

PROJECT AI

CHẨN ĐOÁN BỆNH LIÊN QUAN ĐẾN PHỔI BẰNG THUẬT TOÁN CNN

SVTH: TRẦN HOÀNG BẢO PHƯƠNG

MSSV: 19146244

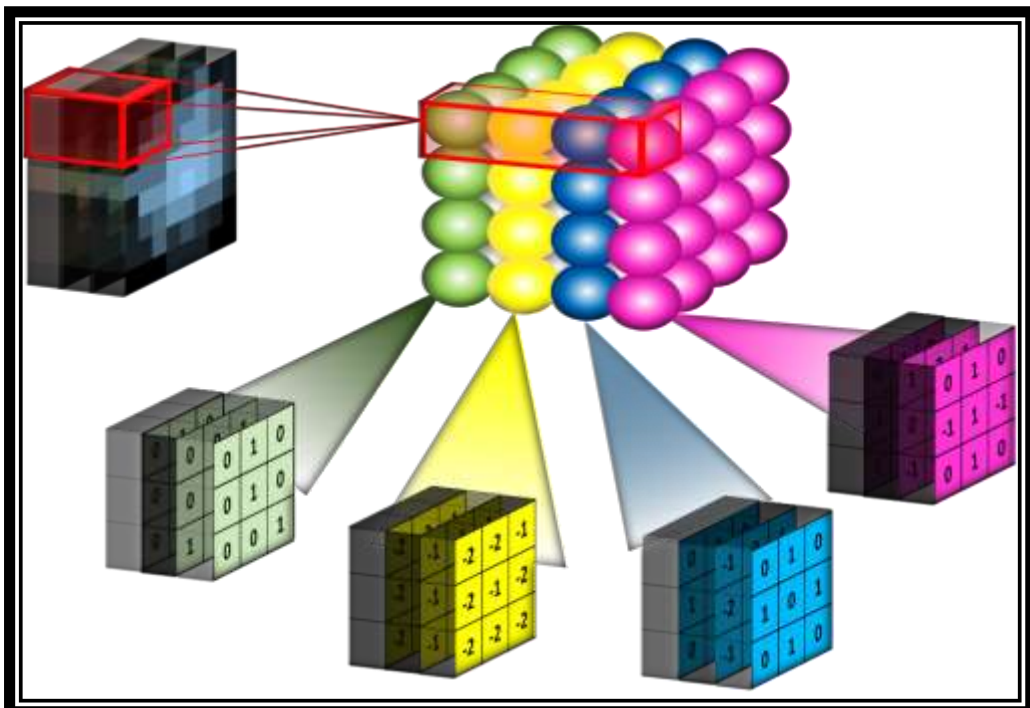
GVHD: PSG.TS NGUYỄN TRƯỜNG THỊNH

ĐẠI HỌC SƯ PHẠM KỸ THUẬT

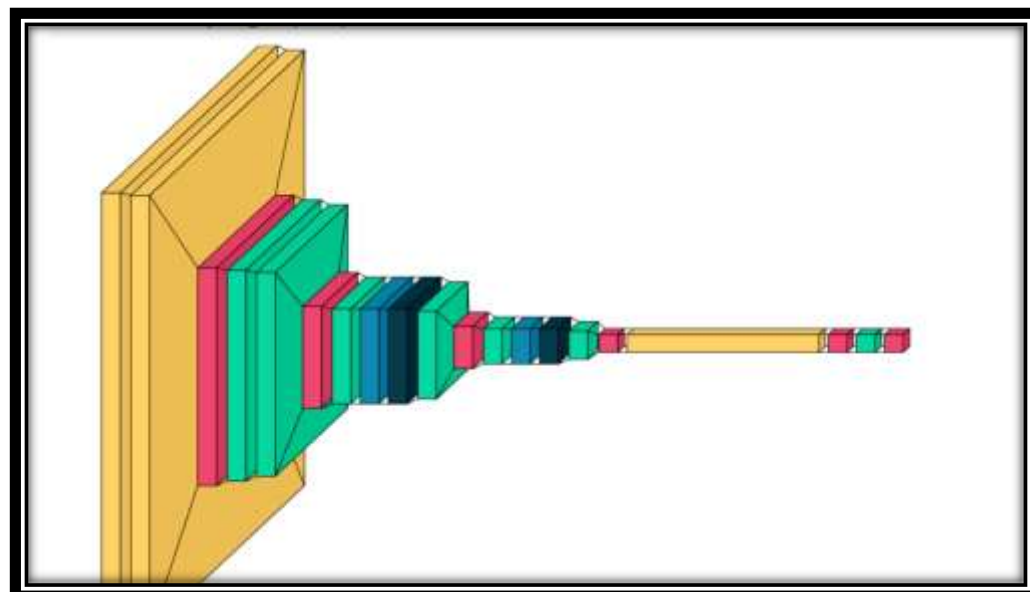
KIẾN TRÚC MẠNG CONVOLUTION NERUAL NETWORK

Xây dựng thuật toán AI bằng ngôn ngữ python dựa trên nền tảng miễn phí là google colab để train model chuẩn đoán bệnh viêm phổi thông qua ảnh X-rays. Việc dự báo không thể đảm bảo chính xác 100% nhưng có thể có một số trường hợp ảnh hưởng đến xấu về bệnh, cùng với việc tạo website như một công cụ trợ dự đoán kết quả nhằm kiểm tra và xem xét liệu kết quả có bệnh hay không

Trong phần đầu tiên của mạng CNN, một hình ảnh đầu vào có kích thước $128 \times 128 \times 3$ được thu thập và gửi đến khối đầu vào. Sau một khối trung gian gồm ba lần lặp, cuối cùng nó được xuất ra bởi khối đầu vào. Xuất dữ liệu tính năng viêm phổi bốn chiều và chuyển đổi dữ liệu thành dữ liệu ba chiều có thể được đọc bởi lớp CNN và gửi nó đến nửa sau của kiến trúc mô hình.. Dữ liệu đầu ra được kết nối đầy đủ thông qua một lớp dày đặc và số lượng tế bào thần kinh trong ba lớp đầu tiên giảm với tốc độ giảm 2 lần theo từng lớp. Đồng thời, với chức năng kích hoạt RELU, các đối tượng địa phương trước đó được tập hợp lại thành một đồ thị hoàn chỉnh thông qua ma trận trọng số. Trong lớp cuối cùng, phân loại dự đoán được thực hiện bằng cách chọn chức năng kích hoạt softmax.



HÌNH 1: ẢNH MINH HỌA CẤU TRÚC CNN



HÌNH 2: CÁC LỚP LAYER ĐƯỢC TẠO

```
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation("relu"))
model.add(Conv2D(256, (3, 3)))
model.add(Activation("relu"))

model.add(Conv2D(128, (3, 3)))
model.add(Activation("relu"))
model.add(SeparableConv2D(2, 2))
model.add(Conv2D(64, (3, 3)))
model.add(Activation("relu"))
model.add(SeparableConv2D(2, 2))

model.add(Conv2D(256, (2, 2)))
model.add(Activation("relu"))
model.add(MaxPooling2D(2, 2))

model.add(Flatten())
model.add(Dense(128))
model.add(Dropout(0.25))
model.add(Dense(3))
model.add(Activation("softmax"))
# Display our model
model.summary()

# Hiển thị hàm con_model.summary() dưới dạng biểu đồ
from tensorflow.keras.utils import plot_model
plot_model(model, to_file='model_final.png', show_shapes = True)
```

HÌNH 3: CODE VỀ KIẾN TRÚC MÔ HÌNH ĐƯỢC TẠO RA

Trong quá trình đào tạo. Hàm MaxPooling2D () dùng để lấy feature nổi bật nhất, như lấy thông số lớn nhất trong lúc so sánh (dùng max) và giúp giảm tham số (parameter) khi huấn luyện. Sau đó Flatten () để được fully connected giữa nút thần kinh ở các lớp ẩn. Tạo đầu ra tương ứng với 128 điểm cuối cùng giai báo kết quả đầu ra, chú ý cần 3 tham số chính phụ thuộc vào 3 đầu vào để xây dựng 3 đầu ra khác nhau. Trước khi huấn luyện cần compile các phân lớp CNN lại để hoàn thành một thể thống nhất.

THÔNG TIN VỀ TẬP DỮ LIỆU

RAW DATA INPUT

DATA PREPROCESSING

Datasets	Normal	Pneumonia	Lung Cancer	Total
TRAIN	1089	1215	1345	3649
TEST	413	460	450	1323
VALIDATION	8	8	50	66

Bảng phân chia dữ liệu gồm 3 lớp chính của dữ liệu

```
# preprocessing the training dataset
# tăng cường dữ liệu
print('train set: ')
data_gen = ImageDataGenerator(rescale=1.0 / 255.0,
                               rotation_range=10,
                               width_shift_range=0.2,
                               height_shift_range=0.2,
                               shear_range=0.2,
                               horizontal_flip=False,
                               zoom_range=0.3)

train_set = data_gen.flow_from_directory(directory=train_folder,
                                         target_size=(224,224),
                                         batch_size=32,
                                         shuffle=False,
                                         class_mode='categorical')

print('test set:')
test_set = data_gen.flow_from_directory(directory=test_folder,
                                       target_size=(224,224),
                                       batch_size=32,
                                       shuffle=False,
                                       class_mode='categorical')

print('Validation set:')
val_set = data_gen.flow_from_directory(directory=val_folder,
                                       target_size=(224,224),
                                       batch_size=32,
                                       shuffle=False,
                                       class_mode='categorical')
```

HÌNH 4: ẢNH TẬP DATASETS CỦA BÀI TOÁN

HÌNH 5: ẢNH VỀ TIỀN XỬ LÝ DỮ LIỆU

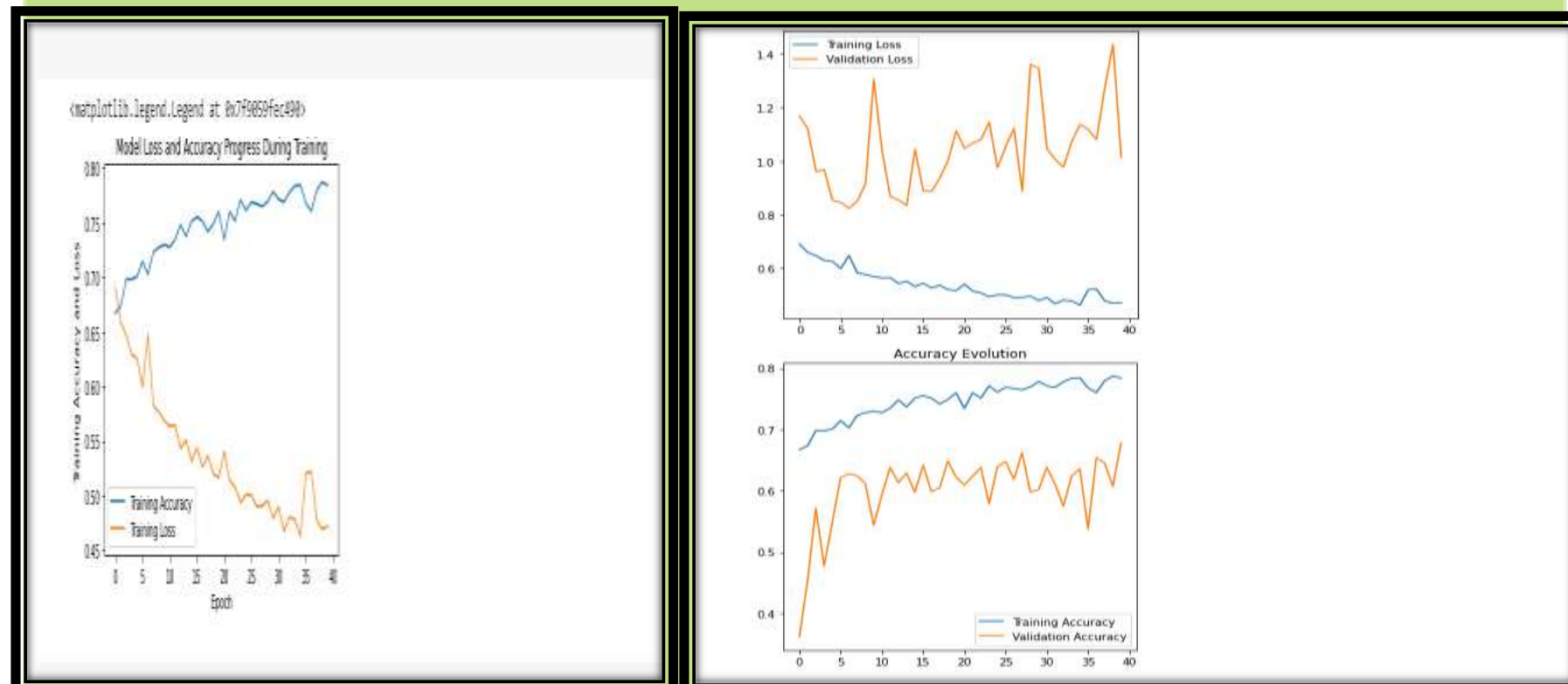
Tập dữ liệu gồm 3 lớp. Tôi sẽ sàng lọc lại dữ liệu mình đã có và tập cho mình 1 datasets riêng biệt. Gồm 3 lớp chính là người bình thường, người bị viêm phổi và người bị ung thư phổi. Đây là mục đích chính của đề tài AI lần này. Với mỗi ảnh có phần đuôi là *.jpeg, tập dữ liệu ảnh chụp X-rays lồng ngực gồm 3 tập “train”, “test”, “validation”.

KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH DỰ TRÊN DỰ ĐOÁN

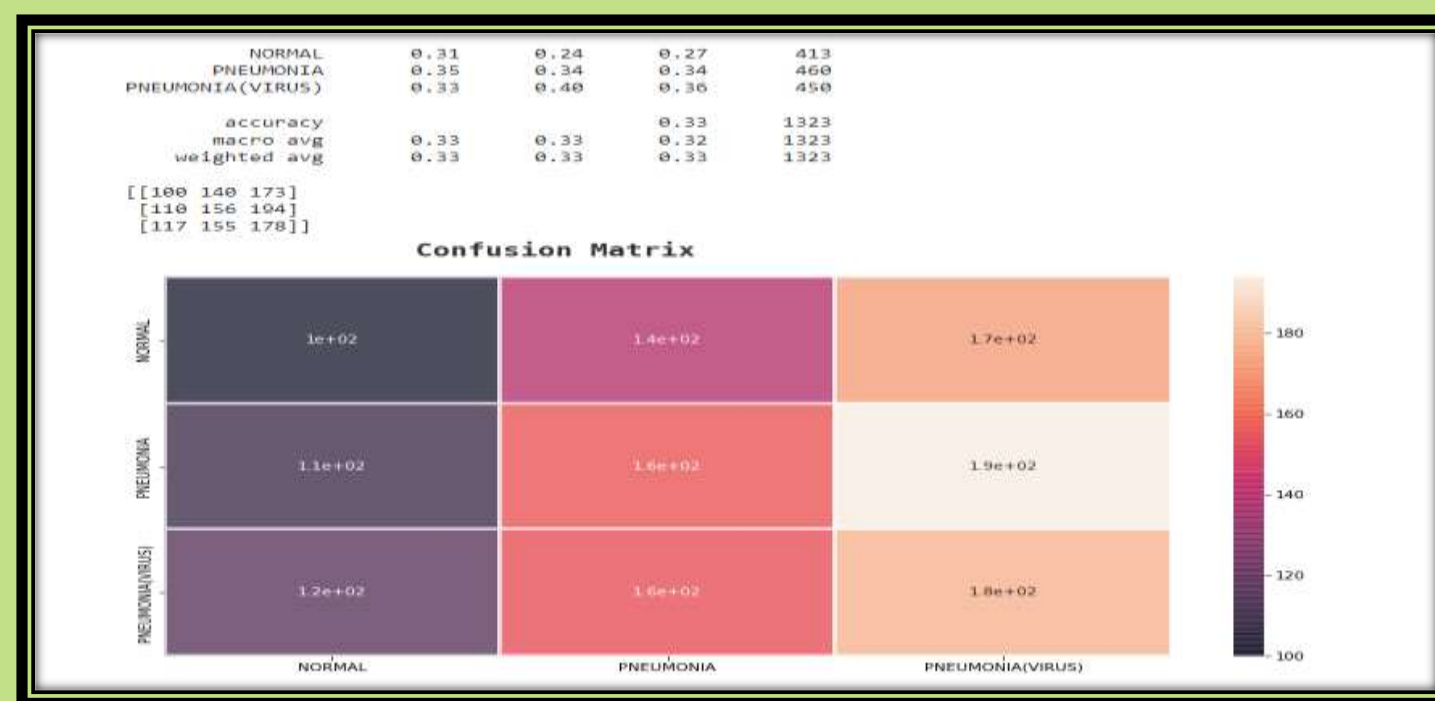
A. KẾT QUẢ ĐÀO TẠO

Về đồ thị sự chính xác (accuracy) : Dù có sự giao động, tuy nhiên đường biểu diễn sự chính xác của mô hình huấn luyện tăng dần từ 40% đến hơn 80% đây là biểu diễn khá đúng cho mô hình thực tế. Sự chính xác trên tập huấn luyện với tập đánh giá không có chênh lệch nhiều chứng tỏ đã giảm bớt tình trạng huấn luyện quá (overfitting).

Về đồ thị hàm mất mát (loss): theo mong muốn thì đồ thị này cần giảm giảm xuống tối đa nhất, tuy nhiên trong quá trình huấn luyện do đồ thị có giảm nhưng không nhiều như mong đợi và có sự giao động cụ thể. Hàm mất mát cả trên tập huấn luyện và tập đánh giá tương đồng một vài khía cạnh.

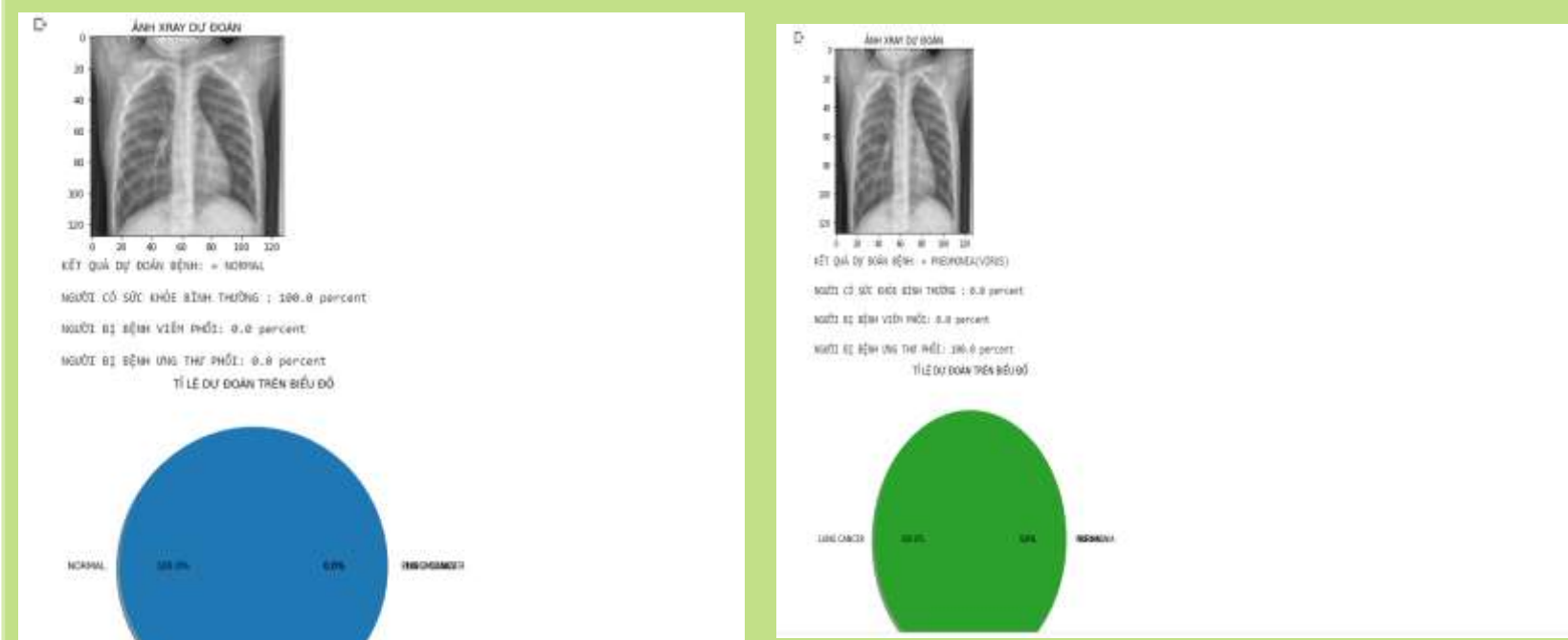


HÌNH 6: SƠ ĐỒ ĐÁNH GIÁ HÀM MẤT MÁT VÀ ĐỘ CHÍNH XÁC TẬP TRAIN VÀ VALIDATION



HÌNH 7: ĐÁNH GIÁ TRÊN MA TRẬN LỖI

B. KẾT QUẢ DỰ ĐOÁN



HÌNH 9: ĐÁNH GIÁ CỦA BỆNH UNG THƯ PHỔI VÀ BỆNH NHÂN BÌNH THƯỜNG

