

TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN

**HỌC PHẦN: CÔNG NGHỆ THÔNG TIN TRONG
CHUYỂN ĐỔI SỐ**

**TÊN ĐỀ TÀI: HỆ THỐNG NHẬN DIỆN THIẾT BỊ
BẢO HỘ CỦA CÔNG NHÂN TẠI CÔNG TRƯỜNG**

Triệu Khắc Tuấn Khoa

Đỗ Trọng Nguyên

Nguyễn Công Thành

Ngành: Công nghệ Thông tin

Chuyên ngành: Hệ thống thông tin

Giảng viên hướng dẫn: ThS. Lê Trung Hiếu

Khoa: Công nghệ Thông tin

HÀ NỘI, 08/2025

LỜI NÓI ĐẦU

Trong quá trình công nghiệp hóa, hiện đại hóa đất nước, các công trình xây dựng ngày càng nhiều, đóng góp quan trọng cho sự phát triển kinh tế – xã hội. Tuy nhiên, đi cùng với đó là những vấn đề về an toàn lao động, đặc biệt là các tai nạn đáng tiếc xảy ra do người lao động không sử dụng đúng hoặc không trang bị đầy đủ các thiết bị bảo hộ cá nhân (PPE). Đây là thách thức không nhỏ đối với các doanh nghiệp xây dựng cũng như cơ quan quản lý.

Trong bối cảnh đó, sự phát triển mạnh mẽ của công nghệ trí tuệ nhân tạo (AI) và thị giác máy tính (Computer Vision) mở ra nhiều cơ hội ứng dụng vào thực tiễn, trong đó có việc giám sát và kiểm tra tự động việc tuân thủ an toàn lao động. Với mong muốn vận dụng kiến thức đã học vào giải quyết bài toán thực tế, nhóm chúng em lựa chọn đề tài “Phát hiện thiết bị bảo hộ lao động bằng AI tại công trường”.

Đề tài tập trung nghiên cứu, xây dựng và thử nghiệm mô hình AI có khả năng phát hiện các thiết bị bảo hộ cơ bản như mũ bảo hộ, áo phản quang, đồng thời nhận diện tình huống vi phạm. Qua đó, hệ thống có thể hỗ trợ cán bộ quản lý theo dõi an toàn công nhân một cách nhanh chóng, chính xác, giảm thiểu rủi ro và nâng cao ý thức chấp hành quy định an toàn lao động. Kết quả nghiên cứu không chỉ mang tính học thuật mà còn có giá trị ứng dụng thực tế, góp phần thiết thực vào việc đảm bảo an toàn cho người lao động và hướng đến một môi trường làm việc hiện đại, văn minh và an toàn hơn.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu của đề tài.....	1
1.3 Phạm vi nghiên cứu	1
CHƯƠNG 2. CÔNG NGHỆ SỬ DỤNG.....	2
2.1 YOLOv8.....	2
2.2 Thư viện Ultralytics	3
2.3 Python	4
2.4 Streamlit.....	4
CHƯƠNG 3. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG ...	6
3.1 Tổng quan hệ thống	6
3.2 Sơ đồ khối hệ thống	6
3.3 Cấu trúc mã nguồn.....	7
3.4 Triển khai và thử nghiệm mô hình.....	8
CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	13

DANH MỤC HÌNH VẼ

Hình 2.1	Mô hình YOLOv8.	2
Hình 2.2	Thư viện Ultralytics	3
Hình 2.3	Python	4
Hình 2.4	Streamlit	5
Hình 3.1	Sơ đồ khối hệ thống phát hiện PPE	6
Hình 3.2	Cấu trúc mã nguồn	8

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

An toàn lao động tại công trường luôn là yếu tố then chốt để bảo vệ tính mạng và sức khỏe của công nhân. Tuy nhiên, việc giám sát thủ công thường tốn nhiều nhân lực, thời gian và dễ bỏ sót vi phạm. Với sự phát triển của trí tuệ nhân tạo và thị giác máy tính, việc ứng dụng các mô hình nhận diện tự động, đặc biệt là YOLOv8 kết hợp thư viện Ultralytics, có thể hỗ trợ phát hiện nhanh chóng và chính xác các vi phạm về thiết bị bảo hộ cá nhân (PPE). Đề tài này tập trung xây dựng hệ thống nhận diện PPE từ video, nhằm nâng cao hiệu quả giám sát, giảm rủi ro tai nạn và góp phần xây dựng môi trường làm việc an toàn hơn tại công trường.

1.2 Mục tiêu của đề tài

Mục tiêu của đề tài là nghiên cứu, xây dựng và triển khai một hệ thống phát hiện thiết bị bảo hộ lao động (PPE) dựa trên công nghệ thị giác máy tính và trí tuệ nhân tạo. Cụ thể, hệ thống ứng dụng mô hình YOLOv8 kết hợp thư viện Ultralytics để nhận diện các thiết bị như mũ bảo hộ, áo phản quang, găng tay và kính bảo hộ trong hình ảnh hoặc video.

Thông qua quá trình thử nghiệm, đề tài hướng tới các mục tiêu:

- Tự động hóa quá trình giám sát PPE nhằm giảm sự phụ thuộc vào kiểm tra thủ công.
- Nâng cao độ chính xác trong phát hiện và phân loại PPE.
- Hỗ trợ cảnh báo sớm khi phát hiện vi phạm quy định an toàn.
- Tạo nền tảng cho việc mở rộng hệ thống giám sát an toàn lao động ở quy mô lớn.

1.3 Phạm vi nghiên cứu

Đề tài tập trung nghiên cứu và ứng dụng mô hình YOLOv8 kết hợp thư viện Ultralytics để phát hiện các thiết bị bảo hộ lao động (PPE) như mũ bảo hộ, áo phản quang, găng tay và kính bảo hộ trong hình ảnh và video. Hệ thống chỉ xử lý dữ liệu từ bộ dữ liệu công khai và một số video mô phỏng môi trường công trường, chưa mở rộng sang các loại PPE khác như giày bảo hộ hay mặt nạ phòng độc. Quá trình triển khai được giới hạn ở mức thử nghiệm trên máy tính cá nhân, chưa tích hợp vào hệ thống giám sát thực tế quy mô lớn, và không nghiên cứu các thuật toán thị giác máy tính ngoài YOLOv8.

CHƯƠNG 2. CÔNG NGHỆ SỬ DỤNG

2.1 YOLOv8

YOLOv8 (You Only Look Once phiên bản 8) là một mô hình phát hiện đối tượng tiên tiến do Ultralytics phát triển, ra mắt vào đầu năm 2023. Mặc dù hiện nay đã có những phiên bản mới hơn, YOLOv8 vẫn được sử dụng rộng rãi nhờ tốc độ xử lý cao, độ chính xác tốt, cùng kiến trúc mạng được cải tiến so với các thế hệ trước. Ngoài khả năng phát hiện đối tượng, YOLOv8 còn hỗ trợ các tác vụ như phân đoạn ảnh, phân loại và theo dõi đối tượng, đáp ứng nhu cầu đa dạng trong lĩnh vực thị giác máy tính.



Hình 2.1: Mô hình YOLOv8.

+ Nguyên lý hoạt động của YOLOv8

YOLOv8 hoạt động dựa trên cơ chế phát hiện đối tượng theo thời gian thực bằng cách chia toàn bộ hình ảnh đầu vào thành một lưới (grid) và dự đoán trực tiếp các hộp giới hạn (bounding boxes) cùng xác suất thuộc về từng lớp đối tượng trong một lần xử lý duy nhất. Khi nhận ảnh hoặc khung hình từ video, YOLOv8 sử dụng một mạng nơ-ron tích chập (CNN) với kiến trúc được tối ưu hóa để trích xuất đặc trưng ở nhiều cấp độ. Sau đó, mô hình áp dụng các đầu ra dự đoán để xác định vị trí (tọa độ hộp), kích thước, độ tin cậy (confidence score) và nhãn của từng đối tượng.

YOLOv8 còn tích hợp các kỹ thuật như anchor-free detection, Non-Maximum Suppression (NMS) để loại bỏ các dự đoán trùng lặp, và có thể thực hiện đồng thời nhiều tác vụ như phát hiện (detection), phân đoạn (segmentation) hoặc phân loại (classification). Nhờ vậy, mô hình đạt tốc độ xử lý cao, độ chính xác tốt, và phù hợp cho các ứng dụng yêu cầu thời gian thực như giám sát an toàn lao động, giao

thông, hoặc hệ thống an ninh.

2.2 Thư viện Ultralytics

Thư viện Ultralytics là một nền tảng mã nguồn mở mạnh mẽ, được phát triển và duy trì bởi công ty Ultralytics, nhằm cung cấp các công cụ tối ưu cho việc huấn luyện, triển khai và ứng dụng các mô hình thị giác máy tính, đặc biệt là các phiên bản YOLO (từ YOLOv3, YOLOv5, YOLOv8 đến YOLOv9). Thư viện này được viết chủ yếu bằng Python, tích hợp sâu với các framework học sâu phổ biến như PyTorch, cho phép người dùng dễ dàng xây dựng, tinh chỉnh (fine-tune) và triển khai mô hình trên nhiều môi trường khác nhau, từ máy tính cá nhân đến hệ thống đám mây và thiết bị nhúng.



Hình 2.2: Thư viện Ultralytics .

Ultralytics dựa trên việc cung cấp một môi trường làm việc thống nhất, giúp người dùng dễ dàng thực hiện từ khâu chuẩn bị dữ liệu, huấn luyện, đánh giá cho đến triển khai mô hình. Khi sử dụng, thư viện sẽ tự động xử lý dữ liệu đầu vào bằng các kỹ thuật tiền xử lý và tăng cường dữ liệu như thay đổi kích thước, chuẩn hóa, xoay, lật hay điều chỉnh độ sáng để cải thiện độ chính xác. Mô hình được khởi tạo có thể là các kiến trúc YOLO có sẵn hoặc mô hình đã huấn luyện trước, được tối ưu hóa trên nền tảng PyTorch để đảm bảo tốc độ và hiệu quả. Quá trình huấn luyện được thực hiện thông qua cơ chế tối ưu siêu tham số, tính toán hàm mất mát và cập nhật trọng số bằng backpropagation, đồng thời theo dõi trực quan bằng TensorBoard hoặc Weights và Biases.

Sau khi huấn luyện, Ultralytics đánh giá mô hình dựa trên các chỉ số mAP, precision và recall, đồng thời hỗ trợ xuất sang nhiều định dạng như ONNX, TensorRT, CoreML để triển khai trên nhiều nền tảng khác nhau. Nhờ cách hoạt động này, Ultralytics giúp rút ngắn thời gian phát triển, đồng thời đảm bảo mô hình đạt hiệu suất cao và dễ dàng ứng dụng vào các bài toán thực tế như giám sát an ninh, phát hiện vật thể hay nhận dạng thiết bị bảo hộ lao động.

2.3 Python

Python là ngôn ngữ lập trình bậc cao, nổi tiếng nhờ cú pháp đơn giản, dễ đọc và khả năng mở rộng mạnh mẽ. Python được ứng dụng rộng rãi trong nhiều lĩnh vực như phân tích dữ liệu, phát triển phần mềm, trí tuệ nhân tạo, học máy và tự động hóa. Hệ sinh thái Python bao gồm hàng nghìn thư viện và framework mạnh mẽ như NumPy, Pandas, TensorFlow, PyTorch, hỗ trợ tối đa cho các tác vụ tính toán và nghiên cứu.



Hình 2.3: Python

- Công dụng:

Python được sử dụng rộng rãi trong phát triển phần mềm, phân tích dữ liệu, trí tuệ nhân tạo, web, tự động hóa, và nhiều lĩnh vực khác. Thư viện phong phú như NumPy, Pandas, TensorFlow, PyTorch cho phép xử lý hiệu quả từ dữ liệu cơ bản đến các mô hình học sâu phức tạp.

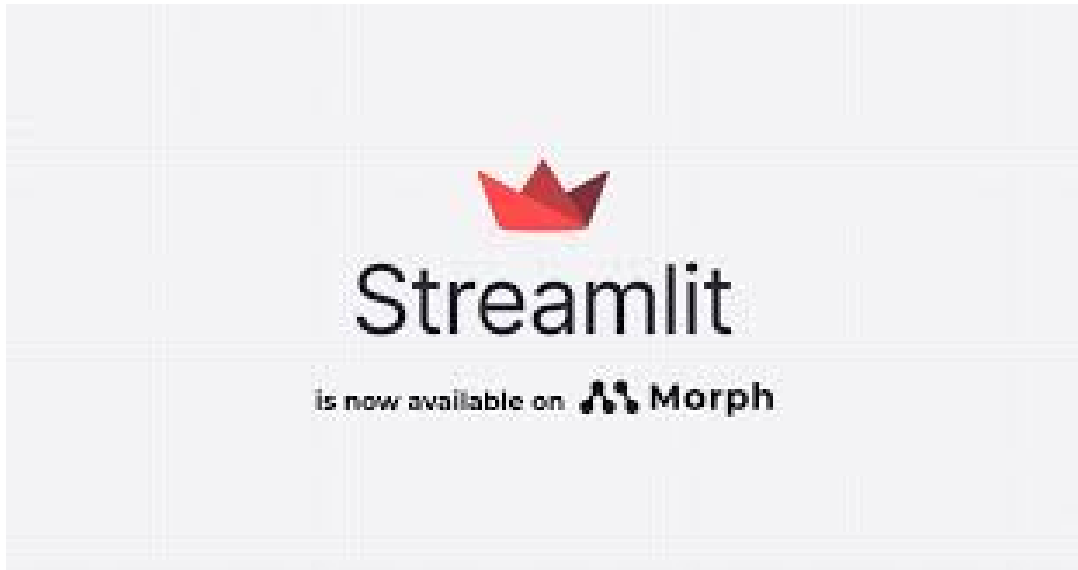
- Cách thức hoạt động:

Python hoạt động dưới dạng một trình thông dịch (interpreter). Khi người dùng viết mã, trình thông dịch Python sẽ đọc từng dòng lệnh, phân tích cú pháp (parsing) và thực thi ngay lập tức. Điều này giúp thử nghiệm và phát triển nhanh chóng, đồng thời dễ dàng phát hiện và sửa lỗi. Python có thể chạy độc lập trên nhiều hệ điều hành hoặc tích hợp vào các nền tảng trực tuyến như Google Colab để tận dụng tài nguyên tính toán từ xa.

2.4 Streamlit

Streamlit được sử dụng để xây dựng giao diện web cho hệ thống. Công nghệ này cho phép phát triển ứng dụng một cách nhanh chóng, trực quan và dễ dàng tương tác. Nhờ Streamlit, kết quả nhận diện từ mô hình có thể được hiển thị ngay trên

trang web, đồng thời hỗ trợ hiển thị báo cáo thống kê vi phạm PPE, giúp người dùng dễ dàng theo dõi và đánh giá.



Hình 2.4: Streamlit

CHƯƠNG 3. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

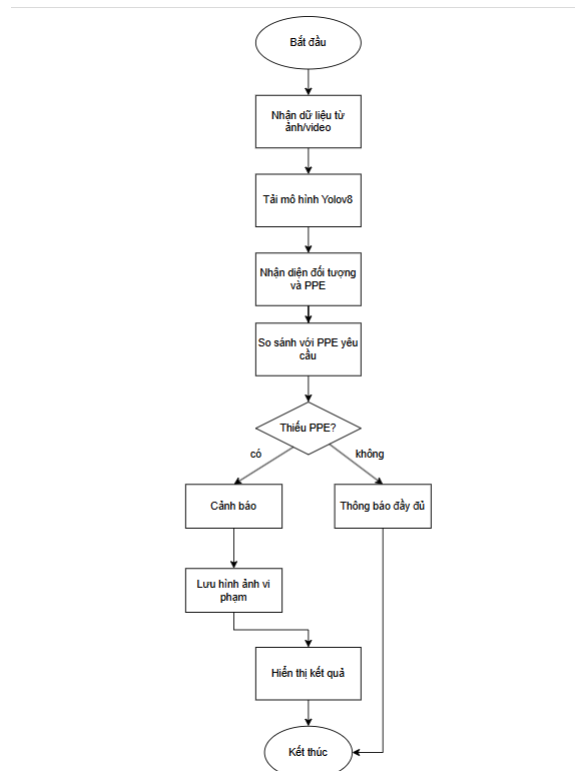
3.1 Tổng quan hệ thống

Hệ thống nhận diện thiết bị bảo hộ lao động (PPE) được xây dựng nhằm phát hiện và phân loại các đối tượng như mũ bảo hộ, áo phản quang, kính bảo hộ, găng tay,... trong môi trường công trường. Về tổng thể, hệ thống hoạt động dựa trên mô hình học sâu YOLOv8 kết hợp thư viện Ultralytics để xử lý hình ảnh và xuất kết quả nhận diện theo thời gian thực.

Quy trình của hệ thống bao gồm các bước chính: thu thập và xử lý dữ liệu hình ảnh, huấn luyện mô hình nhận diện, triển khai mô hình trên nền tảng hỗ trợ (Google Colab), và đánh giá hiệu quả thông qua các chỉ số chính xác, tốc độ xử lý và độ tin cậy. Phần tiếp theo của chương sẽ trình bày chi tiết từng thành phần và quy trình này.

3.2 Sơ đồ khối hệ thống

Hệ thống phát hiện thiết bị bảo hộ lao động (PPE) được thiết kế dựa trên mô hình YOLO. Quy trình hoạt động tổng quát được thể hiện qua sơ đồ khối bên dưới.



Hình 3.1: Sơ đồ khối hệ thống phát hiện PPE

Trước hết, dữ liệu đầu vào được thu nhận từ camera giám sát hoặc video tại công trường. Sau đó, dữ liệu hình ảnh sẽ được đưa vào mô hình YOLO PPE đã huấn luyện để tiến hành xử lý và phát hiện đối tượng. Ở bước này, hệ thống sẽ nhận diện các loại thiết bị bảo hộ như mũ bảo hộ, áo phản quang, khẩu trang, găng tay,...

Khi mô hình phát hiện đối tượng, kết quả sẽ được kiểm tra và phân loại nhằm xác định người lao động có tuân thủ đúng quy định an toàn hay không. Nếu phát hiện vi phạm (thiếu hoặc không sử dụng PPE bắt buộc), hệ thống sẽ ghi nhận lại hình ảnh và cảnh báo. Ngược lại, nếu đầy đủ PPE, kết quả sẽ được hiển thị bình thường.

Nhờ đó, hệ thống có thể hỗ trợ việc giám sát an toàn lao động, giảm thiểu rủi ro và nâng cao hiệu quả quản lý tại công trường.

3.3 Cấu trúc mã nguồn

Hệ thống được cài đặt với cấu trúc mã nguồn đơn giản, gồm các tệp chính sau:

ppe.py: Đây là file chương trình chính, chứa mã nguồn thực thi hệ thống nhận diện thiết bị bảo hộ (PPE). File này sẽ gọi mô hình YOLOv8, xử lý video/hình ảnh đầu vào và hiển thị kết quả nhận diện.

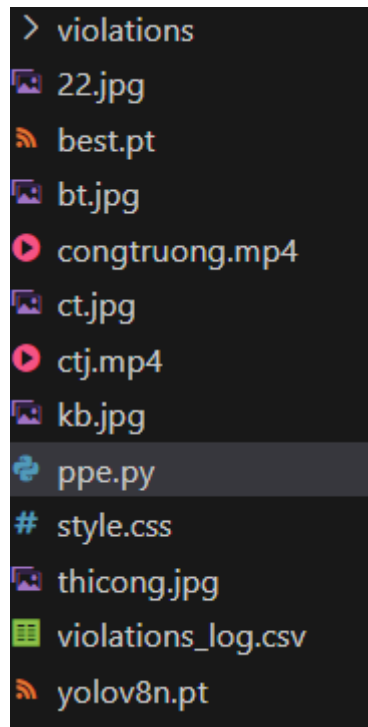
best.pt và yolov8n.pt: Đây là các file mô hình đã huấn luyện. Trong đó, best.pt là mô hình đã được huấn luyện và tinh chỉnh riêng cho bài toán nhận diện PPE, còn yolov8n.pt là mô hình gốc YOLOv8n (phiên bản nhẹ) được dùng làm nền tảng trước khi fine-tune.

style.css: File định dạng giao diện, chứa các quy tắc CSS để tạo bố cục, màu sắc và kiểu dáng cho hệ thống, giúp giao diện trực quan và thân thiện với người dùng.

violationslog.csv: File nhật ký (log) lưu lại kết quả phát hiện vi phạm PPE. Mỗi dòng trong file này ghi lại thông tin về thời gian, loại vi phạm và các chi tiết liên quan, phục vụ cho việc theo dõi và thống kê.

Các file video (ctj.mp4, congtruong.mp4): Đây là dữ liệu mẫu được sử dụng để kiểm thử hệ thống. Các hình ảnh và video này mô phỏng tình huống thực tế tại công trường, cho phép mô hình kiểm tra khả năng phát hiện và phân loại các thiết bị bảo hộ.

violations (thư mục): Thư mục này chứa các hình ảnh/video được hệ thống ghi lại sau khi phát hiện vi phạm PPE. Đây là bằng chứng trực quan để quản lý công trường xem xét và xử lý.



Hình 3.2: Cấu trúc mã nguồn

3.4 Triển khai và thử nghiệm mô hình

+ Dưới đây là toàn bộ mã nguồn của hệ thống nhận diện PPE:

```

1 import streamlit as st
2 from ultralytics import YOLO
3 import cv2
4 import os
5 from datetime import datetime
6 import tempfile
7
8
9 model = YOLO("best.pt") # Load model PPE
10 save_dir = "violations"
11 os.makedirs(save_dir, exist_ok=True)
12 required_ppe = {"hardhat", "safety vest", "mask"}
13
14
15 st.set_page_config(page_title="PPE Detection", page_icon="🚧", layout="wide")
16
17 with open("style.css") as f:
18     st.markdown(f"<style>{f.read()}</style>", unsafe_allow_html=True)
19
20 st.markdown('<p class="big-title">🚧 PPE Detection System</p>', unsafe_allow_html=True)
21 st.markdown('<p class="big-subtitle">Hệ thống nhận diện thiết bị bảo hộ của công nhân tại công trường </p>', unsafe_allow_html=True)
22 st.write("----")
23
24 # Sidebar
25 st.sidebar.header("⚙️ Cài đặt hệ thống")
26 uploaded_file = st.sidebar.file_uploader("📁 Tải video công trường", type=["mp4", "avi", "mov"])
27 save_interval = st.sidebar.slider("⌚ Thời gian lưu ảnh vi phạm (giây)", 3, 15, 5)
28 conf_threshold = st.sidebar.slider("🎯 Ngưỡng tin cậy", 0.3, 0.9, 0.5)
29
30 # Main layout
31 col1, col2 = st.columns([1, 2])
32
33 with col1:
34     st.subheader("📄 Thông tin")
35     st.info(
36         f"""
37         **PPE bắt buộc:**

```

```

33 with col1:
34     st.subheader("📄 Thông tin")
35     st.info(
36         f"""
37         **PPE bắt buộc**:
38         - 🧢 Mũ bảo hộ
39         - 🦺 Áo phản quang
40         - 🧢 Khẩu trang
41         """)
42     )
43
44
45 with col2:
46     if uploaded_file is not None:
47         tfile = tempfile.NamedTemporaryFile(delete=False)
48         tfile.write(uploaded_file.read())
49         video_path = tfile.name
50
51         # Hiển thị video gốc
52         st.video(video_path)
53
54         if st.button("▶ Bắt đầu nhận diện", type="primary"):
55             cap = cv2.VideoCapture(video_path)
56
57             last_save_time = None
58             stframe = st.empty()
59             saved_images = [] # danh sách ảnh vi phạm đã lưu
60
61             while cap.isOpened():
62                 ret, frame = cap.read()
63                 if not ret:
64                     break
65
66                 results = model(frame)
67                 violation_found = False
68
69                 for result in results:
70                     boxes = result.boxes
71                     names = model.names
72
73                     for box in boxes:
74                         cls_id = int(box.cls[0])
75                         label = names[cls_id].lower()
76                         conf = float(box.conf[0])
77                         if conf < conf_threshold:
78                             continue
79                         x1, y1, x2, y2 = map(int, box.xyxy[0])
80
81                         color = (0, 255, 0)
82                         if "no" in label:
83                             color = (0, 0, 255)
84                         cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
85                         cv2.putText(frame, f"{label} {conf:.2f}", (x1, y1 - 5),
86                                     cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 1)
87
88                         if "person" in label:
89                             pid = len(person_ppe) + 1
90                             person_ppe[pid] = set()
91                             person_boxes[pid] = (x1, y1, x2, y2)
92                         elif any(ppe in label for ppe in required_ppe):
93                             if len(person_ppe) > 0:
94                                 person_ppe[list(person_ppe.keys())[-1]].add(label)
95
96             for pid, ppe_set in person_ppe.items():
97                 missing = required_ppe - ppe_set
98                 if missing:
99                     violation_found = True
100                     x1, y1, x2, y2 = person_boxes[pid]
101                     cv2.putText(frame, f"Thiếu: {' '.join(missing)}", (x1, y2 + 20),
102                                 cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
103
104             if violation_found:
105                 now = datetime.now()
106                 if last_save_time is None or (now - last_save_time).total_seconds() >= save_interval:
107                     timestamp = now.strftime("%Y%m%d_%H%M%S")
108                     save_path = os.path.join(save_dir, f"phathienvipham_{timestamp}.jpg")
109                     cv2.imwrite(save_path, frame)
110                     saved_images.append(save_path)
111                     st.sidebar.error(f"🚨 Ảnh vi phạm đã lưu: {save_path}")
112                     last_save_time = now
113
114             frame_resized = cv2.resize(frame, (720, 400))
115             stframe.image(frame_resized, channels="BGR")
116
117             cap.release()
118             st.success("✅ Nhận diện hoàn tất!")
119
120             # Hiển thị ảnh vi phạm trong web
121             if saved_images:
122                 st.subheader("📁 Ảnh vi phạm đã phát hiện")
123                 cols = st.columns(3)
124                 for i, img_path in enumerate(saved_images):
125                     cols[i % 3].image(img_path, use_container_width=True)
126
127 # Footer
128 st.markdown('<p class="footer">© 2025 PPE Detection System - Demo</p>', unsafe_allow_html=True)

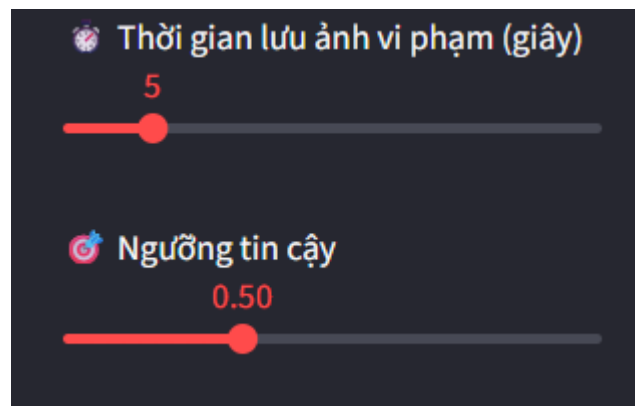
```

+ Các bước để chạy demo nhận diện PPE:

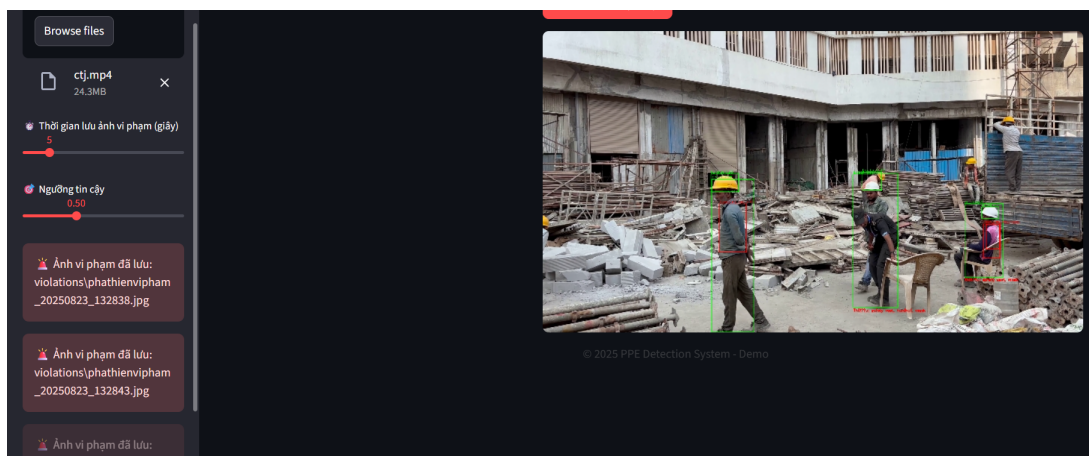
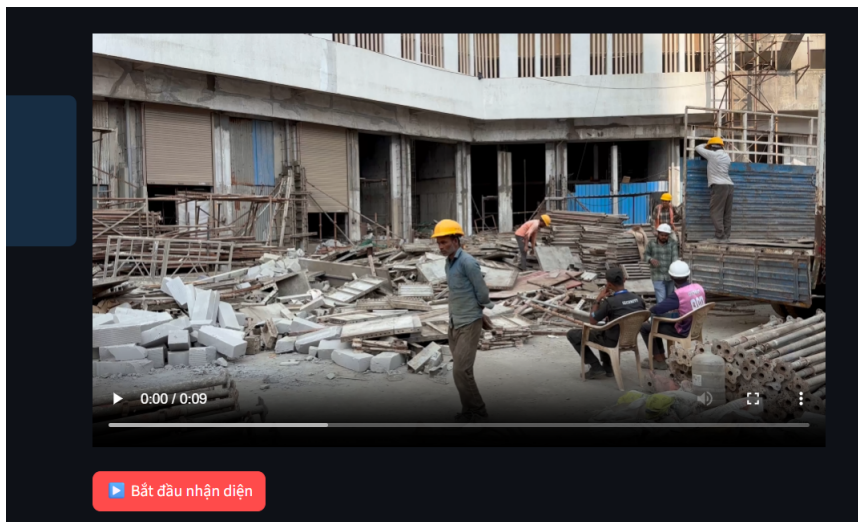
- Người dùng chọn một video công trường cần kiểm tra bằng cách tải file lên thông qua giao diện Streamlit. Các định dạng được hỗ trợ: MP4, AVI, MOV.



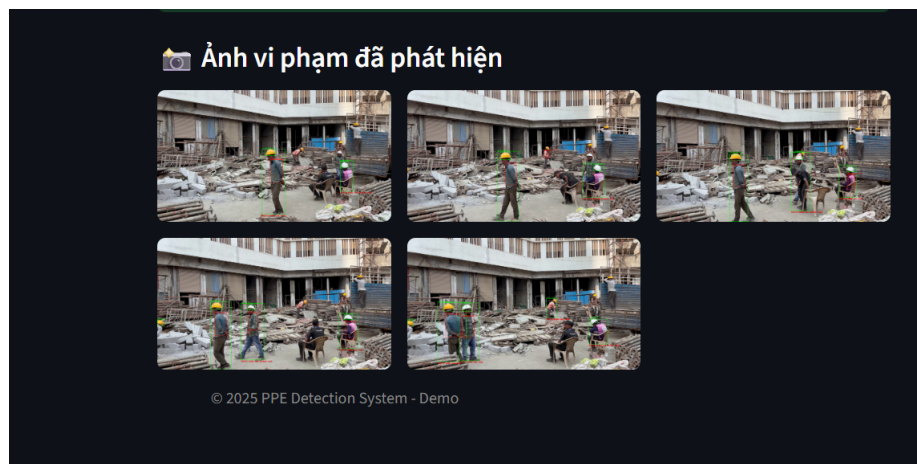
- Người dùng cấu hình tham số cho video để hệ thống Ở bước này, người dùng cần tiến hành thiết lập các tham số quan trọng để hệ thống giám sát hoạt động ổn định và chính xác. Tham số thứ nhất là thời gian lưu ảnh vi phạm (tính theo giây), đây là khoảng thời gian giữa hai lần hệ thống ghi nhận và lưu lại hình ảnh một vi phạm.



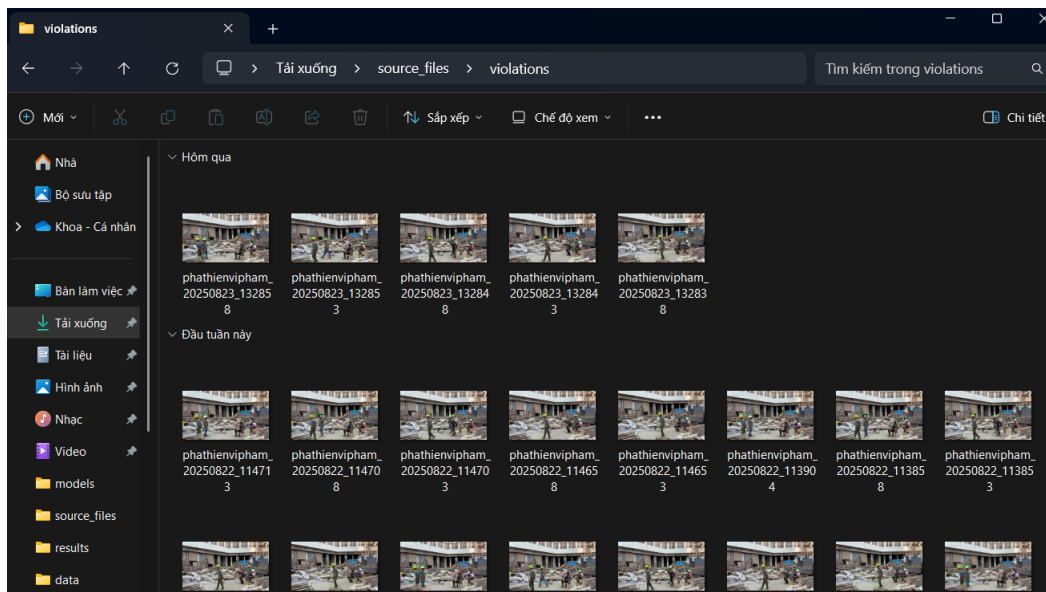
Sau khi hoàn tất việc cấu hình các tham số, người dùng tiến hành chạy hệ thống bằng cách nhấn vào nút “Bắt đầu nhận diện” trên giao diện. Tại thời điểm này, hệ thống sẽ kích hoạt mô hình trí tuệ nhân tạo đã được huấn luyện và bắt đầu thực hiện quá trình phát hiện công nhân cũng như các thiết bị bảo hộ cá nhân (PPE) như mũ bảo hộ, găng tay, áo phản quang, kính bảo hộ hay giày an toàn.



+ Kết quả đạt được



- Ngoài ra sau khi nhận diện hoàn tất, tất cả ảnh vi phạm đều được lưu trữ tại thư mục violations, nhờ đó chúng ta có thể xem một cách rõ ràng hơn.



Hệ thống nhận diện thiết bị bảo hộ lao động (PPE Detection System) sau khi triển khai đã cho thấy khả năng hoạt động hiệu quả trong môi trường thực tế. Khi kết nối với camera giám sát, hệ thống tự động phát hiện và đánh dấu các trường hợp công nhân vi phạm quy định an toàn, chẳng hạn như không đội mũ bảo hộ, không mang găng tay hoặc thiếu các trang bị cần thiết khác. Các khung hình vi phạm được hiển thị trực quan và đồng thời được lưu trữ lại, giúp cho cán bộ quản lý dễ dàng theo dõi, kiểm tra và tổng hợp báo cáo.

CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Hệ thống nhận diện thiết bị bảo hộ lao động (PPE Detection System) được xây dựng đã chứng minh tính ứng dụng thực tiễn trong việc hỗ trợ quản lý an toàn tại công trường. Với khả năng tự động phát hiện, hiển thị và lưu lại hình ảnh các trường hợp vi phạm quy định về PPE, hệ thống góp phần nâng cao hiệu quả giám sát, giảm tải công việc cho cán bộ quản lý và hạn chế rủi ro tai nạn lao động.

Quá trình thử nghiệm cho thấy hệ thống hoạt động ổn định, có thể nhận diện đồng thời nhiều đối tượng trong môi trường phức tạp với độ chính xác tương đối cao. Đây là minh chứng cho sự kết hợp hiệu quả giữa công nghệ thị giác máy tính và trí tuệ nhân tạo trong lĩnh vực đảm bảo an toàn lao động.

Trong tương lai, hệ thống có thể được mở rộng và hoàn thiện theo nhiều hướng để tăng cường hiệu quả và tính ứng dụng. Trước hết, cần nâng cao độ chính xác của mô hình nhận diện thông qua việc huấn luyện trên tập dữ liệu lớn và đa dạng hơn, bao gồm nhiều loại trang bị bảo hộ và tình huống thực tế khác nhau. Tiếp theo, hệ thống có thể được tích hợp với nền tảng IoT và các cảm biến thông minh để cung cấp cảnh báo thời gian thực qua thiết bị di động hoặc hệ thống quản lý trung tâm.

Ngoài ra, việc phát triển giao diện trực quan, thân thiện với người dùng, cũng như bổ sung các chức năng báo cáo thống kê chi tiết theo ca, ngày, tháng sẽ giúp ích nhiều cho công tác quản lý. Về lâu dài, hệ thống có thể được triển khai ở quy mô lớn trong các ngành công nghiệp, kết nối với cơ sở dữ liệu doanh nghiệp nhằm hình thành một giải pháp quản lý an toàn toàn diện, hiện đại và hiệu quả.

TÀI LIỆU THAM KHẢO

- [1] <https://github.com/snehilsanyal/Construction-Site-Safety-PPE-Detection>
- [2] "Personal Protective Equipment Detection: A Deep-Learning-Based and Sustainable Approach" T. Wang, X. Liu, J. Joshi