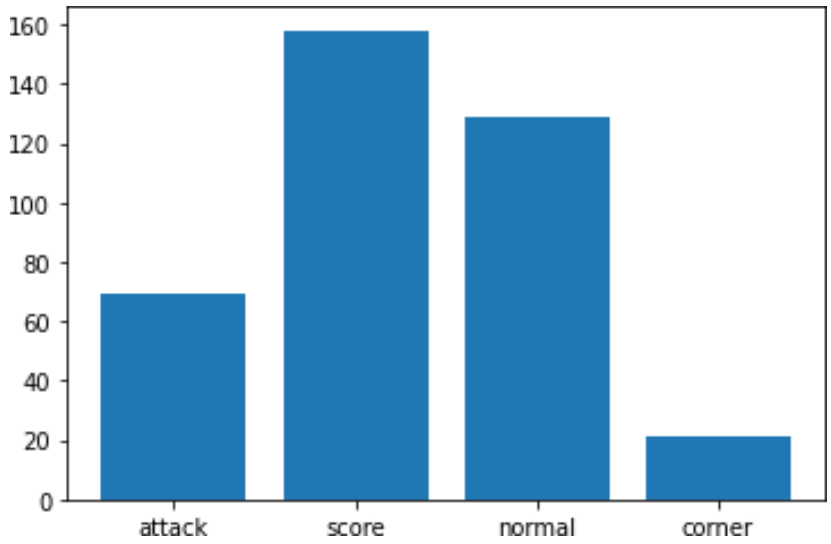
  - Bộ data test



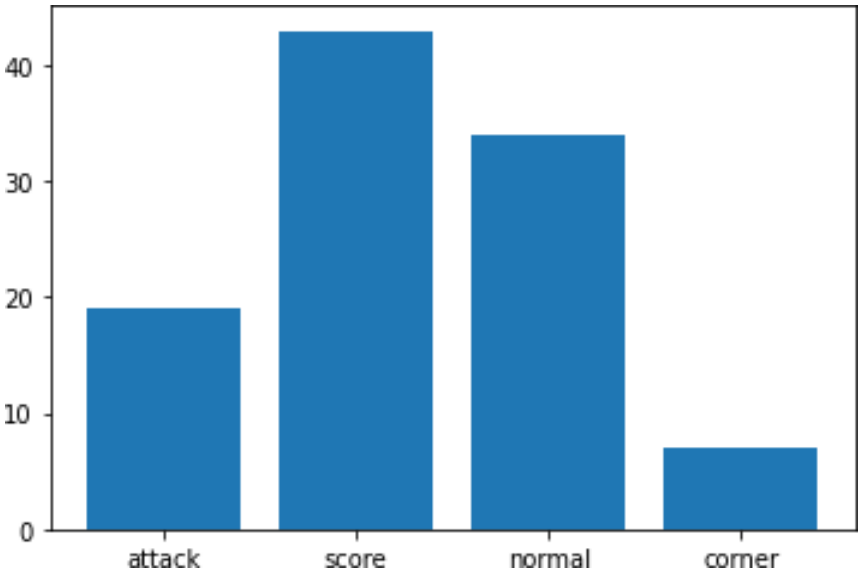
- Bộ thứ hai gồm: 480 video với độ dài ở trong khoảng 3-6s. Các video này chỉ cắt đúng tình huống ghi bàn, tấn công nguy hiểm mà bình luận viên hét lên hoặc tiếng khán giả sôi động lên. Ở bộ này, để giảm nguy cơ overfitting, đối với mỗi video, nhóm chỉ lấy 5-7 tình huống highlight và không phải highlight. Cụ thể là gồm 201 video có tình huống ghi bàn, 163 video tình huống bình thường, 28 video tình

huống phạt góc và 88 video tình huống tấn công nguy hiểm. Được chia làm 2 bộ train, test như sau:

- Train: attack (69), score (158), normal (129), corner (21)



- Test: attack (19), score (43), normal (34), corner (7)



IV. Tiền xử lý dữ liệu

1. Trích xuất audio của từng video

Ý tưởng của việc sử dụng đặc trưng âm thanh để phân loại là do nhóm nhận thấy trong các trận đấu bóng đá, các tình huống chính của trận đấu (ghi bàn, tấn công nguy hiểm, …) thì các bình luận viên thường bình luận to (có thể hét lên) và liên tục hoặc là tiếng khán giả sôi nổi hơn bình thường.

Để sử dụng được đặc trưng âm thanh này, nhóm sẽ chuyển video (đuôi ".mp4") thành file âm thanh (đuôi ".wav"). Sau đó, để tiện cho việc huấn luyện mô hình, nhóm tiếp tục chuyển đổi file âm thanh thành các biểu đồ spectrogram[1] .

2. Resize ảnh

Vì bộ dữ liệu được thu thập từ những nguồn khác nhau, để đảm bảo chuẩn input đầu vào cho các mô hình học sâu, nhóm quyết định resize ảnh nhỏ hơn để việc đào tạo được nhanh hơn, đồng thời thống nhất được một input đầu vào cho mô hình là kích thước 224x224.

3. Kéo dãn audio

Kéo dán audio sẽ giúp các đặc trưng được thể hiện rõ ràng hơn, làm rõ hơn các đặc trưng của mỗi lớp. Cụ thể là sẽ kéo dài các khoảng mà bình luận viên nói lớn, làm nổi bật cho lớp highlight, kéo dài khoảng bình thường sẽ làm nổi bật lớp non-highlight.

4. Lọc tiếng ồn

Trong video input, có nhiều tình huống camera lia đến khán giả, tiếng hò reo rất lớn, nhưng đây không phải là diễn biến chính mà nhóm hướng đến, nên nhóm sẽ tiến hành lọc bớt tiếng ồn để các đặc trưng được trở nên rõ ràng hơn.

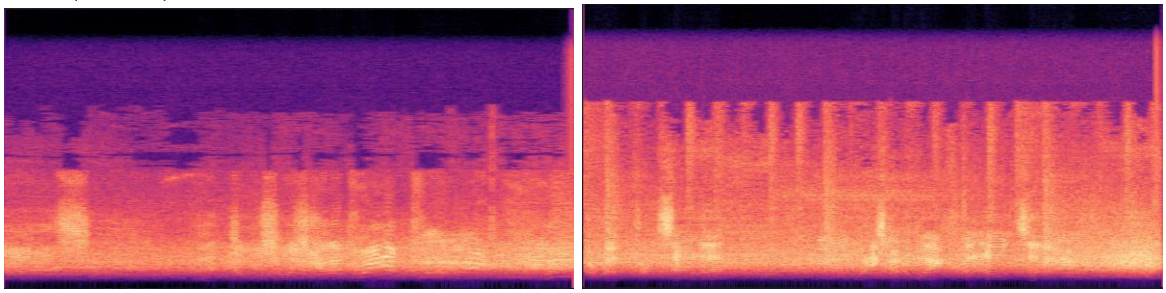
V. Huấn luyện mô hình & đánh giá

1. Sử dụng các đặc trưng về âm thanh

Như đã trình bày ở trên, nhận thấy khi tình huống càng hấp dẫn, bình luận viên, khán giả có xu hướng nói càng lớn. Âm càng to dẫn đến màu biểu thị trên spectrogram càng sáng, đây chính là đặc trưng mà nhóm chúng em hướng đến để phân loại.

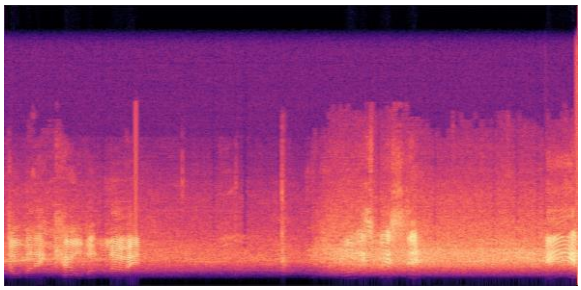
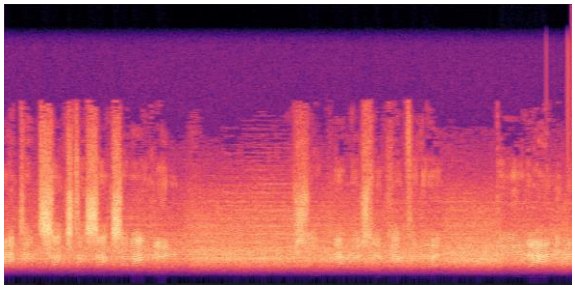
Visualize:

- Ghi bàn (Score):

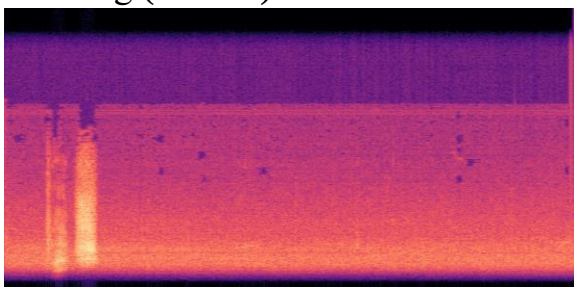
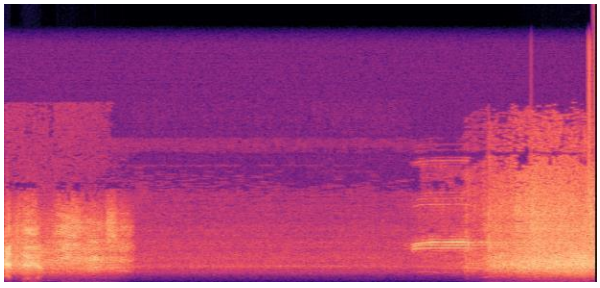


- Tấn công (Attack)

[1] Biểu đồ spectrogram là một cách trực quan để biểu thị cường độ tín hiệu hoặc độ lớn của tín hiệu theo thời gian ở các tần số khác nhau hiện diện trong một dạng sóng cụ thể. Spectrogram dùng các màu khác nhau để biểu thị cường độ của mỗi tần số (màu càng sáng thì năng lượng càng cao).

- Bình thường (normal)

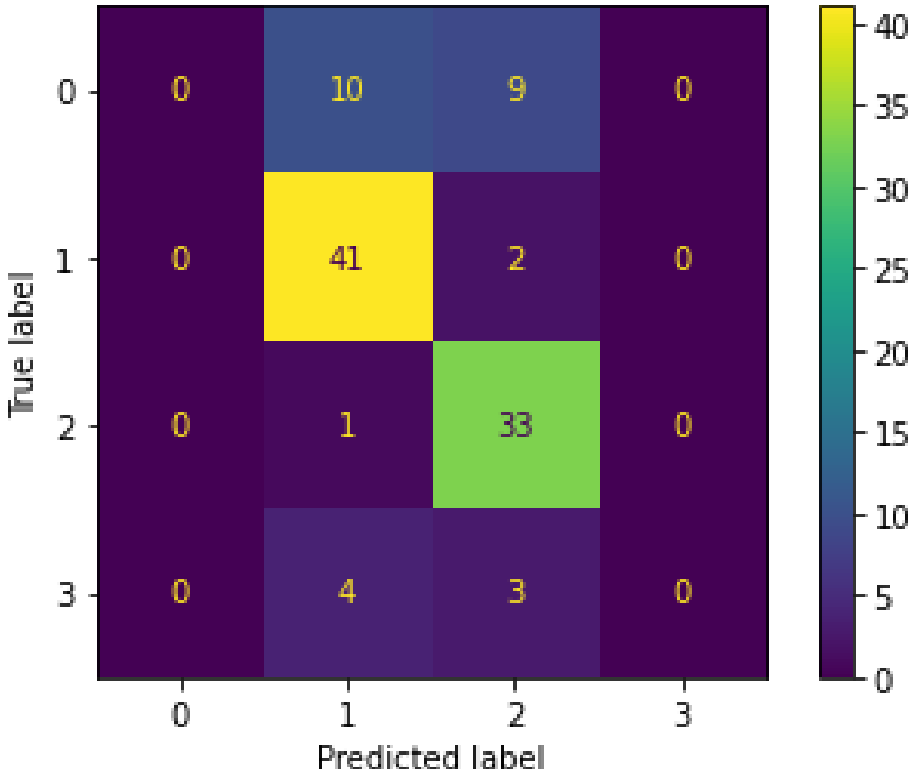
Sau khi đã trích xuất được những đặc trưng cần thiết, nhóm chúng em tiến hành đào tạo mô hình trên SVM và DenseNet-169.

- SVM là một thuật toán học có giám sát, thường được sử dụng trong các bài toán phân loại. Đây là một trong những thuật toán cổ điển, ban đầu được tìm ra bởi Vladimir N. Vapnik và dạng chuẩn hiện nay sử dụng soft margin được tìm ra bởi Vapnik và Corinna Cortes năm 1995.
- DenseNet là một mô hình học sâu, thường được sử dụng trong các bài toán nhận diện hình ảnh. Được ra mắt vào năm 2016 và là một phiên bản nâng cấp của ResNet. DenseNet nhận diện rất tốt với các bài toán nhận diện hình ảnh, khi mô hình này đã out performance các mô hình khác trước đó như ResNet, VGG16 khi đánh giá trên tập ImageNet.

Dưới đây là kết quả khi tiến hành đào tạo và đánh giá dữ liệu trên SVM và DenseNet-169.

Với thuật toán SVM

- Accuracy trên tập test: 0.7184
- Accuracy trên tập train: 0.6976
- Confusion matrix:



- Classification report:

	precision	recall	f1-score	support
attack	0.00	0.00	0.00	19
score	0.73	0.95	0.83	43
normal	0.70	0.97	0.81	34
corner	0.00	0.00	0.00	7
accuracy			0.72	103
macro avg	0.36	0.48	0.41	103
weighted avg	0.54	0.72	0.61	103

Với mô hình DenseNet-169:

- Kết quả evaluate trên tập test sau khi train với 30 epochs: 0.51

- Classification report:

	precision	recall	f1-score	support
attack	0.27	0.74	0.40	19
score	0.00	0.00	0.00	7
normal	0.88	0.65	0.75	34
corner	0.80	0.19	0.30	43
accuracy			0.43	103
macro avg	0.49	0.39	0.36	103
weighted avg	0.68	0.43	0.45	103

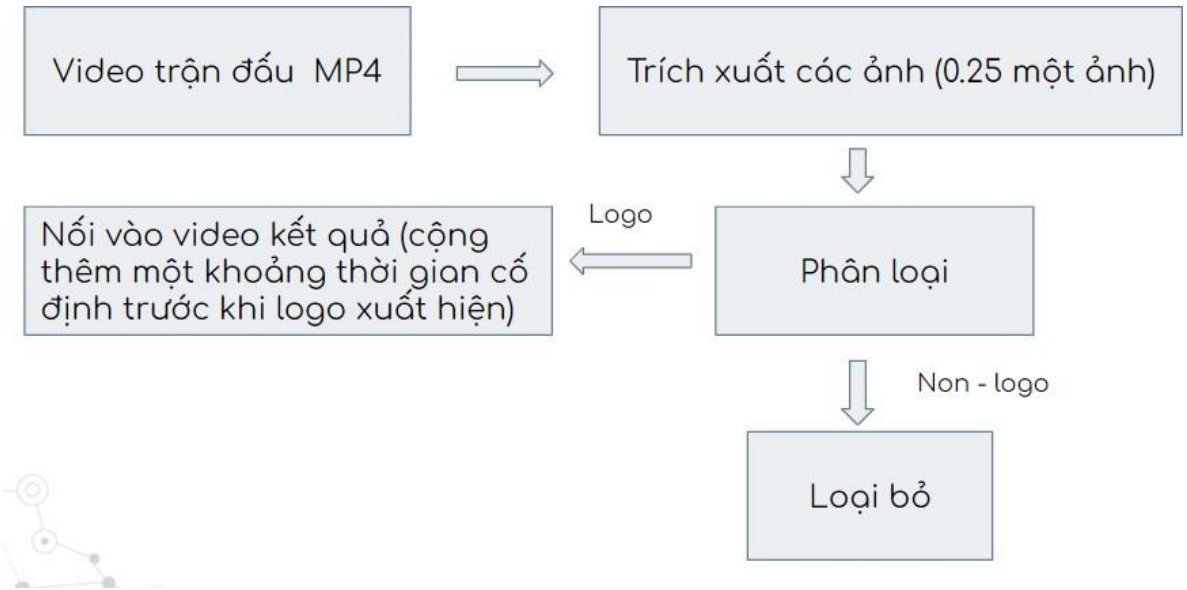
2. Sử dụng các đặc trưng ảnh

Vì dữ liệu nhóm thu thập được là các video mp4 gồm cả hình ảnh và âm thanh, do đó nhóm muốn tận dụng phần còn lại của dữ liệu (hình ảnh) để thực hiện phân loại.

Trong quá trình xem lại các trận đấu và cắt nhỏ video nhóm nhận thấy một đặc điểm đó là: trong các video trận đấu bóng đá, mỗi khi có một tình huống highlight xuất hiện thì editor thường cho chiếu lại tình huống đó với các góc nhìn khác nhau, kèm với đó là 1 logo chuyển tiếp (xuất hiện trước khi tình huống được chiếu lại).

Hình ảnh logo được sử dụng trước (sau) khi phát lại đoạn highlight

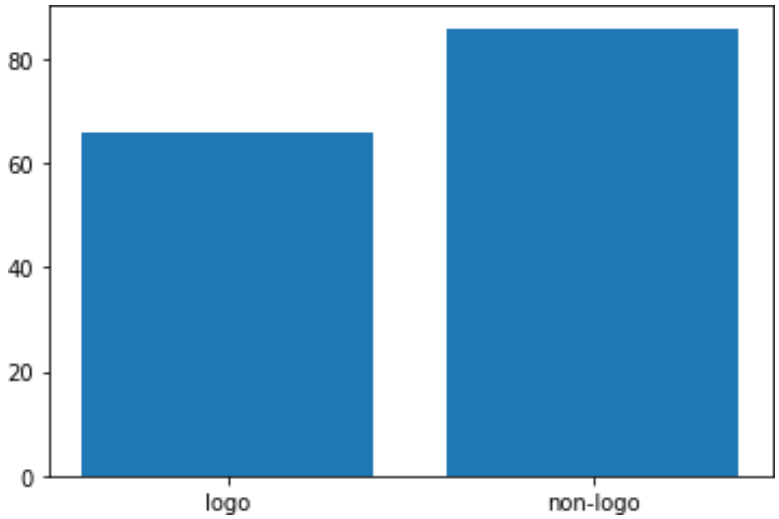
Do đó, nhóm nghĩ ra ý tưởng tìm kiếm những đoạn mà logo xuất hiện và đánh dấu thời gian xuất hiện của nó, và nối một khoảng thời gian tùy chọn để tạo video tóm tắt. Quá trình nhóm thực hiện như sau:



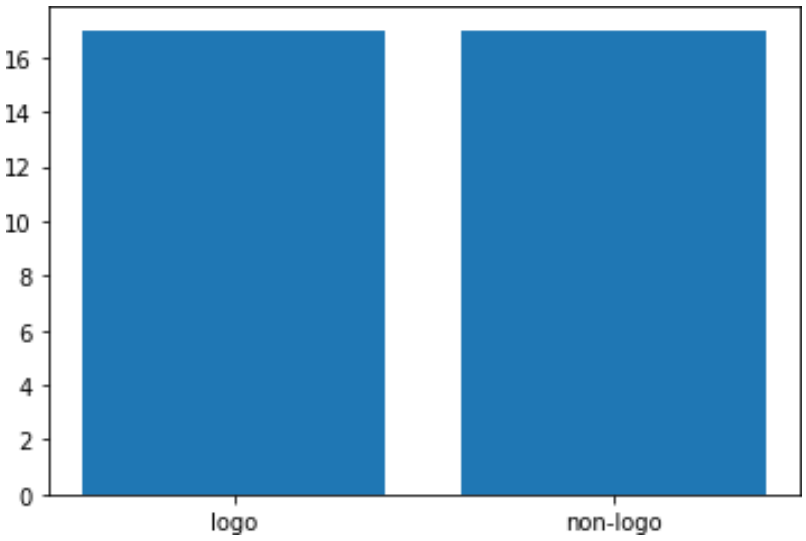
Nhóm tiếp tục xây dựng thêm 1 bộ data phụ để phục vụ cho việc này.

- Bộ dữ liệu được nhóm xây dựng bằng cách xem các video trận đấu và capture lại ảnh logo xuất hiện trong video.
- Bộ dữ liệu gồm 85 ảnh logo của 5 giải đấu (world cup 2018, England FA Cup, Bundesliga, Seria, England Premier League) và 104 ảnh không phải là logo (ảnh các cầu thủ, trọng tài, khán giả, huấn luyện viên, sân bóng).

  Tập train: logo (66), non-logo (86)



  Tập test: logo (17), non-logo (17)



Visualize một vài ảnh trong tập data:

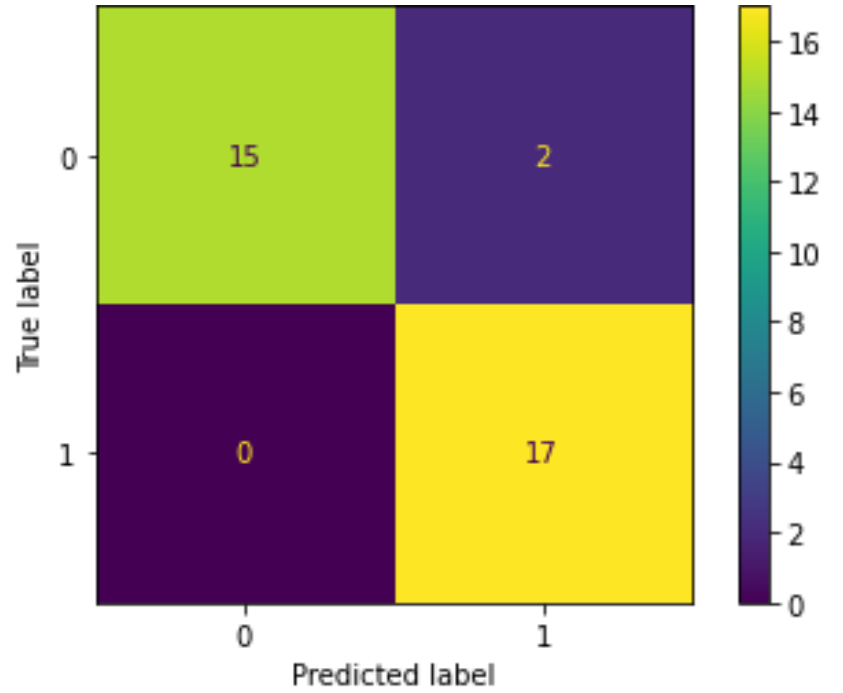
Các ảnh logo:

Các ảnh non-logo:

Sau khi thực hiện huấn luyện và đánh giá trên SVM và DenseNet-169, nhóm thu được kết quả như sau:

Kết quả của SVM:

- Độ chính xác trên tập test: 0.9411
- Độ chính xác trên tập train: 1.0
- Confusion matrix:



- Classification report:

	precision	recall	f1-score	support
logo	1.00	0.88	0.94	17
non-logo	0.89	1.00	0.94	17
accuracy			0.94	34
macro avg	0.95	0.94	0.94	34
weighted avg	0.95	0.94	0.94	34

Kết quả của DenseNet-169 khi train với 10 epoch: độ chính xác đạt 88.24%

```
model.evaluate(X_test,y_test)
```

```
2/2 [==============================] - 32s 3s/step - loss: 1.0678 - accuracy: 0.8824
[1.06781804561615, 0.8823529481887817]
```

	precision	recall	f1-score	support
logo	0.81	1.00	0.89	17
non-logo	1.00	0.76	0.87	17
accuracy			0.88	34
macro avg	0.90	0.88	0.88	34
weighted avg	0.90	0.88	0.88	34

3. Sử dụng đặc trưng ảnh với mô hình tuần tự LSTM

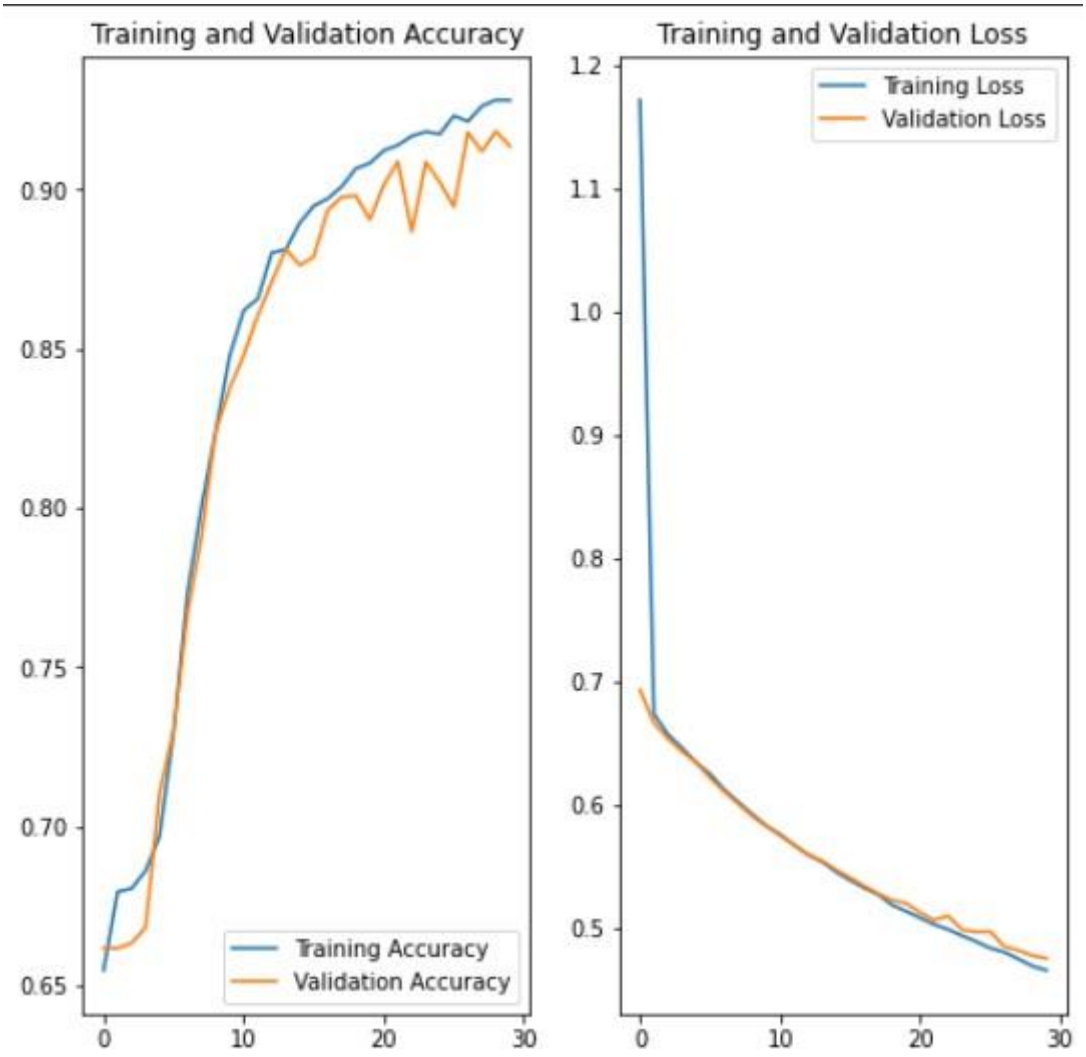
Sau khi tham khảo các bài báo khác, nhóm nhận thấy mô hình tuần tự (sequence models) khá phù hợp để giải quyết bài toán mà nhóm đặt ra.

Mô hình tuần tự mà nhóm chúng em sử dụng là LSTM. Đây là một mô hình thần kinh được phát triển dựa trên mạng RNN. Đối với mạng CNN thông thường, các feature sẽ không được chia sẻ trong mạng thì với LSTM, trọng số được chia sẻ theo thời gian trong mạng. Điều này có thể giúp ích rất nhiều trong việc đào tạo các mô hình để xử lí các bài toán có dữ liệu liên tục.

Sau khi tiến hành trích xuất đặc trưng mỗi video về dạng mà LSTM có thể học được (vector với số chiều là 20x2048), nhóm tiến hành training và evaluate thì thu được kết quả như sau:

Kết quả cuối cùng khi train mô hình với số epoch = 30:

```
Epoch 22/30
44/44 [==============================] - ETA: 0s - loss: 0.5025 - accuracy: 0.9138
Epoch 00022: val_loss did not improve from 0.47662
44/44 [==============================] - 4s 94ms/step - loss: 0.5025 - accuracy: 0.9138 - val_loss: 0.5061 - val_accuracy: 0.9087
Epoch 23/30
43/44 [============================>.] - ETA: 0s - loss: 0.4980 - accuracy: 0.9172
Epoch 00023: val_loss did not improve from 0.47662
44/44 [==============================] - 4s 94ms/step - loss: 0.4985 - accuracy: 0.9167 - val_loss: 0.5095 - val_accuracy: 0.8867
Epoch 24/30
43/44 [============================>.] - ETA: 0s - loss: 0.4933 - accuracy: 0.9182
Epoch 00024: val_loss did not improve from 0.47662
44/44 [==============================] - 4s 95ms/step - loss: 0.4933 - accuracy: 0.9180 - val_loss: 0.4982 - val_accuracy: 0.9085
Epoch 25/30
43/44 [============================>.] - ETA: 0s - loss: 0.4882 - accuracy: 0.9178
Epoch 00025: val_loss did not improve from 0.47662
44/44 [==============================] - 4s 94ms/step - loss: 0.4886 - accuracy: 0.9173 - val_loss: 0.4965 - val_accuracy: 0.9022
Epoch 26/30
43/44 [============================>.] - ETA: 0s - loss: 0.4835 - accuracy: 0.9230
Epoch 00026: val_loss did not improve from 0.47662
44/44 [==============================] - 4s 95ms/step - loss: 0.4834 - accuracy: 0.9230 - val_loss: 0.4968 - val_accuracy: 0.8945
Epoch 27/30
43/44 [============================>.] - ETA: 0s - loss: 0.4804 - accuracy: 0.9210
Epoch 00027: val_loss did not improve from 0.47662
44/44 [==============================] - 4s 95ms/step - loss: 0.4801 - accuracy: 0.9213 - val_loss: 0.4845 - val_accuracy: 0.9178
Epoch 28/30
43/44 [============================>.] - ETA: 0s - loss: 0.4747 - accuracy: 0.9260
Epoch 00028: val_loss did not improve from 0.47662
44/44 [==============================] - 4s 95ms/step - loss: 0.4745 - accuracy: 0.9261 - val_loss: 0.4817 - val_accuracy: 0.9118
Epoch 29/30
43/44 [============================>.] - ETA: 0s - loss: 0.4690 - accuracy: 0.9278
Epoch 00029: val_loss did not improve from 0.47662
44/44 [==============================] - 4s 94ms/step - loss: 0.4688 - accuracy: 0.9280 - val_loss: 0.4772 - val_accuracy: 0.9182
Epoch 30/30
43/44 [============================>.] - ETA: 0s - loss: 0.4653 - accuracy: 0.9283
Epoch 00030: val_loss improved from 0.47662 to 0.47507, saving model to /content/LSTM.h5
44/44 [==============================] - 4s 97ms/step - loss: 0.4654 - accuracy: 0.9279 - val_loss: 0.4751 - val_accuracy: 0.9134
```



Sau đó, nhóm tiến hành thu thập thêm tập test (gồm 237 video non-highlight và 27 video hightlight) để đánh giá thì thu được kết quả như sau:

```
[ ]  model.evaluate(X_test,y_test) # evaluate by data test

     9/9 [==============================] - 0s 34ms/step - loss: 0.8411 - accuracy: 0.5047
     [0.8410683870315552, 0.5047348737716675]
```

VI. Nhận xét

1. Kết quả thực nghiệm

- Để kiểm tra tính thiết thực của đồ án, nhóm tiến hành thu thập them dữ liệu để kiểm tra mô hình có thực sự đạt hiệu quả tốt không. Ở đây nhóm sẽ sử dụng mô hình SVM (sử dụng trích xuất đặc trưng audio) để tiến hành đánh giá.
- Về việc thu thập thêm dữ liệu: nhóm đã thu thập được 3 video, với mỗi video có độ dài là 45 phút và thực hiện gán nhãn cho mỗi video. Với mỗi video, nhóm sẽ chia ra từng giây và đánh giá giây đó có nằm trong diễn biến chính (highlight) hay không, nếu có, label ở giây đó sẽ mang nhãn "1", ngược lại sẽ mang nhãn "0".

Kết quả thu được sau khi dự đoán:

- Trận 1:

	precision	recall	f1-score	support
non-highlight	0.98	0.54	0.70	2662
highlight	0.21	0.93	0.34	346
accuracy			0.59	3008
macro avg	0.60	0.74	0.52	3008
weighted avg	0.90	0.59	0.66	3008

- Trận 2:

	precision	recall	f1-score	support
non-highlight	0.96	0.98	0.97	2698
highlight	0.41	0.25	0.31	151
accuracy			0.94	2849
macro avg	0.68	0.61	0.64	2849
weighted avg	0.93	0.94	0.93	2849

- Trận 3:

	precision	recall	f1-score	support
non-highlight	0.99	0.61	0.75	2647
highlight	0.13	0.92	0.23	173
accuracy			0.63	2820
macro avg	0.56	0.76	0.49	2820
weighted avg	0.94	0.63	0.72	2820

2. Đánh giá kết quả

- Hầu hết các model của nhóm thực nghiệm đều chỉ đạt ở mức tương đối, vẫn cần phải cải tiến mới có thể đem vào sử dụng được.
- Mô hình tìm kiếm khung ảnh logo cho kết quả tốt nhất, nhưng model này gặp hạn chế là chỉ làm được cho một số giải nhất định, những giải đấu chưa được train thì sẽ không thể tóm tắt video được, và cần phải bổ sung để huấn luyện lại.
- Mô hình đánh giá bằng âm thanh rất phụ thuộc vào chất lượng của bình luận viên, nếu bình luận viên bình luận bùng nổ, sôi động những lúc gay cấn thì độ chính xác của mô hình sẽ cao, nếu bình luận viên bình luận không sôi động thì mô hình cũng sẽ dự đoán không tốt lắm.
- Mô hình đánh giá bằng âm thanh và đặc trưng với mô hình tuần tự như LSTM tuy kết quả không cao những vẫn có thể tiếp tục cải thiện và phát triển thêm sau này.

3. Nguyên nhân & nhận xét

- Bộ dữ liệu nhóm làm vẫn chưa được phong phú: vì việc xem và cắt từng đoạn video nhỏ mất rất nhiều thời gian, để model học được chính xác thì cần phải có đủ đa dạng góc quay, cũng như áo đấu của các cầu thủ, các tình huống tương tự có thể xảy ra.
- Không đủ cấu hình cho việc đào tạo: Do nhóm chỉ sử dụng colab bình thường, bị giới hạn ở thời gian train GPU cũng như ram cũng chỉ giới hạn ở 12GB nên việc đào tạo với bộ dữ liệu lớn có gặp nhiều đôi chút khó khăn. Giải pháp có thể là chia nhỏ data hoặc mua bản premium.
- Model, các siêu tham số cấu hình trong mạng neural vẫn chưa phải là tốt nhất cho bài toán của nhóm: do thời gian có hạn, việc training khá mất nhiều thời gian nên việc tuning các siêu tham số mất khá nhiều thời gian.
- Các class trong bộ data của nhóm vẫn chưa được đồng đều: việc kiếm các highlight mất rất nhiều thời gian so với non-highlight, nhóm tin nếu tạo thêm cho bộ data đủ lớn, các class cân bằng nhau về số lượng thì chất lượng có thể sẽ được cải thiện.
- Có thể có nhiễu xuất hiện trong bộ data: Khi phân loại bằng audio, có một số pha tấn công khá nguy hiểm, nhưng bình luận viên lại có vẻ rất thản nhiên, dẫn đến làm lệch mô hình dự đoán; ở các pha cướp được bóng bất ngờ, tuy không phải là một trong những diễn biến chính mà nhóm quan tâm, nhưng bình luận viên lại tỏ ra hào hứng, điều này cũng dẫn đến cho mô hình dự đoán lệch.

VII. Hướng phát triển

- Ở tương lai gần, có thể tăng cường dữ liệu cho bộ dataset của nhóm, tìm hiểu thêm nhiều phương pháp trích xuất đặc trưng để cho việc đào tạo được đúng hơn, thử nghiệm với các bộ siêu tham số để tìm ra bộ tốt nhất.
- Có thể tăng tính chính xác cho mô hình bằng cách kết hợp cả 2 phương pháp (trích xuất đặc trưng ảnh, trích xuất đặc trưng audio) để tiến hành đào tạo mới.
- Nếu đề tài này phát triển tốt, có thể ứng dụng sang các đề tài tương tự như tóm tắt diễn biến của một bộ phim, một chương trình truyền hình, hoặc có thể kết hợp với các kiến thức khác ở xử lí ngôn ngữ tự nhiên để tóm tắt video của một video bài giảng, bài diễn thuyết.

TÀI LIỆU THAM KHẢO

https://librosa.org/doc/latest/index.html

https://www.kaggle.com/mauriciofigueiredo/methods-for-sound-noise-reduction

https://towardsdatascience.com/audio-deep-learning-made-simple-part-1-state-of-the-art-techniques-da1d3dff2504

https://towardsdatascience.com/audio-deep-learning-made-simple-part-2-why-mel-spectrograms-perform-better-aad889a93505

https://towardsdatascience.com/audio-deep-learning-made-simple-part-3-data-preparation-and-augmentation-24c6e1f6b52

https://towardsdatascience.com/audio-deep-learning-made-simple-sound-classification-step-by-step-cebc936bbe5

https://zulko.github.io/moviepy/ref/ref.html

https://nttuan8.com/bai-14-long-short-term-memory-lstm/

https://www.researchgate.net/publication/332682552_Soccer_Video_Summarization_Using_Deep_Learning

https://arxiv.org/pdf/1709.08421.pdf