1. Tổng quan về MobileNets

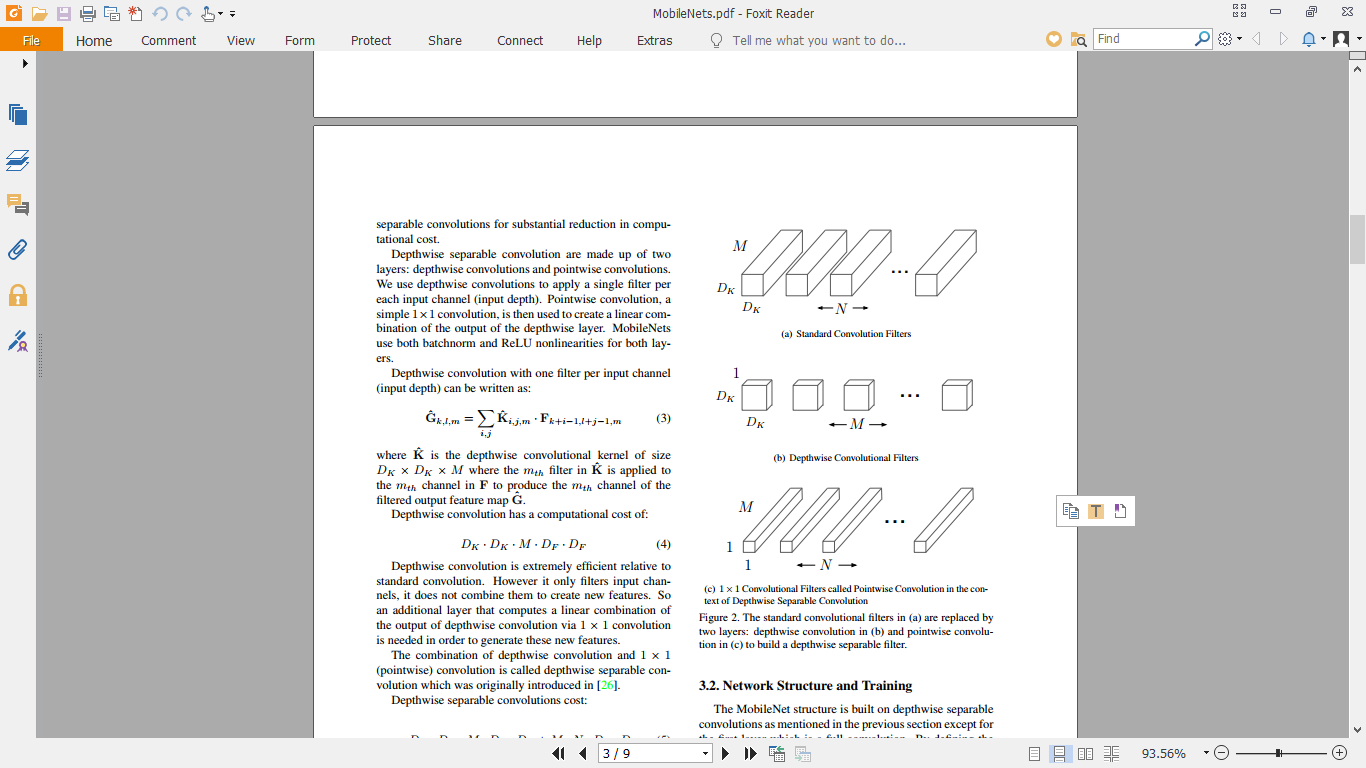
Trong quá trình phát triển mạng neural tích chập sâu (deep convolutional neural networks), Mobinets là một mô hình có hiệu quả được sử dụng trong các ứng dụng có tầm nhìn di động và nhúng bởi kích thước mạng nhỏ mà vẫn đảm bảo hiệu năng cao vầ giảm độ trễ. Mobinets dựa trên kiến trúc streamlined (thuật ngữ chỉ một dạng có tốc độ được gia tăng và dễ dàng di chuyển), sử dụng các tích chập có thể tách sâu (depthwise separable convolutions) để xây dựng các mạng neural sâu có trọng số nhỏ.

Các phần tiếp theo sẽ trình bày kiến trúc Mobilenets, 2 tham số tăng cường có thể cân bằng độ trễ và độ chính xác. Các tham số này cho phép các người xây dựng lựa chọn mô hình có kích thước phù hợp cho các ứng dụng cụ thể dựa trên các ràng buộc của bài toán.

2. Kiến trúc MobileNets

2.1 Depthwise separable convolutions

MobileNet được xây dựng dựa trên depthwise separable convolution, là một dạng của factorized convolutions. Factorized convolution là một covolution mà phân giải một convolution cơ bản thành một depthwise convolution và một 1x1 convolution (được gọi là pointwise convolution). Đối với MobileNets, depthwise convolution sử dụng một bộ lọc đơn cho mỗi kênh đầu vào. Sau đó pointwise convolution sử dụng 1x1 convolution để kết hợp các đầu ra của depthwise convolution. Một convolution cơ bản vừa lọc vừa kết hợp các đầu vào vào trong một tập các đầu ra mới chỉ trong một bước.. Depthwise separable convolution chia tập này thành 2 lớp, 1 lớp để lọc và 1 lớp để kết hợp. Quá trình phân giải này có ảnh hưởng mạnh trong việc giảm thiểu khối lượng tính toán và kích thước mô hình MobileNets. Hình 1 thể hiện cách mà một convolution cơ bản được phân giải thành một depthwise convolution và một 1x1 convolution



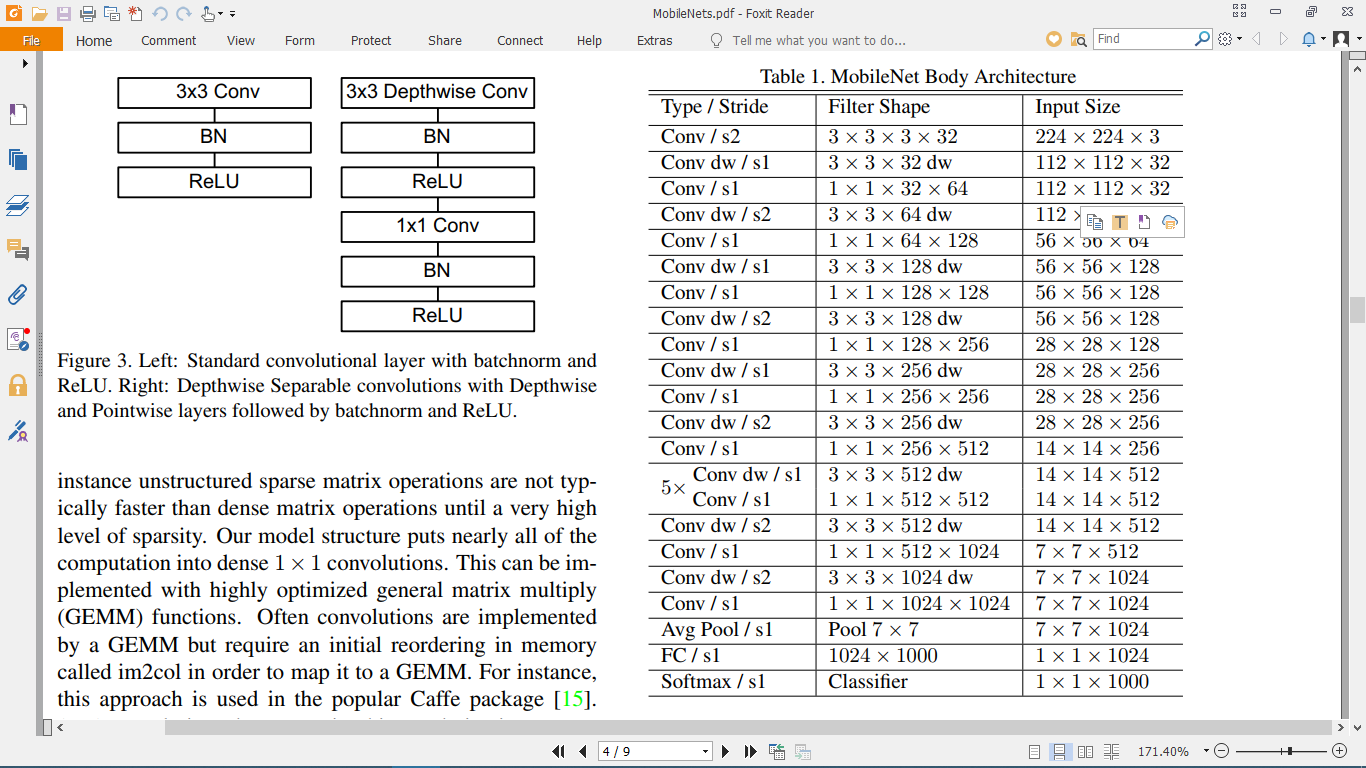
Hình 1. Convolution cơ bản (a) được phân giải thành depthwise convolution (b) và pointwise convolution (c)

Trên hình 1, M là số kênh đầu vào, N là số kênh đầu ra, là số chiều không gian của hạt nhân convolution được giả định là khối vuông.

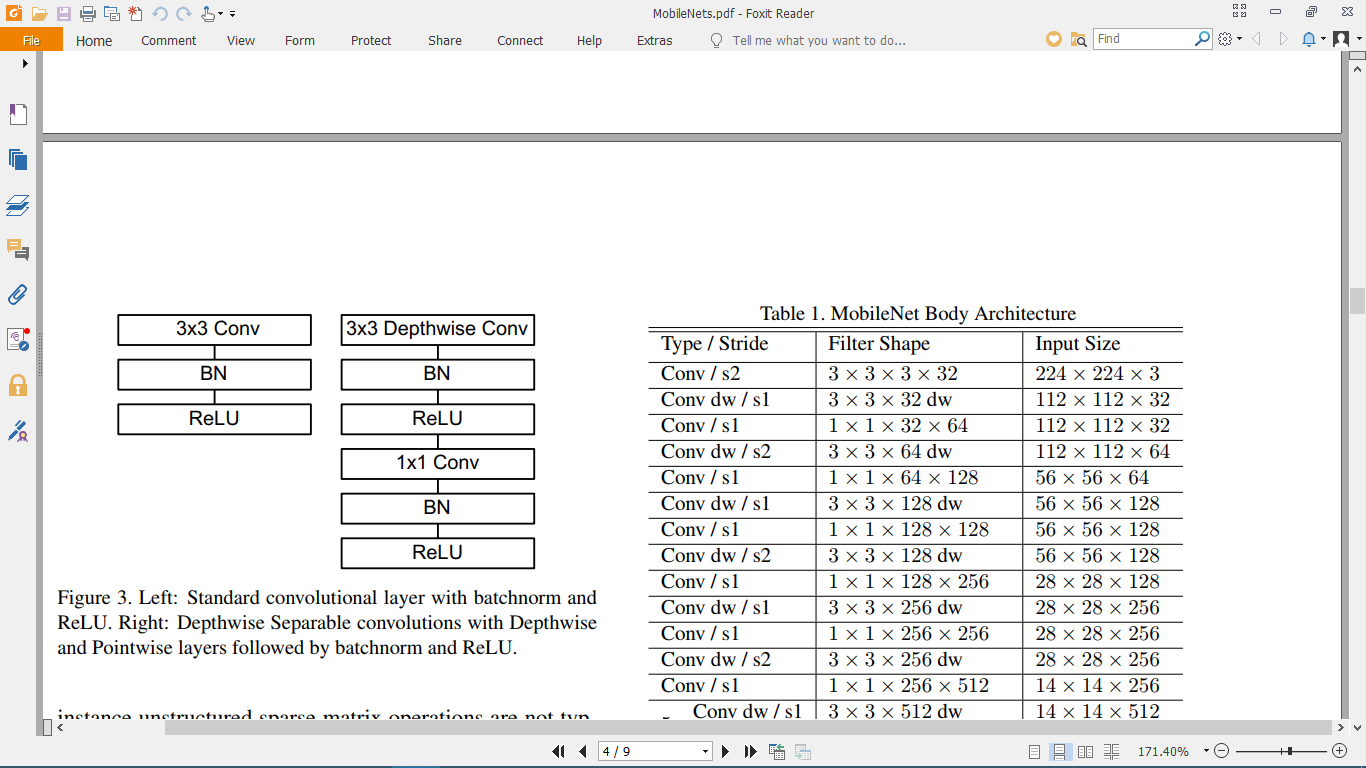
2.2 Cấu trúc mạng và huấn luyện mạng

Kiến trúc MobileNet được thể hiện trong bảng 1

Bảng 1. Kiến trúc MobileNet



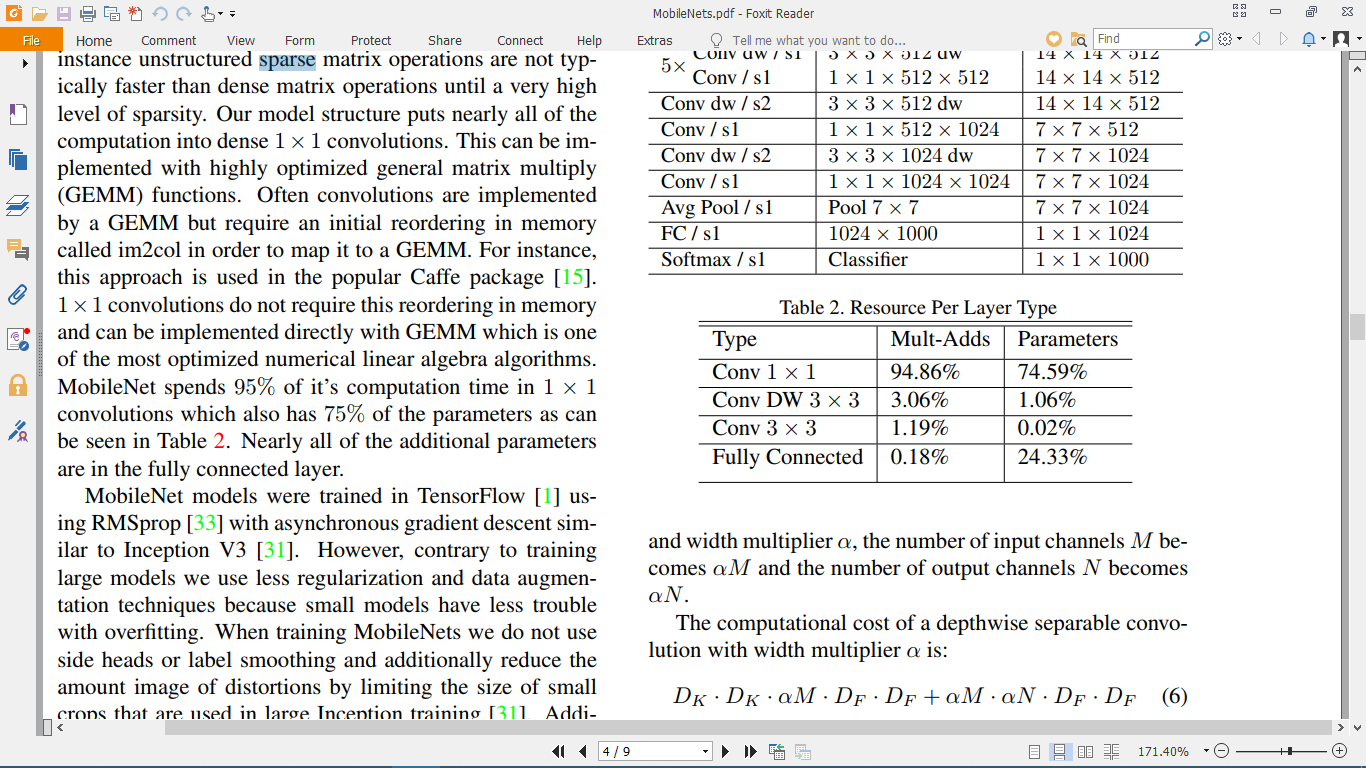
Cột thứ nhất là dạng kiến trúc/bước nhảy của lớp, cột thứ hai là kích thước bộ lọc, cột thứ ba là kích thước đầu vào. Tất cả các lớp được theo sau bởi 1 batchnorm và tính phi tuyển ReLU với ngoại lệ là lớp kết nối đầy đủ, lớp đó không có tính phi tuyến và cấp dữ liệu vào lớp softmax để phân loại. Hình 2 so sánh một lớp có convolution thông thường, batchnorm và tính phi tuyến ReLU với lớp factorized có depthwise convolution, 1x1 convolution cũng như batchnorm và ReLU sau mỗi lớp chập. Quá trình down sampling được xử lí với strided convolution ở trong depthwise convolution như ở lớp đầu tiên. Quá trình pooling cuối cùng giảm độ phân giải không gian xuống 1 trước layer kết nối đầy đủ. MobileNet có 28 layers



Hình 2. Convolution cơ bản với batchnorm và ReLu (trái) và depthwise separable convolution với các lớp depthwise và pointwise theo sau bởi batchnorm và ReLu (phải)

MobiNet dành 95% khối lượng tính toàn vào các 1x1 convolution, nơi có gần 75% tham số (bảng 2). Gần như toàn bộ các tham số khác nằm trong lớp kết nối đầy đủ

Bảng 2. Các tài nguyên mỗi lớp hjhụhjhjhjsssslơllớplớp



MobiNet được luyện tập trong TensorFlow sử dụng RMSprop với gradient descent không đồng bộ. Tuy nhiên, trái với các mô hình luyện tập lớn, chúng ta sử dụng ít hơn các kĩ thuật chính quy và tăng cường dữ liệu vì các mô hình nhỏ gặp ít vấn đề hơn với overfitting. Khi luyện tập MobileNets chúng ta có thể giảm thiểu các hình ảnh bị méo bằng các giới hạn kích thước cắt ảnh được sử dụng. Ngoài ra, một điều rất quan trọng là đặt các trọng số nhỏ hoặc không có trọng số trên các bộ lọc depthwise vì ở đó có khá ít tham số

3. Hai tham số tăng cường

3.1 Width Multiplier: Mô hình nhỏ hơn

Mặc dù kiến trúc MobileNet đã nhỏ và ít trễ, song nhiều khi trong các trường hợp cụ thể có thể đòi hỏi mô hình nhỏ hơn và ít tính toán hơn, ở đây chúng ta dùng một tham số α được gọi là width multiplier, mục đính để giảm kích thước mạng đồng đều ở mỗi layer. Đối với một lớp, số lượng kênh đầu vào M sẽ trở thành αM, số lượng kênh đầu ra N sẽ trở thành αN, với 0<α1, thường α được chọn là 0.75, 0.5, 0.25. Tham số α có thể giảm thiểu khá nhiều khối lượng tính toán và số lượng tham số xuống xấp xỉ lần. Tham số α có thể được áp dụng với bất cứ cấu trúc mô hình nào để xác định mô hình nhỏ hơn với độ chính xác chấp nhận được, cân bằng giữa độ trễ và kích thước mạng

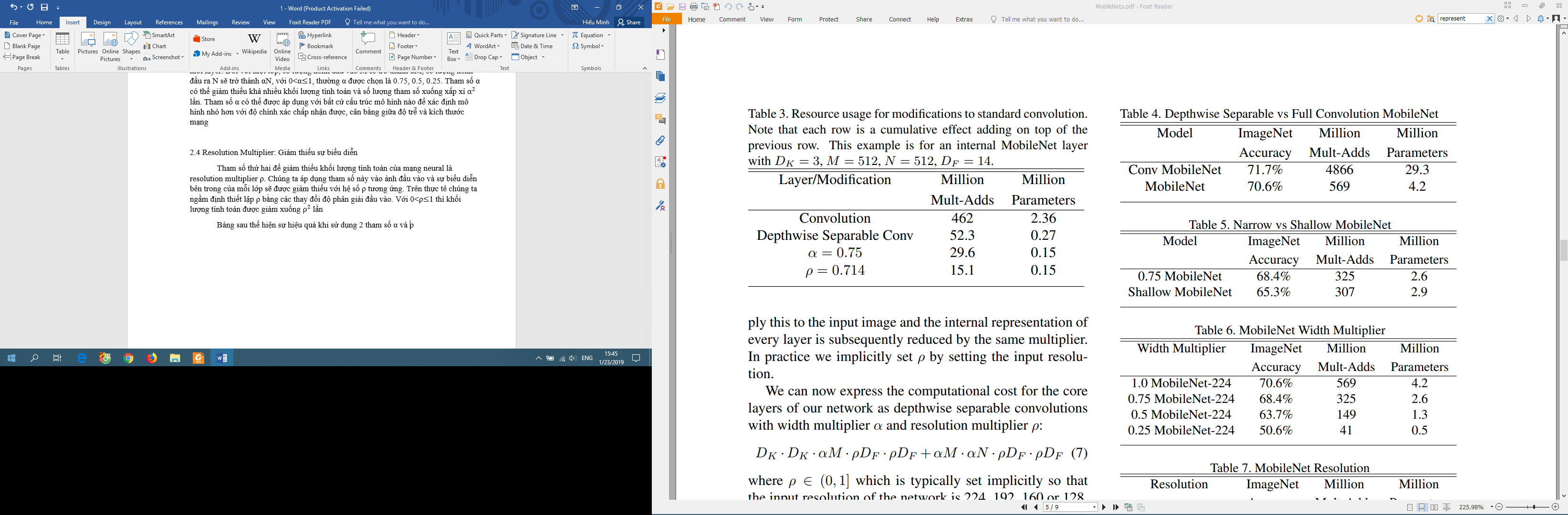
3.2 Resolution Multiplier: Giảm thiểu sự biểu diễn

Tham số thứ hai để giảm thiểu khối lượng tính toán của mạng neural là resolution multiplier ρ. Chúng ta áp dụng tham số này vào ảnh đầu vào và sự biểu diễn bên trong của mỗi lớp sẽ được giảm thiểu với hệ số ρ tương ứng. Trên thực tê chúng ta ngầm định thiết lập ρ bằng các thay đổi độ phân giải đầu vào. Với 0<ρ1 thì khối lượng tính toán được giảm xuống lần

3.3 Hiệu quả khi sử dụng 2 tham số α và ρ

Bảng 3 thể hiện sự hiệu quả khi sử dụng 2 tham số α và ρ

Bảng 3. Sử dụng tài nguyên cho convolution cơ bản có chỉnh sửa

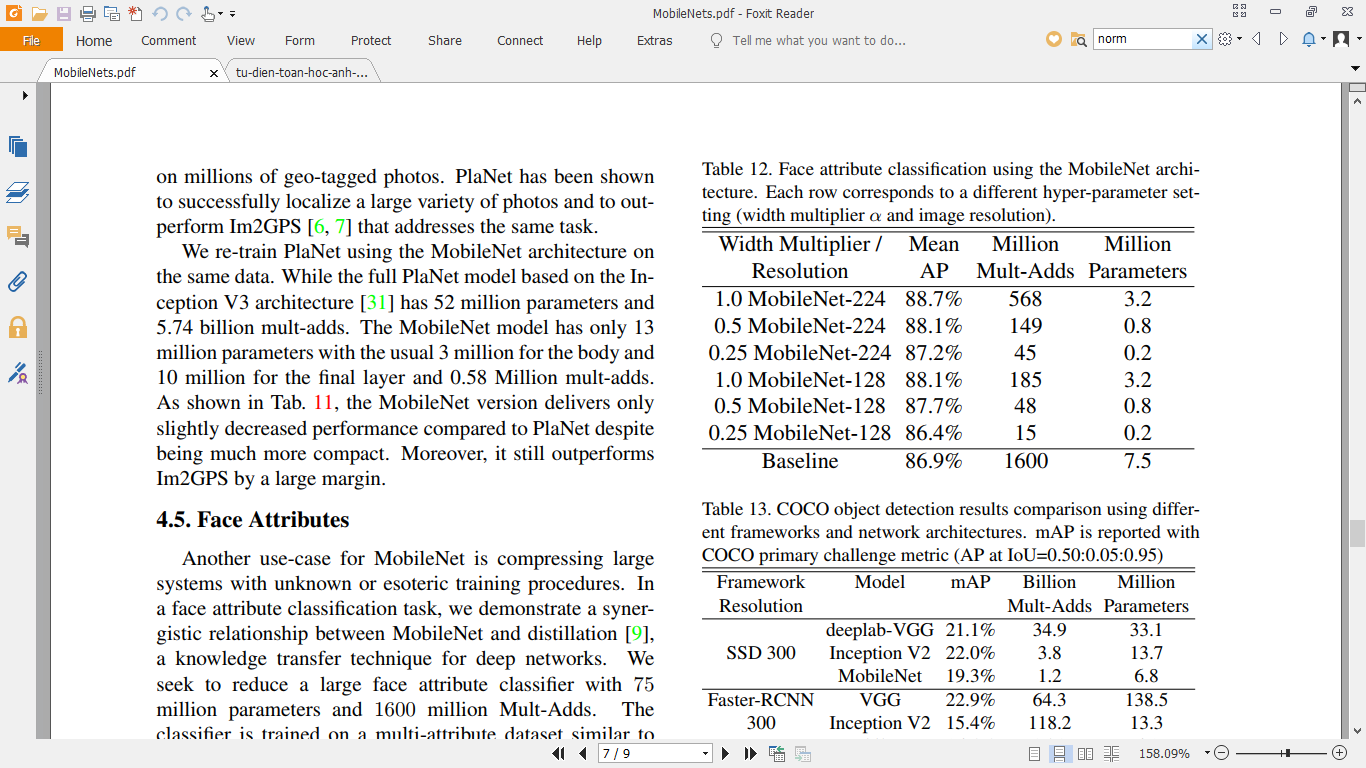


4 hàng thể hiện số phép nhân-cộng và tham số trong một lớp convolution đầy đủ, depthwise separable convolution, và khi áp dụng 2 tham số α, ρ.

4. Ứng dụng của MobileNet trong nhận diện các thuộc tính khuôn mặt

MobileNet có thể được sử dụng để nén một hệ thống lớn với các thủ tục luyện tập ẩn. Trong tác vụ phân loại thuộc tính khuôn mặt, chúng ta sẽ chứng minh mối quan hệ giữa MobileNet và distillation, một kĩ thuật chuyển đổi đối với mạng sâu. Sử dụng MobileNet sẽ giảm số lượng phân lớp thuộc tính khuôn mặt với 75 triệu tham số và 1.6 tỉ phép nhận-cộng. Kết hợp với khả năng mở rộng của luyện tập distillatinon và các tham số kĩ thuật của MobileNet, hệ thống sau cùng thể hiện được hiệu năng vượt trội (bảng 4) Có thể thấy từ bảng 4 tính ưu việt của MobileNet: nó đạt được độ chính xác trung bình (mean average precision – Mean AP) chấp nhận được trong khi chỉ tiêu thụ 1% số lượng phép nhân-cộng (khi sử dụng tham số α=0.5 và giảm độ phân giải ảnh đầu vào từ 224 xuống còn 128).

Bảng 4. Nhận diện các thuộc tính khuôn mặt sử dụng kiến trúc Mobinet



TLTK: Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*, Google Inc, 2017