Báo cáo đề tài KHKT

Chẩn đoán ung thư da sử dụng Học sâu (dEEp learning)



# Thành viên nhóm

* Nguyễn Vũ Khánh Uy (10/6): Code giao diện và xử lí sản phẩm.
* Bùi Minh Huy (10/8): Góp ý sản phẩm.

# Vấn đề hiện nay hiện nay

* Theo thống kê năm 2015, thế giới có 8,8 triệu người chết do ung thư, trong đó 70% số ca tử vong xảy ra ở các nước thu nhập thấp và trung bình. Mỗi năm, Việt Nam có hơn 160000 ca mắc mới và khoảng 94000 người tử vong vì ung thư.
* Trong đó, ung thư da là loại ung thư thường gặp nhất, nhiều hơn tất cả các loại ung thư khác cộng lại. Ở một nước thu nhập cao như Hoa Kì, ước tính hằng năm có hơn một triệu người nhiễm loại bệnh này. Theo thống kê gần đây nhất thì đã có 9320 ca tử vong cho đến giữa năm 2018 ở Hoa Kì.
* Hằng năm, người dân Hoa Kì dành 2.7 tỷ đô la mỹ cho điều trị ung thư da.
* Ung thư da là vấn nạn trên toàn thế giới.

# Hình thành ý tưởng

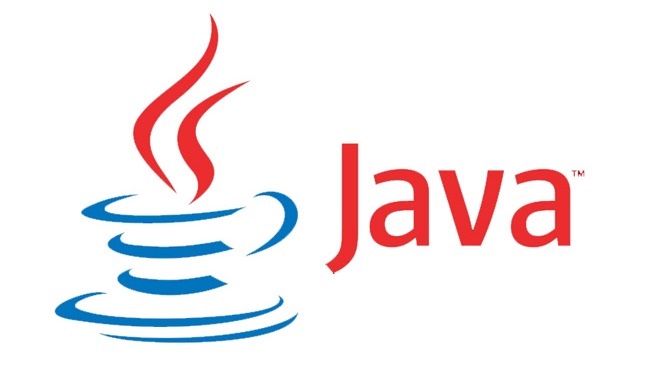
* Ung thư da hoàn toàn có thể chữa khỏi với tỉ lệ sống sót lên đến 98% nếu phát hiện sớm và có thể kéo dài sự sống từ 5 đến trên 10 năm tùy thể trạng từng người.
* Tuy vậy, dấu hiệu nhận biết ung thư da khá mơ hồ và không rõ rang với một số người (dấu hiệu của bệnh ung thư da giai đoạn đầu thường chỉ là những hạt nốt ruồi).
* Một lý do khác là do công việc bận rộn, chỉ được nghỉ vào ngày chủ nhật (các cơ sở y tế thường không khám vào ngày này) nên nhiều người bỏ qua đến khi bệnh trở nặng thì đã quá muộn.
* Hiện nay, hầu như mọi người đều có một (có người còn có nhiều hơn) chiếc điện thoại thông minh. Đặc biệt, ở Việt Nam đa số đều dùng điện thoại android và hệ sinh thái smartphone còn có rất nhiều tiềm năng phát triển.
* Những chiếc smartphone đang ngày càng được phát triển về cấu hình, hiện nay iphone X của Apple đã có thể xử lí trí tuệ nhân tạo để nhận diện khuông mặt (điều mà cách đây 3 – 4 chưa ai nghĩ tới kể cả với những chiếc máy tính). Theo dự đoán tương lai hầu hết các điện thoại thông đều có cấu hình đủ để xử lí trí tuệ nhân tạo(AI) đã không còn xa và có thể là năm 2020(4 năm sau).
* Tạo ra một ứng dụng chẩn đoán ung thư da trên điện thoại di động là một giải pháp tối ưu và là tương lai tất yếu của ngành y học, khi bác sĩ không còn phải chẩn đoán mọi việc mà chỉ giám sát máy móc làm việc.

# Lịch trình phát triển

* Nghiên cứu deep learning, bệnh ung thư da. Nguồn: tutorial trên youtube, các blog, các tài liệu deep learning và y học (1 tháng).
* Chuẩn bịvà chọn lọc dự liệu cần thiết để training model (4 ngày).
* Chọn framework sử dụng giữa tensorflow và pytorch, giữa keras và tflearn và đưa ra lựa chọn sử dụng tensorflow & keras để tạo ra model. Nguồn: thông qua đánh giá các điểm mạnh điểm yếu của các framework qua các blog và tài liệu (2 ngày).
* Xây dựng model đầu tiên và kiểm thử (6 ngày).
* Tối ưu hóa model và kiểm thử (5 ngày).
* Chuyển đổi model sang dạng phù hợp(.tflite) để chạy trên smartphone (2 ngày).
* Nhúng vào nền tảng android – Xây dựng giao diện (1 ngày).
* Nhúng vào nền tảng android – Xử lí chẩn đoán (2 ngày).
* Viết báo cáo (1 đêm).
* Tổng cộng: 1 tháng 22 ngày 1 đêm

Giới thiệu công nghệ & công cụ sử dụng

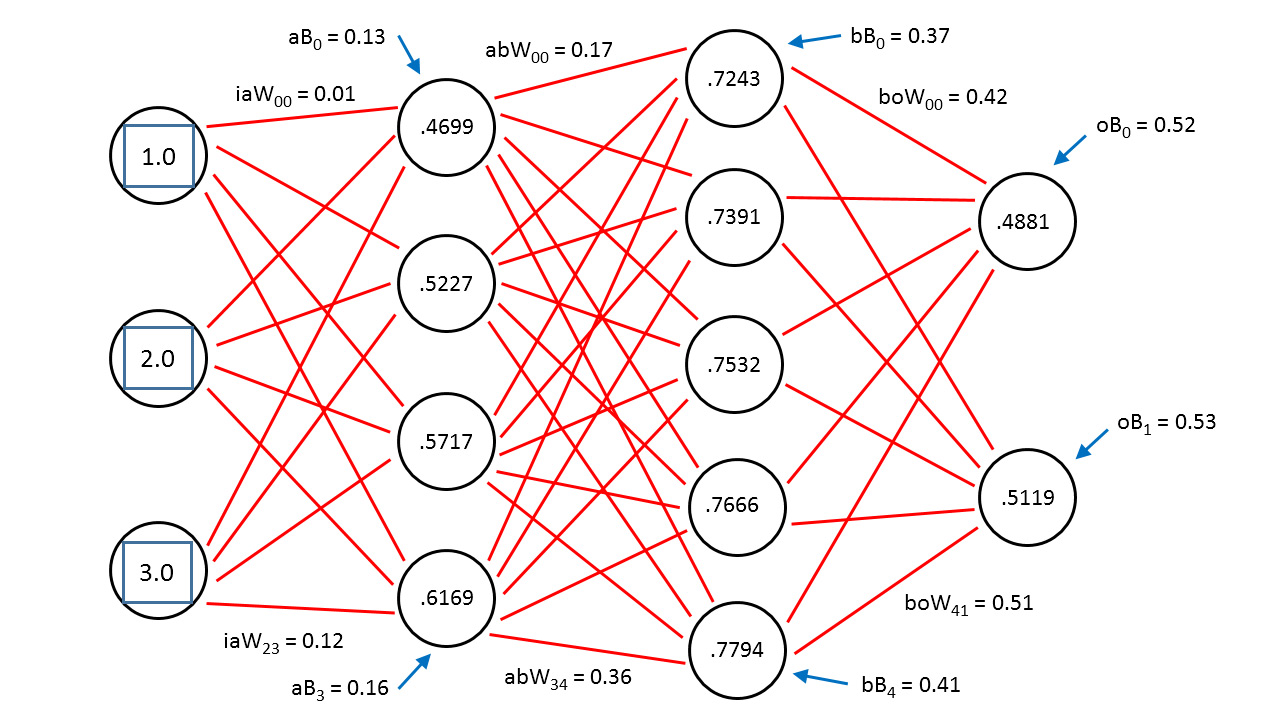
* Chuẩn bị dữ liệu: <https://isic-archive.com/> Cung cấp rất nhiều ảnh về bệnh ung thư da được sắp xếp theo mục (ác tính và lành tính).
* Ngôn ngữ lập trình sử dụng:
  + Java: là một ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp (class). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy. Trước đây, Java chạy chậm hơn những ngôn ngữ dịch thẳng ra mã máy như C và C++, nhưng sau này nhờ công nghệ "biên dịch tại chỗ" - Just in time compilation, khoảng cách này đã được thu hẹp, và trong một số trường hợp đặc biệt Java có thể chạy nhanh hơn. Java chạy nhanh hơn những ngôn ngữ thông dịch như Python, Perl, PHP gấp nhiều lần. Java chạy tương đương so với C#, một ngôn ngữ khá tương đồng về mặt cú pháp và quá trình dịch/chạy. Cú pháp Java được vay mượn nhiều từ C & C++ nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn. Trong Java, hiện tượng rò rỉ bộ nhớ hầu như không xảy ra do bộ nhớ được quản lý bởi Java Virtual Machine (JVM) bằng cách tự động "dọn dẹp rác". Người lập trình không phải quan tâm đến việc cấp phát và xóa bộ nhớ như C, C++. Tuy nhiên khi sử dụng những tài nguyên mạng, file IO, database (nằm ngoài kiểm soát của JVM) mà người lập trình không đóng (close) các streams thì rò rỉ dữ liệu vẫn có thể xảy ra.



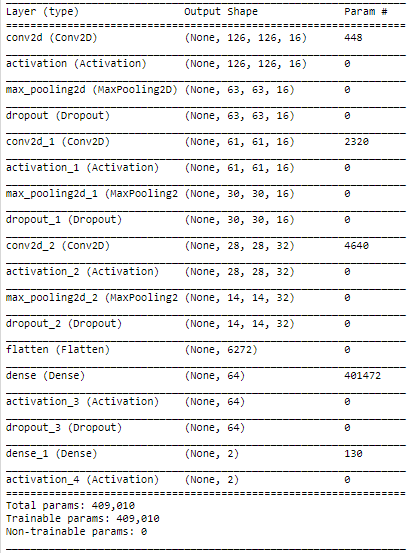
* + Python: là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, Van Rossum đã từ chức Leader trong cộng đồng ngôn ngữ Python sau 30 năm lãnh đạo. Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý. Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, nó đã "bành trướng" sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.



* Deep learning:
  + Tổng quan: Mạng nơ-ron nhân tạo hay thường gọi ngắn gọn là mạng nơ-ron là một mô hình toán học hay mô hình tính toán được xây dựng dựa trên các mạng nơ-ron sinh học. Nó gồm có một nhóm các nơ-ron nhân tạo (nút) nối với nhau, và xử lý thông tin bằng cách truyền theo các kết nối và tính giá trị mới tại các nút (cách tiếp cận connectionism đối với tính toán). Trong nhiều trường hợp, mạng nơ-ron nhân tạo là một hệ thống thích ứng (adaptive system) tự thay đổi cấu trúc của mình dựa trên các thông tin bên ngoài hay bên trong chảy qua mạng trong quá trình học. Trong thực tế sử dụng, nhiều mạng nơ-ron là các công cụ mô hình hóa dữ liệu thống kê phi tuyến. Chúng có thể được dùng để mô hình hóa các mối quan hệ phức tạp giữa dữ liệu vào và kết quả hoặc để tìm kiếm các dạng/mẫu trong dữ liệu.
  + Sử dụng: Có lẽ lợi thế lớn nhất của mạng nơ-ron nhân tạo (Artificial Neural Network- ANN) là khả năng được sử dụng như một cơ chế xấp xỉ hàm tùy ý mà 'học' được từ các dữ liệu quan sát. Tuy nhiên, sử dụng chúng không đơn giản như vậy, và một sự hiểu biết tương đối tốt về các lý thuyết cơ bản là điều cần thiết. Chọn mô hình: điều này sẽ phụ thuộc vào cách trình bày dữ liệu và các ứng dụng. Mô hình quá phức tạp có xu hướng dẫn đến những thách thức trong việc học. Thuật toán học: có rất nhiều sự thỏa thận giữa các thuật toán học. Hầu hết các thuật toán sẽ làm việc tốt với các siêu tham số (hyperparameter) đúng để huấn luyện trên một tập hợp dữ liệu cố định cụ thể. Tuy nhiên, việc lựa chọn và điều chỉnh một thuật toán để huấn luyện trên dữ liệu không nhìn thấy yêu cầu một số lượng đáng kể các thử nghiệm. Mạnh mẽ: Nếu các mô hình, hàm chi phí và thuật toán học được lựa chọn một cách thích hợp, thì ANN sẽ cho kết quả có thể vô cùng mạnh mẽ. Với việc thực hiện chính xác, ANN có thể được sử dụng một cách tự nhiên học trực tuyến và các ứng dụng tập dữ liệu lớn. Việc thực thi đơn giản của chúng và sự tồn tại của chủ yếu là địa phương phụ thuộc được thể hiện trong cấu trúc cho phép triển khai nhanh chóng, song song trong phần cứng.



Cấu trúc model hiện tại đang sử dụng (Hình dưới)



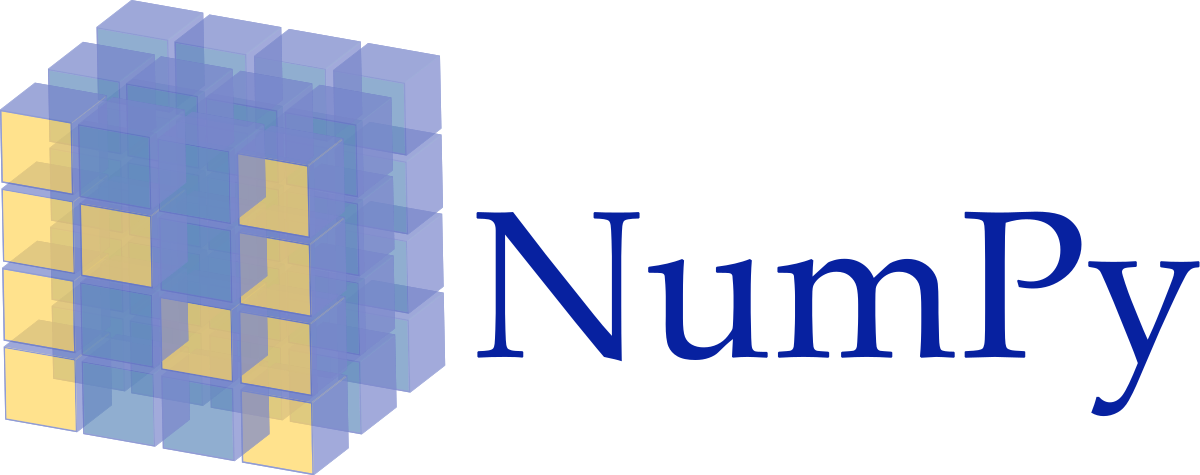
* Xây dựng model deep learning:
  + Tensorflow: là một thư viện phần mềm mã nguồn mở dành cho máy học trong nhiều loại hình tác vụ nhận thức và hiểu ngôn ngữ. Nó hiện đang được sử dụng cho cả nghiên cứu lẫn sản xuất bởi 50 đội khác nhau trong hàng tá sản phẩm thương mại của Google, như nhận dạng giọng nói, Gmail, Google Photos, và tìm kiếm, nhiều trong số đó đã từng sử dụng chương trình tiền nhiệm DistBelief của nó. TensorFlow nguyên thủy được phát triển bởi đội Google Brain cho mục đích nghiên cứu và sản xuất của Google và sau đó được phát hành theo giấy phép mã nguồn mở Apache 2.0 vào ngày 9/11/2015.



* + Keras: là một library được phát triển vào năm 2015 bởi François Chollet, là một kỹ sư nghiên cứu deep learning tại google. Nó là một open source cho neural network được viết bởi ngôn ngữ python. keras là một API bậc cao có thể sử dụng chung với các thư viện deep learning nổi tiếng như tensorflow (được phát triển bởi google), CNTK(được phát triển bởi microsoft),theano(người phát triển chính Yoshua Bengio). Keras có một số ưu điểm như: Dễ sử dụng, xây dựng model nhanh, có thể chạy trên cả cpu và gpu, hỗ trợ xây dựng CNN , RNN và có thể kết hợp cả 2.



* + NumPy: là một thư viện cho ngôn ngữ lập trình Python, thêm hỗ trợ cho các mảng lớn, đa chiều và ma trận, cùng với một bộ sưu tập lớn các hàm toán học cấp cao để hoạt động trên các mảng này. Tổ tiên của NumPy, Numeric, ban đầu được tạo ra bởi Jim Hugunin với sự đóng góp từ một số nhà phát triển khác.



* + Matplotlib: là một thư viện vẽ sơ đồ cho ngôn ngữ lập trình Python và phần mở rộng toán học số của nó là NumPy. Nó cung cấp một API hướng đối tượng để nhúng các ô vào các ứng dụng bằng cách sử dụng các bộ công cụ GUI có mục đích chung như Tkinter, wxPython, Qt hoặc GTK +



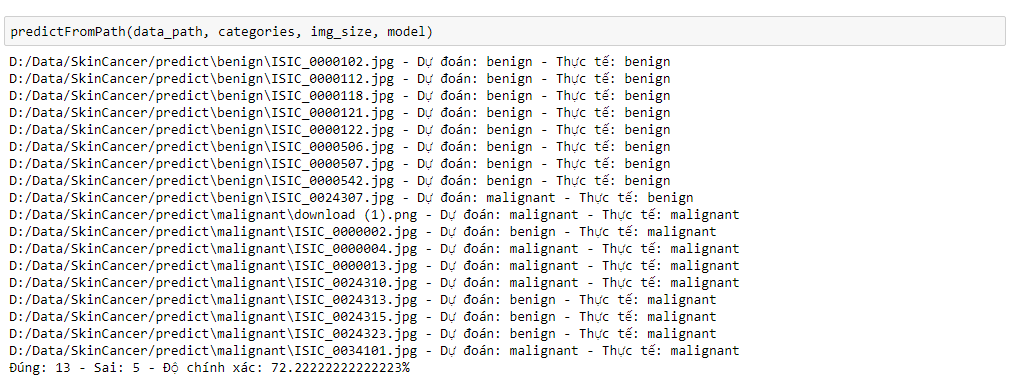
* Chuyển đổi sang dạng phù hợp (.tflite): Do một số lỗi về môi trường lập trình nên máy chạy hệ điều hành windows khi chuyển đổi model sẽ gặp một số lỗi. Vì vậy tôi sử dụng môi trường linux trên colab (<https://colab.research.google.com/>) để covert model.
* Công cụ lập trình:
  + Jupyter Notebook: Là ứng dụng tính toán khoa học mã nguồn mở, interactive, hỗ trợ hơn 40 ngôn ngữ lập trình, trong đó có python (jupyter = julia + python + R), chạy trên nền web cho phép chạy interactive python (tương tự ipython). Hơn thế nữa, nó còn hỗ trợ vẽ các đồ thị, biểu đồ, hỗ trợ viết 1 "notebook" bằng cách sử dụng Markdown.
  + Sublime Text 3: là một trình soạn thảo mã nguồn đa nền tảng độc quyền với giao diện lập trình ứng dụng Python. Nó nguyên bản hỗ trợ nhiều ngôn ngữ lập trình và ngôn ngữ đánh dấu, và các chức năng có thể được thêm vào bởi người dùng với các plugin, thường được xây dựng và duy trì theo các giấy phép phần mềm tự do.
  + Android Studio: là môi trường phát triển tích hợp (IDE) chính thức dành cho phát triển nền tảng Android. Nó được ra mắt vào ngày 16 tháng 5 năm 2013 tại hội nghị Google I/O. Android Studio được phát hành miễn phí theo giấy phép Apache Licence 2.0. Android Studio ở giai đoạn truy cập xem trước sớm bắt đầu từ phiên bản 0.1 vào tháng 5.2013, sau đó bước vào giai đoạn beta từ phiên bản 0.8 được phát hành vào tháng 6 năm 2014. Phiên bản ổn định đầu tiên được ra mắt vào tháng 12 năm 2014, bắt đầu từ phiên bản 1.0. Dựa trên phần mềm IntelliJ IDEA của JetBrains, Android Studio được thiết kế đặc biệt để phát triển ứng dụng Android. Nó hỗ trợ các hệ điều hành Windows, Mac OS X và Linux, và là IDE chính thức của Google để phát triển ứng dụng Android gốc để thay thế cho Android Development Tools (ADT) dựa trên Eclipse.

# Khó khăn khi phát triển

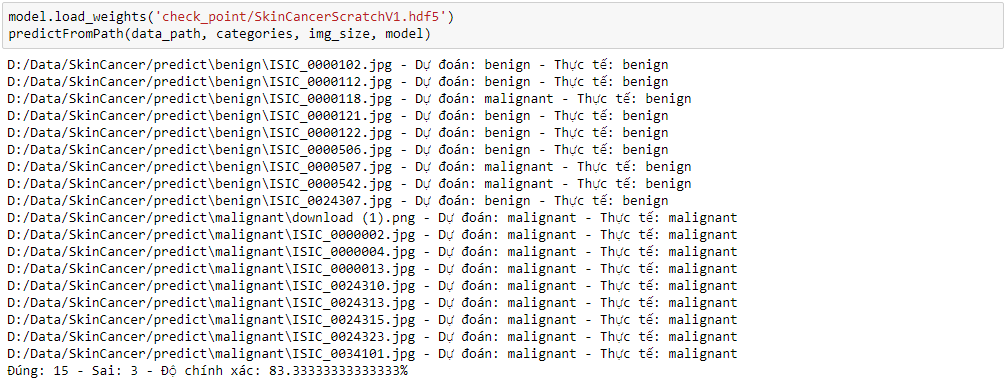
* Tài liệu nghiên cứu về deep learning viết bằng tiếng Việt gần như không có, các blog thì khá ít nên khi code gặp bug thì phải đọc tài liệu tiếng Anh. Tài liệu nghiên cứu về ung thư da thì có khá nhiều phiên bản tiếng Việt nhưng viết không nhất quán, đọc khá khó hiểu.
* Nguồn dữ liệu của bệnh ung thư da khá hạn chế, có những bộ dữ liệu nhiều nhưng không ở dạng cần thiết. Ngay cả data tôi sử dụng ở <https://isic-archive.com> tuy khá tốt nhưng vẫn có những bức ảnh của loại này nhưng lại được xếp ở loại kia, có nhiều bức ảnh trùng lặp, sự chênh lệnh giữa số bức ảnh ác tính và lành tính quá lớn(2000 và 19000) và file ảnh khá nặng dẫn đến phải chọn lọc bỏ bớt khá công phu.
* Các framework đều có những điểm mạnh yếu riêng nên đối với một người chưa có nhiều kinh nghiệm khá khó lựa chọn.
* Xây dựng, kiểm thử và chuyển đổi model rất hay gặp phải những lỗi môi trường vì tensorflow chỉ hỗ trợ thực sự tốt trên linux còn với windows và macos thì khá hay gặp bug.
* Deploy model trên smartphone vẫn hay gặp một số lỗi nhỏ.

# Kết quả thu được

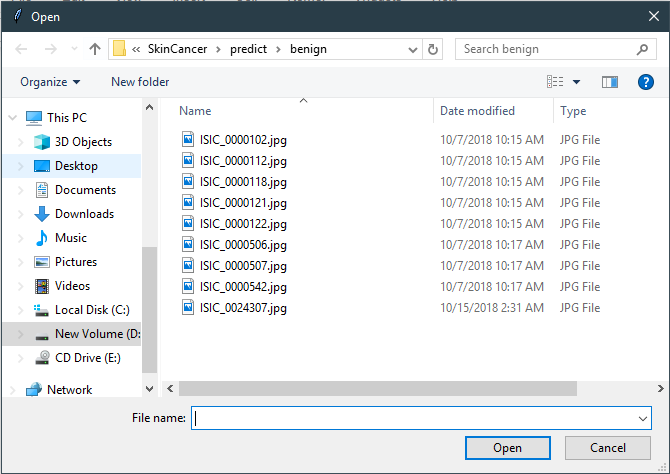
* Sau 20 epoch: tạo ra được một model chuẩn đoán với độ chính xác 66,67% (Vì một số vấn đề nên hình ảnh kết quả lúc đầu bị mất).
* Sau 40 epoch: tạo ra được một model chuẩn đoán với độ chính xác 72,22%



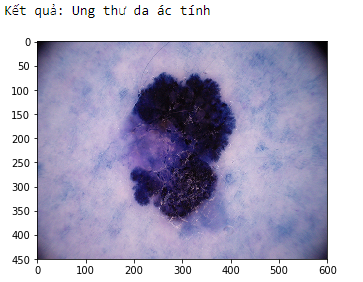
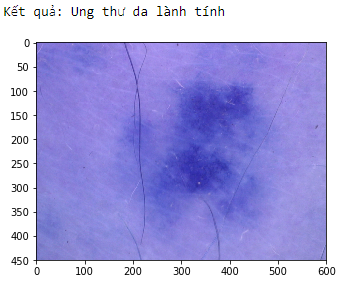
* Sau 60 epoch (model hiện tại): tạo ra được model chuẩn đoán với độ chính xác 83,33%



Do chưa làm kịp phần android nên tôi có làm một mẫu thiết kế thử nghiệm đơn giản, khi chạy code sẽ hiện lên một hộp thoại cho phép chọn ảnh:



Và sau đây là kết quả sau 2 lần thử nghiệm (Do thử nghiệm trên jupyter notebook nên ảnh có màu xanh tím nhưng vẫn không ảnh hưởng đến kết quả chẩn đoán):

Ở đây tôi không traning thêm vì sau 60 epoch, dù training thêm 10 lần nhưng model có dấu hiệu không cải thiện độ chính xác nên trong tương lai tôi sẽ xem xét các phương pháp tối ưu khác để tăng độ chính xác của model.

# Định hướng phát triển

* Bổ sung thêm dữ liệu vào bộ dữ liệu để tăng độ chính xác.
* Cân nhắc training thêm để tăng độ chính xác model.
* Thiết kế thêm một số tính năng nhỏ để hoàn thiện app hơn.
* Tối ưu hóa model với transfer learning và fine tune sử dụng pre-trained model như là InceptionV3 hoặc VGG16(cũng có thể sẽ thử với ResNet).
* Sớm hoàn thành phần phát triển app android.
* Phát triển hỗ trợ sang các nền tảng khác như IOS, Web, Desktop app.
* Làm thêm một số tính năng nhỏ như tra cứu thông tin, cách điều trị về bệnh.



*“Người dân khỏe mạnh là tài sản lớn nhất với mọi quốc gia.”*

* Winston Churchill -

# KẾT THÚC

Cảm ơn đã dành thời gian đọc báo cáo.