**Bản tóm tắt Project – Pose Estimation**

1. **Tổng quan về LSTM và bài toán Pose Estimation**
2. **LSTM**

Long Short-Term Memory (LSTM) là một loại kiến trúc mạng neural đặc biệt trong lĩnh vực học sâu (deep learning), thuộc loại Recurrent Neural Network (RNN). LSTM được phát triển để giải quyết một vấn đề lớn trong RNN truyền thống, đó là vấn đề "vanishing gradient" (biến mất gradient).

Cấu trúc của LSTM:

* Cell State: LSTM sử dụng một bộ nhớ gọi là cell state, có thể coi như là một đường dẫn dọc qua toàn bộ chuỗi. Cell state được điều chỉnh thông qua các cổng (gates) khác nhau.
* Forget Gate: Cổng này quyết định thông tin nào cần được loại bỏ khỏi cell state.
* Input Gate: Cổng này xác định thông tin mới nào sẽ được thêm vào cell state.
* Output Gate: Cổng này quyết định giá trị nào từ cell state sẽ được xuất ra ngoài.

LSTM đã trở thành một công cụ mạnh mẽ trong việc xử lý các bài toán liên quan đến dữ liệu tuần tự và giải quyết hiệu quả các vấn đề mà RNN truyền thống gặp phải. Sự linh hoạt và khả năng ghi nhớ dài hạn của nó làm cho LSTM trở thành một trong những lựa chọn phổ biến nhất trong lĩnh vực học sâu và trí tuệ nhân tạo.

1. **Pose Estimation**

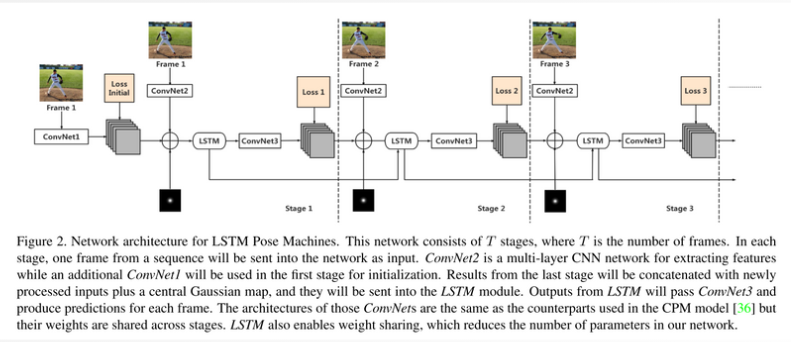
Pose Estimation là một bài toán quan trọng trong lĩnh vực thị giác máy tính (Computer Vision) và trí tuệ nhân tạo (AI), liên quan đến việc nhận diện và định vị các bộ phận cơ thể (pose) của người hoặc động vật trong một hình ảnh hoặc video. Mục tiêu chính của bài toán là xác định vị trí các điểm khớp (keypoints) đại diện cho các bộ phận chính của cơ thể, chẳng hạn như đầu, vai, khuỷu tay, đầu gối, và hông.

1. **Điểm nổi bật của LSTM so với các mô hình trước đó**

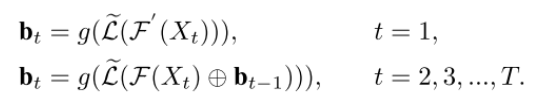
LSTM nổi bật trong việc xử lý bài toán Pose Estimation nhờ vào khả năng ghi nhớ dài hạn và ngắn hạn, khả năng xử lý thông tin theo chuỗi, và sự kết hợp hiệu quả với các mô hình CNN.

Điểm nổi bật nhất phải kể đến là khả năng ghi nhớ thông tin lâu dài của LSTM. LSTM được thiết kế để giải quyết vấn đề vanishing gradient (mất mát gradient) mà các mô hình trước đó, như RNN truyền thống, thường gặp phải khi xử lý các chuỗi dài. Điều này rất quan trọng trong Pose Estimation vì các tư thế của con người thường có mối liên hệ với nhau trong suốt chuỗi các khung hình video. Nhờ vào cơ chế "forget gate" và "input gate", LSTM có khả năng chọn lọc thông tin nào cần được lưu giữ hoặc quên đi, giúp mô hình học được các mẫu chuyển động phức tạp và duy trì thông tin quan trọng trong nhiều khung hình.

1. **Kiến trúc của mô hình**

****

* LSTM được chia thành nhiều stage khác nhau như đã đề cập ở trên. Ở mỗi stage sẽ có 1 hàm loss. Tác dụng của các hàm loss này là giúp cho mô hình học và cải thiện những đặc trưng trích xuất được từ stage trước. Giúp mô hình không chỉ tối ưu hoá 1 lần ở đầu ra cuối cùng mà tối ưu hoá qua nhiều bước trung gian.
* ConvNet1: được sử dụng ở giai đoạn đầu tiên để trích xuất các cơ sở cụ thể từ khung đầu tiên của chuỗi.
* Từ stage thứ hai trở đi, ConvNet2 được sử dụng để trích xuất các đặc trưng mới từ frame hiện tại của chuỗi. Nó cũng giúp tinh chỉnh các đặc trưng đã được học từ frame trước.
* ConvNet3 nhận đầu ra từ LSTM và tinh chỉnh đặc trưng này để tạo ra dự đoán cuối cùng cho mỗi frame.



* Trong công thức sau thì Convnet1, Convnet2 là F(Xt). Và Convnet3 là g().
* Luồng hoạt động :

+) Ảnh đầu vào (input) sẽ được đưa qua mạng Convnet1 để trích xuất các đặc trưng căn bản từ ảnh. Sau đó đặc trưng sẽ qua 1 hàm loss và 1 mạng ConvNetv2 để tinh chỉnh và học các đặc trưng sâu hơn. Ở mỗi stage thì mô hình sẽ lấy lại output của stage trước đó kết hợp với các đặc trưng học được ở stage hiện tại để LSTM học tiếp (Convnetv3).

+) Từ đó sẽ giúp cho model có khả năng học được đầu đủ các đặc trưng của nhiều frame liên tiếp (video như trong project lần này).

+) Vậy ta có thể tổng kết lại luồng hoạt động như sau:

* + ConvNet1 trích xuất đặc trưng cơ bản từ frame đầu tiên.
  + LSTM tích hợp thông tin qua các frame theo thời gian.
  + ConvNet2 và ConvNet3 giúp tinh chỉnh và tối ưu hóa đặc trưng từ LSTM để đưa ra dự đoán chính xác hơn.
  + Các hàm loss giúp tối ưu hóa mô hình ở từng stage, đảm bảo rằng mô hình học tập và cải thiện liên tục qua từng bước.

1. **Quy trình các bước tiến hành:**

B1: Thu thập data những động tác mới.

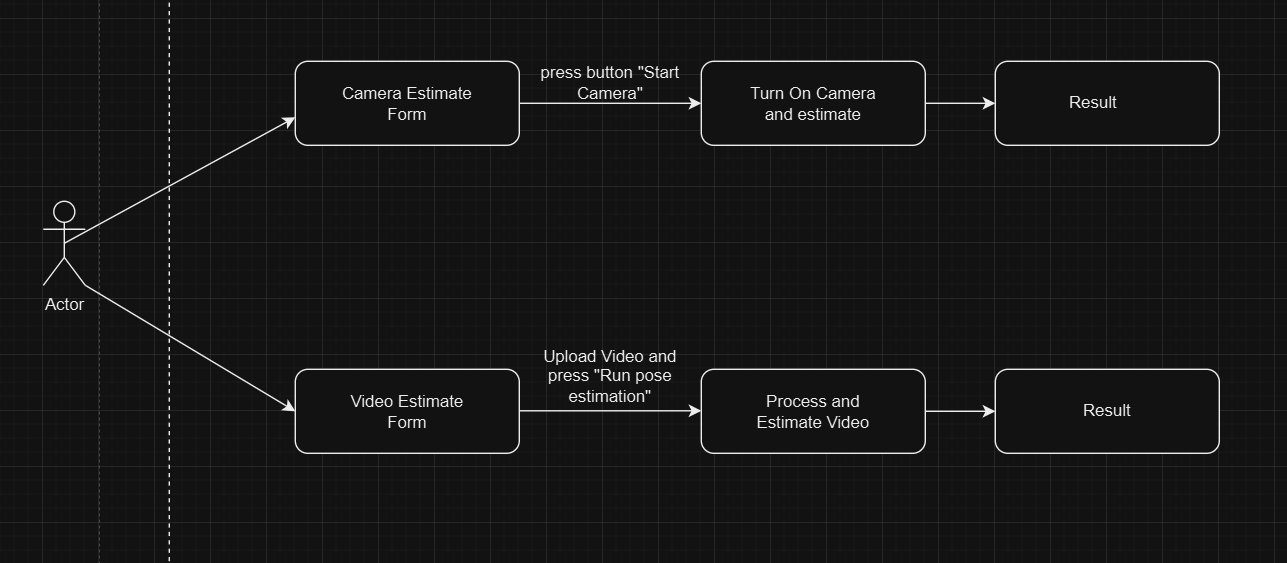
B2: Tinh chỉnh lại lớp Dense cuối cùng cho phù hợp với số lượng class mới.

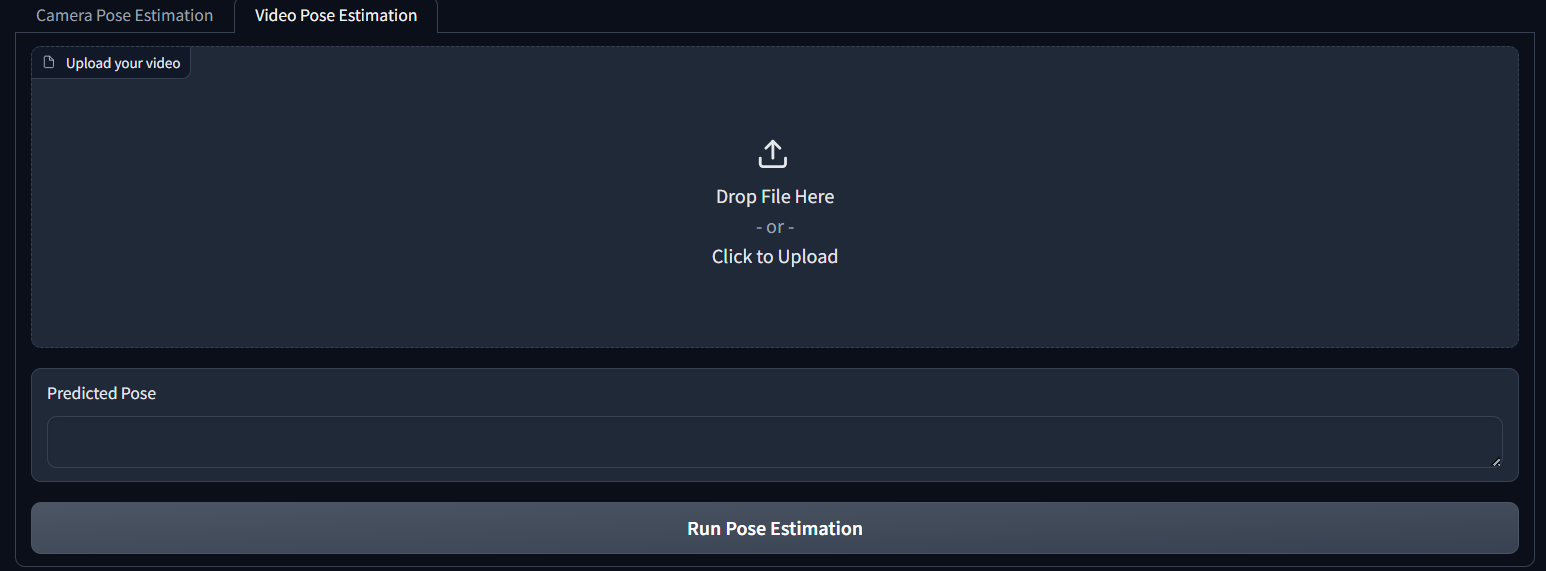
B3: Train model.

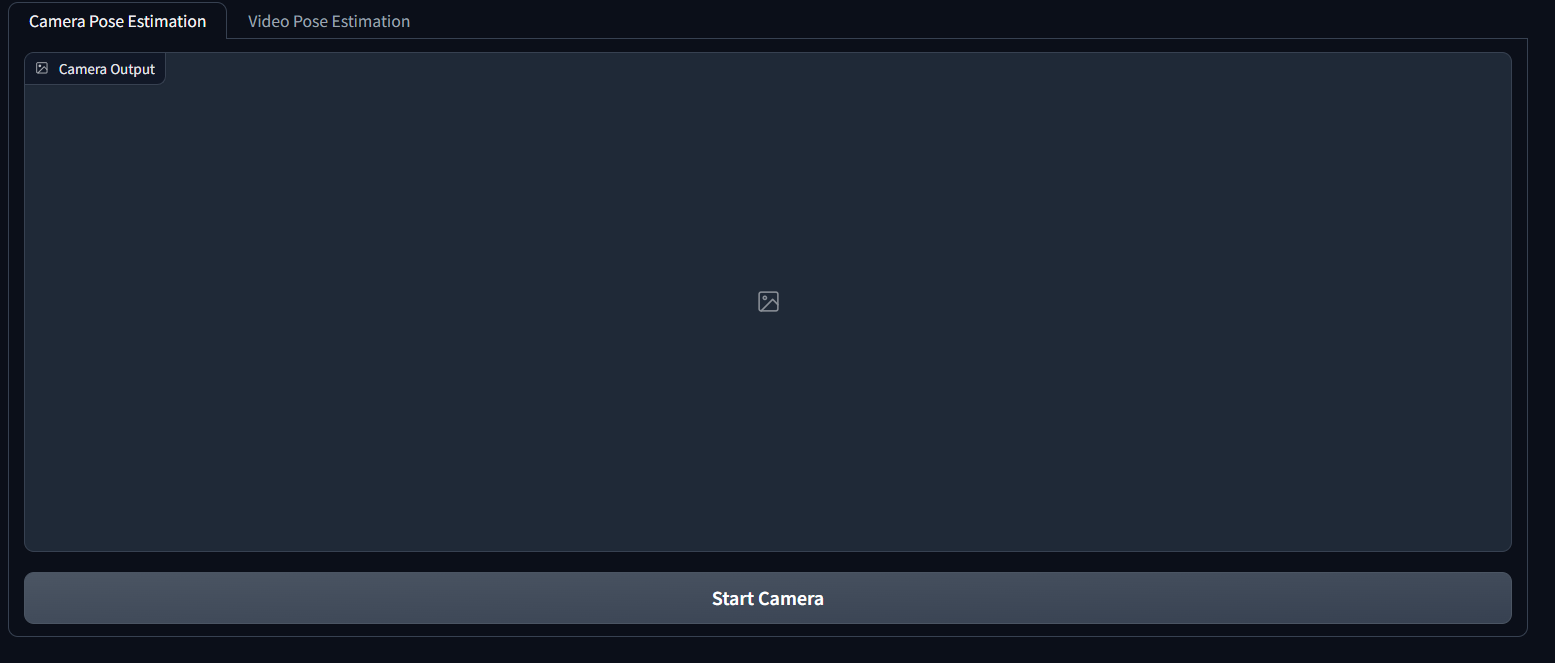
B4: Tạo giao diện bằng Gradio..

B5: Nhận diện động tác thông qua giao diện được tạo bằng Gradio.

1. **Work Flow Diagram và Hình ảnh giao diện**







1. **Những điểm cần cải thiện**

* Vẫn bị trùng lặp giữa những động tác gần giống nhau.
* Xử lý với 2 người trở lên trong cùng 1 khung ảnh.
* Model chỉ ở mức nhận diện được những động tác, chưa thể đáp ứng được việc. phát hiện những lỗi sai từ các động tác.
* Chỉ số FPS còn thấp.