

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Hệ thống lọc bỏ Video nội dung xấu cho trẻ em dưới  
14 tuổi**

**NGÔ QUỐC DŨNG**

dung.nq183896@sis.hust.edu.vn

**Ngành Công nghệ thông tin và truyền thông  
Chuyên ngành Công nghệ thông tin**

**Giảng viên hướng dẫn:** Ths. Nguyễn Tiến Thành

Chữ ký GVHD

**Khoa:** Công nghệ phần mềm

**Trường:** Công nghệ thông tin và Truyền thông

**HÀ NỘI, 06/2022**

# LỜI CẢM ƠN

Lời đầu tiên, em xin chân thành cảm ơn Trường Công nghệ Thông tin và Truyền thông – Đại học Bách Khoa Hà Nội đã tạo điều kiện cho em và các bạn được học tập và tiếp xúc với những kiến thức quan trọng của chương trình học để chúng em tự tin và sẵn sàng với những công việc sau này.

Em xin gửi lời cảm ơn sâu sắc đến ThS. Nguyễn Tiên Thành – người đã dùn dắt chúng em từ những ngày đầu vào trường cho đến tận bây giờ. Đặc biệt, thầy còn là người đồng hành hướng dẫn em trong quá trình nghiên cứu, xây dựng và hoàn thành Đồ án Tốt nghiệp. Không có sự đóng góp của thầy, em đã không thể hoàn thiện đồ án này như hiện giờ.

Ngoài ra, em cũng xin cảm ơn các thầy cô trong trường đã dạy bảo tận tình và giúp đỡ em trong suốt quá trình học tập tại trường, tạo nền tảng vững chắc để em trở thành nguồn nhân lực chất lượng cao trong những công việc sau này.

Cuối cùng, em xin cảm ơn gia đình, bạn bè và đồng nghiệp – những người đã luôn bên cạnh động viên và hỗ trợ em khi em gặp khó khăn và tạo động lực để em theo đuổi những thành tựu mới trong quá trình học tập tại Trường Công nghệ Thông tin và Truyền thông – Đại học Bách Khoa Hà Nội.

Do thời gian có hạn, cũng như kiến thức chuyên môn còn non kém, ĐATN của em sẽ không tránh khỏi những thiết sót và hạn chế. Em rất mong nhận được những đánh giá, nhận xét từ thầy cô và các bạn, để từ đây nhận thấy những ưu, nhược điểm của bản thân, đúc kết và xây dựng kinh nghiệm làm việc một cách vững vàng hơn.

Em xin chân thành cảm ơn!

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Hiện nay với sự phổ biến của các thiết bị thông minh và Internet, việc chia sẻ các video giải trí và học tập trên các nền tảng lớn như Youtube, Tiktok hay thậm chí là Facebook,... đang ngày một phát triển. Tuy nhiên, việc tràn lan những nội dung như vậy trên mạng xã hội cũng tiềm ẩn những nguy cơ như tin giả, nạn bắt nạt trên mạng và đặc biệt hơn là tổn thương tâm lý đối với người xem.

Để tránh phát tán những nội dung không phù hợp trên không gian mạng, nhiều nền tảng lớn như Youtube đã triển khai cơ chế riêng để lọc các video được đăng tải bằng machine learning cùng với sự giám sát của con người. Đây là một biện pháp khá hữu hiệu khi đa số các video không phù hợp được loại bỏ nhanh trước khi tiếp xúc với người dùng. Tuy nhiên, do số lượng người dùng lớn cộng với các video có độ dài khác nhau nên việc kiểm duyệt tự động thường được thực hiện không đồng bộ (tức là có thể đăng video xong thì mới được kiểm duyệt) dẫn đến vẫn còn khả năng số ít người dùng có thể đã xem video trước khi nó bị xóa bỏ.

Nhận thấy tiềm năng của hướng tiếp cận này, em quyết định xây dựng một website chia sẻ video giả lập sử dụng chính công nghệ deep learning và computer vision để giúp người quản trị kiểm soát các nội dung không phù hợp, với sự khác biệt cơ bản là các video khi được đăng tải sẽ phải trải qua các bước sàng lọc rõ ràng trước khi được đăng tải.

Sở dĩ em chọn cách giải quyết này bởi nó sẽ tăng tính phòng ngừa với các video nguy hiểm và đảm bảo các video được chọn là phù hợp với tiêu chuẩn chung của trang web. Để làm được điều này, em sẽ sử dụng một Deep learning Model giải quyết bài toán Image classification sau đó chia video thành từng ảnh để phân loại. Video sẽ chỉ được đăng tải khi không có chứa nội dung sexy, porn hoặc hentai.

Qua đồ án này, em đã đề xuất một hướng tiếp cận chặt chẽ hơn trong việc kiểm duyệt nội dung, kết quả đạt được tuy không hoàn hảo nhưng đã cung cấp cái nhìn chung về việc sử dụng AI trong việc kiểm soát các video trên mạng. .

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án .....	3
<b>CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....</b>	<b>5</b>
2.1 Khảo sát hiện trạng .....	5
2.2 Tổng quan chức năng .....	6
2.2.1 Biểu đồ use case tổng quát .....	6
2.2.2 Biểu đồ use case phân rã tương tác với Video .....	7
2.2.3 Biểu đồ use case phân rã quản lý Video.....	7
2.2.4 Biểu đồ use case phân rã quản lý người dùng .....	8
2.2.5 Quy trình nghiệp vụ .....	9
2.3 Đặc tả chức năng .....	9
2.3.1 Đặc tả use case Đăng nhập.....	9
2.3.2 Đặc tả use case Đăng Video .....	10
2.3.3 Đặc tả use case Duyệt yêu cầu đăng Video .....	11
2.3.4 Đặc tả use case Báo cáo vi phạm .....	12
2.3.5 Đặc tả use case Chặn Video vi phạm.....	13
2.4 Yêu cầu phi chức năng .....	14
<b>CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....</b>	<b>15</b>
3.1 Deep Learning.....	15
3.2 Convolutional Neural Network (CNN).....	17
3.3 Tensorflow.....	20

3.4 OpenCV .....	20
3.5 Django.....	21
<b>CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ .....</b>	<b>22</b>
4.1 Thiết kế kiến trúc.....	22
4.1.1 Lựa chọn kiến trúc phần mềm .....	22
4.1.2 Thiết kế tổng quan.....	23
4.1.3 Thiết kế chi tiết gói .....	24
4.2 Thiết kế chi tiết.....	26
4.2.1 Thiết kế giao diện .....	26
4.2.2 Thiết kế lớp .....	28
4.2.3 Thiết kế cơ sở dữ liệu .....	37
4.3 Xây dựng ứng dụng.....	39
4.3.1 Thư viện và công cụ sử dụng.....	39
4.3.2 Kết quả đạt được .....	40
4.3.3 Minh họa các chức năng chính .....	41
4.4 Kiểm thử.....	51
4.4.1 Đăng Video .....	51
4.4.2 Duyệt báo cáo vi phạm .....	56
4.4.3 Tổng kết .....	57
4.5 Triển khai .....	58
<b>CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....</b>	<b>59</b>
5.1 Thu thập và xử lý dữ liệu để huấn luyện Model.....	59
5.1.1 Đặt vấn đề .....	59
5.1.2 Giải pháp .....	59
5.1.3 Kết quả đạt được .....	63

5.2 Xây dựng Model.....	63
5.2.1 Đặt vấn đề .....	63
5.2.2 Giải pháp .....	64
5.2.3 Kết quả .....	67
5.3 Xử lý Background Task .....	68
5.3.1 Đặt vấn đề .....	68
5.3.2 Giải pháp .....	69
5.3.3 Kết quả .....	70
<b>CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>71</b>
6.1 Kết luận .....	71
6.2 Hướng phát triển.....	71



## DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ Use Case Tổng quan . . . . .	6
Hình 2.2	Biểu đồ Use Case Tương tác với Video . . . . .	7
Hình 2.3	Biểu đồ Use Case Quản lý Video . . . . .	7
Hình 2.4	Biểu đồ Use Case Quản lý người dùng . . . . .	8
Hình 2.5	Biểu đồ Hoạt động đăng Video . . . . .	9
Hình 3.1	Hình ảnh minh họa một mạng Nơ-ron . . . . .	16
Hình 3.2	Minh họa cấu trúc và cách huấn luyện một mạng Nơ-ron . . . . .	17
Hình 3.3	Minh họa cửa sổ trượt . . . . .	18
Hình 3.4	Cấu trúc mô hình CNN . . . . .	19
Hình 4.1	Minh họa cấu trúc MVT . . . . .	22
Hình 4.2	Biểu đồ gói . . . . .	23
Hình 4.3	Thiết kế gói . . . . .	24
Hình 4.4	Thiết kế gói Blog . . . . .	24
Hình 4.5	Thiết kế gói Users . . . . .	25
Hình 4.6	Trang chủ . . . . .	27
Hình 4.7	Đăng Video . . . . .	27
Hình 4.8	Xem Video . . . . .	27
Hình 4.9	Xem thông tin người dùng . . . . .	28
Hình 4.10	Lớp Views (Blog) . . . . .	28
Hình 4.11	Biểu đồ trạng thái Video . . . . .	30
Hình 4.12	Lớp ReportDetailView . . . . .	31
Hình 4.13	Lớp PostCreateView . . . . .	33
Hình 4.14	Lớp Views (Users) . . . . .	34
Hình 4.15	Biểu đồ trình tự UC đăng Video . . . . .	36
Hình 4.16	Biểu đồ trình tự UC Duyệt yêu cầu đăng Video . . . . .	36
Hình 4.17	Biểu đồ thực thể liên kết . . . . .	37
Hình 4.18	Form đăng Video . . . . .	41
Hình 4.19	Bài đăng đã được nộp . . . . .	41
Hình 4.20	Bài đăng đang được xử lý . . . . .	42
Hình 4.21	Bài đăng đã được duyệt bởi hệ thống . . . . .	43
Hình 4.22	Bài đăng xuất hiện trên trang chủ . . . . .	43
Hình 4.23	Bài đăng bị từ chối bởi hệ thống . . . . .	44
Hình 4.24	Bài đăng đang chờ duyệt bằng tay . . . . .	44
Hình 4.25	Bài đăng bị từ chối bởi Quản trị viên . . . . .	45

Hình 4.26	Yêu cầu đăng Video . . . . .	45
Hình 4.27	Form từ chối yêu cầu . . . . .	46
Hình 4.28	Video bị từ chối . . . . .	46
Hình 4.29	Video được chấp nhận . . . . .	47
Hình 4.30	Xem bài đăng . . . . .	47
Hình 4.31	Form báo cáo vi phạm . . . . .	48
Hình 4.32	Báo cáo vi phạm thành công . . . . .	48
Hình 4.33	Xem báo cáo vi phạm . . . . .	49
Hình 4.34	Form chặn Video . . . . .	50
Hình 4.35	Video đã bị chặn . . . . .	50
Hình 4.36	Form xóa báo cáo vi phạm . . . . .	51
Hình 4.37	Báo cáo vi phạm đã bị xóa . . . . .	51
Hình 5.1	Mô tả Nsfw_data_scraper trên Github . . . . .	60
Hình 5.2	Minh họa DupeGuru . . . . .	61
Hình 5.3	Sắp xếp file trong File Explorer . . . . .	62
Hình 5.4	Đổi định dạng file trong Command Prompt . . . . .	62
Hình 5.5	Biểu đồ hoạt động lọc File lỗi . . . . .	63
Hình 5.6	Công cụ NSFW JS . . . . .	64
Hình 5.7	Sử dụng Dropout . . . . .	66
Hình 5.8	Hàm matsu với Regularization . . . . .	67
Hình 5.9	Áp dụng L2 Regularization . . . . .	67
Hình 5.10	Hàm matsu mới . . . . .	67
Hình 5.11	Ma trận nhầm lẫn . . . . .	68

## DANH MỤC BẢNG BIỂU

Bảng 2.1	Đặc tả use case Đăng nhập . . . . .	9
Bảng 2.2	Đặc tả use case Đăng Video . . . . .	10
Bảng 2.3	Đặc tả use case Duyệt yêu cầu đăng Video . . . . .	11
Bảng 2.4	Đặc tả use case Báo cáo vi phạm . . . . .	12
Bảng 2.5	Đặc tả use case Chặn Video vi phạm . . . . .	13
Bảng 4.1	MVT và MVC . . . . .	22
Bảng 4.2	Operation của lớp Views . . . . .	28
Bảng 4.3	Method của lớp Views . . . . .	29
Bảng 4.4	Attribute của lớp ReportDetailView . . . . .	31
Bảng 4.5	Operation của lớp ReportDetailViews . . . . .	32
Bảng 4.6	Attribute của lớp PostCreateView . . . . .	33
Bảng 4.7	Operation của lớp PostcreateView . . . . .	34
Bảng 4.8	Operation của lớp Views (Users) . . . . .	35
Bảng 4.9	Method của lớp Views (Users) . . . . .	35
Bảng 4.10	Bảng User . . . . .	37
Bảng 4.11	Bảng Profile . . . . .	38
Bảng 4.12	Bảng Post . . . . .	38
Bảng 4.13	Bảng Report . . . . .	39
Bảng 4.14	Danh sách thư viện và công cụ sử dụng . . . . .	40
Bảng 4.15	Danh sách ngôn ngữ sử dụng . . . . .	40
Bảng 4.16	Danh sách Package . . . . .	40
Bảng 4.17	Test Case Đăng Video hợp lệ . . . . .	51
Bảng 4.18	Test Case Đăng Video vi phạm . . . . .	52
Bảng 4.19	Test Case Đăng File ảnh . . . . .	53
Bảng 4.20	Test Case Đăng File không đúng định dạng . . . . .	54
Bảng 4.21	Test Case Đăng Video bởi người dùng bị chặn . . . . .	55
Bảng 4.22	Test Case Chặn Video bị báo cáo . . . . .	56
Bảng 4.23	Test Case Xóa báo cáo vi phạm . . . . .	57
Bảng 4.24	Thông tin cấu hình máy . . . . .	58
Bảng 4.25	Kết quả triển khai thử nghiệm . . . . .	58
Bảng 5.1	Số ảnh thu được sau khi chạy lần đầu . . . . .	60
Bảng 5.2	Số ảnh thu được sau khi chạy lần 2 . . . . .	61
Bảng 5.3	Bảng Background_task/Background_task_completetask . . .	69



## **DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT**

<b>Thuật ngữ</b>	<b>Ý nghĩa</b>
API	Giao diện lập trình ứng dụng (Application Programming Interface)
EUD	Phát triển ứng dụng người dùng cuối(End-User Development)
GWT	Công cụ lập trình Javascript bằng Java của Google (Google Web Toolkit)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
IaaS	Dịch vụ hạ tầng

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Trẻ vị thành niên sớm là nhóm trẻ có độ tuổi từ 10 đến 13 tuổi, đây là lứa tuổi rất dễ bị ảnh hưởng bởi các hành động tiêu cực từ bạn bè, thầy cô và đặc biệt là trên mạng xã hội. Trong thời gian dịch bệnh Covid diễn biến phức tạp vừa qua, mạng xã hội và các nền tảng giải trí online khác đang ngày một thịnh hành và có ảnh hưởng lớn đến giới trẻ. Khi ở một mình không có người giám sát, trẻ vị thành niên sớm dễ trở thành nạn nhân của các bài đăng độc hại trên mạng xã hội, đặc biệt là các Video khi ngày càng có nhiều mạng xã hội chuyển sang phát tán nội dung này.

Trong bối cảnh mạng xã hội đang ngày càng phát triển đến mức bão hòa, các trang mạng xã hội đang dần chuyển hướng từ nhu cầu chia sẻ và giao tiếp thông thường bằng chữ viết và hình ảnh sang chia sẻ các nội dung dạng video. Bằng chứng là Tiktok và Youtube là hai mạng xã hội lớn nhất hiện nay có nội dung chính là các video. Hay thậm chí các trang như Facebook, Instagram cũng đang dần chuyển sang làm nhiều các nội dung video hơn với các chức năng Story, Shorts và Watch.

Tuy các video mang đến nhiều thông tin hơn cho người xem nhưng chúng cũng tiềm ẩn những nguy cơ mất an toàn trên không gian mạng nếu nội dung không được sàng lọc kỹ càng. Không như những bài viết thông thường, các video có chứa lượng lớn thông tin trực quan từ hình ảnh đến âm thanh. Những video vì vậy sẽ cho người xem thấy được sự việc rõ ràng và sinh động hơn như thể chính mình đang ở hiện trường. Đây là một công cụ vô cùng mạnh mẽ để tuyên truyền và phổ biến thông tin từ giáo dục, giải trí lành mạnh đến bạo lực, tội phạm... và nhiều hơn nữa. Khi chúng ta tiếp cận với các video trên mạng, ta có nguy cơ bị ảnh hưởng bởi các nội dung tiêu cực, sai trái gây tổn thương tâm lý hay tệ hơn là làm sai lệch nhận thức.

Nếu đảm bảo được các video là phù hợp với tiêu chuẩn cộng đồng và không gây ảnh hưởng đến an ninh, an toàn trên không gian mạng, ta có thể bảo vệ được người dùng trước các hành vi quấy phá. Mạng xã hội tuy có nhiều mặt tích cực nhưng cũng đem đến những rủi ro lớn đối với người dùng, một trong số đó là các nội dung độc hại. Bằng cách loại bỏ những video không phù hợp, ta sẽ góp phần xây dựng một cộng đồng văn minh, lịch sự hơn và tận dụng được những lợi ích của mạng xã hội một cách tối đa.

## 1.2 Mục tiêu và phạm vi đề tài

Hiện nay, với các nền tảng chính chia sẻ nội dung video như Youtube đã có áp dụng trí tuệ nhân tạo hay cụ thể hơn là deep learning vào phân loại và lọc nội dung không phù hợp một cách chặt chẽ. Tiktok là một trong những mạng xã hội lớn nhất thế giới nhưng những nội dung trên Tiktok được kiểm duyệt một cách lỏng lẻo theo quy định của trang mạng. Theo đó những nội dung nhạy cảm ở mức độ vừa phải vẫn có thể xuất hiện trên trang của người dùng tùy theo mục đích của người đăng tải. Đối với Facebook hay các mạng xã hội khác không tập trung chính vào video thì phần lớn thuật toán và trí tuệ nhân tạo sẽ được áp dụng vào những hành vi không phù hợp của người dùng thay vì chính nội dung của các bài đăng. Ví dụ như Facebook kiểm soát những tin nhắn và comments chặt chẽ hơn nhiều so với Youtube.

Hạn chế chính của những ứng dụng trên là hướng tiếp cận giải quyết nhanh chóng vi phạm thay vì phòng ngừa trước khi vi phạm xảy ra. Bộ lọc của Youtube không tự động lọc các video trước khi đăng tải mà chỉ quét những video xuất hiện trên trang chủ. Công nghệ của Google đã phát triển rất nhiều dẫn đến tốc độ xử lý video nhanh hơn đáng kể so với trước đây nhưng sự tăng trưởng của Youtube cũng đang trở thành một thách thức khi mỗi giờ có rất nhiều các đoạn clip được đăng tải với tổng thời lượng khoảng 300 tiếng. Như vậy xác suất để một video không phù hợp đến được với người dùng là vẫn còn có thể xảy ra.

Với mong muốn thử nghiệm một hướng tiếp cận an toàn hơn trong kiểm soát nội dung, em muốn phát triển trang Web chia sẻ video với chức năng chính là lọc các đoạn clip trước khi được đăng tải theo quy trình rõ ràng. Như vậy các nội dung trên Web sẽ được đảm bảo bởi bộ lọc và cả sự giám sát của con người, giảm thiểu tối đa vi phạm và phòng chống được sai phạm trước khi xảy ra. Tất nhiên, về mặt hiệu năng cách làm này gần như là bất khả thi trong thực tế do số lượng video được đăng có thể rất lớn dẫn đến nghẽn server hoặc ít ra là gây nên thời gian chờ rất dài để đăng tải một video. Vì vậy, đồ án này em sẽ tập trung hơn đến sức mạnh của Deep Learning trong xử lý ảnh và kiểm tra tính hữu dụng của nó trong kiểm duyệt nội dung nói chung.

## 1.3 Định hướng giải pháp

Để xây dựng bộ lọc video, em sẽ giải quyết bài toán Image Classification sử dụng mạng CNN rồi áp dụng từ hình ảnh vào video bằng cách quét qua từng khung hình trong đoạn phim và tính toán xác suất lớn nhất của từng nhãn. Các nhãn sẽ được chia thành Neutral - nội dung không có vi phạm (không phải tranh vẽ hay hoạt hình), Drawings – nội dung bằng tranh vẽ, hoạt hình hay anime không có vi

phạm, Hentai – truyện tranh hay phim hoạt hình 18+, Sexy – nội dung người thật có xu hướng nhạy cảm 13+, Porn – nội dung người thật 18+. Model của em sẽ được Transfer Learning theo mạng InceptionV3 với các lớp Dense ở cuối và được train với số lượng ảnh là khoảng 140000 ảnh.

Khi tích hợp vào trang Web để lọc nội dung, em sẽ sử dụng bộ lọc một cách tự động mỗi khi có video mới được đăng tải. Video sẽ chỉ được đăng lên trang chủ sau khi đã được gán nhãn bởi bộ lọc. Nếu video là phù hợp (Drawings hoặc Neutral) sẽ được tự động đăng lên trang chủ, ngược lại nếu video ko phù hợp (Hentai, Sexy hoặc Porn) sẽ được gán nhãn tương ứng và không được đăng. Trong trường hợp bộ lọc phát hiện nhầm hoặc đánh nhầm sai người dùng có thể yêu cầu phê duyệt bằng tay đến Admin.

Như vậy, qua đồ án này em đã đề xuất một hướng giải quyết đơn giản nhưng an toàn cho bài toán kiểm duyệt nội dung video trên không gian mạng. Kết quả đạt được cho thấy chất lượng của các đoạn clip được chia sẻ phụ thuộc rất nhiều đến model và phương pháp được sử dụng để lọc video. Model hiện tại tuy chưa được tốt nhưng cũng đã cho thấy được sức mạnh của Deep learning trong xử lý và phân loại video. Tuy vậy, do quy mô bài toán nhỏ nên giải pháp có phần đơn thuần, chưa áp dụng được vào thực tiễn khi số lượng video và người dùng đăng tải lớn.

#### 1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về quá trình khảo sát và phân tích yêu cầu, trong đó có quá trình phân tích thực trạng và định hình các nhu cầu, chức năng cần thiết của chương trình. Chương này có quan hệ mật thiết với bài toán được giới thiệu ở trên.

Trong Chương 3, em giới thiệu về các công nghệ sử dụng, đặc biệt là mạng CNN và tổng quan về Deep Learning trong Computer Vision. Ngoài ra em cũng có đề cập qua đến framework Django được sử dụng để xây dựng trang Web.

Ở chương 4, em sẽ trình bày cấu trúc cụ thể của chương trình với các biểu đồ lớp thiết kế và cơ sở dữ liệu. Với công nghệ sử dụng ở chương 3, em cũng sẽ trình bày cách triển khai và vận hành ứng dụng cũng như kết quả đạt được.

Tiếp đến chương 5 em sẽ nêu lên những đóng góp nổi bật trong bài toán bao gồm quá trình thu thập và xử lý dữ liệu, train model cũng như fine-tune để cải thiện model. Chương này có mục đích làm nổi bật chức năng quan trọng nhất của trang web và quá trình xây dựng nó.

Cuối cùng ở chương 6 em sẽ tổng hợp lại những kết quả đạt được từ những nội dung ở trên và đánh giá độ hoàn thiện cũng như quyết định hướng phát triển trong

tương lai.

## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

### 2.1 Khảo sát hiện trạng

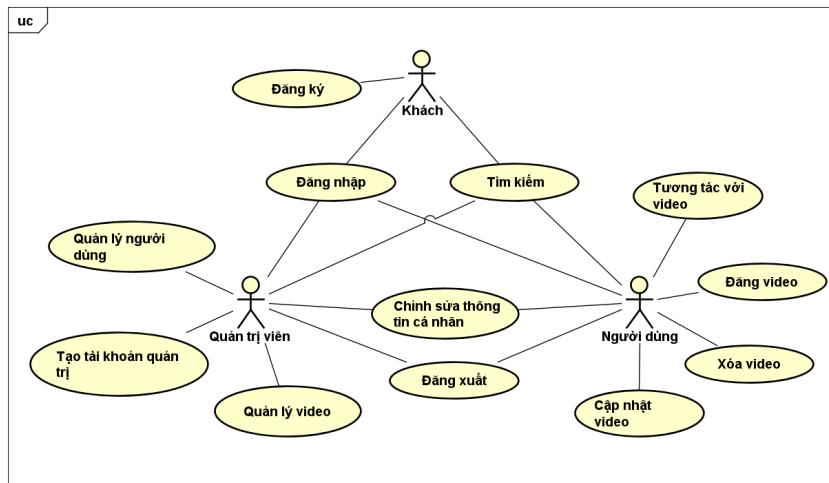
Với các nền tảng chia sẻ video trực tuyến hiện nay, yêu cầu cơ bản là phải cho phép người dùng đăng tải và chia sẻ video cũng như xem các video được chia sẻ trên trang chủ. Ngoài ra, để tiện lợi cho người dùng và đẩy mạnh tương tác giữa người với người hay người dùng với trang web, các hệ thống hiện nay cũng yêu cầu thêm các chức năng như Like/Dislike, Share, Comment và Follow/Subscribe. Để đề cao tính thời sự và tăng tương tác người dùng, các mạng xã hội còn sử dụng chức năng Livestream để phát trực tiếp video. Tuy nhiên khi phát trực tiếp video thì việc quản lý nội dung cũng trở nên khó khăn hơn và sẽ phụ thuộc lớn vào báo cáo của các người dùng khác để nhận biết vi phạm. Cuối cùng, để tạo lợi nhuận cho người sáng tạo nội dung cũng như nền tảng chia sẻ video cần có phương thức thanh toán riêng cho phép các nhà tài trợ đăng quảng cáo hoặc người dùng nhận tiền lãi từ mạng xã hội. Các phương thức thanh toán này có khác biệt lớn giữa các mạng xã hội khác nhau và rất khó đi sâu vào tìm hiểu.

Như đã nêu ở Chương 1, các bộ lọc nội dung video trên mạng xã hội thường có sự kết hợp giữa AI và con người ở các mức độ khác nhau. Nếu so sánh giữa các nền tảng phổ biến thì có thể thấy bộ lọc của Youtube có chất lượng vượt trội hơn hẳn khi lọc các video. Tuy nhiên, để đảm bảo tiện lợi cho người dùng thì các video có khả năng không được phân loại trước khi đăng tải nên chưa giải quyết được nguy cơ tiềm ẩn khi một video vi phạm được đăng lên.

Từ những quan sát trên, em hướng tới tạo nên một trang Web mô phỏng tập trung chính vào tính năng lọc nội dung. Trong đó khi đăng tải video sẽ phải được bộ lọc tự động phân loại trước khi thông qua, đồng thời có cho phép kiểm tra lại bởi con người để đảm bảo không có trường hợp loại nhầm. Trang Web cũng phải đảm bảo được yêu cầu chính của một nền tảng chia sẻ Video là có chức năng đăng tải, chia sẻ video cũng như xem video. Tuy nhiên do bộ lọc tập trung vào các nội dung video nên em sẽ lược bỏ một vài chức năng tương tác với người dùng như Comment, Share... Tính năng Livestream do khó kiểm soát trong kiểm duyệt nội dung nên cũng sẽ không được thêm vào.

## 2.2 Tổng quan chức năng

### 2.2.1 Biểu đồ use case tổng quát



**Hình 2.1:** Biểu đồ Use Case Tổng quan

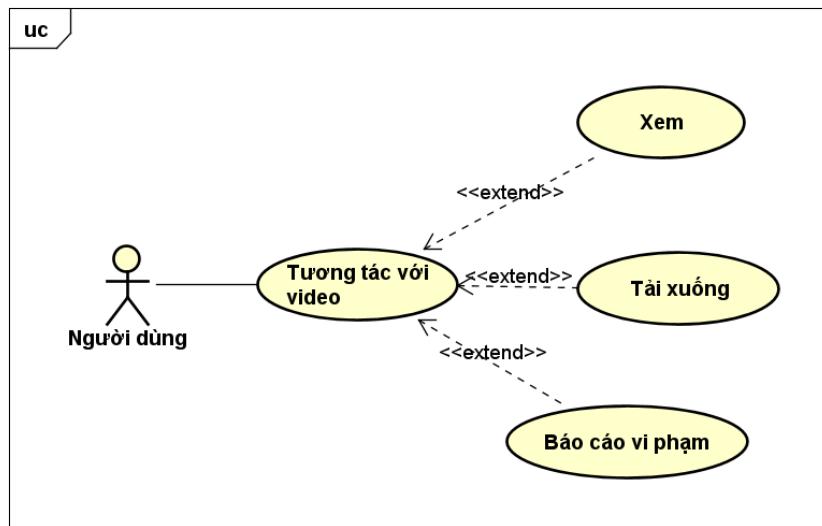
Các tác nhân chính bao gồm Khách vãng lai, Người dùng và Quản trị viên.  
Trong đó:

Khách vãng lai là đối tượng truy cập vào trang web mà không cần tài khoản. Khách vãng lai có quyền tìm kiếm video, kênh của người dùng và xem các video trên web. Ngoài ra khách vãng lai còn có thể đăng ký tài khoản người dùng hoặc đăng nhập với tài khoản đã có.

Người dùng là đối tượng chính sử dụng trang web khi đã đăng nhập vào tài khoản người dùng. Người dùng có quyền đăng video, chỉnh sửa video, xóa video, tìm kiếm video, kênh, xem video và tương tác với video.

Quản trị viên là đối tượng quản lý trang web sử dụng tài khoản quản trị viên. Quản trị viên có các quyền điều hành trang web như xử lý yêu cầu đăng video, chặn video, khóa/mở khóa tài khoản người dùng và tạo tài khoản quản trị viên.

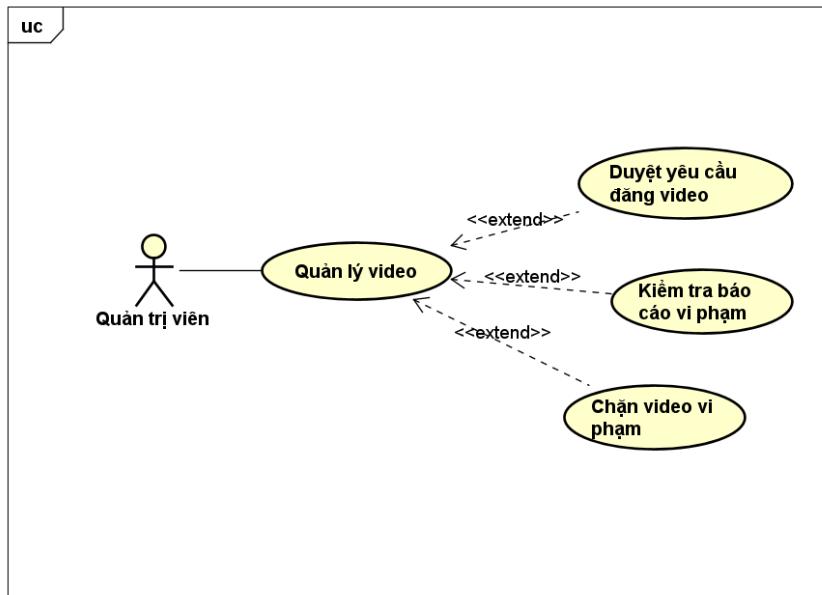
### 2.2.2 Biểu đồ use case phân rã tương tác với Video



**Hình 2.2:** Biểu đồ Use Case Tương tác với Video

Khi tương tác với video người dùng có thể xem video, người dùng cũng có thể tải xuống file video tương ứng hoặc báo cáo vi phạm để đưa yêu cầu đến quản trị viên khi thấy nội dung không phù hợp.

### 2.2.3 Biểu đồ use case phân rã quản lý Video

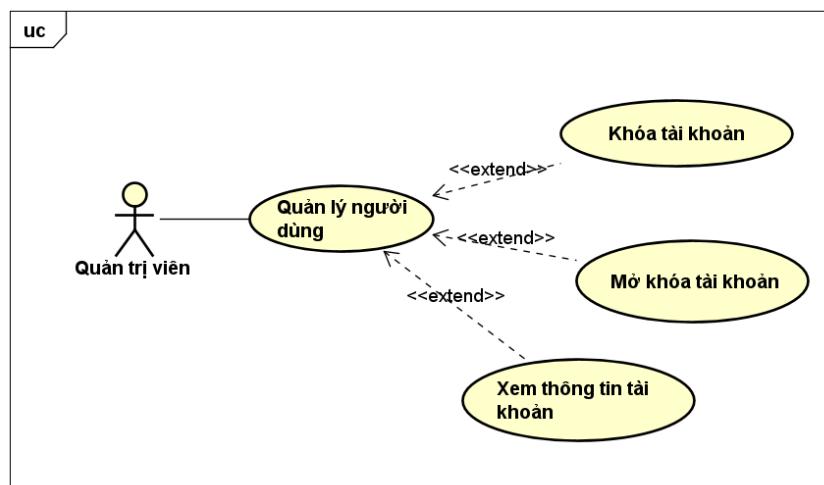


**Hình 2.3:** Biểu đồ Use Case Quản lý Video

Quản trị viên có quyền giám sát các video trên trang web. Khi có yêu cầu đăng video không được duyệt tự động, người dùng có thể yêu cầu trực tiếp đến quản trị viên. Mặt khác khi có người dùng báo cáo có video vi phạm không được phân loại

chính xác, quản trị viên cũng có thể kiểm tra yêu cầu và quyết định có chặn video đó hay không. Quản trị viên cũng có thể chặn video vi phạm bất kì sau khi xem trên trang chủ.

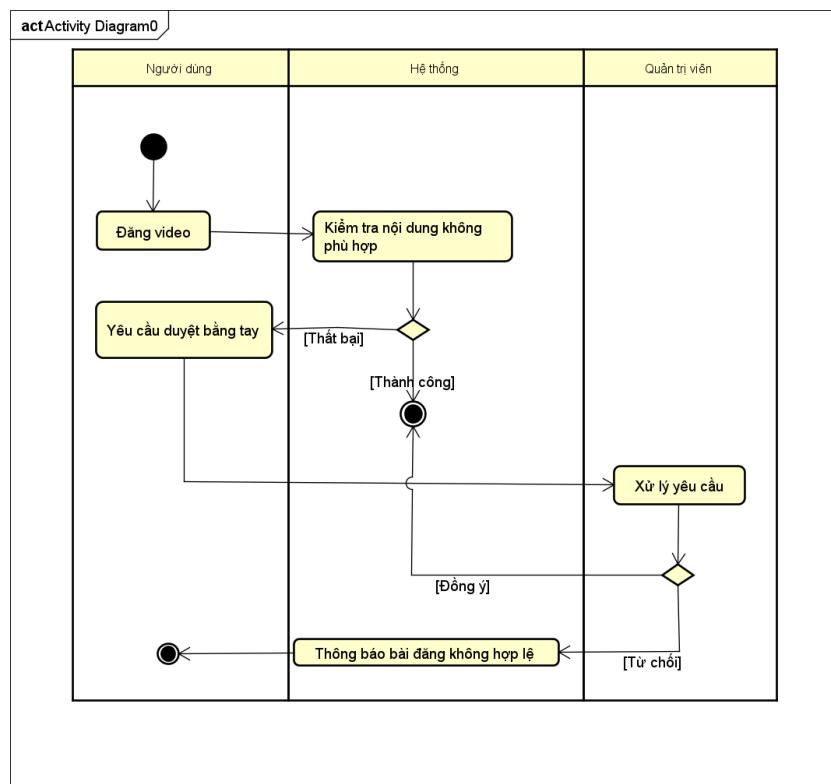
#### 2.2.4 Biểu đồ use case phân rã quản lý người dùng



**Hình 2.4:** Biểu đồ Use Case Quản lý người dùng

Quản trị viên có quyền xem thông tin người dùng, khóa tài khoản người dùng trong trường hợp có hành vi vi phạm và cũng có thể mở khóa tài khoản đã khóa. Khi tài khoản người dùng bị khóa người dùng đó sẽ không thể đăng video cho đến khi tài khoản được mở khóa bởi người quản trị.

### 2.2.5 Quy trình nghiệp vụ



**Hình 2.5:** Biểu đồ Hoạt động đăng Video

Quy trình nghiệp vụ đăng video như sau: Người dùng tạo bài đăng mới và đăng lên Web, hệ thống sẽ tự động quét video và phân loại nội dung, nếu phù hợp video sẽ được tự động đăng, nếu bị lọc người dùng có thể yêu cầu duyệt bằng tay đến quản trị viên.

## 2.3 Đặc tả chức năng

### 2.3.1 Đặc tả use case Đăng nhập

**Bảng 2.1:** Đặc tả use case Đăng nhập

<b>Mã Use case</b>	<b>UC001</b>
<b>Tên Use case</b>	<b>Đăng nhập</b>
<b>Tác nhân</b>	Khách
<b>Mục đích sử dụng</b>	Khách đăng nhập vào hệ thống
<b>Sự kiện kích hoạt</b>	Khách chọn chức năng đăng nhập
<b>Tiền điều kiện</b>	Không

<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Khách chọn chức năng Đăng nhập</li> <li>2. Hệ thống hiển thị giao diện đăng nhập</li> <li>3. Khách nhập username và password</li> <li>4. Khách yêu cầu đăng nhập</li> <li>5. Hệ thống kiểm tra Khách đã nhập các trường bắt buộc nhập hay chưa</li> <li>6. Hệ thống kiểm tra username và password có hợp lệ do Người dùng đăng ký với hệ thống hay không, Hệ thống kiểm tra vai trò của Khách trong hệ thống</li> <li>7. Hệ thống thông báo đăng nhập thành công và chuyển sang giao diện trang chủ, giao diện phù hợp với vai trò trong hệ thống.</li> </ol>
<b>Luồng sự kiện thay thế</b>	<ol style="list-style-type: none"> <li>6a. Hệ thống thông báo lỗi: cần nhập các trường bắt buộc nếu khách nhập thiếu</li> <li>7a. Hệ thống thông báo lỗi: Username hoặc mật khẩu không đúng nếu không tìm thấy username hoặc password trong hệ thống</li> </ol>
<b>Hậu điều kiện</b>	Hệ thống thông báo đăng nhập thành công

### 2.3.2 Đặc tả use case Đăng Video

Bảng 2.2: Đặc tả use case Đăng Video

<b>Mã Use case</b>	<b>UC002</b>
<b>Tên Use case</b>	<b>Đăng Video</b>
<b>Tác nhân</b>	Người dùng
<b>Mục đích sử dụng</b>	Người dùng đăng video lên trang chủ
<b>Sự kiện kích hoạt</b>	Người dùng chọn tạo bài đăng mới
<b>Tiền điều kiện</b>	Không

<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng chọn chức năng tạo bài mới</li> <li>2. Hệ thống hiển thị giao diện đăng bài</li> <li>3. Người dùng nhập tên bài đăng, mô tả và gán file video</li> <li>4. Người dùng yêu cầu đăng bài</li> <li>5. Hệ thống kiểm tra Người dùng đã nhập các trường bắt buộc nhập hay chưa</li> <li>6. Hệ thống kiểm tra thông tin bài đăng có hợp lệ không và file đã đúng định dạng chưa</li> <li>7. Hệ thống tải lên video và thông báo với người dùng video đang chờ xử lý</li> <li>8. Hệ thống tiến hành quét video và cập nhật trạng thái bài đăng thành đang xử lý</li> <li>9. Hệ thống thông báo video hợp lệ và cập nhật trạng thái bài đăng lên trang chủ</li> </ol>
<b>Luồng sự kiện thay thế</b>	<ol style="list-style-type: none"> <li>5a. Hệ thống thông báo lỗi: cần nhập các trường bắt buộc nếu khách nhập thiếu</li> <li>6a. Hệ thống thông báo lỗi: định dạng file không phải video hoặc ảnh</li> <li>10a. Hệ thống thông báo video không phù hợp</li> <li>10b. Người dùng chọn yêu cầu duyệt bằng tay đến quản trị viên</li> <li>10c. Hệ thống thông báo bài đăng được duyệt hoặc bị từ chối</li> </ol>
<b>Hậu điều kiện</b>	Hệ thống thông báo bài đăng phù hợp và đã được đăng tải

### 2.3.3 Đặc tả use case Duyệt yêu cầu đăng Video

Bảng 2.3: Đặc tả use case Duyệt yêu cầu đăng Video

<b>Mã Use case</b>	<b>UC003</b>
<b>Tên Use case</b>	<b>Duyệt yêu cầu đăng Video</b>
<b>Tác nhân</b>	Quản trị viên

<b>Mục đích sử dụng</b>	Quản trị viên kiểm tra lại yêu cầu đăng video đã bị lọc bởi hệ thống
<b>Sự kiện kích hoạt</b>	Quản trị viên chọn xem yêu cầu đăng bài
<b>Tiền điều kiện</b>	Người dùng yêu cầu duyệt bài đăng bằng tay
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Quản trị viên chọn xem yêu cầu đăng bài cần duyệt</li> <li>2. Hệ thống hiển thị giao diện bài đăng và vi phạm</li> <li>3. Quản trị viên xem nội dung video và chọn Đồng ý hay Từ chối yêu cầu</li> <li>4. Hệ thống tự động đăng bài nếu được Đồng ý và trả yêu cầu về người dùng nếu từ chối</li> <li>5. Hệ thống thông báo bài đăng đã được Duyệt hoặc Từ chối</li> </ol>
<b>Luồng sự kiện thay thế</b>	Không
<b>Hậu điều kiện</b>	Hệ thống cập nhật tình trạng bài đăng đến người dùng. Nếu Đồng ý, bài đăng được đăng lên trang chủ

#### 2.3.4 Đặc tả use case Báo cáo vi phạm

Bảng 2.4: Đặc tả use case Báo cáo vi phạm

<b>Mã Use case</b>	<b>UC004</b>
<b>Tên Use case</b>	<b>Báo cáo vi phạm</b>
<b>Tác nhân</b>	Người dùng
<b>Mục đích sử dụng</b>	Người dùng báo cáo có video vi phạm và yêu cầu quản trị viên kiểm tra
<b>Sự kiện kích hoạt</b>	Người dùng chọn báo cáo vi phạm
<b>Tiền điều kiện</b>	Bài đăng đã được đăng tải trên trang chủ

<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Người dùng chọn chức năng báo cáo vi phạm</li> <li>2. Hệ thống hiển thị giao diện báo cáo vi phạm</li> <li>3. Người dùng chọn lý do vi phạm</li> <li>4. Người dùng yêu cầu gửi báo cáo</li> <li>5. Hệ thống kiểm tra Người dùng đã nhập lý do vi phạm hay chưa</li> <li>6. Hệ thống tạo báo cáo vi phạm</li> <li>7. Hệ thống thông báo báo cáo đã được gửi đến Quản trị viên</li> </ol>
<b>Luồng sự kiện thay thế</b>	<ol style="list-style-type: none"> <li>5a. Hệ thống thông báo lỗi: cần nhập lý do vi phạm</li> <li>6a. Hệ thống thông báo lỗi: người dùng đã báo cáo Video trước đó</li> </ol>
<b>Hậu điều kiện</b>	Hệ thống tạo báo cáo vi phạm mới đến Quản trị viên

### 2.3.5 Đặc tả use case Chặn Video vi phạm

Bảng 2.5: Đặc tả use case Chặn Video vi phạm

<b>Mã Use case</b>	<b>UC005</b>
<b>Tên Use case</b>	<b>Chặn Video</b>
<b>Tác nhân</b>	Quản trị viên
<b>Mục đích sử dụng</b>	Quản trị viên chặn video vi phạm trên trang chủ
<b>Sự kiện kích hoạt</b>	Quản trị viên chọn chặng video vi phạm
<b>Tiền điều kiện</b>	Bài đăng đã được đăng tải trên trang chủ

<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Quản trị viên chọn chức năng chặn video vi phạm</li> <li>2. Hệ thống hiển thị giao diện báo cáo vi phạm</li> <li>3. Quản trị viên chọn lý do vi phạm</li> <li>4. Quản trị viên yêu cầu chặn video</li> <li>5. Hệ thống kiểm tra Quản trị viên đã nhập lý do vi phạm hay chưa</li> <li>6. Hệ thống cập nhật trạng thái video thành bị chặn</li> <li>7. Hệ thống thông báo video vi phạm đã bị chặn</li> </ol>
<b>Luồng sự kiện thay thế</b>	<p>5a. Hệ thống thông báo lỗi: cần nhập lý do vi phạm</p>
<b>Hậu điều kiện</b>	Hệ thống cập nhật trạng thái video thành bị chặn

## 2.4 Yêu cầu phi chức năng

Chương trình yêu cầu tính chất chẽ trong kiểm duyệt nội dung nên cần có tính bảo mật cao, phân quyền rõ ràng: Quản trị viên có quyền cao nhất, tiếp đến là Người dùng và Khách vãng lai.

Vì bài toán giả lập nên quy mô Server nhỏ chỉ đáp ứng được một vài yêu cầu đăng video một lúc với thời gian xử lý tùy thuộc vào độ dài video và nội dung (nếu có vi phạm). Ngoài ra cơ sở dữ liệu Sqlite3 là một cơ sở dữ liệu nhúng nên số lượng người dùng và bài đăng cũng bị giới hạn ở mức vài trăm. Sqlite3 cũng không cho phép truy cập đồng thời vào cơ sở dữ liệu.

## CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

### 3.1 Deep Learning

Trong lĩnh vực nghiên cứu AI, Machine Learning đã đạt được thành công đáng kể trong những năm gần đây – cho phép máy tính vượt qua hoặc tiến gần đến việc kết hợp hiệu suất của con người trong các lĩnh vực từ nhận dạng khuôn mặt đến nhận dạng giọng nói và ngôn ngữ.

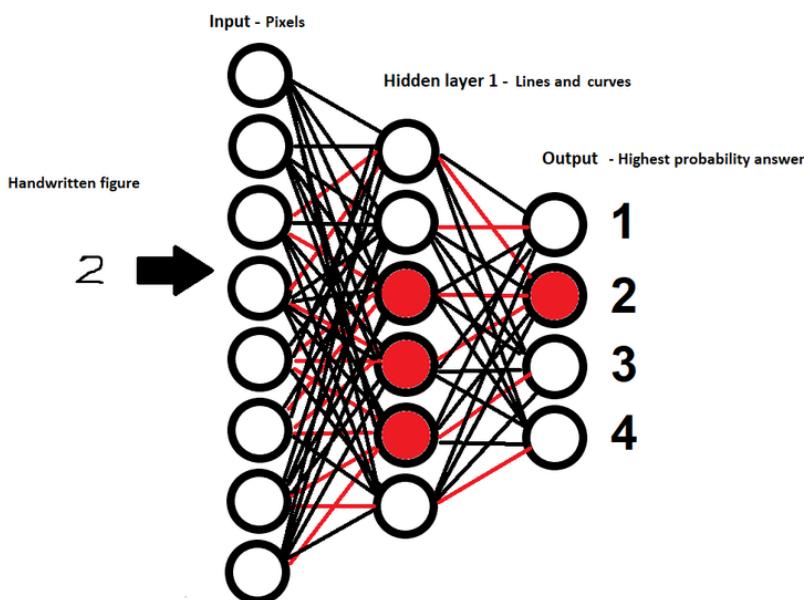
Machine Learning là quá trình dạy máy tính thực hiện một nhiệm vụ, thay vì lập trình nó làm thế nào để thực hiện nhiệm vụ đó từng bước một. Khi kết thúc đào tạo, một hệ thống Machine Learning sẽ có thể đưa ra dự đoán chính xác khi được cung cấp dữ liệu.

Machine Learning thường được chia thành học có giám sát, trong đó máy tính học bằng ví dụ từ dữ liệu được gắn nhãn và học không giám sát, trong đó các máy tính nhóm các dữ liệu tương tự và xác định chính xác sự bất thường.

Deep Learning là một tập hợp con của Machine Learning, có khả năng khác biệt ở một số khía cạnh quan trọng so với Machine Learning nông truyền thống, cho phép máy tính giải quyết một loạt các vấn đề phức tạp không thể giải quyết được.

Để giải quyết vấn đề này, máy tính cần phải có khả năng đối phó với sự đa dạng lớn trong cách thức trình bày dữ liệu. Mỗi chữ số từ 0 đến 9 có thể được viết theo vô số cách: kích thước và hình dạng chính xác của mỗi chữ số viết tay có thể rất khác nhau tùy thuộc vào người viết và trong hoàn cảnh nào. Đối phó với sự biến đổi của các tính năng này và sự lộn xộn tương tác lớn hơn giữa chúng, là nơi học tập sâu và mạng nơ-ron sâu trở nên hữu ích.

Mạng nơ-ron là các mô hình toán học có cấu trúc được lấy cảm hứng lỏng lẻo từ bộ não. Mỗi nơ-ron trong mạng nơ-ron là một hàm toán học lấy dữ liệu thông qua đầu vào, biến đổi dữ liệu đó thành dạng dễ điều chỉnh hơn và sau đó phun ra thông qua đầu ra. Bạn có thể nghĩ về các nơ-ron trong một mạng lưới thần kinh như được sắp xếp theo lớp, như hình dưới đây.

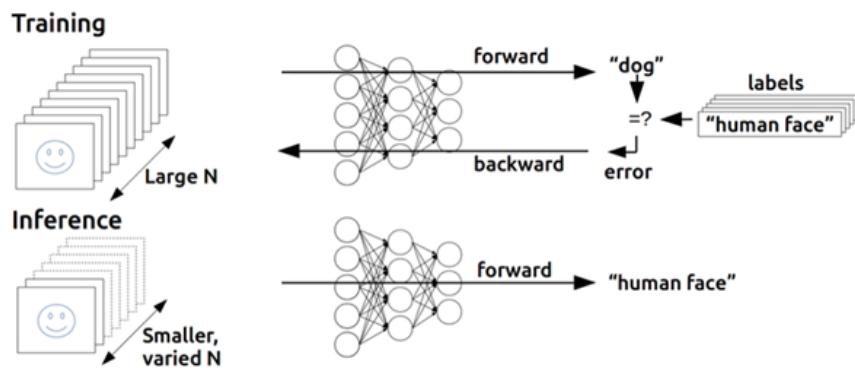


**Hình 3.1:** Hình ảnh minh họa một mạng Nơ-ron

Tất cả các mạng nơ-ron đều có một lớp đầu vào, trong đó dữ liệu ban đầu được đưa vào và một lớp đầu ra, tạo ra dự đoán cuối cùng. Nhưng trong một mạng nơ-ron sâu, sẽ có nhiều “lớp tế bào” ẩn giữa các lớp đầu vào và đầu ra, mỗi lớp cho dữ liệu vào nhau. Do đó, thuật ngữ “Deep” trong “Deep Learning” và “mạng nơ-ron sâu”, nó liên quan đến số lượng lớn các lớp ẩn – thường lớn hơn ba – tại trung tâm của các mạng thần kinh này.

Có thể thấy, đầu ra của một lớp là đầu vào của lớp tiếp theo trong mạng, với dữ liệu chảy qua mạng từ đầu vào đến đầu ra.

Quá trình học tập được thực hiện bằng cách mạng có thể thay đổi tầm quan trọng của các liên kết giữa các nơ-ron trong mỗi lớp. Mỗi liên kết có một giá trị đính kèm được gọi là trọng số, nó sẽ sửa đổi giá trị được tạo ra bởi một nơron khi nó truyền từ lớp này sang lớp kế tiếp. Bằng cách thay đổi giá trị của các trọng số này và một giá trị liên quan được gọi là sai lệch, có thể nhấn mạnh hoặc làm giảm tầm quan trọng của các liên kết giữa các nơ-ron trong mạng.



**Hình 3.2:** Minh họa cấu trúc và cách huấn luyện một mạng Nơ-ron

Mô hình học được các liên kết giữa các nơ-ron rất quan trọng trong việc đưa ra dự đoán thành công trong quá trình đào tạo. Ở mỗi bước trong quá trình đào tạo, mạng sẽ sử dụng một hàm toán học để xác định mức độ chính xác của dự đoán mới nhất của nó so với dự kiến. Hàm này tạo ra một loạt các giá trị lỗi, do đó hệ thống có thể sử dụng để tính toán cách mô hình nên cập nhật giá trị của các trọng số được gắn vào mỗi liên kết, với mục đích cuối cùng là cải thiện độ chính xác của các dự đoán của mạng. Mức độ mà các giá trị này sẽ được thay đổi được tính bởi một chức năng tối ưu hóa, chẳng hạn như giảm độ dốc và những thay đổi đó được đẩy lùi trên toàn mạng vào cuối mỗi chu kỳ đào tạo trong một bước gọi là lan truyền ngược.

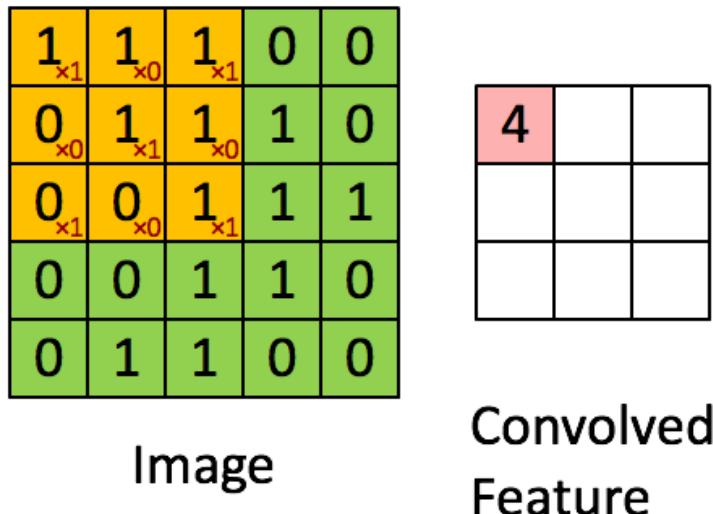
Trải qua nhiều, rất nhiều chu kỳ đào tạo và với sự trợ giúp của việc điều chỉnh tham số thủ công không thường xuyên, mạng sẽ tiếp tục обуч để tạo dự đoán tốt hơn và tốt hơn cho đến khi nó đạt gần với độ chính xác cao nhất.

Về cơ bản, Deep Learning cho phép Machine Learning giải quyết một loạt các vấn đề phức tạp mới – chẳng hạn như nhận dạng hình ảnh, ngôn ngữ và lời nói – bằng cách cho phép máy móc tìm hiểu cách các tính năng trong dữ liệu kết hợp thành các dạng trừu tượng ngày càng cao hơn.

Trong bài toán phân loại ảnh, Deep Learning được sử dụng với các nơ-ron là các pixels hay tập hợp các pixels được đưa qua mạng CNN mà em sẽ đề cập ở phần tiếp theo. Kết quả cuối cùng sẽ dự đoán được xác suất tương ứng của từng nhãn trong ảnh.

### 3.2 Convolutional Neural Network (CNN)

Convolution là một cửa sổ trượt (Sliding Windows) trên một ma trận như mô tả hình 3.3:

**Hình 3.3:** Minh họa cửa sổ trượt

Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature. Trong hình ảnh ví dụ trên, ma trận bên trái là một hình ảnh trắng đen được số hóa. Ma trận có kích thước  $5 \times 5$  và mỗi điểm ảnh có giá trị 1 hoặc 0 là giao điểm của dòng và cột.

Convolution hay tích chập là nhân từng phần tử trong ma trận 3. Sliding Window hay còn gọi là kernel, filter hoặc feature detect là một ma trận có kích thước nhỏ như trong ví dụ trên là  $3 \times 3$ .

Convolution hay tích chập là nhân từng phần tử bên trong ma trận  $3 \times 3$  với ma trận bên trái. Kết quả được một ma trận gọi là Convolved feature được sinh ra từ việc nhân ma trận Filter với ma trận ảnh  $5 \times 5$  bên trái.

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

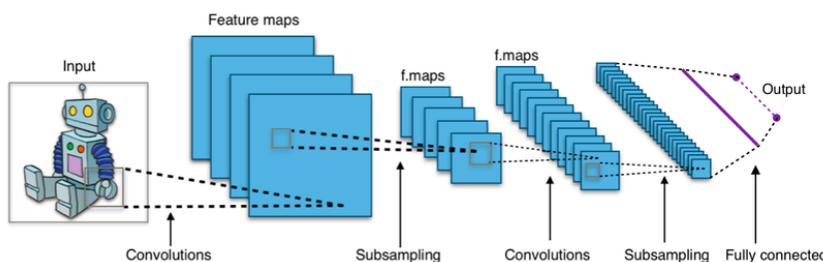
Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (training) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.



**Hình 3.4:** Cấu trúc mô hình CNN

Trong mô hình CNN có 2 khía cạnh cần quan tâm là tính bất biến (Location Invariance) và tính kết hợp (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các góc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.

Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.

Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên.

Mạng CNN đặc biệt có lợi trong việc xử lý hình ảnh bởi tính không gian của nó. Với bài toán phân loại ảnh, mạng CNN được sử dụng phổ biến bởi tính đơn giản và hiệu quả. Cụ thể là trong bài toán phân loại video theo từng khung hình mà em đang giải quyết, em có thể sử dụng các mạng CNN đã được huấn luyện từ trước để làm lớp trích chọn đặc trưng, sau đó huấn luyện thêm các lớp dense ở cuối để nhận được kết quả phù hợp với bài toán. Đây còn gọi là kỹ thuật Transfer Learning. Sau

khi train được model hội tụ, em tiếp tục huấn luyện lại một số lớp của mạng trích chọn đặc trưng để thu được kết quả tốt hơn. Đây là quá trình Fine-tune.

### 3.3 Tensorflow

Với sự bùng nổ của lĩnh vực Trí Tuệ Nhân Tạo - AI trong thập kỷ vừa qua, machine learning và deep learning rõ ràng cũng phát triển theo cùng. Và ở thời điểm hiện tại, Tensorflow chính là thư viện mã nguồn mở cho machine learning nổi tiếng nhất thế giới, được phát triển bởi các nhà nghiên cứu từ Google. Việc hỗ trợ mạnh mẽ các phép toán học để tính toán trong machine learning và deep learning đã giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn nhiều.

Các hàm được dựng sẵn trong thư viện cho từng bài toán cho phép TensorFlow xây dựng được nhiều neural network. Nó còn cho phép bạn tính toán song song trên nhiều máy tính khác nhau, thậm chí trên nhiều CPU, GPU trong cùng 1 máy hay tạo ra các dataflow graph – đồ thị luồng dữ liệu để dựng nên các model. Nếu bạn muốn chọn con đường sự nghiệp trong lĩnh vực A.I. này, nắm rõ những điều cơ bản của TensorFlow thực sự rất quan trọng.

Được viết bằng C++ và thao tác interface bằng Python nên phần performance của TensorFlow cực kỳ tốt. Đối tượng sử dụng nó cũng đa dạng không kém: từ các nhà nghiên cứu, nhà khoa học dữ liệu và dĩ nhiên không thể thiếu các lập trình viên.

Kiến trúc Tensorflow hoạt động được chia thành 3 phần: Tiền xử lý dữ liệu, dựng model, train và ước tính model.

Với bài toán là phân loại ảnh dựa trên các nhãn, em sử dụng Tensorflow để compile, train và fine-tune model dựa trên mạng InceptionV3 với số epoch là 100, số step trong mỗi epoch là 500 và sử dụng metrics là accuracy. Kết quả thu được một model có khả năng gán nhãn ảnh với các nhãn tương ứng.

Từ model thu được, em tiếp tục sử dụng Tensorflow để dự đoán nhãn của các ảnh mới dùng để lọc video theo từng frame.

### 3.4 OpenCV

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV được viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOs lẫn Android, iOS OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần.

Opencv có rất nhiều ứng dụng như: Nhận dạng ảnh, xử lý hình ảnh, phục hồi hình ảnh/video, thực tế ảo và các ứng dụng khác.

Trong bài toán phân loại video, em chủ yếu sử dụng OpenCv để xử lý dữ liệu đầu vào, cắt video thành từng frame ảnh và resize hình ảnh, chuyển về dạng mảng.

### 3.5 Django

Django là một web framework miễn phí mã nguồn mở được viết bằng Python. Django sử dụng mô hình Model-View-Template (MVT). Mô hình MVT được sử dụng trong khi tạo một ứng dụng với Tương tác người dùng. Mô hình này bao gồm code HTML với Django Template Language (DTL). Controller là mã được viết để kiểm soát sự tương tác giữa Model và View và Django tự động xử lý được công việc này. Bất cứ khi nào người dùng gửi request, nó xử lý request của người dùng đó bằng Model, View và Template. Nó hoạt động như một Controller để kiểm tra xem nó có khả dụng hay không bằng cách ánh xạ URL và nếu URL ánh xạ thành công thì View sẽ bắt đầu tương tác với Model và gửi lại Template cho người dùng dưới dạng response.

Mục tiêu chính của Django là đơn giản hóa việc tạo các website phức tạp có sử dụng cơ sở dữ liệu. Django tập trung vào tính năng “có thể tái sử dụng” và “có thể tự chạy” của các component, tính năng phát triển nhanh, không làm lại những gì đã làm.

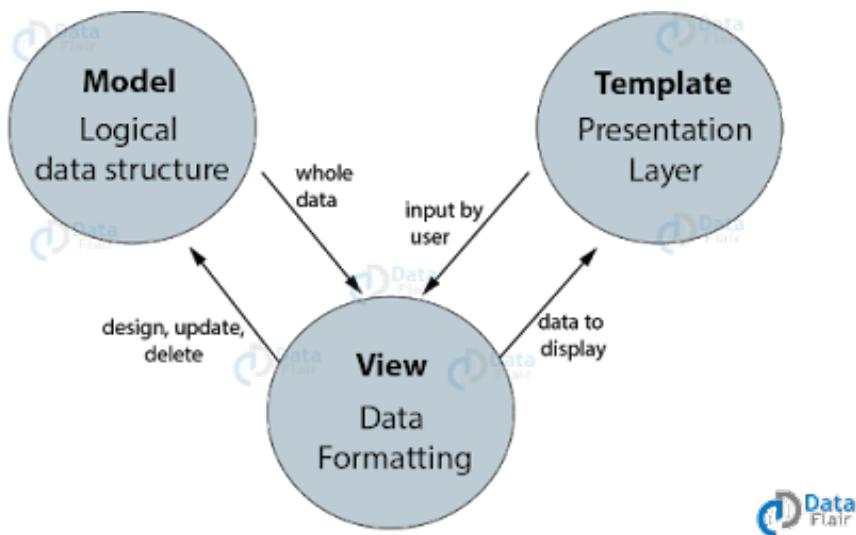
Với bài toán không quá phức tạp và việc tích hợp model deep learning trong phân loại video, em chọn ngôn ngữ lập trình là Python và framework Django để đơn giản hóa công việc lập trình web và tập trung vào bài toán xử lý nhận dạng video.

## CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

### 4.1 Thiết kế kiến trúc

#### 4.1.1 Lựa chọn kiến trúc phần mềm

Vì sử dụng Framework Django nên kiến trúc em sử dụng sẽ là mô hình MVT. Mô hình MVT là biến thể của mô hình MVC. MVT là viết tắt của Model-View-Template. Về bản chất MVT và MVC đều giống nhau, đều chia ứng dụng ra thành ba thành phần: phần xử lý logic, phần hiển thị và phần thao tác với cơ sở dữ liệu.



Hình 4.1: Minh họa cấu trúc MVT

Sự khác biệt của mô hình MVT và MVC được minh họa trong bảng sau:

Bảng 4.1: MVT và MVC

Mô hình MVC	Mô hình MVT	Chức năng
Model	Model	Làm việc với cơ sở dữ liệu
View	Template	Hiển thị giao diện
Controller	View	Xử lý logic

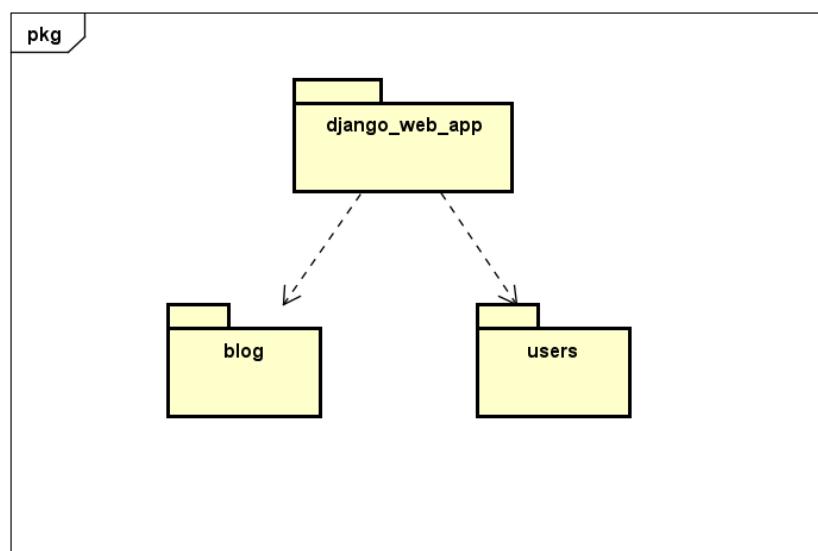
Template nghĩa là bản mẫu, là công cụ để tạo ra các giao diện web. Trong một bản mẫu sẽ có những phần cố định (static) và những phần được bổ sung sau (dynamic). Django có hai công cụ để tạo các template là DTL và Jinja2. Để tạo ra các template, bạn sẽ sử dụng cú pháp của một trong hai công cụ là DTL hoặc Jinja2, sau đó chuyển template cho template engine để chuyển sang mã HTML.

Nhiệm vụ của Model là thực hiện quản lý dữ liệu. Nghĩa là Model sẽ kiểm soát mọi sự thay đổi của dữ liệu cũng như việc giao tiếp CRUD giữa BE và Database.

Mục đích của View là xử lý logic nghiệp vụ thông qua các url được định nghĩa. Khi có một request từ client, View sẽ là trung tâm để thực hiện các yêu cầu của request đó và sẽ trả về (response) dữ liệu theo yêu cầu.

Đối với trang Web em thiết kế, do quy mô nhỏ gọn nên em sử dụng sẵn Template Engine của Django và cơ sở dữ liệu nhúng Sqlite3. Ưu điểm của chúng là được hỗ trợ sẵn bởi Django và không cần tùy chỉnh quá nhiều.

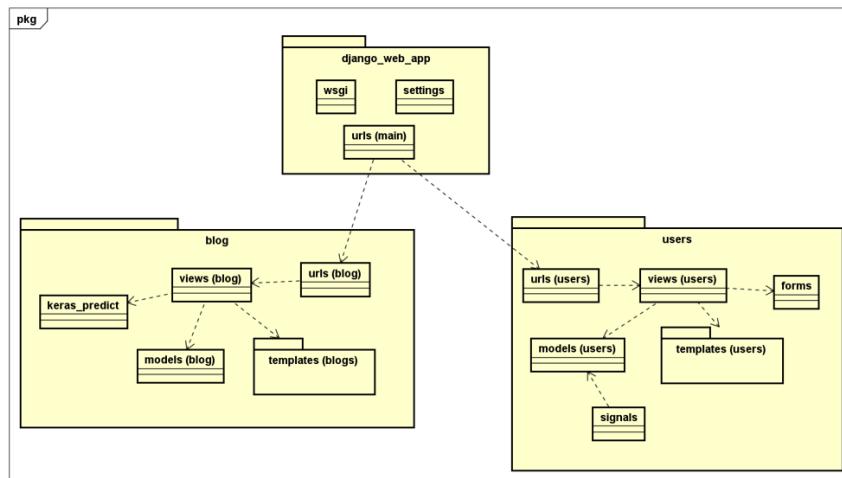
#### 4.1.2 Thiết kế tổng quan



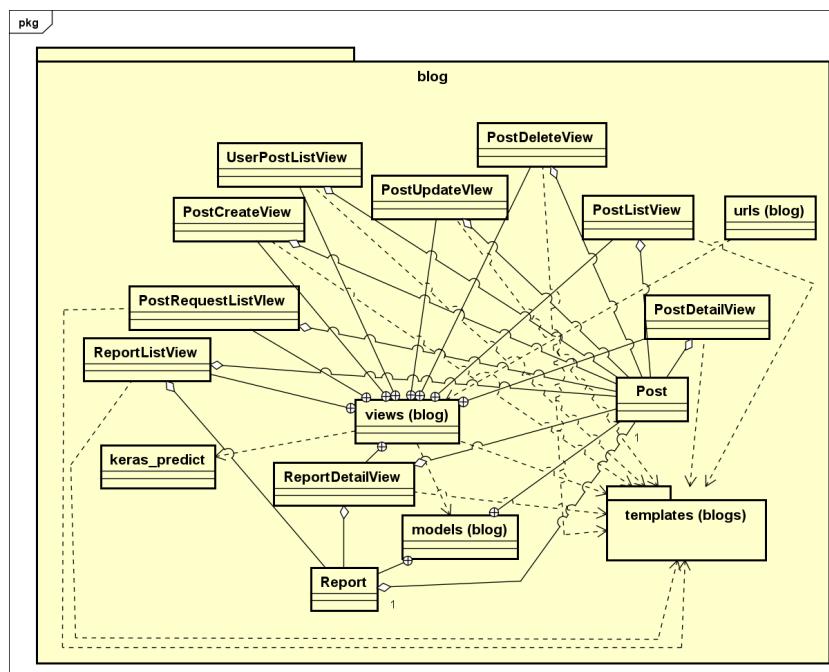
**Hình 4.2:** Biểu đồ gói

Trong Django, với một Project sẽ có một Package chính (ở đây là Django web app) để phụ trách kiểm soát toàn bộ cài đặt và thiết lập của hệ thống. Từ đó, em xây dựng thêm hai App là các package thành phần với cấu trúc MVT riêng biệt của chúng để được sử dụng trong Package chính dưới vai trò là các hệ thống con phục vụ một nhóm chức năng liên quan. Trong đó Package Blog là Package sẽ phụ trách các UC liên quan đến bài đăng còn Users sẽ phụ trách các UC liên quan đến người dùng.

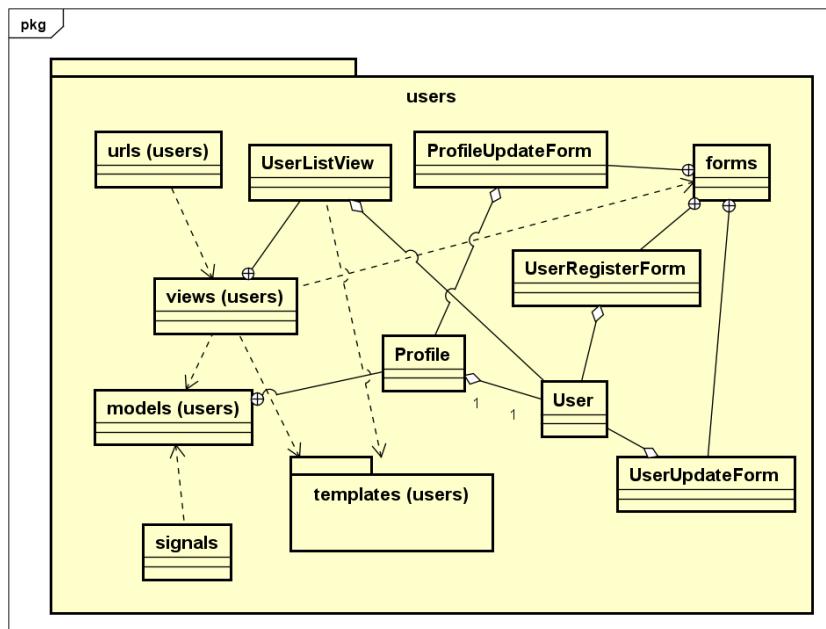
#### 4.1.3 Thiết kế chi tiết gói



**Hình 4.3:** Thiết kế gói



**Hình 4.4:** Thiết kế gói Blog

**Hình 4.5:** Thiết kế gói Users

Như đã nêu ở phần trước, Package Django web app là Package chính phụ trách cài đặt và thiết lập để chạy trang web, các cài đặt này nằm ở file settings.py và wsgi.py. Ngoài ra, để liên kết các Package con hay các hệ thống con vào trang web chính cần có một lớp urls của Package này (urls main) trong đó include các file urls của các Package con.

Đối với các App cần có ít nhất ba thành phần chính là models, views và urls. Ngoài urls ra thì em sử dụng chính kiến trúc MVT để đặt các file html vào phần templates sau đó sử dụng chúng trong views. Các Object trong cơ sở dữ liệu cũng được thao tác trong phần views.

Với App Blog, em có thêm lớp Keras predict để tích hợp bộ lọc video. Còn với Users sẽ sử dụng thêm Forms và Signals để tùy chỉnh các Form tương tác với cơ sở dữ liệu và cài đặt các hàm Callback.

Trong Django, lớp Views đóng vai trò như một Package View trong mô hình MVT. Nó sẽ chứa tất cả các lớp và phương thức tương ứng với các API ở phía Back End trong gói Control của mô hình MVC.

Có 2 cách để định nghĩa một API trong Views: Sử dụng các lớp con kế thừa Generic View của Django được định nghĩa sẵn trong Framework hoặc định nghĩa các hàm với tham số là Http Request sẽ được gửi tới và trả về Http Response tương ứng.

Tương tự như Views, lớp Models cũng đóng vai trò như một Package. Điểm khác duy nhất là Models chỉ định nghĩa các đối tượng trong cơ sở dữ liệu qua các

lớp con.

Trong Django có định nghĩa sẵn một số lớp Model bao gồm User với đầy đủ chức năng quản lý người dùng nên em sử dụng lớp này như ở trên để định nghĩa thêm lớp Profile của người dùng.

## 4.2 Thiết kế chi tiết

### 4.2.1 Thiết kế giao diện

#### a, Thông tin màn hình

Độ phân giải: 1920 x 1080 pixels

Kích thước màn hình: 24 inch/ 27 inch

Số lượng màu hỗ trợ: 16,777,216 màu

#### b, Chuẩn hóa giao diện

##### Screen

Vị trí button: Góc trên hoặc góc dưới bên phải của khung

Vị trí của message: Ở trên cùng theo chiều dọc và ở giữa theo chiều ngang màn hình

Vị trí của screen title: Title ở góc trên bên trái màn hình

##### Control

Text: Dùng font mặc định của Bootstrap

Xử lý check input: Kiểm tra xem input có empty hay không

Dịch chuyển màn hình: Có sử dụng Pop-up khi điền Form báo cáo hoặc chặn Video. Ngoài ra các màn hình là tách biệt và không có các khung chồng lên nhau.

##### Color

Màu nền (màu chủ đạo) là màu trắng

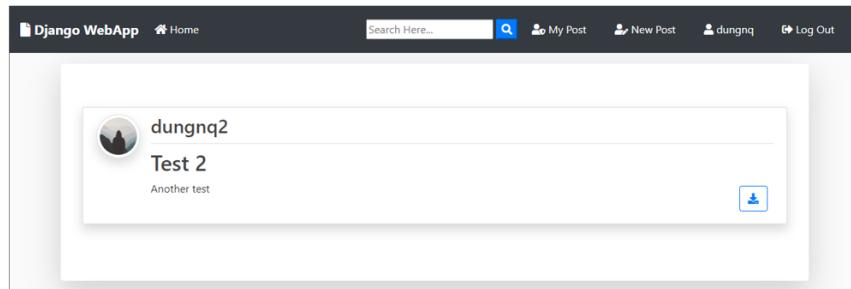
Thanh điều khiển (màu phụ) là màu xám

Các nút/thông báo thành công có màu xanh lá, chặn/bị chặn có màu đỏ

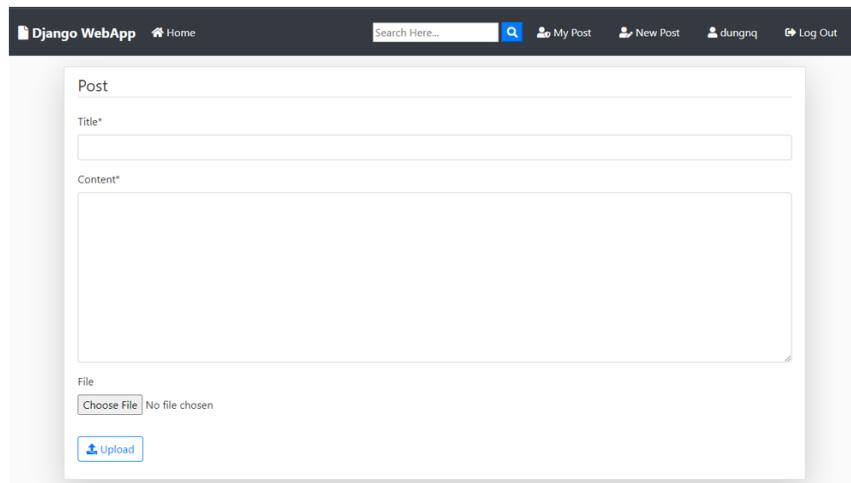
Thông báo thông thường có màu xám

Nút thông thường có màu xanh hoặc xám

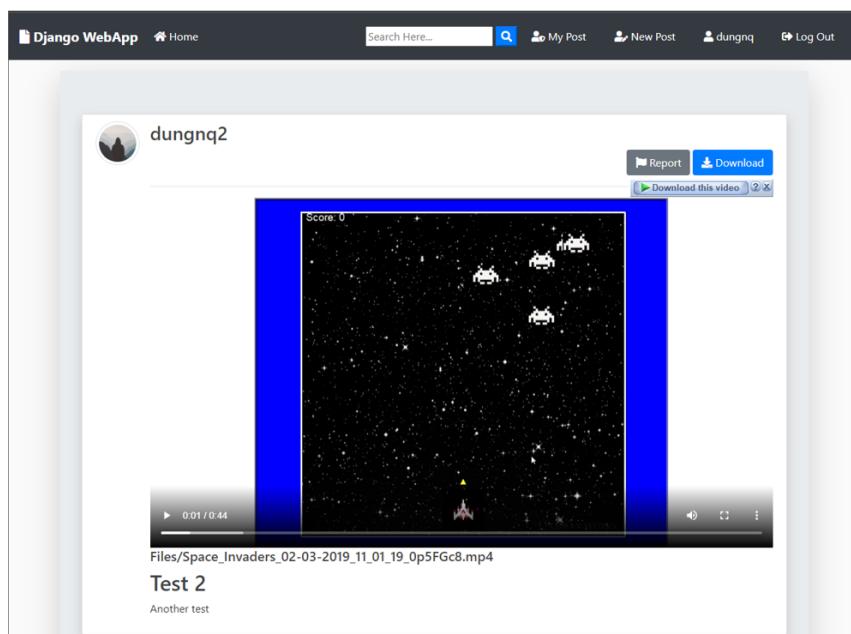
c, Hình ảnh minh họa



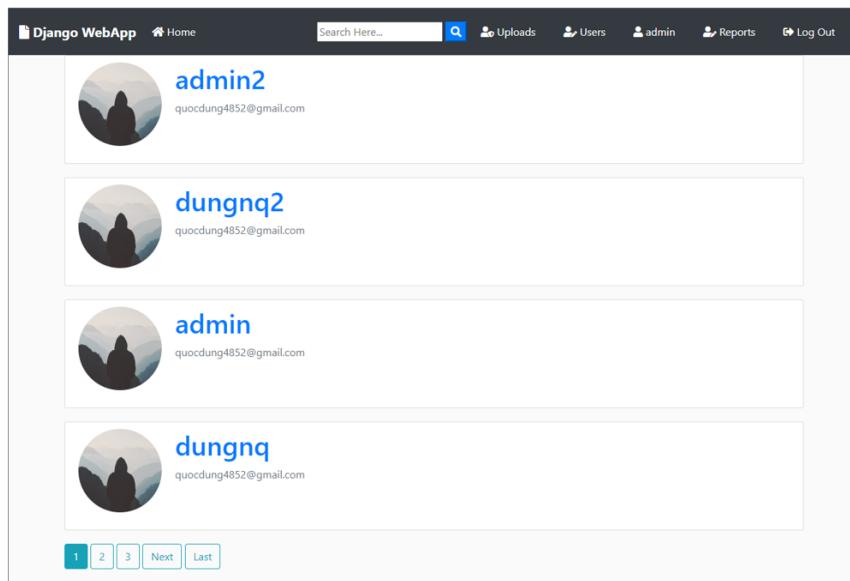
Hình 4.6: Trang chủ



Hình 4.7: Đăng Video

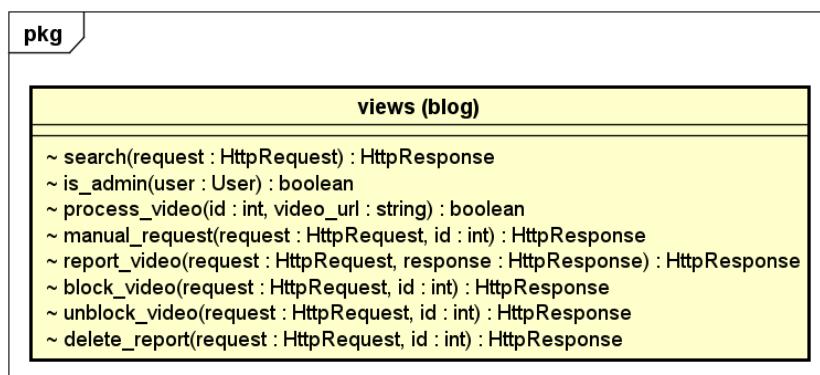


Hình 4.8: Xem Video

**Hình 4.9:** Xem thông tin người dùng

### 4.2.2 Thiết kế lớp

#### a, Lớp Views (Blog)

**Hình 4.10:** Lớp Views (Blog)

#### Attribute

*Không có*

#### Operation

**Bảng 4.2:** Operation của lớp Views

Tên	Kiểu trả về	Mô tả (Mục đích)
search	HttpResponse	Tìm kiếm bài đăng và người dùng tương ứng với từ khóa tìm kiếm

manual_request	HttpResponse	Yêu cầu duyệt Video bằng tay đến Admin (thực hiện bởi người dùng)
report_video	HttpResponse	Báo cáo Video vi phạm tiêu chuẩn đến Admin (thực hiện bởi người dùng)
block_video	HttpResponse	Chặn video vi phạm (thực hiện bởi Admin)
unblock_video	HttpResponse	Bỏ chặn video vi phạm (thực hiện bởi Admin)
delete_report	HttpResponse	Xóa các báo cáo vi phạm của người dùng trên một bài đăng (thực hiện bởi Admin)

*Parameter:*

- Request: Thông điệp Http Request được gửi đến từ người dùng, có thể chứa các thông tin nếu là Http Post Request
- Id: Khóa chính của bài đăng/video tương ứng trong cơ sở dữ liệu

*Exception:*

- Not found 404 nếu không tìm được bài đăng tương ứng
- Forbidden nếu không có quyền truy cập view

### Method

**Bảng 4.3:** Method của lớp Views

Tên	Kiểu trả về	Mô tả (Mục đích)
is_admin	Boolean	Kiểm tra xem người dùng có phải là Admin không
process_video	Boolean	Phân loại Video và lọc sử dụng Model Deep Learning

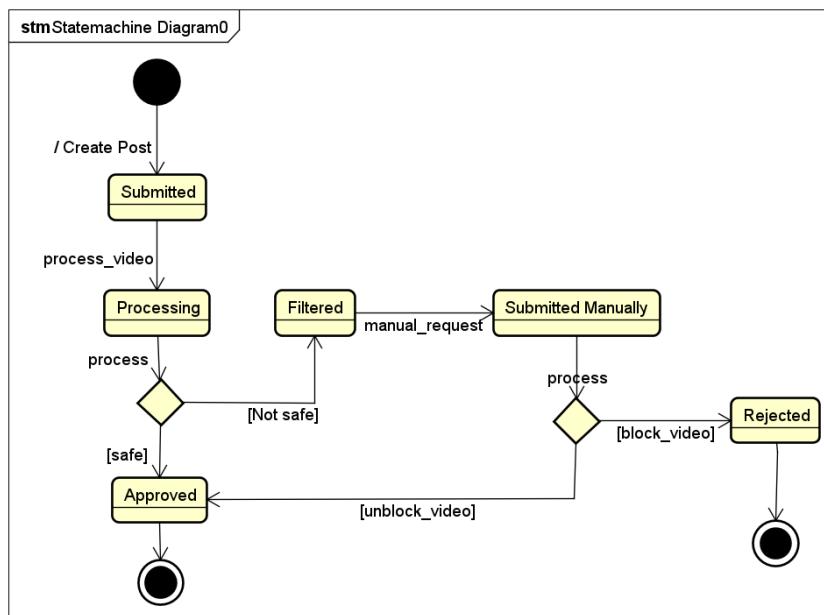
*Parameter:*

- User: Đối tượng người dùng trong Model
- Id: Khóa chính của bài đăng/video tương ứng trong cơ sở dữ liệu
- Video\_url: Đường dẫn đến Video trên Server

*Exception:*

*Không*

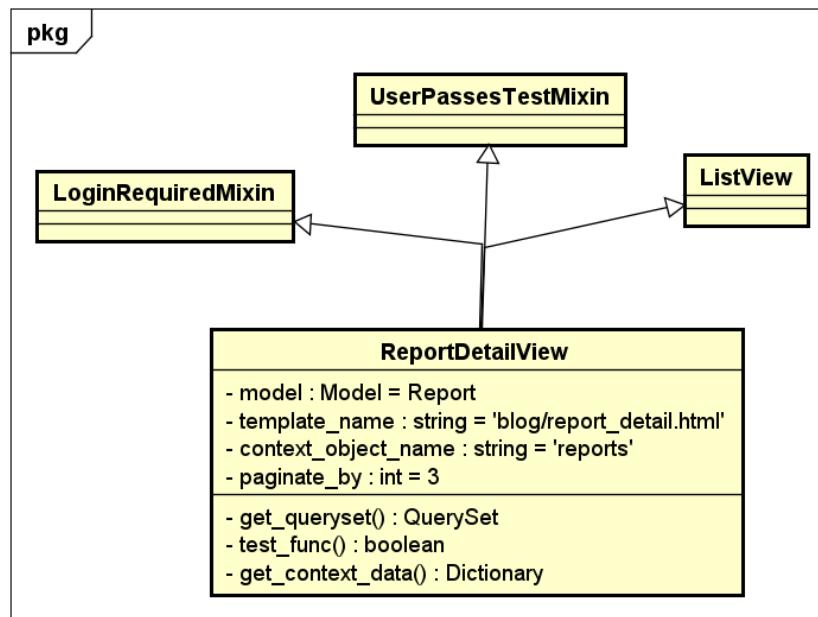
## State



**Hình 4.11:** Biểu đồ trạng thái Video

Khi Video được đăng tải bài đăng sẽ ở vào trạng thái Submitted, lúc này nếu Admin thực hiện yêu cầu process-video thành công thì Video sẽ được chuyển sang trạng thái Processing. Nếu hệ thống tự động phân loại Video là phù hợp thì Video sẽ được Approved đồng nghĩa với việc được đăng tải lên trang chủ. Nếu hệ thống phân loại Video là không phù hợp nó sẽ ở trạng thái Filtered. Khi đó người dùng có thể yêu cầu Manual Request để Admin duyệt bằng tay và chuyển sang trạng thái Submitted Manually. Admin khi nhìn thấy yêu cầu sẽ quyết định xem phê duyệt hay từ chối Video. Nếu từ chối thì Video này sẽ ở trạng thái Rejected và không được đăng tải.

## b, Lớp ReportDetailView



**Hình 4.12:** Lớp ReportDetailView

Trong Python, một lớp có thể kế thừa từ nhiều lớp khác. Với Django, khi muốn định nghĩa API trong Views sử dụng lớp con ta cần kế thừa một trong số các lớp GenericViews được định nghĩa sẵn trong Framework (ở đây là ListView). Ngoài ra em còn sử dụng thêm cả lớp LoginRequiredMixin và UserPassesTestMixin để kiểm soát quyền truy cập đến API này. Khi sử dụng GenericView, ta chỉ cần định nghĩa các thuộc tính như model và template-name là Django sẽ tự động sinh ra View tương ứng bằng cách lấy trực tiếp Model và đưa vào Template. Nhưng do em muốn tùy chỉnh việc lấy Model nên cần override thêm hàm get-queryset để trả về những đối tượng mong muốn.

### Attribute

**Bảng 4.4:** Attribute của lớp ReportDetailView

Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
model	Model	Report	Model Report trong lớp Models của Blog

template_name	String	blog/report_detail.html	Đường dẫn đến file Html tương ứng trong thư mục template
context_object_name	String	reports	Tên của biến trong context sẽ được truyền vào template khi dùng template engine
paginate_by	int	3	Số bảng trong một trang (phục vụ phân trang)

## Operation

**Bảng 4.5:** Operation của lớp ReportDetailView

Tên	Kiểu trả về	Mô tả (Mục đích)
get_queryset	QuerySet	Truy vấn đến cơ sở dữ liệu trong Model để lấy các bảng Report tương ứng với Video đang xem
test_func	Boolean	Kiểm tra xem người truy cập có phải là Admin không
get_context_data	Dictionary	Gửi các dữ liệu tương ứng vào Context để truyền đến Template Engine

*Parameter:*

*Không*

*Exception:*

- Not found 404 nếu không tìm được bài đăng tương ứng
- Forbidden nếu không có quyền truy cập view

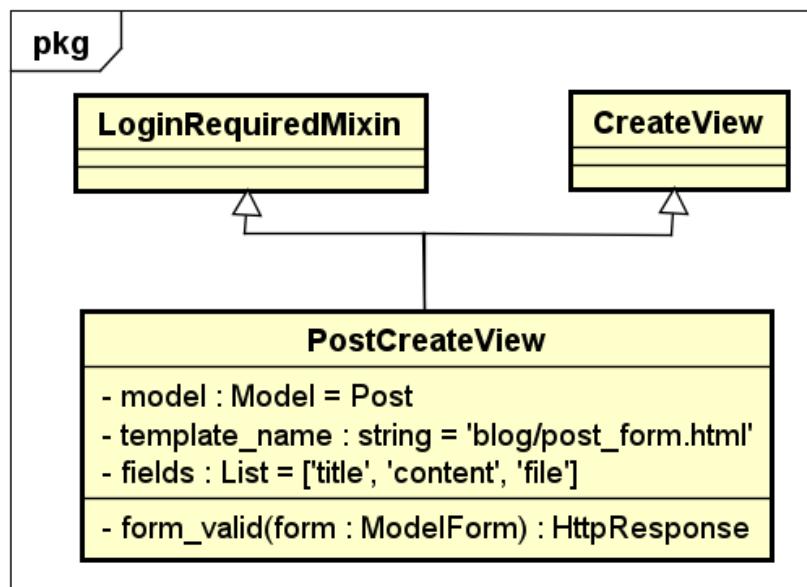
## Method

*Không*

## State

*Không*

### c, Lớp PostCreateView



**Hình 4.13:** Lớp PostCreateView

Khi một lớp con kết thừa lớp CreateView, nó sẽ tự động sử dụng một lớp ModelForm có sẵn của Django để tạo Form giúp người dùng tạo một bảng mới trong cơ sở dữ liệu hay một đối tượng mới trong Models. Ở đây em sử dụng thuộc tính fields để chỉ rõ những trường sẽ xuất hiện trong form và sử dụng thêm hàm form-valid để override mặc định của Django là sẽ tự động lưu đối tượng mới vào Models để thêm các xử lý cho Video.

#### Attribute

**Bảng 4.6:** Attribute của lớp PostCreateView

Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
model	Model	Post	Model Post trong lớp Models của Blog
template_name	String	blog/post_form.html	Đường dẫn đến file Html tương ứng trong thư mục template

fields	List	[‘title’, ‘content’, ‘file’]	Tên của các trường sẽ xuất hiện trong form để người dùng nhập các giá trị cho bảng mới
--------	------	------------------------------	--

## Operation

**Bảng 4.7:** Operation của lớp PostCreateView

Tên	Kiểu trả về	Mô tả (Mục đích)
form_valid	HttpResponse	Xử lý dữ liệu được điền trong form bởi người dùng và trả về kết quả

*Parameter:*

- Form: Form của Model tương ứng (Post)

*Exception:*

- Forbidden nếu không có quyền truy cập view

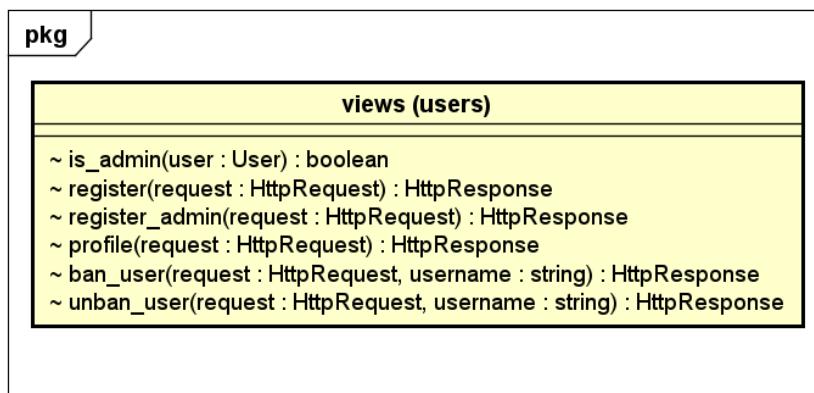
## Method

*Không*

## State

*Không*

## d, Lớp Views (Users)



**Hình 4.14:** Lớp Views (Users)

## Attribute

*Không*

## Operation

**Bảng 4.8:** Operation của lớp Views (Users)

Tên	Kiểu trả về	Mô tả (Mục đích)
register	HttpResponse	Đăng ký tài khoản người dùng
register_admin	HttpResponse	Đăng ký tài khoản Admin (chỉ thực hiện bởi Admin)
profile	HttpResponse	Xem thông tin cá nhân
ban_user	HttpResponse	Chặn người dùng, không cho phép người dùng đăng bài (thực hiện bởi Admin)
unban_user	HttpResponse	Bỏ chặn người dùng (thực hiện bởi Admin)

*Parameter:*

- Request: Thông điệp Http Request được gửi đến từ người dùng, có thể chứa các thông tin nếu là Http Post Request
- Username: Tên người dùng của tài khoản đích

*Exception:*

- Not found 404 nếu không tìm được người dùng tương ứng
- Forbidden nếu không có quyền truy cập view

## Method

**Bảng 4.9:** Method của lớp Views (Users)

Tên	Kiểu trả về	Mô tả (Mục đích)
is_admin	Boolean	Kiểm tra xem người dùng có phải là Admin không

*Parameter:*

- User: Đối tượng người dùng trong Model

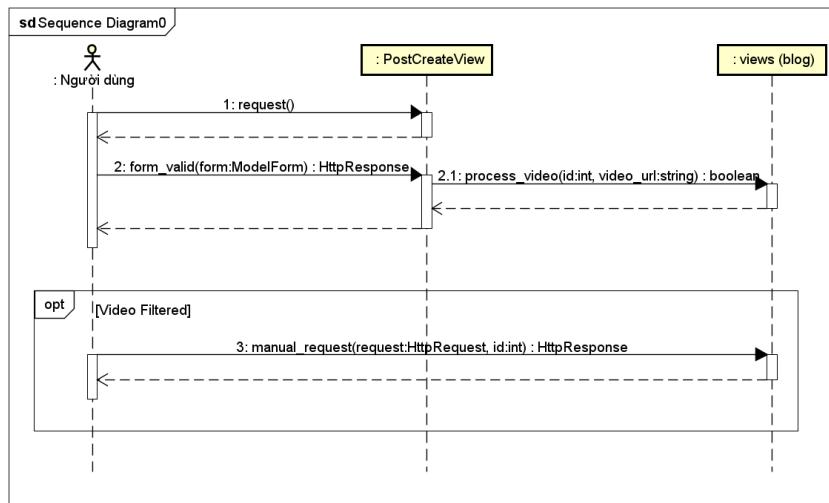
*Exception:*

*Không*

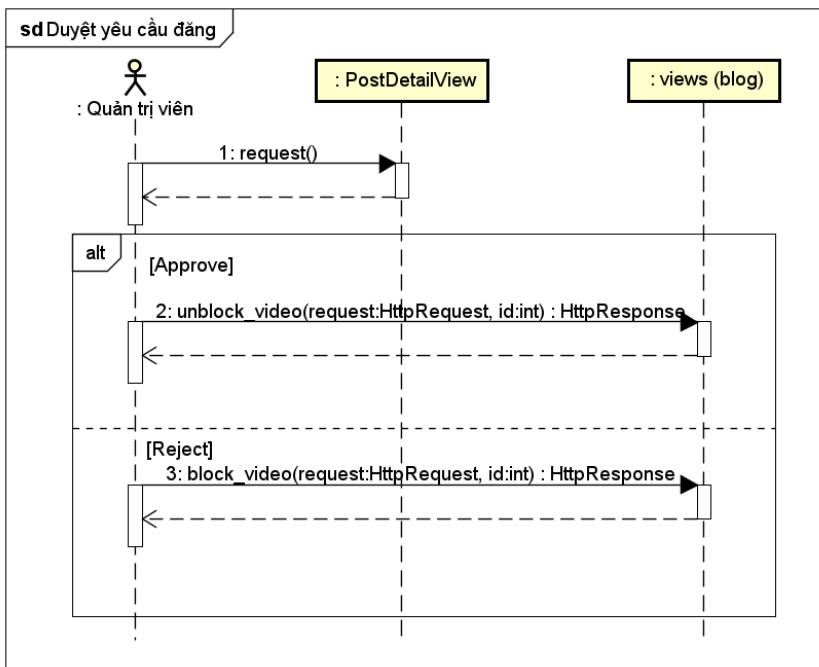
## State

*Không*

### e, Biểu đồ trình tự các Use Case quan trọng



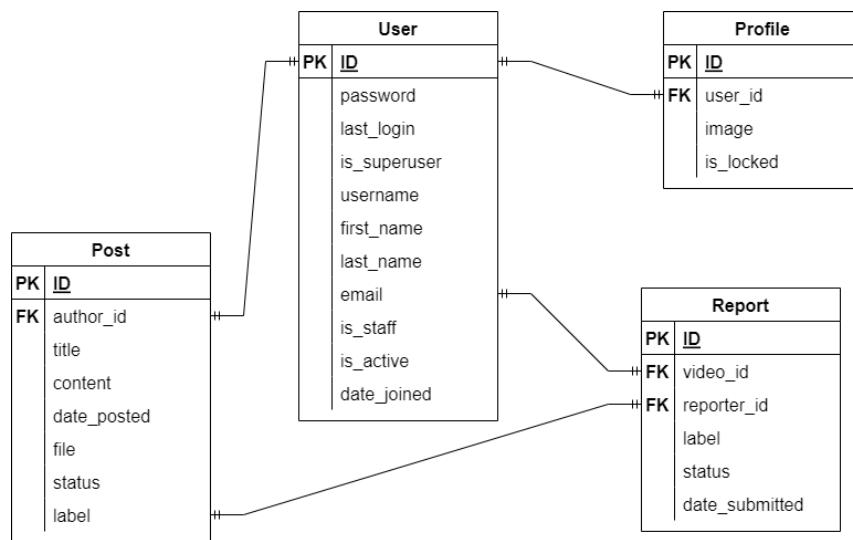
Hình 4.15: Biểu đồ trình tự UC đăng Video



Hình 4.16: Biểu đồ trình tự UC Duyệt yêu cầu đăng Video

### 4.2.3 Thiết kế cơ sở dữ liệu

#### a, Biểu đồ thực thể liên kết



Hình 4.17: Biểu đồ thực thể liên kết

Bảng người dùng đã được định nghĩa sẵn trong Django với các trường chứa thông tin cá nhân, quyền và trạng thái của người dùng. Bảng Profile dùng để thêm ảnh đại diện cho tài khoản và kiểm tra tài khoản có bị khóa hay không. Hai bảng Post và Report đều có trường status để chứa thông tin trạng thái và label chứa thông tin phân loại nhãn.

#### b, Thiết kế chi tiết từng bảng

Bảng 4.10: Bảng User

#	PK	FK	Tên cột	Kiểu dữ liệu	Bắt buộc	Mô tả
1	x		id	integer	có	id, auto increment
2			password	varchar(128)	có	Kiểm tra người dùng có phải quản trị viên
3			last_login	datetime	không	Mật khẩu được lưu dưới dạng băm
4			is_superuser	bool	có	Kiểm tra người dùng có phải quản trị viên
5			username	varchar(150)	có	Tên tài khoản
6			first_name	varchar(30)	có	Tên người dùng

7			last_name	varchar(150)	có	Họ người dùng
8			email	varchar(254)	có	Email người dùng
9			is_staff	bool	có	Kiểm tra người dùng có phải nhân viên
10			is_active	bool	có	Kiểm tra người dùng có còn hoạt động
11			date_joined	datetime	có	Ngày đăng ký tài khoản

**Bảng 4.11:** Bảng Profile

#	PK	FK	Tên cột	Kiểu dữ liệu	Bắt buộc	Mô tả
1	x		id	integer	có	id, auto increment
2			image	varchar(100)	có	Đường dẫn đến hình ảnh đại diện
3		x	user_id	integer	có	Khóa ngoài của người dùng tương ứng
4			is_locked	bool	có	Kiểm tra người dùng có bị khóa tài khoản không

**Bảng 4.12:** Bảng Post

#	PK	FK	Tên cột	Kiểu dữ liệu	Bắt buộc	Mô tả
1	x		id	integer	có	id, auto increment
2			title	varchar(100)	có	Tiêu đề bài đăng
3			content	text	có	Nội dung bài đăng
4			date_posted	datetime	có	Ngày đăng bài
5		x	author_id	integer	có	Khóa ngoài đến người dùng thực hiện đăng bài
6			file	varchar(100)	có	Đường dẫn đến Video được đăng

7			status	varchar(1)	có	Trạng thái của bài đăng
8			label	text	có	Nhãn phân loại Video

**Bảng 4.13:** Bảng Report

#	PK	FK	Tên cột	Kiểu dữ liệu	Bắt buộc	Mô tả
1	x		id	integer	có	id, auto increment
2			date_submitted	datetime	có	Ngày báo cáo
3		x	reporter_id	integer	có	Khóa ngoài đến người dùng gửi báo cáo
4		x	video_id	integer	có	Khóa ngoài đến bài đăng bị báo cáo
5			status	varchar(1)	có	Trạng thái của báo cáo
6			label	text	có	Nhãn phân loại Video

### 4.3 Xây dựng ứng dụng

#### 4.3.1 Thư viện và công cụ sử dụng

Mục đích	Công cụ	Địa chỉ URL
Code Editor	Visual Studio Code	<a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a> v1.69.2
Ngôn ngữ lập trình Back-end	Python v3.10.4	<a href="https://www.python.org/">https://www.python.org/</a>
Thư viện Frame- work Web	Django v3.0	<a href="https://www.djangoproject.com/">https://www.djangoproject.com/</a>
Thư viện CSS	Bootstrap v4.3.1	<a href="https://getbootstrap.com/">https://getbootstrap.com/</a>
Thư viện hàng đợi phía Server	Django- background-tasks	<a href="https://django-background-tasks.readthedocs.io/en/latest/">https://django-background-tasks.readthedocs.io/en/latest/</a>
Thư viện Deep Learning	Tensorflow v2.8.0	<a href="https://www.tensorflow.org/">https://www.tensorflow.org/</a>

Scripts tự động tải dữ liệu ảnh	Nsfw_data_scraper	<a href="https://github.com/alex000kim/nsfw_data_scraper/">https://github.com/alex000kim/nsfw_data_scraper/</a>
Công cụ tìm kiếm file trùng lặp	Duplicate-file-finder	<a href="https://github.com/Qarj/duplicate-file-finder">https://github.com/Qarj/duplicate-file-finder</a>
App tìm kiếm file ảnh có nội dung giống nhau	DupeGuru	<a href="https://dupeguru.voltaicideas.net/">https://dupeguru.voltaicideas.net/</a>
Vẽ biểu đồ UML	Astah UML v8.5.0	<a href="https://astah.net/">https://astah.net/</a>

**Bảng 4.14:** Danh sách thư viện và công cụ sử dụng

### 4.3.2 Kết quả đạt được

Sử dụng các công cụ có sẵn trong Framework Django để debug và chạy Server, trang Web có thể chạy thành công trên máy local với đầy đủ các tính năng đã cài đặt. Ứng dụng được chia thành 2 App chính là Blog và Users với các nhóm chức năng liên quan tới bài đăng và người dùng. Việc sử dụng Django cũng giúp cho quá trình chạy Server nhanh gọn và có thể Debug dễ dàng.

Tên Ngôn ngữ	Số file	Số dòng code
Html	15	832
Python	33	1037
CSS	1	85

**Bảng 4.15:** Danh sách ngôn ngữ sử dụng

Tên Package	Số file	Số dòng code	Dung lượng
Django_web_app	4	187	14.8KB
Blog	25	1348	193MB
Users	18	388	28.1KB
Toàn bộ mã nguồn	49	2328	194.56MB

**Bảng 4.16:** Danh sách Package

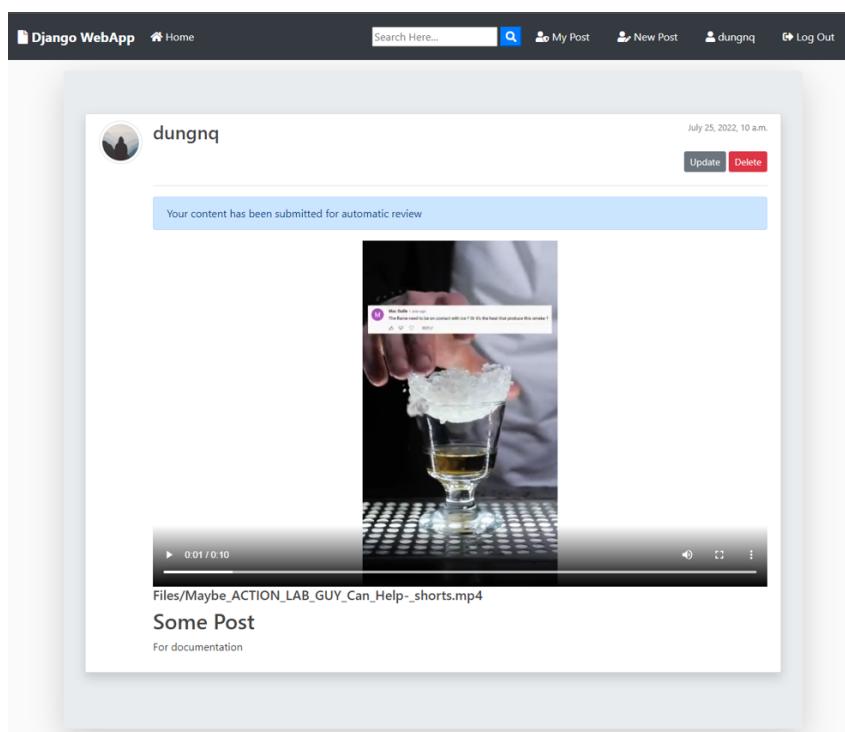
### 4.3.3 Minh họa các chức năng chính

#### a, Đăng Video

The screenshot shows a web browser window with the title 'Django WebApp'. At the top, there's a navigation bar with links for 'Home', 'Search Here...', 'My Post', 'New Post', 'dungnq', and 'Log Out'. Below the navigation bar is a form titled 'Post'. The form has three main sections: 'Title\*' with a text input field, 'Content\*' with a large text area, and 'File\*' with a file upload section containing a 'Choose File' button and a message 'No file chosen'. At the bottom of the form is a blue 'Upload' button.

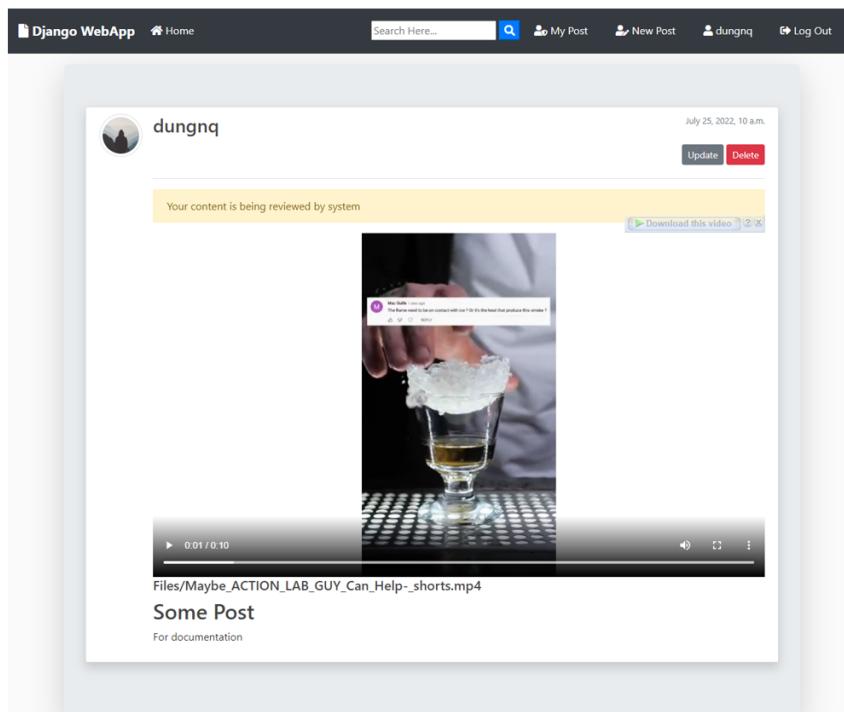
**Hình 4.18:** Form đăng Video

Khi muốn đăng Video, người dùng chọn nút New Post trên thanh điều khiển để hiện ra Form tạo Post. Người dùng cần điền đầy đủ các thông tin của bài đăng trước khi nhấn Upload. Sau khi tạo bài đăng thành công, bài đăng sẽ ở trạng thái Submitted và người dùng có thể kiểm tra trạng thái bài đăng bất cứ lúc nào bằng cách vào trang My Post. Người dùng cũng có thể cập nhật và xóa bài đăng. Tuy nhiên không thể thay đổi Video trong Post.



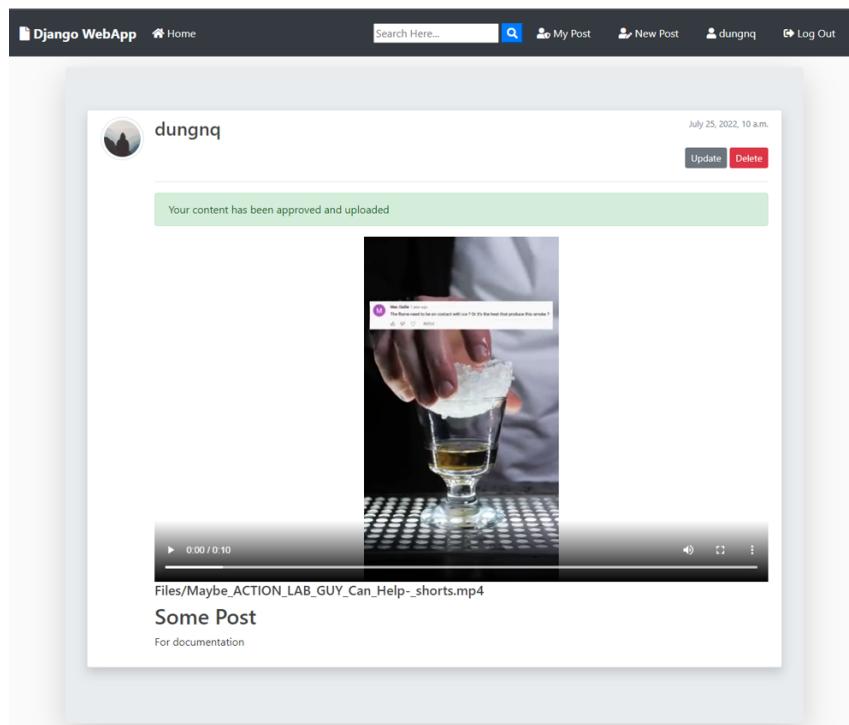
**Hình 4.19:** Bài đăng đã được nộp

Bài đăng sẽ được tự động phân loại bởi hệ thống và được chuyển thành trạng thái Processing khi hệ thống đang trong quá trình phân loại.

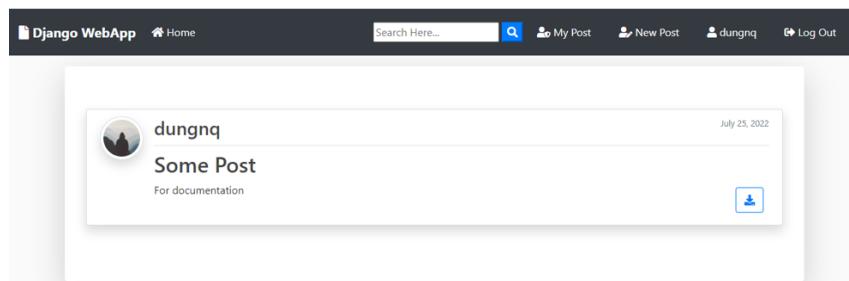


**Hình 4.20:** Bài đăng đang được xử lý

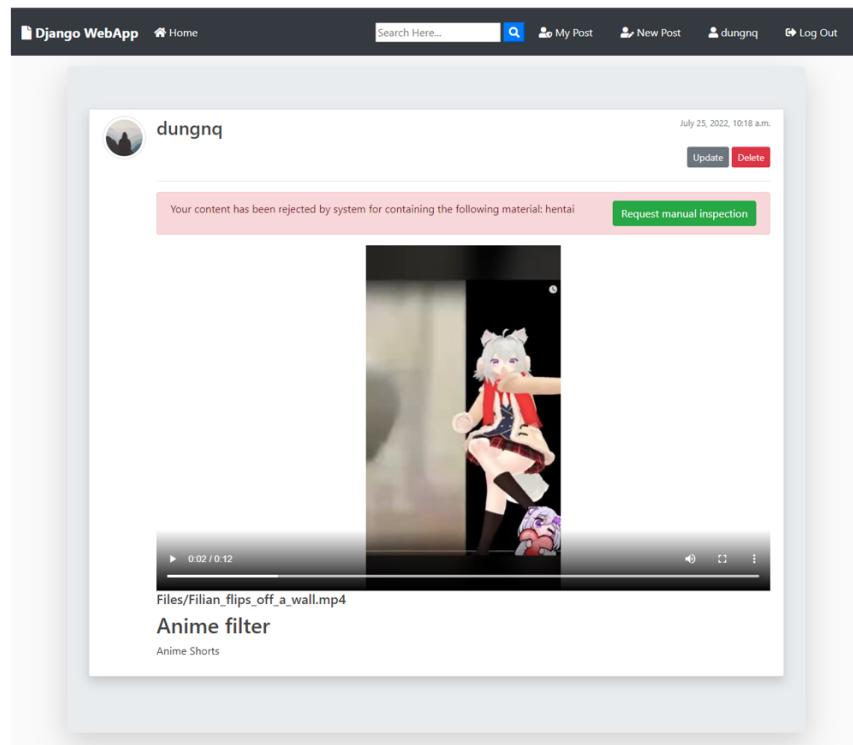
Sau khi bài đăng đã được xử lý, nếu được duyệt thì bài đăng sẽ ở trạng thái Approved và tự động xuất hiện trên trang chủ. Ngược lại, người dùng có thể sử dụng chức năng yêu cầu duyệt bằng tay để gửi yêu cầu đến Admin.



**Hình 4.21:** Bài đăng đã được duyệt bởi hệ thống

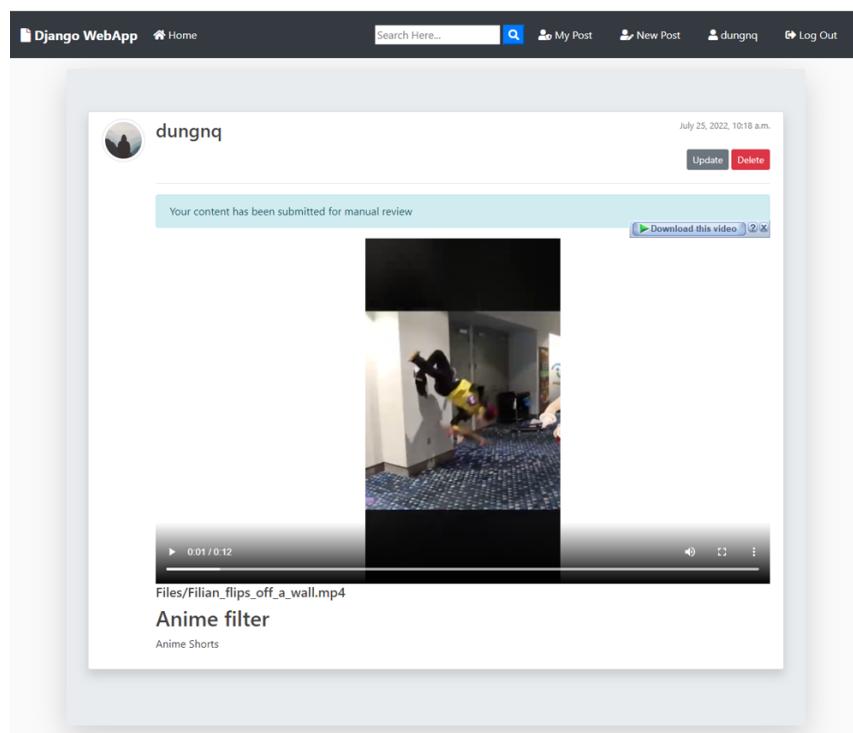


**Hình 4.22:** Bài đăng xuất hiện trên trang chủ

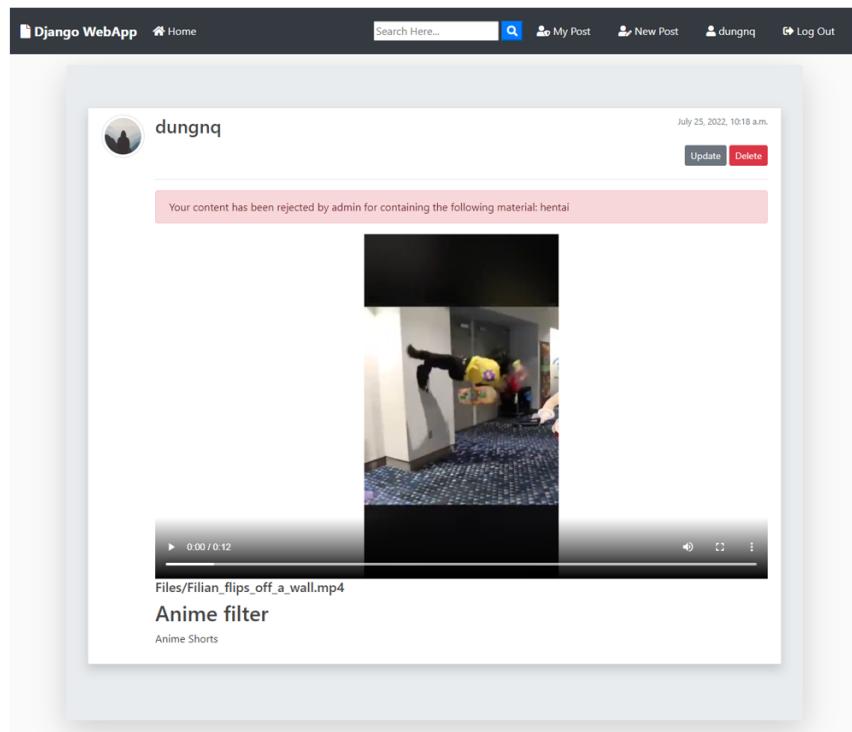


**Hình 4.23:** Bài đăng bị từ chối bởi hệ thống

Sau khi chọn yêu cầu duyệt bằng tay, Video sẽ lại chuyển sang trạng thái chờ duyệt, lúc này nếu Admin từ chối yêu cầu thì bài đăng sẽ không thể xuất hiện lên trang chủ cho đến khi quản trị viên chấp thuận. Ngược lại bài đăng sẽ lại được chuyển sang trạng thái Approved và đăng tải.

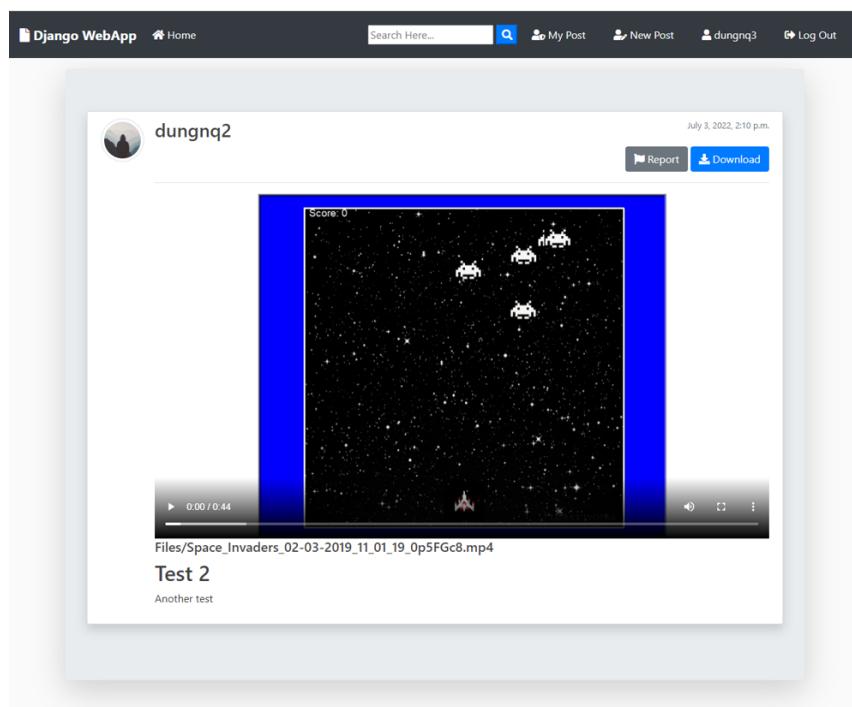


**Hình 4.24:** Bài đăng đang chờ duyệt bằng tay



**Hình 4.25:** Bài đăng bị từ chối bởi Quản trị viên

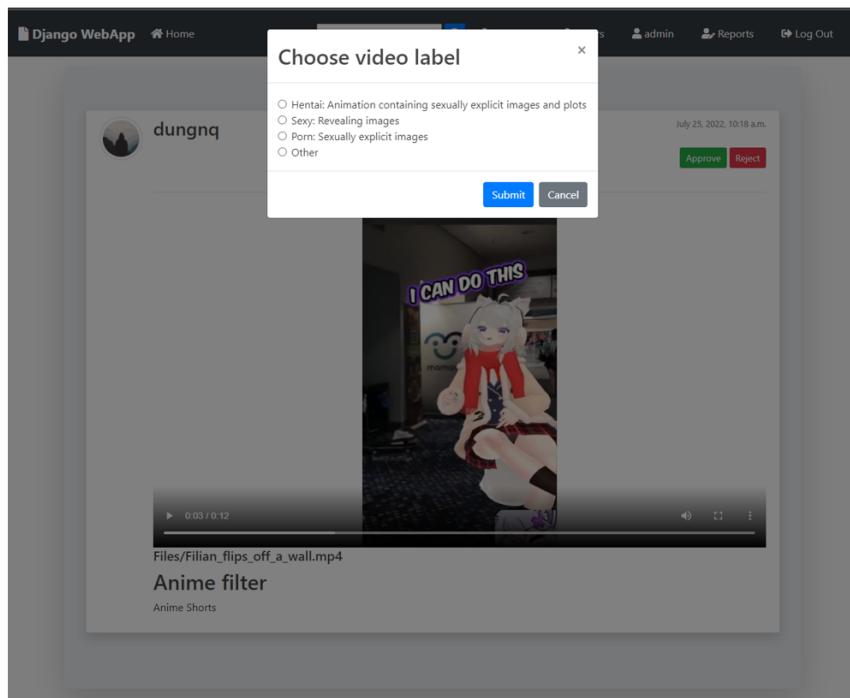
### b, Duyệt bài đăng



**Hình 4.26:** Yêu cầu đăng Video

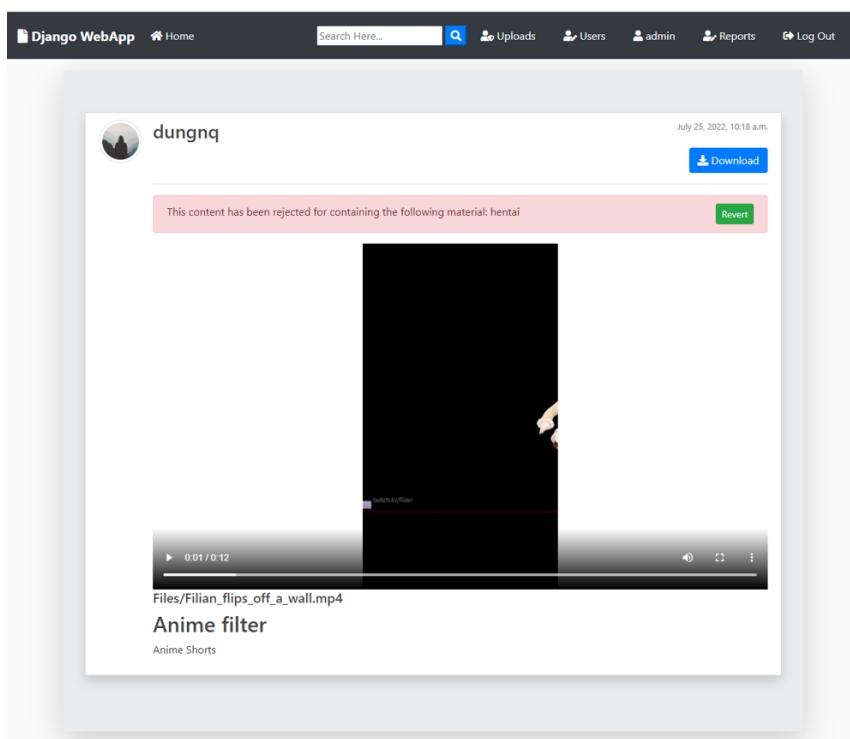
Quản trị viên sau khi đăng nhập vào hệ thống có thể nhấn nút Uploads để xem các yêu cầu đăng Video bị hệ thống từ chối. Sau khi chọn đồng ý hoặc từ chối trạng thái của bài đăng tương ứng sẽ được tự động cập nhật. Nếu bấm từ chối, quản trị

viên cần phải nêu rõ lý do vi phạm ở Form sau.

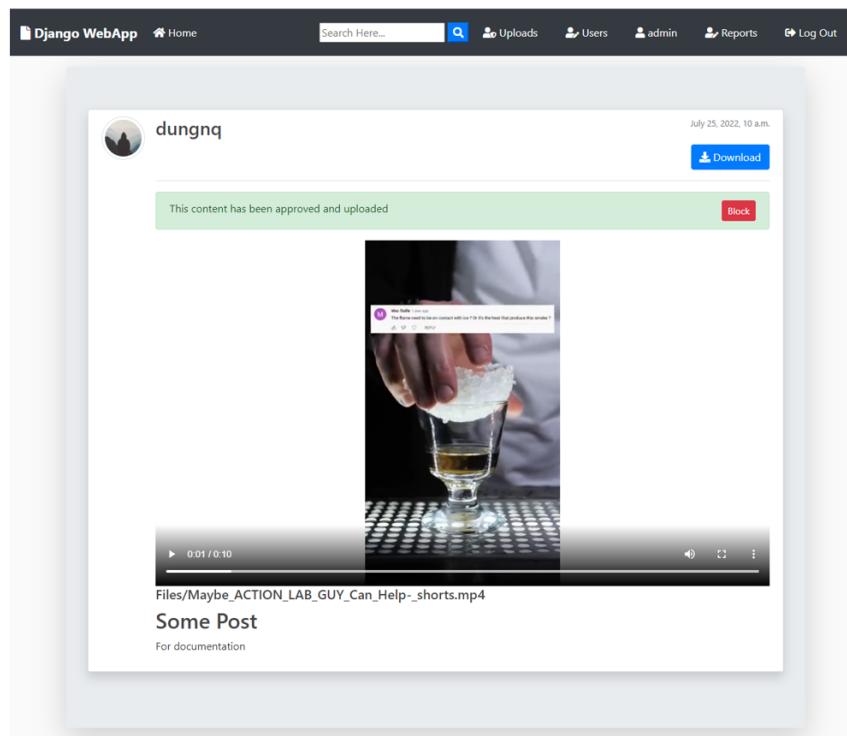


**Hình 4.27:** Form từ chối yêu cầu

Sau khi từ chối, quản trị viên vẫn có quyền chấp thuận lại Video và ngược lại, sau khi đồng ý quản trị viên vẫn có quyền chặn Video nếu thấy vi phạm.

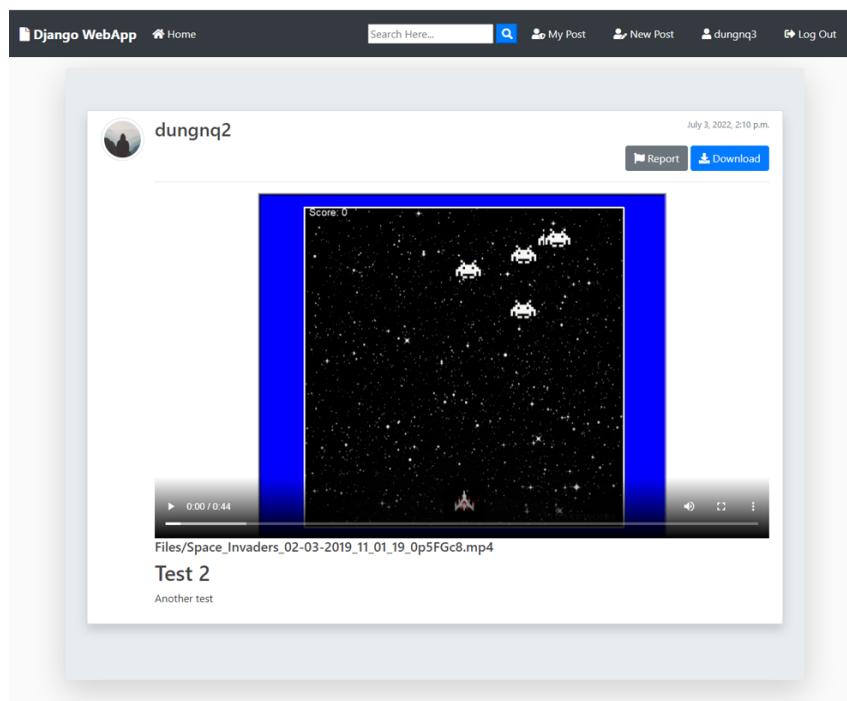


**Hình 4.28:** Video bị từ chối



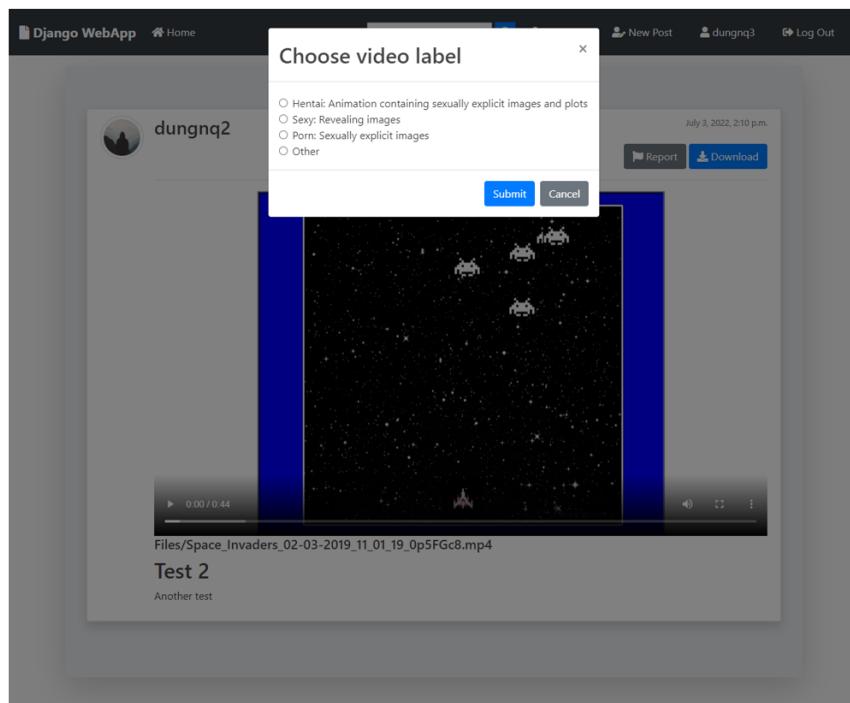
**Hình 4.29:** Video được chấp nhận

### c, Báo cáo Video



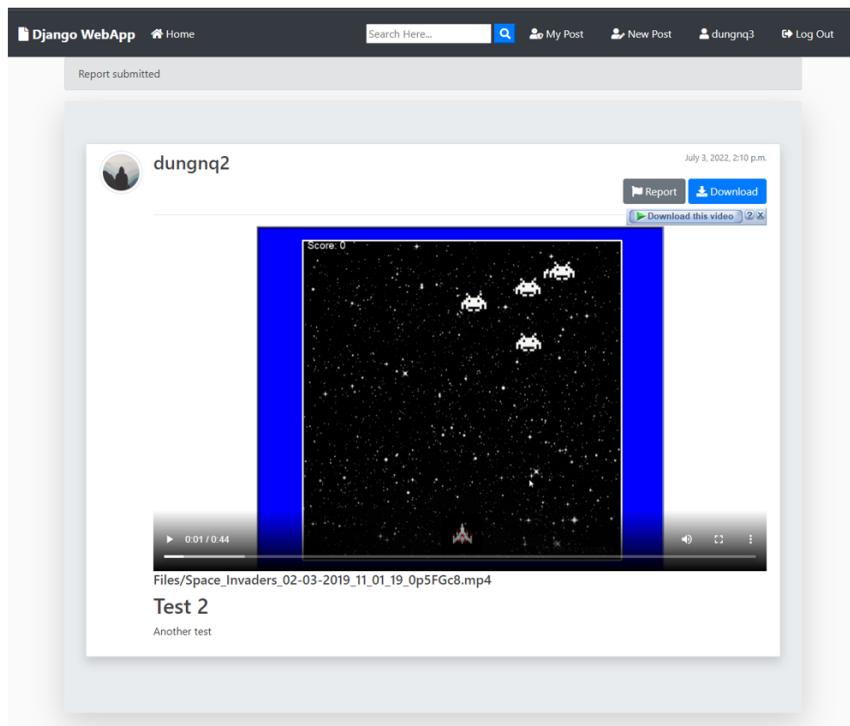
**Hình 4.30:** Xem bài đăng

Người dùng khi xem bài đăng có thể ấn nút Report để báo cáo vi phạm. Khi đó Form báo cáo sẽ hiện ra để người dùng chọn loại nội dung vi phạm.



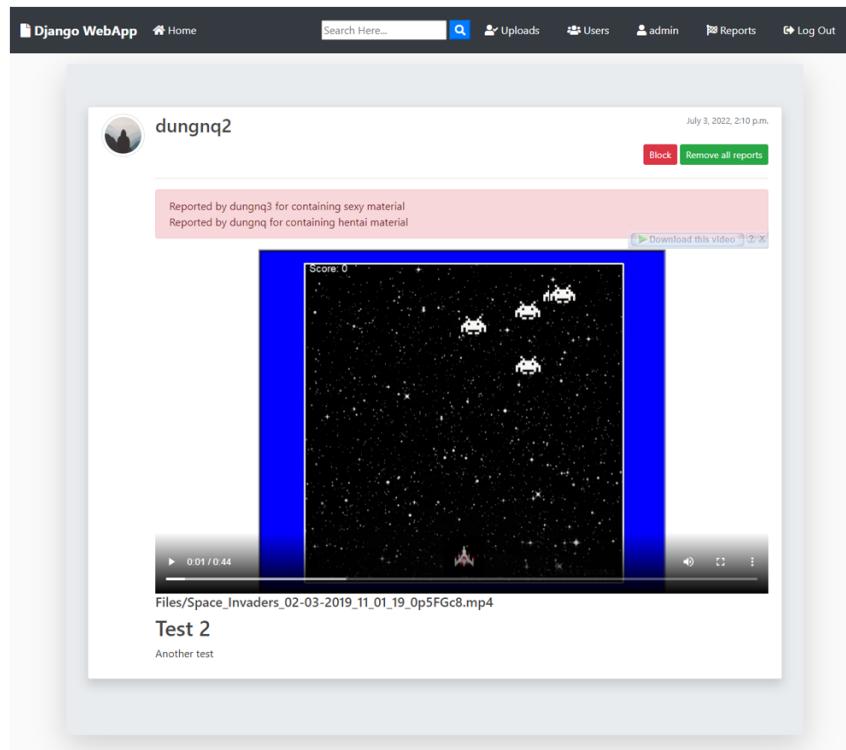
**Hình 4.31:** Form báo cáo vi phạm

Người dùng cần điền Form này trước khi bấm Submit để gửi báo cáo đến Quản trị viên.



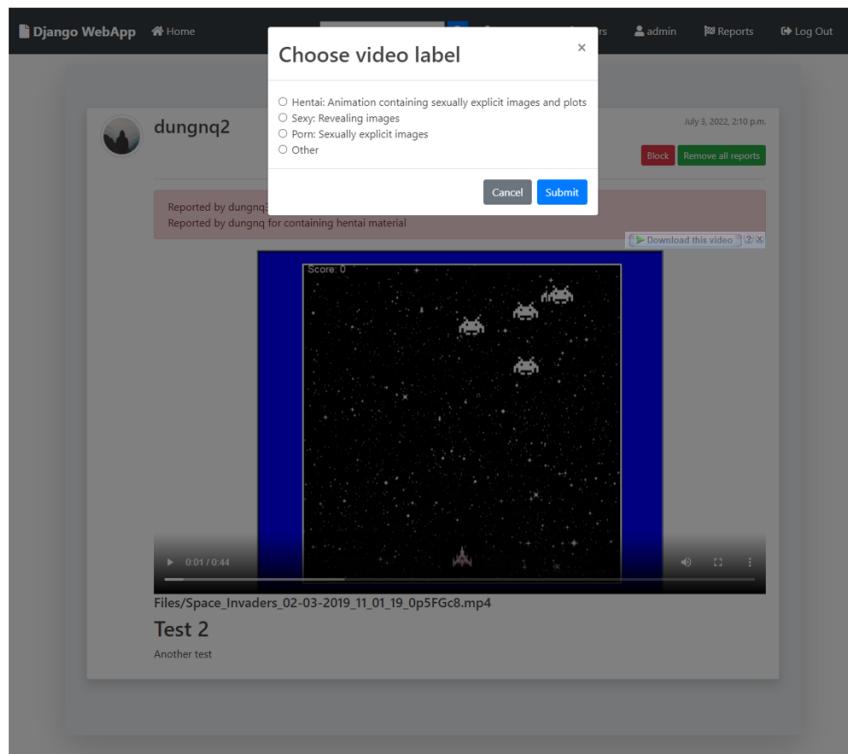
**Hình 4.32:** Báo cáo vi phạm thành công

#### d, Duyệt báo cáo vi phạm

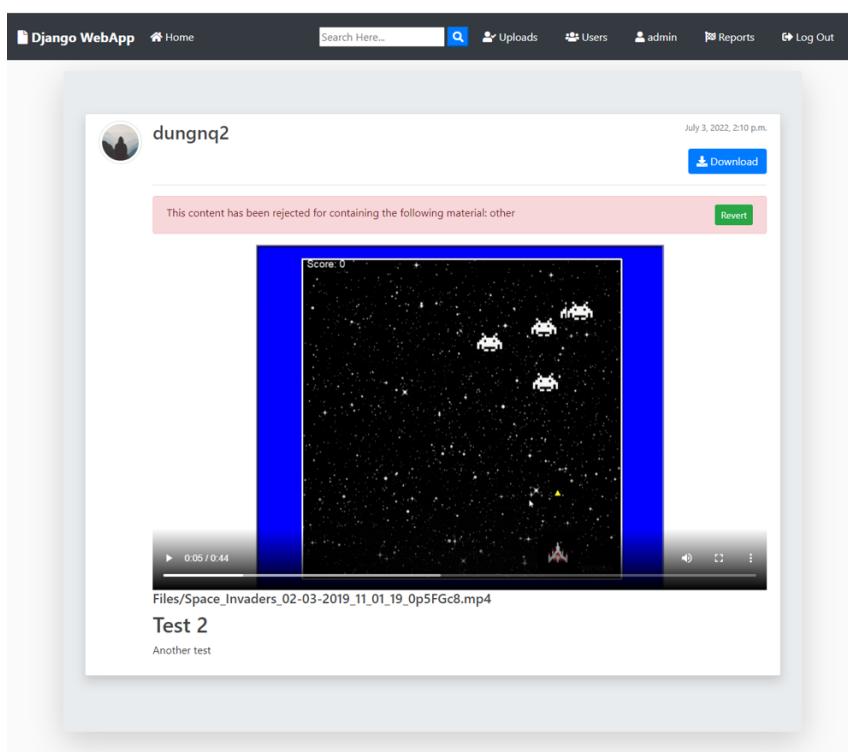


**Hình 4.33:** Xem báo cáo vi phạm

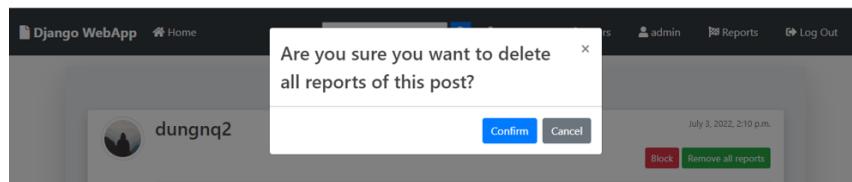
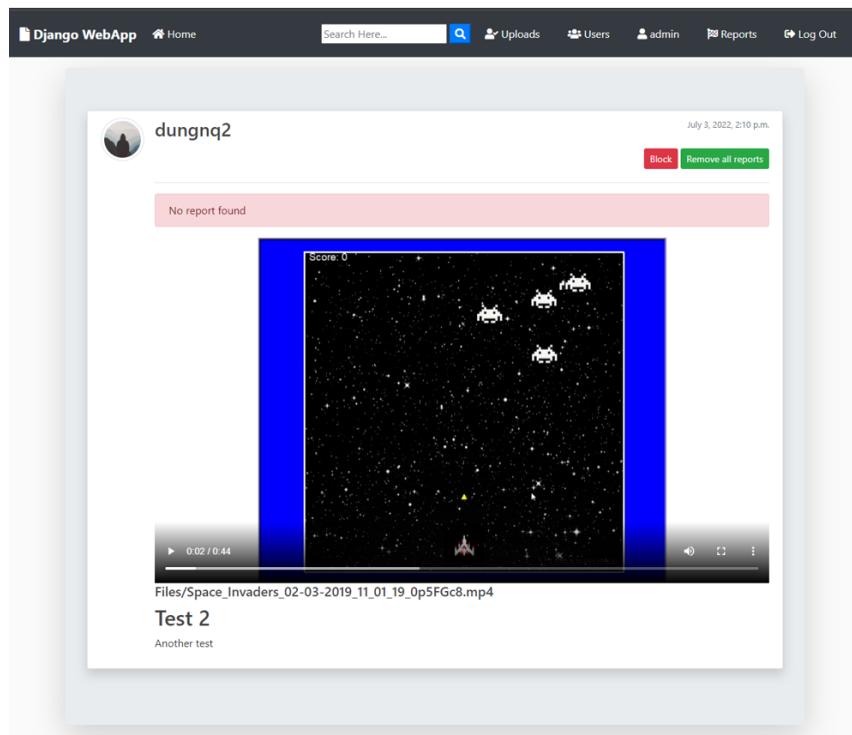
Quản trị viên có thể chọn mục Reports trên thanh điều khiển để xem các báo cáo vi phạm. Khi mở một bài đăng bị báo cáo, tất cả các báo cáo của bài đăng sẽ được hiển thị. Lúc này Quản trị viên có thể chọn chặn Video vi phạm hoặc xóa hết các báo cáo. Nếu chọn chặn Video, quản trị viên vẫn có thể ngưng chặn Video. Tuy nhiên khi xóa các báo cáo sẽ không thể khôi phục được. Muốn xóa tất cả các báo cáo, quản trị viên cần điền Form xác nhận.



**Hình 4.34:** Form chặn Video



**Hình 4.35:** Video đã bị chặn

**Hình 4.36:** Form xóa báo cáo vi phạm**Hình 4.37:** Báo cáo vi phạm đã bị xóa

## 4.4 Kiểm thử

### 4.4.1 Đăng Video

**Bảng 4.17:** Test Case Đăng Video hợp lệ

<b>Mã Test Case</b>	TC001	
<b>Tên Test Case</b>	<b>Đăng Video hợp lệ</b>	
<b>Mô tả</b>	Kiểm tra tính năng đăng Video	
<b>Ngày Test</b>	26/7/2022	
<b>Kết quả</b>	Đạt	
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập vào hệ thống	
<b>Dữ liệu Test</b>	<b>Dữ liệu</b>	<b>Giá trị</b>
	Title	Test Post
	Content	Test valid video
	File	shorts.mp4

<b>Kịch bản</b>	Kiểm tra khi nhập đầy đủ các trường và Video hợp lệ, bộ lọc duyệt và đăng video				
<b>Các bước Test</b>	<b>Bước</b>	<b>Chi tiết</b>	<b>Mong đợi</b>	<b>Thực tế</b>	<b>Kết quả</b>
	<b>1</b>	Chọn New Post	Hiển thị Form đăng bài	Như mong đợi	Đạt
	<b>2</b>	Điền thông tin Post	Thông tin được điền	Như mong đợi	Đạt
	<b>3</b>	Nhấn Upload	Bài đăng được chuyển sang trạng thái Submitted rồi sau đó được duyệt và đăng tải	Như mong đợi	Đạt

**Bảng 4.18:** Test Case Đăng Video vi phạm

<b>Mã Test Case</b>	TC002				
<b>Tên Test Case</b>	<b>Đăng Video vi phạm</b>				
<b>Mô tả</b>	Kiểm tra tính năng đăng và lọc Video				
<b>Ngày Test</b>	26/7/2022				
<b>Kết quả</b>	Đạt				
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập vào hệ thống				
<b>Dữ liệu Test</b>	<b>Dữ liệu</b>	<b>Giá trị</b>			
	Title	Test Filter			
	Content	Test Sexy video			
	File	Women's Swimsuits.mp4			
<b>Kịch bản</b>	Kiểm tra khi đăng Video không phù hợp, bộ lọc từ chối bài đăng				
<b>Các bước Test</b>	<b>Bước</b>	<b>Chi tiết</b>	<b>Mong đợi</b>	<b>Thực tế</b>	<b>Kết quả</b>
	<b>1</b>	Chọn New Post	Hiển thị Form đăng bài	Như mong đợi	Đạt
	<b>2</b>	Điền thông tin Post	Thông tin được điền	Như mong đợi	Đạt

	<b>3</b>	Nhấn Up-load	Bài đăng được chuyển sang trạng thái Submitted rồi sau đó bị lọc bởi hệ thống	Như mong đợi	Đạt
--	----------	--------------	---	--------------	-----

**Bảng 4.19:** Test Case Đăng File ảnh

<b>Mã Test Case</b>	TC003				
<b>Tên Test Case</b>	<b>Đăng File ảnh</b>				
<b>Mô tả</b>	Kiểm tra tính năng đăng và lọc bài với File là dạng ảnh có vi phạm				
<b>Ngày Test</b>	26/7/2022				
<b>Kết quả</b>	Đạt				
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập vào hệ thống				
<b>Dữ liệu Test</b>	<b>Dữ liệu</b>	<b>Giá trị</b>			
	Title	Test Image			
	Content	Test NSFW image			
	File	censoredh4.png			
<b>Kịch bản</b>	Kiểm tra khi đăng ảnh, bộ lọc duyệt và phân loại như video				
<b>Các bước Test</b>	<b>Bước</b>	<b>Chi tiết</b>	<b>Mong đợi</b>	<b>Thực tế</b>	<b>Kết quả</b>
	<b>1</b>	Chọn New Post	Hiển thị Form đăng bài	Như mong đợi	Đạt
	<b>2</b>	Điền thông tin Post	Thông tin được điền	Như mong đợi	Đạt
	<b>3</b>	Nhấn Up-load	Bài đăng được chuyển sang trạng thái Submitted rồi sau đó bị lọc bởi hệ thống	Như mong đợi	Đạt

**Bảng 4.20:** Test Case Đăng File không đúng định dạng

<b>Mã Test Case</b>	TC004				
<b>Tên Test Case</b>	<b>Đăng File không đúng định dạng</b>				
<b>Mô tả</b>	Kiểm tra tính năng kiểm tra và từ chối File là các định dạng khác ngoài ảnh và Video				
<b>Ngày Test</b>	26/7/2022				
<b>Kết quả</b>	Đạt				
<b>Tiền điều kiện</b>	Người dùng đã đăng nhập vào hệ thống				
<b>Dữ liệu Test</b>	<b>Dữ liệu</b>	<b>Giá trị</b>			
	Title	Test Doc			
	Content	Test Word file			
	File	TestPlan.doc			
<b>Kịch bản</b>	Kiểm tra khi đăng file, hệ thống từ chối vì định dạng file không phù hợp				
<b>Các bước Test</b>	<b>Bước</b>	<b>Chi tiết</b>	<b>Mong đợi</b>	<b>Thực tế</b>	<b>Kết quả</b>
	1	Chọn New Post	Hiển thị Form đăng bài	Như mong đợi	Đạt
	2	Điền thông tin Post	Thông tin được điền	Như mong đợi	Đạt
	3	Nhấn Up-load	Bài đăng không được đăng tải do hệ thống từ chối, hiển thị thông báo cần chọn file ảnh hoặc video	Như mong đợi	Đạt

**Bảng 4.21:** Test Case Đăng Video bởi người dùng bị chặn

<b>Mã Test Case</b>	TC005				
<b>Tên Test Case</b>	<b>Đăng Video bởi người dùng bị chặn</b>				
<b>Mô tả</b>	Kiểm tra tính năng đăng bài với người dùng bị khóa tài khoản				
<b>Ngày Test</b>	26/7/2022				
<b>Kết quả</b>	Đạt				
<b>Tiền điều kiện</b>	#	<b>Điều kiện</b>			
	1	Người dùng đã đăng nhập vào hệ thống			
	2	Người dùng đã bị khóa bởi quản trị viên			
<b>Dữ liệu Test</b>	<b>Dữ liệu</b>		<b>Giá trị</b>		
	Title		Test Ban		
	Content		Test banned user		
	File		Shorts.mp4		
<b>Kịch bản</b>	Kiểm tra khi đăng file, hệ thống từ chối vì người dùng đang bị khóa				
<b>Các bước Test</b>	<b>Bước</b>	<b>Chi tiết</b>	<b>Mong đợi</b>	<b>Thực tế</b>	<b>Kết quả</b>
	1	Chọn New Post	Hiển thị Form đăng bài	Như mong đợi	Đạt
	2	Điền thông tin Post	Thông tin được điền	Như mong đợi	Đạt
	3	Nhấn Up-load	Bài đăng không được đăng tải do hệ thống từ chối, hiển thị thông báo người dùng hiện đang bị khóa	Như mong đợi	Đạt

#### 4.4.2 Duyệt báo cáo vi phạm

**Bảng 4.22:** Test Case Chặn Video bị báo cáo

<b>Mã Test Case</b>	<b>TC006</b>				
<b>Tên Test Case</b>	<b>Chặn Video bị báo cáo</b>				
<b>Mô tả</b>	Kiểm tra tính năng chặn video vi phạm đối với Video bị báo cáo				
<b>Ngày Test</b>	26/7/2022				
<b>Kết quả</b>	Đạt				
<b>Tiền điều kiện</b>	#	<b>Điều kiện</b>			
	1	Quản trị viên đã đăng nhập vào hệ thống			
	2	Bài đăng bị báo cáo bởi người dùng			
<b>Dữ liệu Test</b>	<b>Dữ liệu</b>	<b>Giá trị</b>			
	Label	Other			
<b>Kịch bản</b>	Kiểm tra khi chặn Video, Video không còn xuất hiện trên trang chủ và trang báo cáo				
<b>Các bước Test</b>	<b>Bước</b>	<b>Chi tiết</b>	<b>Mong đợi</b>	<b>Thực tế</b>	<b>Kết quả</b>
	1	Chọn Reports	Hiển thị các bài đăng bị báo cáo	Như mong đợi	Đạt
	2	Chọn xem bài đăng	Hiển thị bài đăng và các báo cáo từ người dùng	Như mong đợi	Đạt
	3	Nhấn Block	Hiển thị Form chặn Video	Như mong đợi	Đạt
	4	Điền Form nhãn Video	Nhãn được chọn thành công	Như mong đợi	Đạt
	5	Nhấn Submit	Hệ thống thông báo thành công, xóa Video khỏi trang chủ và trang Reports	Như mong đợi	Đạt

**Bảng 4.23:** Test Case Xóa báo cáo vi phạm

<b>Mã Test Case</b>	TC007				
<b>Tên Test Case</b>	<b>Xóa báo cáo vi phạm</b>				
<b>Mô tả</b>	Kiểm tra tính năng chặn xóa báo cáo vi phạm đối với Video bị báo cáo				
<b>Ngày Test</b>	26/7/2022				
<b>Kết quả</b>	Đạt				
<b>Tiền điều kiện</b>	#	<b>Điều kiện</b>			
	1	Quản trị viên đã đăng nhập vào hệ thống			
	2	Bài đăng bị báo cáo bởi người dùng			
<b>Dữ liệu Test</b>	<b>Không</b>				
<b>Kích bản</b>	Kiểm tra khi xóa các báo cáo vi phạm, Video không còn xuất hiện trong mục Reportss				
<b>Các bước Test</b>	<b>Bước</b>	<b>Chi tiết</b>	<b>Mong đợi</b>	<b>Thực tế</b>	<b>Kết quả</b>
	1	Chọn Re-ports	Hiển thị các bài đăng bị báo cáo	Như mong đợi	Đạt
	2	Chọn xem bài đăng	Hiển thị bài đăng và các báo cáo từ người dùng	Như mong đợi	Đạt
	3	Nhấn Remove All Reports	Hiển thị Form xác nhận	Như mong đợi	Đạt
	4	Nhấn Confirm	Hệ thống xóa bài đăng khỏi mục Reports và thông báo không còn báo cáo vi phạm	Như mong đợi	Đạt

#### 4.4.3 Tổng kết

Em đã thực hiện kiểm thử với 2 UC chính là đăng video và duyệt báo cáo vi phạm. Ở cả hai UC em đều thực hiện kiểm thử theo phương pháp hộp đen. Phương pháp này giúp hạn chế các lỗi do người dùng phát sinh khi sử dụng trang Web với một số trường hợp cụ thể chứ không tập trung vào mã nguồn hay cách hoạt động của trang Web.

Cả 7 trường hợp kiểm thử đều cho kết quả đạt. Từ đó có thể kết luận ban đầu rằng chương trình đã hạn chế được các lỗi dễ xảy ra trong quá trình sử dụng.

#### 4.5 Triển khai

Trang Web được triển khai ngay trên máy Local để tiện mô phỏng và debug. Sau đây là các thông tin về cấu hình máy tính:

#	Cấu hình	Thông tin
1	CPU	Intel Core i5-12600KF
2	GPU	RTX 3060
3	Hệ điều hành	Windows 10

**Bảng 4.24:** Thông tin cấu hình máy

Server được chạy trên máy với triển khai thử nghiệm như sau:

#	Thông tin	Thử nghiệm
1	Số lượng người dùng	10 người
2	Lượng truy cập cùng lúc	3 người
3	Thời gian phản hồi	0,23s
4	Số video xử lý cùng lúc	1 video

**Bảng 4.25:** Kết quả triển khai thử nghiệm

## CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

### 5.1 Thu thập và xử lý dữ liệu để huấn luyện Model

#### 5.1.1 Đặt vấn đề

Với một bài toán phân loại ảnh phức tạp như phân loại nội dung không phù hợp với trẻ vị thành niên sớm, điều quan trọng là phải thu thập được thật nhiều dữ liệu phù hợp để Model có thể học nhằm không những tăng độ chính xác mà còn giúp nó đạt kết quả tốt hơn với những dữ liệu mới.

Trong thực tế khi muốn huấn luyện Model từ các mạng CNN có sẵn để áp dụng và cải tiến cho một bài toán cụ thể thì có thể cần tới hàng trăm nghìn bức ảnh. Khi đó mới đảm bảo tránh bị Underfit với tập dữ liệu huấn luyện.

Việc thu thập những bức ảnh trên mạng là không khó, nhưng để thu thập một số lượng lớn các bức ảnh với cùng một chủ đề là vô cùng tốn thời gian do mỗi trang Web chỉ có một số lượng nhỏ ảnh. Chính vì vậy mà người ta thường hay dùng các bộ Scripts để tự động tìm và tải về những bức ảnh có cùng chủ đề trên mạng hoặc đơn giản hơn là tải về các bộ dữ liệu có sẵn.

Ngay cả khi thu được một số lượng lớn ảnh, chất lượng của từng ảnh cũng cần được đảm bảo để tránh gặp phải những ảnh bị lỗi định dạng, ảnh quá nhỏ, quá lớn hay bị trùng lặp. Quan trọng nhất là các ảnh cần phải được gán đúng nhãn bởi nhiều khi những bức ảnh trên Internet lại không phù hợp với từ khóa tìm kiếm.

#### 5.1.2 Giải pháp

##### a, Thu thập dữ liệu

Bước đầu khi tìm kiếm dữ liệu phù hợp, em đã thử tìm các bộ dữ liệu có sẵn trên Internet nhưng không thu được kết quả. Các bài toán phân loại nội dung như vậy thường có bộ dữ liệu được Private hoặc số lượng ảnh nhỏ nên không phù hợp.

Với cách tiếp cận sử dụng Scripts để tự động lướt Web, em đã tìm được hai bộ Scripts với các đánh giá tích cực từ người dùng. Trong đó, bộ Scripts với đánh giá tốt nhất và cho nhiều ảnh nhất là nsfw\_data\_scraper của alex000kim trên Github.

## NSFW Data Scraper

Note: use with caution - the dataset is noisy

### Description

This is a set of scripts that allows for an automatic collection of *tens of thousands* of images for the following (loosely defined) categories to be later used for training an image classifier:

- `porn` - pornography images
- `hentai` - hentai images, but also includes pornographic drawings
- `sexy` - sexually explicit images, but not pornography. Think nude photos, playboy, bikini, etc.
- `neutral` - safe for work neutral images of everyday things and people
- `drawings` - safe for work drawings (including anime)

Here is what each script (located under `scripts` directory) does:

- `1_get_urls.sh` - iterates through text files under `scripts/source_urls` downloading URLs of images for each of the 5 categories above. The `ripme` application performs all the heavy lifting. The source URLs are mostly links to various subreddits, but could be any website that Ripme supports. Note: I already ran this script for you, and its outputs are located in `raw_data` directory. No need to rerun unless you edit files under `scripts/source_urls`.
- `2_download_from_urls.sh` - downloads actual images for urls found in text files in `raw_data` directory.
- `3_optional_download_drawings.sh` - (optional) script that downloads SFW anime images from the [Danbooru2018](#) database.
- `4_optional_download_neutral.sh` - (optional) script that downloads SFW neutral images from the [Caltech256](#) dataset
- `5_create_train.sh` - creates `data/train` directory and copy all `*.jpg` and `*.jpeg` files into it from `raw_data`. Also removes corrupted images.
- `6_create_test.sh` - creates `data/test` directory and moves `N=2000` random files for each class from `data/train` to `data/test` (change this number inside the script if you need a different train/test split). Alternatively, you can run it multiple times, each time it will move `N` images for each class from `data/train` to `data/test`.

**Hình 5.1:** Mô tả Nsfw\_data\_scraper trên Github

Bộ Scripts này hoạt động bằng cách truy cập vào một loạt các nhóm trên Reddit với nội dung các nhóm thuộc vào một trong năm nhãn ở trên và tải về tất cả các ảnh được đăng tải trên các nhóm này vào các nhãn tương ứng.

Sau khi chạy Scripts trong vòng khoảng 8 tiếng, số lượng ảnh thu được sau tiền xử lý như sau:

#	Nhãn	Số ảnh
1	Drawings	14,367
2	Hentai	8,238
3	Neutrals	34,234
4	Sexy	12,819
5	Porn	56,167

**Bảng 5.1:** Số ảnh thu được sau khi chạy lần đầu

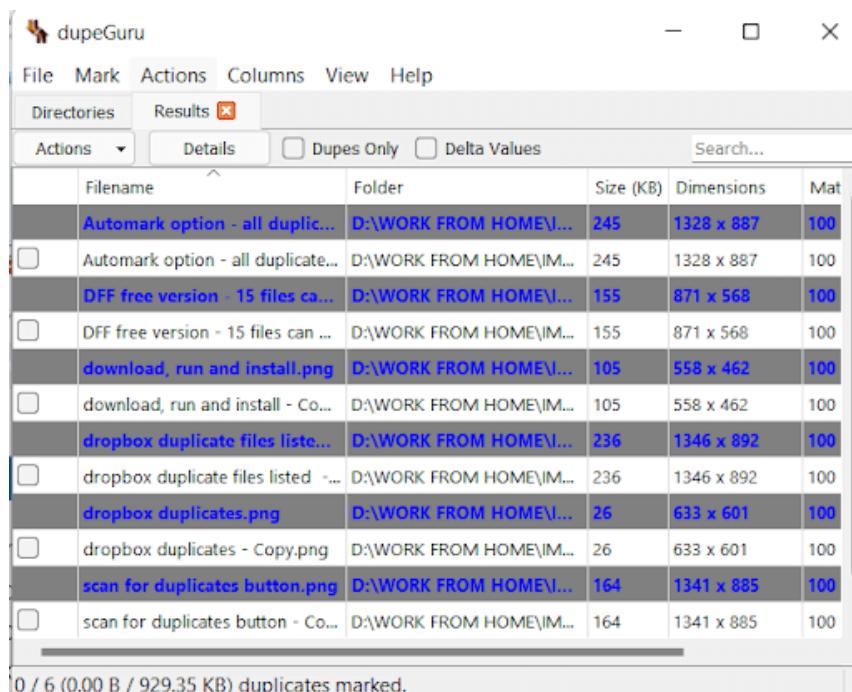
Với số lượng ảnh như trên, sau khi huấn luyện nhận thấy độ chính xác của nhãn Hentai khá thấp so với Drawings và 2 nhãn này thường bị nhầm lẫn với nhau. Để tăng độ chính xác và giúp cân bằng số lượng ảnh giữa các lớp, em đã thêm vào khoảng 20 nhóm Reddit khác với nội dung Hentai và chạy Scripts để bổ sung ảnh. Sau khi chạy Scripts và tiền xử lý ảnh, số ảnh lúc này như sau:

#	Nhãn	Số ảnh
1	Drawings	14,367
2	Hentai	23,372
3	Neutrals	34,234
4	Sexy	12,819
5	Porn	56,167

**Bảng 5.2:** Số ảnh thu được sau khi chạy lần 2**b, Tiền xử lý**

Các ảnh thu được sau khi chạy Scripts đều được tải trực tiếp từ trang Web của Reddit là một mạng xã hội với số lượng người dùng lớn nên có nhiều nhiễu. Chẳng hạn như các bức ảnh được đăng lại bởi nhiều người hoặc có những bức ảnh không liên quan đến nội dung của nhóm. Để loại bỏ các bức ảnh trùng nhau, em sử dụng hai công cụ chính là duplicate-file-finder của Qarj trên Github và DupeGuru.

Với Duplicate-file-finder, em dễ dàng loại bỏ được những file có nội dung giống nhau nhưng tên khác nhau. Tuy nhiên công cụ này không thể tìm được những file có nội dung gần giống nhau của cùng một bức ảnh. Để loại bỏ những ảnh này em cần dùng đến DupeGuru để quét nội dung của từng bức ảnh và so sánh chúng với nhau.

**Hình 5.2:** Minh họa DupeGuru

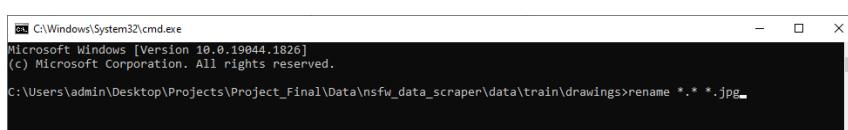
Trong trường hợp các ảnh có kích cỡ quá nhỏ hoặc quá lớn không bị xóa, em

cũng cần dùng đến tính năng sắp xếp file trong File Explorer theo kích cỡ của file. Sau khi sắp xếp, em chỉ cần chọn những file nhỏ hơn 6KB hoặc lớn hơn 10MB để xóa.

Name	Date	Type	Size
96ad0424998f90191...	5/11/2022 9:25 AM	JPG File	7 KB
9bbcffc88d8c0cd80...	5/11/2022 9:24 AM	JPG File	7 KB
f25c1c3cf9a093e4...	5/11/2022 9:25 AM	JPG File	8 KB
4ee2b96a332a25c75...	5/11/2022 9:24 AM	JPG File	9 KB
4cdfb607be1c439e8...	5/11/2022 9:24 AM	JPG File	9 KB
d7d18121ead5345c...	5/11/2022 9:24 AM	JPG File	9 KB
c2678ca7185b69871...	5/11/2022 9:24 AM	JPG File	9 KB
f7973db2ad2ad57da...	5/11/2022 9:23 AM	JPG File	10 KB
d439144e2fa441601...	5/11/2022 9:24 AM	JPG File	11 KB
332860c7a3d42e777...	5/11/2022 9:24 AM	JPG File	11 KB
02a980a743678d2b2...	5/11/2022 9:20 AM	JPG File	12 KB
57ad799625f6648d1...	5/11/2022 9:23 AM	JPG File	13 KB
55abde3543c51e055...	5/11/2022 9:25 AM	JPG File	13 KB
7b7ee0b5f58368d30...	5/11/2022 9:21 AM	JPG File	13 KB
618ffd19acff9e662f2...	5/11/2022 9:23 AM	JPG File	13 KB
578aaef44a919af57b...	5/11/2022 9:23 AM	JPG File	13 KB
d45dba8bdb4740e3...	5/11/2022 9:23 AM	JPG File	13 KB
b50f10754c6f4af765...	5/11/2022 9:24 AM	JPG File	14 KB
da72187514213e356...	5/11/2022 9:24 AM	JPG File	14 KB
35e7544fc0d328081...	5/11/2022 9:25 AM	JPG File	14 KB
e3bb62ff7b9dd43cb...	5/11/2022 9:25 AM	JPG File	14 KB
9603ba9a6dd5e299...	5/11/2022 9:24 AM	JPG File	14 KB
9708cab6c5167162d...	5/11/2022 9:24 AM	JPG File	14 KB

**Hình 5.3:** Sắp xếp file trong File Explorer

Ngoài ra, do các file tải về có định dạng khác nhau (các file bị corrupt đã bị loại bỏ khi chạy scripts) nên cần chuyển toàn bộ về định dạng .jpg. Để chuyển toàn bộ file, em sử dụng Windows Command Prompt để chạy lệnh sau:

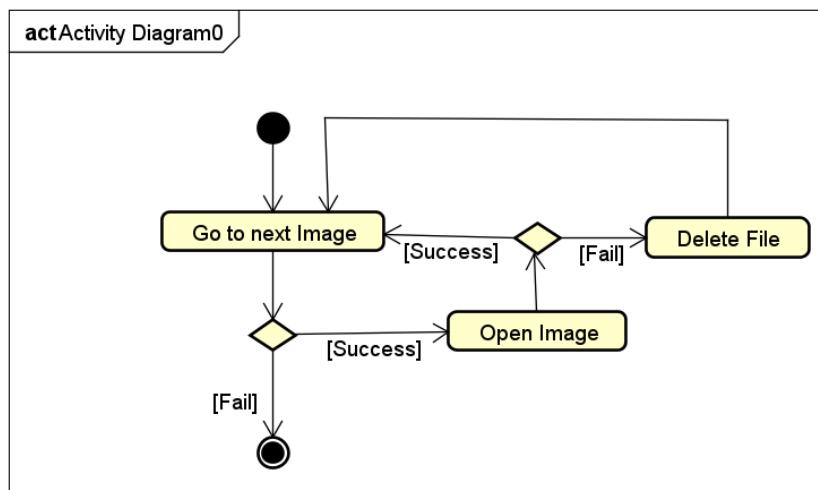


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin\Desktop\Projects\Project_Final\Data\nsfw_data_scraper\data\train\drawings>rename *.* *.jpg
```

**Hình 5.4:** Đổi định dạng file trong Command Prompt

Cuối cùng, để đảm bảo các file ảnh còn lại có thể mở được mà không bị lỗi, em đã viết một đoạn mã bằng Python để kiểm tra và tự động xóa những file không mở được với cách hoạt động như sau:

**Hình 5.5:** Biểu đồ hoạt động lọc File lỗi

Tất nhiên, các bước trên mới chỉ đảm bảo bộ dữ liệu là có thể học được và không có lỗi chứ chưa đảm bảo chất lượng của bộ dữ liệu. Muốn làm vậy sẽ cần phải đi qua từng ảnh và kiểm tra xem nhãn đã đúng hay chưa. Do thời gian có hạn nên em chỉ có thời gian lướt qua các lớp quan trọng như Hentai và Porn.

### 5.1.3 Kết quả đạt được

Sau khi sử dụng Scripts để tải bộ dữ liệu và các công cụ khác để làm sạch dữ liệu, em đã thu được một bộ dữ liệu với tổng cộng khoảng 140,000 ảnh với số lượng ảnh tối thiểu ở một lớp là 12,000 ảnh. Số lượng ảnh này đảm bảo được không xảy ra lỗi trong quá trình huấn luyện và cho Model đủ dữ liệu để học và cải tiến mạng có sẵn.

## 5.2 Xây dựng Model

### 5.2.1 Đặt vấn đề

Với các bài toán Computer Vision nói chung và bài toán nhận dạng ảnh nói riêng, người ta thường sử dụng các mạng trích chọn đặc trưng có sẵn với độ chính xác cao với bài toán tương tự thay vì xây dựng lại Model từ đầu. Phương pháp này được gọi là Transfer Learning và được áp dụng rộng rãi trong các bài toán phân loại ảnh.

Để huấn luyện Model thích nghi với bài toán mới, ta chỉ cần thêm các lớp đầy đủ ở cuối Model và huấn luyện các lớp này trên nền tảng là các lớp trích chọn đặc trưng đã có. Với trường hợp lượng dữ liệu đủ nhiều, sau khi Model đã bão hòa mà độ chính xác vẫn chưa đạt ngưỡng mong muốn ta có thể tiến hành thêm bước Fine-tune: khóa các lớp đầy đủ ở cuối và quay trở lại huấn luyện các lớp trích chọn đặc chung để ứng dụng cải tiến với bài toán mới.

Với bài toán phân loại Video phù hợp và lượng dữ liệu khá lớn, vấn đề là chọn mạng trích chọn đặc trưng nào cho kết quả tốt nhất với bài toán và làm sao để tránh vấn đề Overfitting khi huấn luyện và Fine-tune.

### 5.2.2 Giải pháp

#### a, Chọn mạng trích chọn đặc trưng

Sau khi tìm hiểu và biết đến bài toán tương tự với Model của Gant Laborde trên trang NSFW JS, em quyết định sử dụng mạng InceptionV3 được huấn luyện trên bộ dữ liệu Imagenet để làm Model gốc. Theo Gant Laborde, mạng này có thể giúp Model đạt được độ chính xác đến 93% với bài toán phân loại nội dung phù hợp.



**Hình 5.6:** Công cụ NSFW JS

Mạng Inception-V1 đã dành chiến thắng ở cuộc thi ImageNet vào năm 2015. Kiến trúc này đã giải quyết một câu hỏi lớn trong mạng CNN đó là sử dụng kernel\_size với kích thước bao nhiêu thì hợp lý. Các kiến trúc mạng nơ ron trước đó đều sử dụng các bộ lọc với đa dạng các kích thước 11x11, 5x5, 3x3 cho tới nhỏ nhất là 1x1. Một khám phá được đưa ra bởi bài báo đó là việc cùng kết hợp đồng thời các bộ lọc này vào cùng một block có thể mang lại hiệu quả đó chính là kiến trúc khối Inception. Inception-V3 là kế thừa của Inception-V1 bao gồm 24 triệu tham số. Toàn bộ các layer tích chập của Inception-V3 được sau bởi một layer batch normalization và một ReLU activation. Batch normalization là kỹ thuật chuẩn hóa đầu vào theo từng minibatch tại mỗi layer theo phân phối chuẩn hóa  $N(0,1)$ , giúp

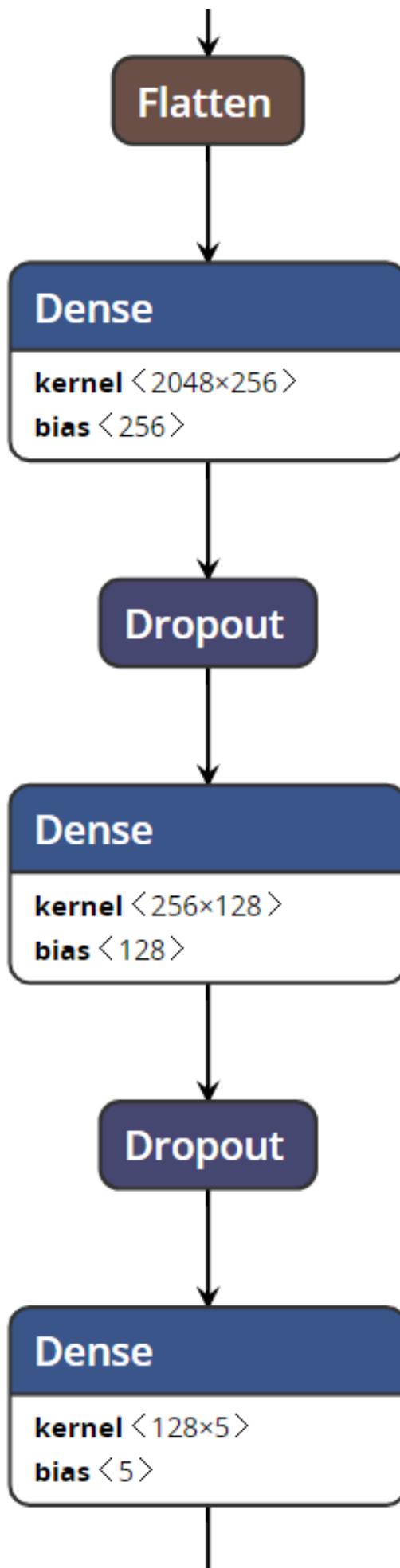
cho quá trình huấn luyện thuật toán nhanh hơn.

Inception-V3 giải quyết được vấn đề thắt cổ chai (representational bottlenecks). Tức là kích thước của các layers không bị giảm một cách đột ngột. Đồng thời Inception-V3 có một cách tính toán hiệu quả hơn nhờ sử dụng phương pháp nhân tố (factorisation methods). Sau khi đã chọn mạng gốc là Inception V3, em tiến hành loại bỏ đi các lớp trên cùng và thêm vào 3 lớp đầy đủ với lần lượt 256, 128 và 5 nút ẩn. Lớp cuối sử dụng hàm kích hoạt là Softmax là lớp đưa ra dự đoán về nhãn của ảnh.

Sau khi đã chọn mạng gốc là Inception V3, em tiến hành loại bỏ đi các lớp trên cùng và thêm vào 3 lớp đầy đủ với lần lượt 256, 128 và 5 nút ẩn. Lớp cuối sử dụng hàm kích hoạt là Softmax là lớp đưa ra dự đoán về nhãn của ảnh.

### b, Tránh Overfit

Để tránh Overfit do lượng dữ liệu lớn dẫn đến Model học thuộc theo dữ liệu đã học mà không dự đoán tốt với các dữ liệu mới, em sử dụng Dropout để loại bỏ ngẫu nhiên một số liên kết giữa các nút. Khi đó các lớp ở cuối có cấu trúc như sau:



Hình 5.7: Sử dụng Dropout

Ngoài ra, em còn sử dụng L2 Regularization ở lớp Dense đầu tiên và thực hiện Augment Dataset bằng cách xoay, xê dịch, cắt, phóng to, đảo ngược và chuyển kênh hình ảnh để đảm bảo bộ dữ liệu được đa dạng và giảm khả năng Overfit.

Kỹ thuật regularization là thêm vào hàm mất mát ( loss ) một đại lượng nữa.

$$J_{\text{reg}}(\theta) = J(\theta) + \lambda R(\theta)$$

**Hình 5.8:** Hàm mất mát với Regularization

Đại lượng này sẽ tác động đến hàm loss. Cụ thể : nếu lamda lớn thì ảnh hưởng của đại lượng thêm vào lên hàm loss cũng lớn và ngược lại nếu lamda nhỏ thì ảnh hưởng của nó lên hàm loss cũng nhỏ. Nhưng lamda cũng không được quá lớn vì nếu quá lớn thì đại lượng thêm vào sẽ lấn át loss dẫn đến mô hình xây dựng sẽ bị sai ( underfitting ).

L2 regularization :

$$R(\mathbf{w}) = \|\mathbf{w}\|_2^2$$

**Hình 5.9:** Áp dụng L2 Regularization

Suy ra loss :

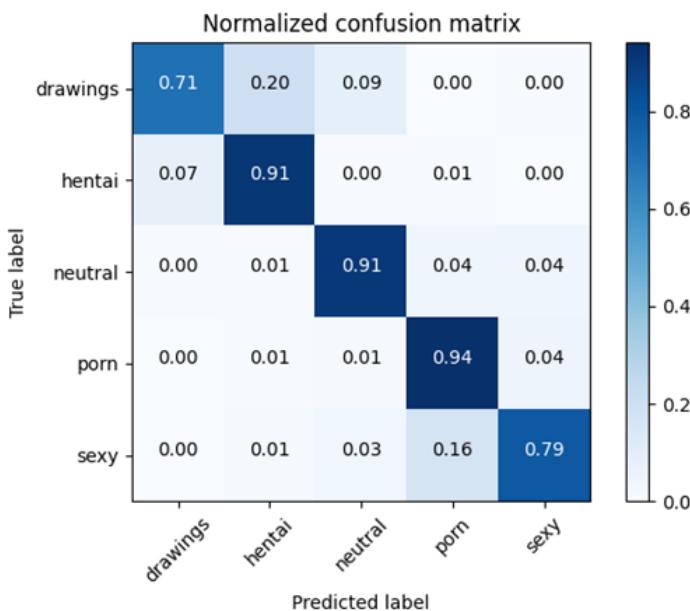
$$J(\mathbf{w}) = \frac{1}{2} \|\mathbf{y} - \mathbf{Xw}\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

**Hình 5.10:** Hàm mất mát mới

Việc tối ưu hóa model cũng đồng nghĩa với việc làm giảm hàm mất mát ( loss ) hay giảm weight . Nên L2 regularization còn được gọi là 'weight decay' ( trọng số tiêu biến ).

### 5.2.3 Kết quả

Sau khi Train và Fine-tune Model với bộ dữ liệu như trên, độ chính xác thu được của Model như sau:

**Hình 5.11:** Ma trận nhầm lẫn

Độ chính xác của Model với tập dữ liệu huấn luyện là 85,2%. Tuy nhiên Recall của các nhãn nhạy cảm như Hentai và Porn là rất cao với khoảng 91%, nhãn Sexy tuy có Recall kém nhưng có thể nhận thấy phần lớn những trường hợp khi gán nhãn nhầm với nội dung Sexy, nhãn đó khả năng cao sẽ là Porn và sẽ vẫn bị loại bỏ. Như vậy, với bài toán đặt trọng tâm vào phát hiện nhầm còn hơn bỏ sót, Model này là rất phù hợp.

Tất nhiên, kết quả này mới chỉ đánh giá trên bộ dữ liệu huấn luyện nên cần có thời gian kiểm chứng tính chính xác trong thực tế. Ngoài ra, việc nhận nhầm tới 20% Drawings là Hentai cũng có nghĩa nhiều nội dung tốt sẽ bị lọc sai. Đây chính là lý do trang Web cho phép người dùng yêu cầu quản trị viên tự duyệt bài đăng khi bị hệ thống từ chối.

### 5.3 Xử lý Background Task

#### 5.3.1 Đặt vấn đề

Với Django hay bất cứ REST Framework nào, việc xử lý các tác vụ phức tạp ở phía Back-End thường được thực hiện độc lập với yêu cầu phản hồi từ hệ thống. Bởi lẽ người dùng cần phải nhận phản hồi gần như ngay lập tức từ phía Server để biết được yêu cầu của mình đã được đáp ứng hay chưa thay vì phải mất thời gian rất lâu để chờ nhận một thông điệp đáp ứng từ Server, chưa kể đến việc gửi đi nhiều thông điệp cùng một lúc có thể dẫn đến nghẽn Server.

Đây cũng chính là vấn đề em gặp phải khi sử dụng bộ lọc Video tự động: người dùng cần được cập nhật ngay về trạng thái bài đăng của mình trong khi việc quét

và phân loại toàn bộ một Video có thể mất đến hàng giờ.

### 5.3.2 Giải pháp

Để giải quyết vấn đề này, phương pháp phổ biến sẽ là dùng hàng đợi Queue để tiếp nhận thông điệp và xử lý chúng một cách có trật tự để đảm bảo mỗi thông điệp từ người dùng đều sẽ được xử lý với tài nguyên và thời gian tối ưu nhất. Tuy nhiên, việc áp dụng hàng đợi đôi khi là quá mức cần thiết với một trang Web chỉ có vài ba người dùng cùng lúc hoặc các tác vụ ở phía Server là đơn giản và không tiềm ẩn nguy cơ mất an toàn khi được thực hiện song song.

Với Django, một giải pháp cực kì đơn giản và tiện lợi là thư viện Django-background-tasks. Thư viện này cho phép Server xử lý các Tasks song song với việc tiếp nhận và phản hồi các yêu cầu từ phía Client bằng một hàng đợi tích hợp trong cơ sở dữ liệu.

Để khai báo một Task, ta chỉ cần dùng Decorator **@background(schedule=x)** với x là thời gian chờ từ lúc tác vụ được gọi đến khi nó được thực hiện. Để làm được điều này, nó sử dụng hai bảng trong cơ sở dữ liệu với các trường dữ liệu giống nhau để lưu thông tin về các tác vụ. Một bảng để lưu những Tác vụ chưa được thực hiện, bảng còn lại lưu các tác vụ đã thực hiện xong.

**Bảng 5.3:** Bảng Background\_task/Background\_task\_completedtask

#	PK	FK	Tên cột	Kiểu dữ liệu	Bắt buộc	Mô tả
1	x		id	integer	có	id, auto increment
2			task_params	text	có	Tham số của hàm
3			task_hash	varchar(40)	có	Mã băm của tác vụ
4			verbose_name	varchar(255)	không	Tên đầy đủ
5			priority	integer	có	Mức độ ưu tiên
6			run_at	datetime	có	Thời điểm bắt đầu chạy
7			repeat	bigint	có	Số lần lặp
8			repeat_until	datetime	không	Lặp lại đến khi nào
9			queue	varchar(190)	không	Hàng đợi
10			attempts	integer	có	Số lần thử
11			failed_at	datetime	không	Thời điểm gặp lỗi
12			last_error	text	có	Lỗi gần nhất
13			locked_by	varchar(64)	không	Tạo bởi

14		locked_at	datetime	không	Thời điểm được tạo
15		creator_object_id	integer unsigned	không	Id đối tượng tạo
16		creator_content_type_id	integer	không	Id kiểu nội dung đối tượng tạo
17		task_name	varchar(190)	có	Tên tác vụ

Để tự động chạy các tác vụ, ta chỉ cần gõ lệnh sau trong Terminal khi chạy Server:

```
python .\manage.py process_tasks
```

### 5.3.3 Kết quả

Sử dụng thư viện Django Background Task, em dễ dàng quản lý và kiểm soát các tác vụ phía Back-End bên Server. Công cụ này không chỉ phù hợp với quy mô bài toán mà còn đảm bảo giúp việc xử lý các Video là kịp thời và bao quát. Tuy nhiên do không có sự phân chia độ ưu tiên hay điều phối các yêu cầu từ người dùng một cách tối ưu nên việc đáp ứng nhiều yêu cầu cùng một lúc từ Client vẫn chưa được thực hiện triệt để.

## CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

Sau quá trình xây dựng và phát triển, trang Web đã có đầy đủ những chức năng cơ bản về chia sẻ Video và kiểm soát nội dung. Tuy nhiên so với các ứng dụng thực tế, Website còn sơ sài và có phần đơn giản. Đặc biệt là về độ ổn định và tốc độ xử lý của Server. Do các Video được xử lý một cách tuần tự nên nếu có nhiều người dùng cùng đăng bài một lúc thì có nguy cơ dẫn đến nghẽn Server hoặc phải chờ rất lâu để xử lý một Video.

Ngoài ra, bộ lọc và phân loại Video có độ chính xác chưa cao dẫn đến dễ phân loại nhầm hay bỏ sót Video. Nếu so với bộ lọc của những trang mạng xã hội nổi tiếng như Youtube thì Model hiện tại không thể nào bắt kịp về mặt chất lượng cũng như hiệu suất.

Tuy vậy, sau khi thực hiện ĐATN, em đã thiết kế và vận hành được một ứng dụng mô phỏng hoàn thiện của một trang mạng xã hội chia sẻ nội dung. Em cũng đã huấn luyện và cải thiện được một mô hình Deep Learning sử dụng mạng CNN để phân loại Video cũng như phát triển trang Web làm nền tảng cho mạng xã hội. Từ những đóng góp này, em đã rút ra được nhiều kinh nghiệm trong việc thiết kế, lập trình và kiểm thử một sản phẩm phần mềm nói chung.

### 6.2 Hướng phát triển

Để hoàn thiện trang Web đã có, em hướng tới việc đầu tiên là sử dụng Socket và các kỹ thuật đa luồng khác để xử lý việc phân loại Video từ phía Server cũng như thông báo đến người dùng theo thời gian thực. Em cũng muốn ứng dụng hệ thống thông báo và tin nhắn giữa các người dùng và thêm các tính năng tương tác nữa để làm trang Web sống động hơn.

Về các cải tiến đến trang Web, thành phần quan trọng nhất là bộ lọc nội dung. Để cải thiện Model với độ chính xác cao hơn, em sẽ hướng tới trước tiên là thu thập thêm thật nhiều dữ liệu và xử lý chúng để Model học được nhiều hơn. Tiếp đó khi Model đã bão hòa, có thể thay đổi các siêu tham số khác để xây dựng Model cải tiến và kiểm tra tính hiệu quả của nó.