**Nhận dạng khuôn mặt người với thông tin không đầy đủ**

1. **Tiền xử lý (Preprocessing)**
   1. **Xác định khuôn mặt trong ảnh**

ảnh được chia thành ma trận ô vuông 7x7, mỗi ô vuông bao gồm một tập các thông tin mà mô hình phải dự đoán.

* 1. **Rút trích mắt trái, mắt phải, mũi, miệng**

Sau khi lấy được khuôn mặt bằng HaarCascades, bằng việc sử dụng thư viện dlib để xác định 68 điểm mốc trên khuôn mặt.

Sau khi xác định được các điểm mốc, nhóm em dễ dàng rút trích các bộ phận như mắt trái, mắt phải, mũi miệng và khuôn mặt.

* 1. **Canh chỉnh lại khuôn mặt**

Đối với khuôn mặt thì cần phải canh chỉnh lại, nên nhóm sử dụng thư viện AlignDlib canh chỉnh lại khuôn mặt dựa trên mắt và mũi của ảnh

Sau khi xong các bước thì tập ảnh đó sẽ được chia ra thành các bộ phận như mắt trái, mắt phải, mũi, miệng và khuôn mặt và lưu lại để cho việc huấn luyện.

1. **Mô hình CNN**

Nhóm em sử dụng thư viện keras của tensorflow và khai báo mô hình Sequential, mỗi layer của mô hình Sequential chỉ cho phép một input tensor và 1 output tensor.

* Kiến trúc CNN gồm 5 lớp Conv, 5 lớp MaxPooling, sau đó là 2 tầng Conv với kernel size = (1,1) thay cho tầng fully conntected. Loss categorical\_crossentropy được sử dụng để dữ liệu dự đoán sẽ trả về là đa lớp (multi-class) vì bài toán là nhận dạng khuôn mặt người với giá trị trả về là 1 class để chỉ người, nên có rất là nhiều class.

Đầu tiên, nhóm em sử dụng 1 cặp lớp gồm 1 Conv2D và MaxPooling để làm giảm đi kích thước ảnh và làm tăng chiều sâu cảu ảnh. Tương tự như vậy tiếp tục với 3 cặp Conv2D và MaxPooling để tiếp tục làm phẳng dữ liệu ảnh ban đầu. Để tránh trường vấn đề overfitting, nhóm em thêm 1 lớp dropout(0.2). Bên cạnh đó EarlyStopping để kết thúc train nếu accuracy không tăng sau một số epochs nhất định. ModelCheckpoint để lưu lại trọng số tốt nhất của model. ReduceLROnPlateau để giảm learning rate sau mỗi lần epoch nhưng không cải thiện được accuracy.

Cấu trúc layer mà nhóm đang sử dụng để xây dựng mô hình

* Lớp Conv2D( 32, kernel (3,3), padding=’same’, activation =’relu’, input\_shape=(128,128,3) )
* Lớp MaxPooling2D(pool\_size=(2,2))
* Lớp Conv2D( 64, kernel (3,3), padding=’same’, activation =’relu’)
* Lớp MaxPooling2D(pool\_size=(2,2))
* Lớp Conv2D( 128, kernel (3,3), padding=’same’, activation =’relu’)
* Lớp MaxPooling2D(pool\_size=(2,2))
* Lớp Conv2D( 256, kernel (3,3), padding=’same’, activation =’relu’ )
* Lớp MaxPooling2D(pool\_size=(2,2))
* Hàm Dropout rate = 0.2
* Lớp Flatten, fully connection
* Lớp Dense (num\_classes, activation=”softmax”)
* Optimizer:Adam, loss=”categorical\_crossentropy”, metrics =”accuracy”

1. **Bộ dữ liệu (Dataset)**

Trong bài này, nhóm em sử dụng bộ dữ liệu Local Faced in the Wild (lfw) cho bài toán. Lý do vì bộ dữ liệu này dễ dàng tải từ Internet do nó đã được công bố rộng rãi. Bộ dữ liệu có hơn 13233 ảnh khuôn mặt người, trong đó có tới 5749 người.

Tuy nhiên vì do có nhiều người, nên bộ dữ liệu có khá nhiều người chỉ có 1 hoặc một vài bức ảnh, trong khi lại có những người có hơn chục ảnh. Do đó dẫn đến mất cân bằng dữ liệu (imbalanced data). Điều này gây khó khăn cho việc phân chia dữ liệu train, test cũng như thiếu thụt dữ liệu để huấn luyện (training) trong model.

1. **Kết quả**

Vẫn đang trong quá trình nghiên cứu nên chưa có kết quả cụ thể.

Dưới đây là kết quả của train riêng của toàn bộ khuôn mặt

