

**Cuộc thi khoa học, kỹ thuật cấp trường dành cho học sinh trung học  
năm học 2022-2023**

Tên dự án dự thi

**SỬ DỤNG TRÍ TUỆ NHÂN TẠO TRONG VIỆC NHẬN  
BIẾT KHỐI U TRONG NÃO VÀ PHỔI**

Lĩnh vực dự thi

**phần mềm hệ thống**

Tác giả: Phạm Đình Trung Hiếu

# Mục lục

<b>I. Lời cảm ơn .....</b>	<b>3</b>
<b>II. Tóm tắt nội dung dự án .....</b>	<b>4</b>
<b>III. Quá trình nghiên cứu .....</b>	<b>4</b>
<b>A. TensorFlow .....</b>	<b>5</b>
<b>B. OpenCV .....</b>	<b>10</b>
<b>IV. Kết quả .....</b>	<b>13</b>
<b>V. Tài liệu tham khảo .....</b>	<b>13</b>

## **I. Lời cảm ơn**

Dự án: Sử dụng trí tuệ nhân tạo để xác định khối u trong não và phổi con người là một dự án phục vụ đời sống của con người, giúp tự động phân loại nhanh chóng những ca bệnh có khối u trong não và phổi mà không cần đến sự can thiệp của con người.

Nội dung dự án gồm:

1. Tóm tắt nội dung dự án.
2. Quá trình nghiên cứu.
3. Kết quả.

## II. Tóm tắt nội dung dự án

Sử dụng trí tuệ nhân tạo để xác định khối u trong não và phổi con người.

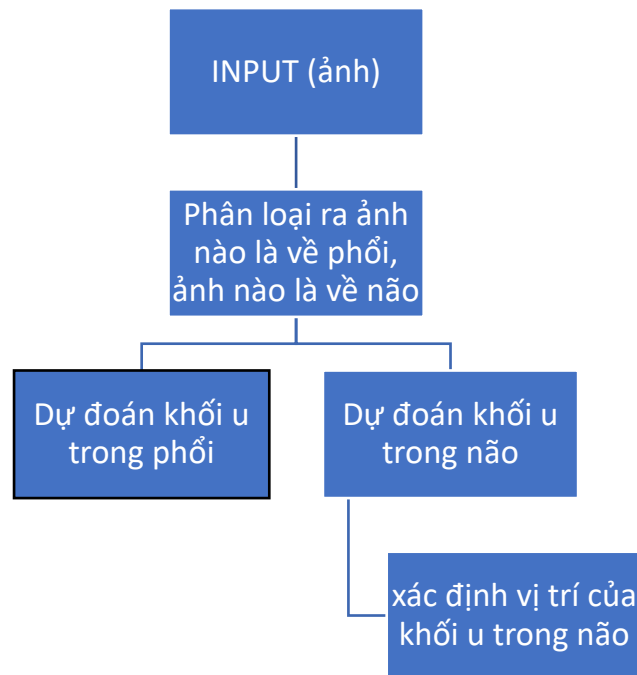
## III. Quá trình nghiên cứu

Sử dụng thư viện TensorFlow của Google để train cho model.

Dùng openCV để xác định vị trí của khối u.

Dữ liệu cho việc nghiên cứu được lấy từ các bệnh viện và trên Kaggle.

### Mô hình



## A. TensorFlow

### Tạo model

```
vgg = VGG16(input_shape=size + [3],
            weights='imagenet',
            include_top=False)
for layer in vgg.layers:
    layer.trainable = False

folders = glob('Datasets/train/*')
temp = Flatten()(vgg.output)
prediction = Dense(len(folders),
                  activation='softmax')(temp)

model = Model(inputs=vgg.input, outputs=prediction)
model.summary()
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
```

### Bộ dữ liệu

Là các hình ảnh xray về não, phổi của người bình thường và có xuất hiện khối u.  
Sơ đồ của bộ dữ liệu:

#### **.\Datasets**

```
\test <bộ dữ liệu để kiểm tra đối chiếu>
    \tumor
    \normal
\train <bộ dữ liệu để training>
    \tumor
    \normal
```



## Tạo các Datasets

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1./255)

training_set = train_datagen.flow_from_directory('Datasets/train',
                                                  target_size=(224, 224),
                                                  batch_size=32,
                                                  class_mode='categorical')

test_set = test_datagen.flow_from_directory('Datasets/test',
                                             target_size=(224, 224),
                                             batch_size=32,
                                             class_mode='categorical')
```

## Tiến hành fit Model

```
model.fit(
    training_set,
    validation_data=test_set,
    epochs=5,
    steps_per_epoch=len(training_set),
    validation_steps=len(test_set)
)

model.save('model-23-9-2022-1.h5')
```

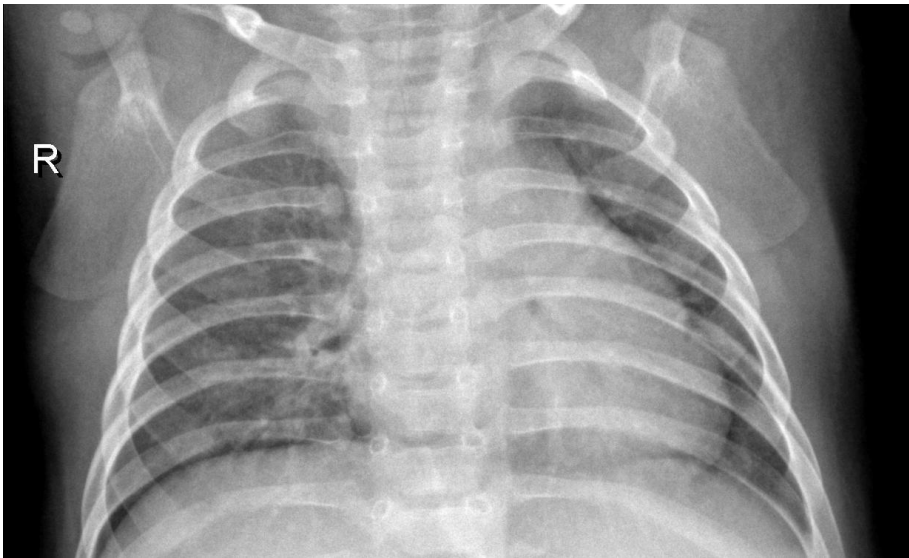
## Tiến hành thử nghiệm trên bộ dữ liệu lớn

Mọi kết quả của máy đưa ra được so sánh với kết quả từ bác sĩ của bệnh viện nếu khớp thì là đúng còn, và ngược lại.

Trường hợp	Số lần thử	Độ chính xác
Não bình thường	97	88%
Não có khối u	155	85%
Phổi bình thường	234	54%

Phổi có khối u	390	93%
----------------	-----	-----

Ta có thể nhận thấy việc nhận biết có độ chính xác không được cao. Qua quá trình kiểm tra thì ta nhận thấy ở hầu hết các bức hình phổi đều bị che đi một phần bởi xương sống (hình phía dưới)



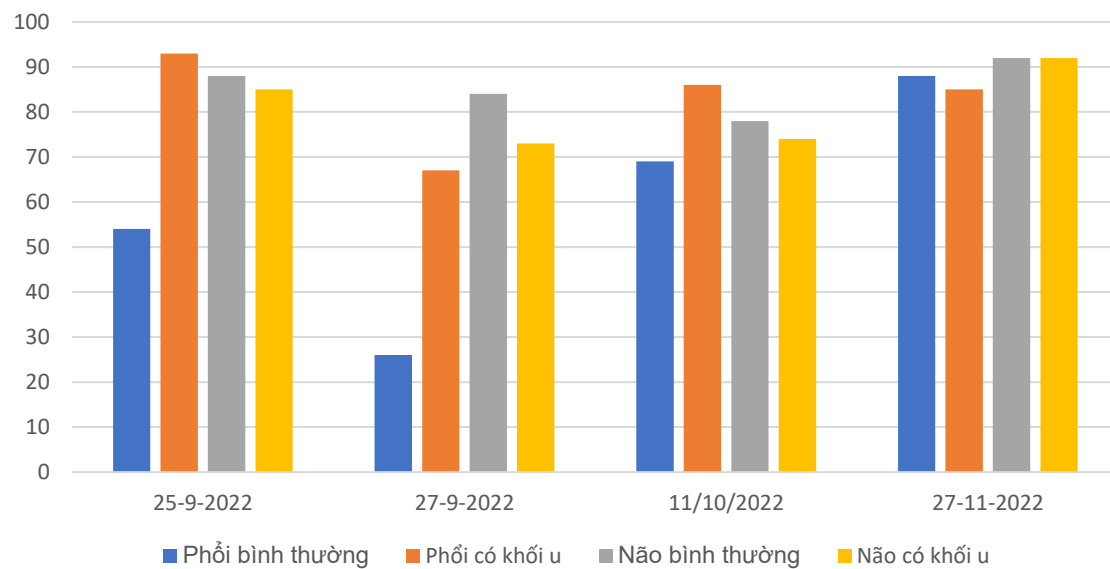
Sau khi kiểm tra lại toàn bộ bộ dữ liệu và bổ xung hình ảnh cho cả phổi và não thì ta thu được kết quả có độ chính xác cao hơn.

Trường hợp	Số lần thử	Độ chính xác
Não bình thường	98	92.5%
Não có khối u	155	92.9%
Phổi bình thường	234	88.0%
Phổi có khối u	390	85.7%

Nhưng kết quả vẫn chưa được cao như mong muốn vì có nhiều trường hợp phổi của bệnh nhân có dấu hiệu bị hình mờ dạng đường hoặc đám và các nốt mờ thường tập trung chủ yếu ở vùng dưới phổi nên dễ làm sai lệch kết quả.



Độ chính xác của các phiên bản model



## B. OpenCV

### Xác định vị trí của khối u trong não

Sau khi thử nghiệm thì **TensorFlow** xác định vị trí của khối u trong não không được chính xác lắm, nên ta sẽ thử nghiệm trên **OpenCV** thì cho ra kết quả chính xác hơn rất nhiều. (hình bên phải là của TensorFlow, bên trái là của openCV)



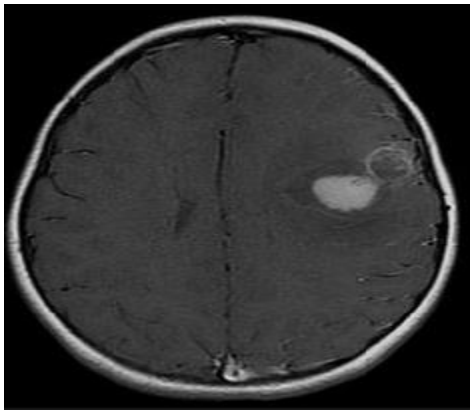
### Tạo Bộ dữ liệu mới

Vì ta chuyển qua dùng openCV nên ta phải tạo ra bộ dữ liệu mới, có dạng như sau:

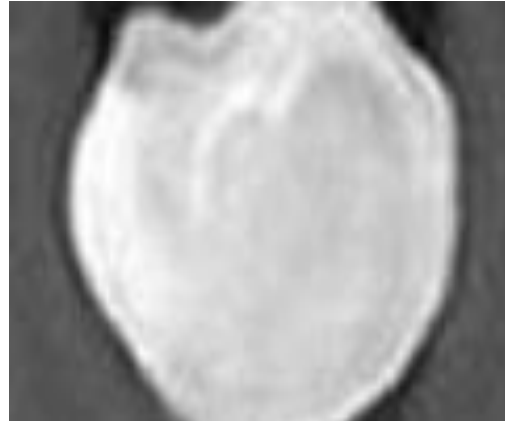
#### ./Datasets

./negative <ảnh chỉ có khối u mà không có các đối tượng khác>

./positive <ảnh có cả khối u và các bộ phận khác của não>



Positive

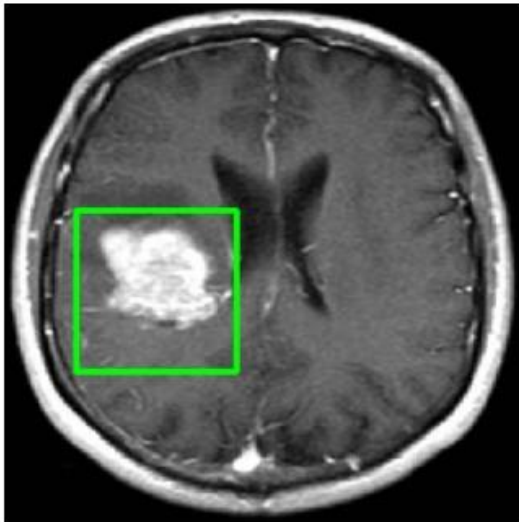


Negative

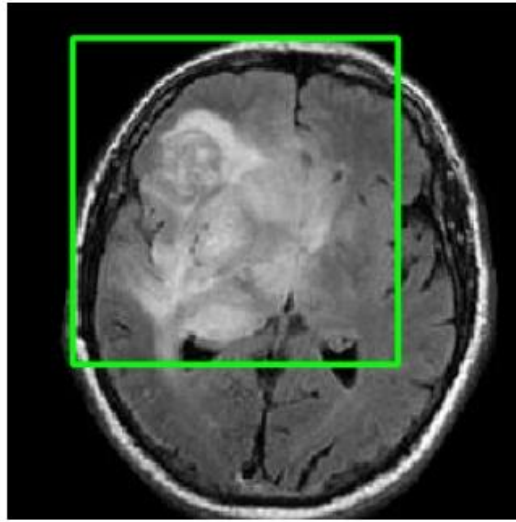
## Tạo CascadeClassifier và xác định khối u

```
def DetectTumor(img):  
    img = load_img(img, target_size=(255, 255))  
    oimg = img  
    img = np.array(img)  
    detector = cv2.CascadeClassifier('./brain.xml')  
    rect = detector.detectMultiScale(img, 1.1, 9)  
    for (x, y, w, h) in rect:  
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)  
    return img
```

Y11.jpg




Y2.jpg




## C. Tạo giao diện để sử dụng


Ta sử dụng thư viện **Streamlit** của **Python** để tạo giao diện một cách đơn giản và trực quan.


Tải ảnh lên


 Drag and drop files here  
Limit 200MB per file • PNG, JPG, JPEG


Browse files


 Y3.jpg 45.6KB





 Y2.jpg 12.1KB



 Y1.jpg 5.0KB



Showing page 1 of 5

Kết Quả

	Tên	loại bệnh	Độ chính xác (%)
4	3 no.jpg	Não bình thường	100.0000
5	8 no.jpg	Não bình thường	100.0000
6	9 no.jpg	Não bình thường	100.0000
7	10 no.jpg	Não bình thường	100.0000
8	15 no.jpg	Não có khối u	100.0000
9	17 no.jpg	Não bình thường	100.0000
10	18 no.jpg	Não bình thường	100.0000
11	Y1.jpg	Não có khối u	100.0000
12	Y2.jpg	Não có khối u	99.9676
13	Y3.jpg	Não có khối u	100.0000

#### **IV. Kết luận**

Máy đã có thể nhận và phân loại khá chính xác tình trạng có khối u hoặc không ở phổi và não.

##### **Hướng phát triển trong tương lai:**

Giúp máy có thể phân loại được cả khối u ác tính và lành tính.

Gợi ý hướng chữa trị cho các bác sĩ.

#### **V. Tài liệu tham khảo**

<https://www.tensorflow.org/tutorials>

<https://www.youtube.com/c/TensorFlow>

Lý thuyết về Mạng chập CNN của Hệ Phạm Văn

<https://www.geeksforgeeks.org/detect-an-object-with-opencv-python/>

<https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection>

<https://www.kaggle.com/datasets/deepball69xt/chest-xray-tumor-lung>

<https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/>