

	VIETTEL AI RACE	TD045
	HỌC CHUYÊN GIAO VÀ TĂNG CƯỜNG DỮ LIỆU	Lần ban hành: 1

1. Transfer learning

Bạn có bài toán cần nhận diện 1000 người nổi tiếng ở Việt Nam, tuy nhiên dữ liệu để train chỉ khoảng 10 ảnh / 1 người. Số lượng dữ liệu là quá ít để train một mô hình CNN hoàn chỉnh.

Bạn tìm trên mạng thấy VGGFace2 dataset có 3.31 triệu ảnh của 9131 người, với trung bình 362.6 ảnh cho mỗi người. Họ làm bài toán tương tự đó là nhận diện ảnh từng người và họ đã train được CNN model với accuracy hơn 99%.

Bạn nhớ ra là trong convolutional neural network, convolutional layer có tác dụng lấy ra các đặc trưng của ảnh, và sau hàng loạt các convolutional layer + pooling layer (ConvNet) thì model sẽ học được các đặc điểm của ảnh, trước khi được cho vào fully connected layer => ConvNet trong VGGFace2 model cũng lấy ra được các đặc điểm của mặt người (tai, mũi, tóc,...) => Ta cũng có thể áp dụng phần ConvNet của VGGFace2 model vào bài toán nhận diện mặt người nổi tiếng ở Việt Nam để lấy ra các đặc điểm của mặt.

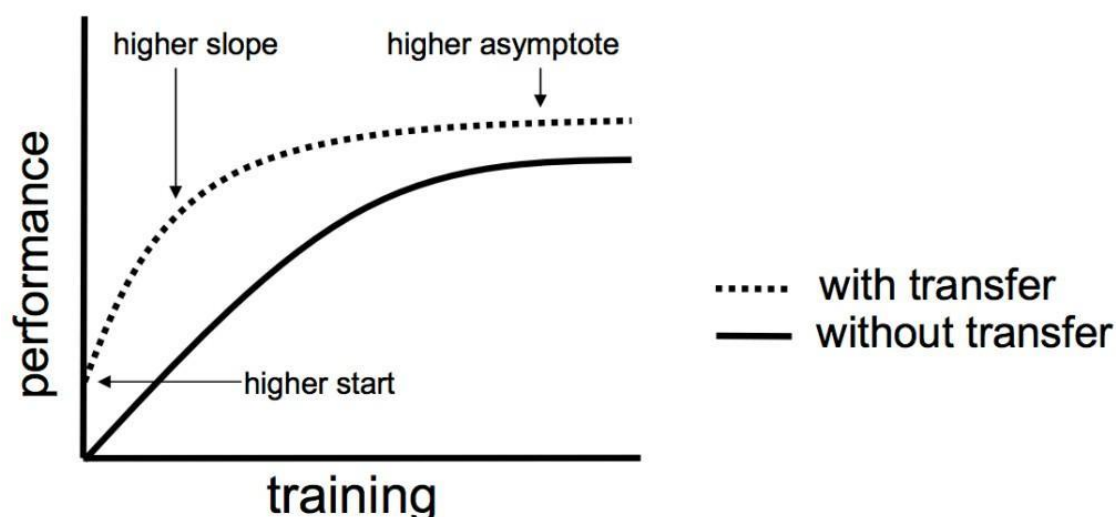
Quá trình sử dụng pre-trained model như trên gọi là transfer learning.

Các pre-trained model được sử dụng thường là các bài toán được train với dữ liệu lớn ví dụ ImageNet, dữ liệu chứa 1.2 triệu ảnh với 1000 thể loại khác nhau.

Có 2 loại transfer learning:

- **Feature extractor:** Sau khi lấy ra các đặc điểm của ảnh bằng việc sử dụng ConvNet của pre-trained model, thì ta sẽ dùng linear classifier (linear SVM, softmax classifier,...) để phân loại ảnh. Hiểu đơn giản thì các đặc điểm ảnh (tai, mũi, tóc,...) giờ như input của bài toán linear regression hay logistic regression.
- **Fine tuning:** Sau khi lấy ra các đặc điểm của ảnh bằng việc sử dụng ConvNet của pre-trained model, thì ta sẽ coi đây là input của 1 CNN mới bằng cách thêm các ConvNet và Fully Connected layer. Lý do là ConvNet của VGGFace 2 model có thể lấy ra được các thuộc tính của mặt người nói chung nhưng người Việt Nam có những đặc tính khác nên cần thêm 1 số Convnet mới để học thêm các thuộc tính của người Việt Nam.

	VIETTEL AI RACE	TD045
	HỌC CHUYỂN GIAO VÀ TĂNG CƯỜNG DỮ LIỆU	Lần ban hành: 1



Hình 11.1: Hiệu quả khi sử dụng transfer learning [2]

Bài toán: Ta muốn nhận diện ảnh của 17 loài hoa, mỗi loài hoa có khoảng 80 ảnh.

Sử dụng pre-trained model là VGG 16 của ImageNet. Mô hình VGG 16 mọi người có thể xem lại bài CNN. Mô hình VGG16 của ImageNet dataset, phân loại ảnh thuộc 1000 thể loại khác nhau. Nên có thể hiểu là nó đủ tổng quát để tách ra các đặc điểm của bức ảnh, cụ thể ở đây là hoa.

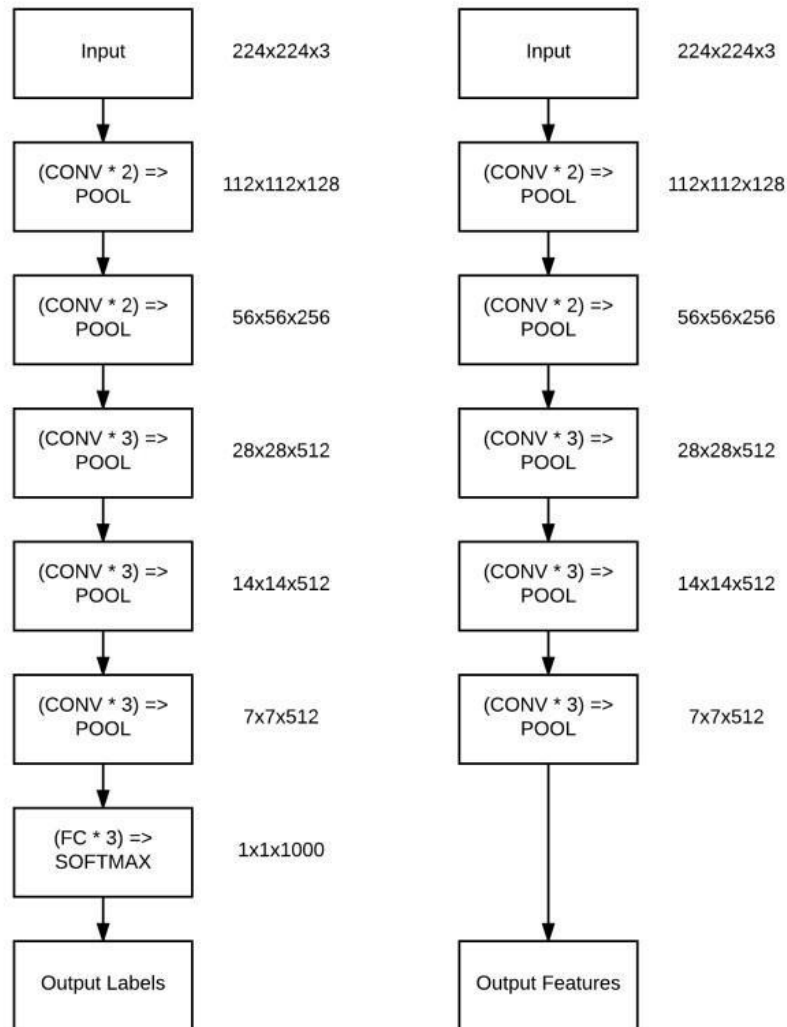
1.1 Feature extractor

Ta chỉ giữ lại phần ConvNet trong CNN và bỏ đi FCs. Sau đó dùng output của ConvNet còn lại để làm input cho Logistic Regression với nhiều output, như trong hình 11.2.

Mô hình logistic regression với nhiều output có 2 dạng:

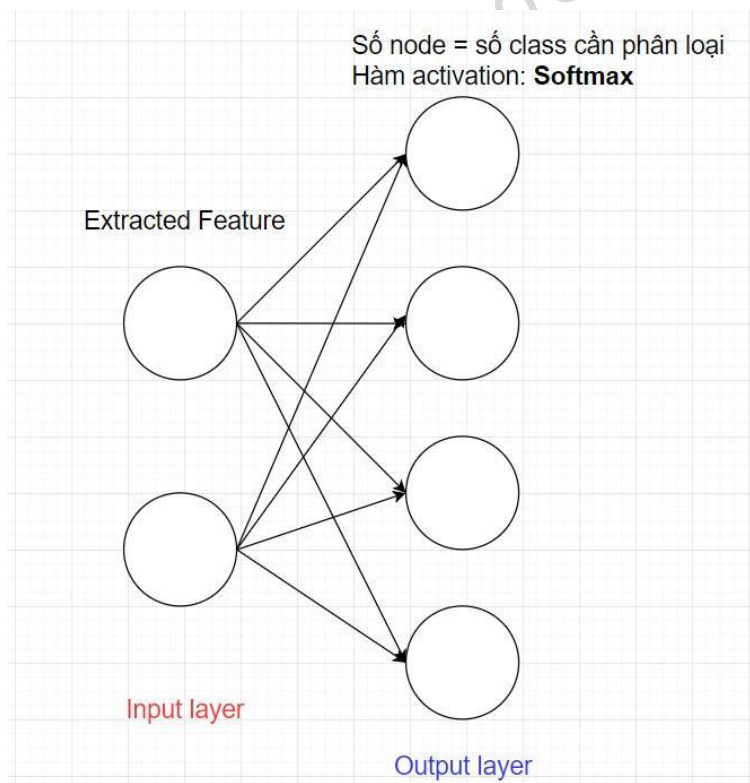
Dạng thứ nhất là một neural network, không có hidden layer, hàm activation ở output layer là softmax function, loss function là hàm categorical-cross entropy, giống như bài phân loại ảnh.

	VIETTEL AI RACE		TD045
	HỌC CHUYỂN GIAO VÀ TĂNG CƯỜNG DỮ LIỆU		Lần ban hành: 1

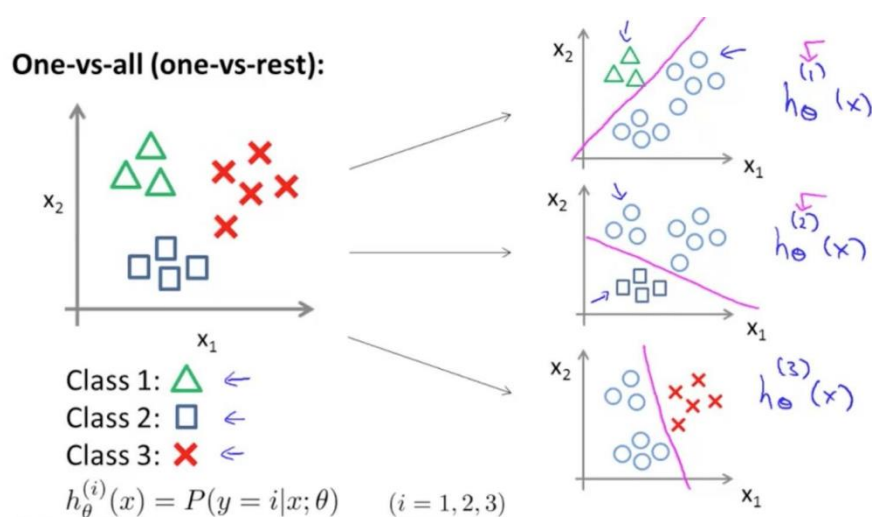


Hình 11.2: Bên trái là mô hình VGG16, bên phải là mô hình VGG16 chỉ bao gồm ConvNet (bỏ Fully Connected layer) [21]

	VIETTEL AI RACE	TD045
	HỌC CHUYỂN GIAO VÀ TĂNG CƯỜNG DỮ LIỆU	Lần ban hành: 1



Dạng thứ hai giống như bài logistic regression, tức là model chỉ phân loại 2 class. Mỗi lần ta sẽ phân loại 1 class với tất cả các class còn lại.



Hình 11.3: 1 vs all classification [13]

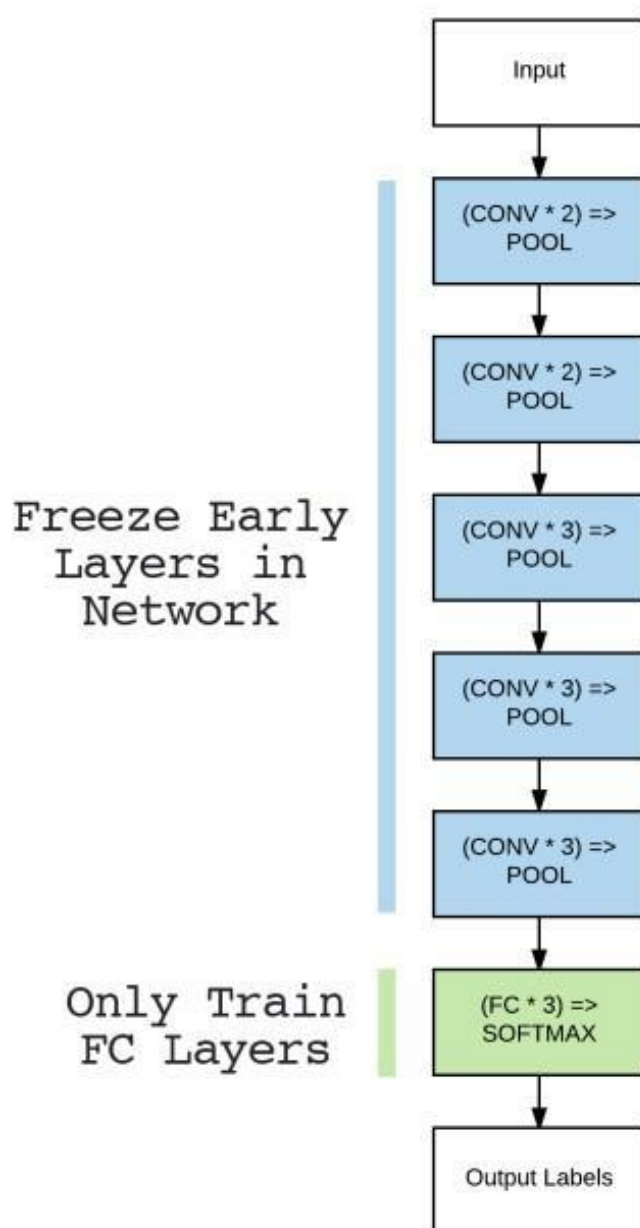
1.2 Fine tuning

Ta chỉ giữ lại phần ConvNet trong CNN và bỏ đi FCs. Sau đó thêm các Fully Connected layer mới vào output của ConvNet như trong hình 11.4.

	VIETTEL AI RACE	TD045
	HỌC CHUYỂN GIAO VÀ TĂNG CƯỜNG DỮ LIỆU	Lần ban hành: 1

Khi train model ta chia làm 2 giai đoạn

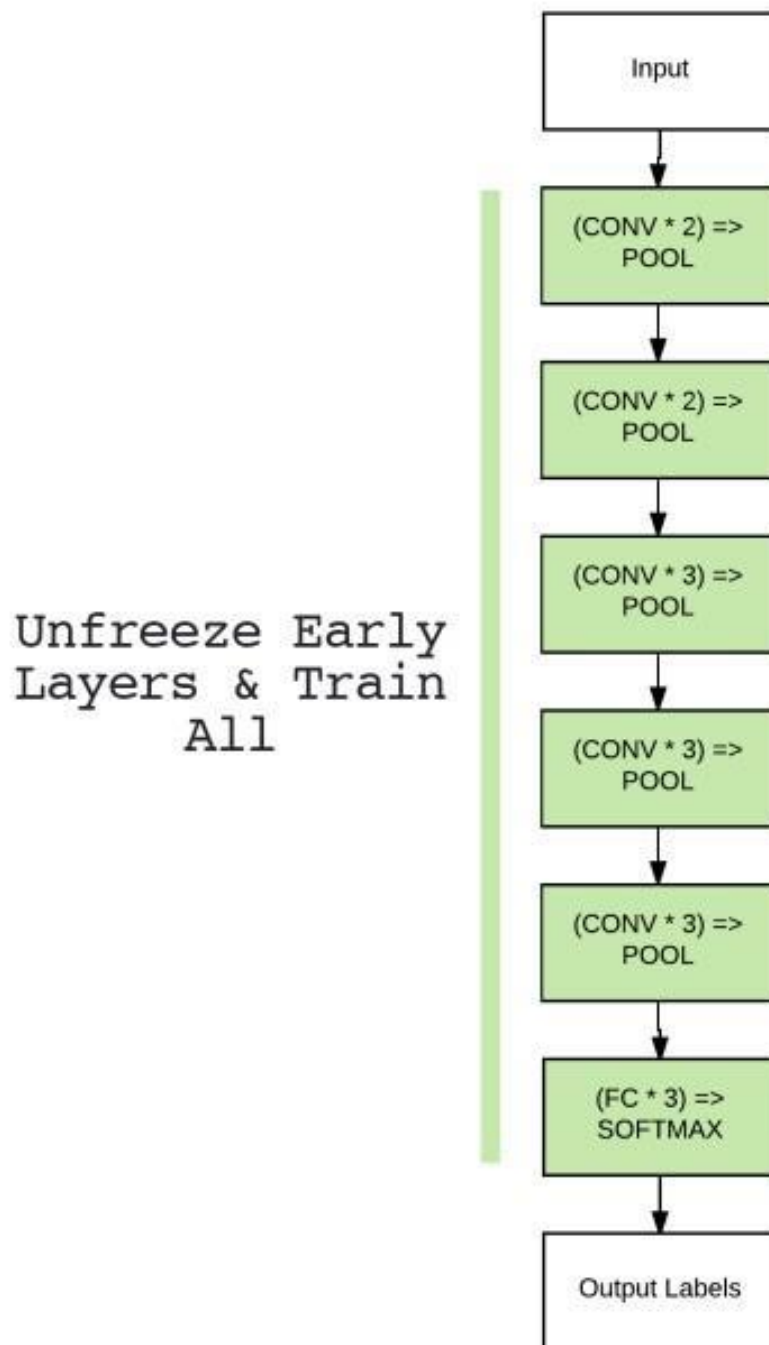
Giai đoạn 1: Vì các Fully Connected layer ta mới thêm vào có các hệ số được khởi tạo ngẫu nhiên tuy nhiên các layer trong ConvNet của pre-trained model đã được train với ImageNet dataset nên ta sẽ không train (đóng băng/freeze) trên các layer trong ConvNet của model VGG16. Sau khoảng 20-30 epoch thì các hệ số ở các layer mới đã được học từ dữ liệu thì ta chuyển sang giai đoạn 2.



Hình 11.5: Freeze các layer của pre-trained model, chỉ train ở các layer mới [21]

	VIETTEL AI RACE	TD045
	HỌC CHUYỂN GIAO VÀ TĂNG CƯỜNG DỮ LIỆU	Lần ban hành: 1

Giai đoạn 2: Ta sẽ unfreeze các layer trên ConvNet của pre-trained model và train trên các layer của ConvNet của pre-trained model và các layer mới. Bạn có thể unfreeze tất cả các layer trong ConvNet của VGG16 hoặc chỉ unfreeze một vài layer cuối tùy vào thời gian và GPU bạn có.



Hình 11.6: Freeze các layer của pre-trained model, chỉ train ở các layer mới [21]

	VIETTEL AI RACE	TD045
	HỌC CHUYÊN GIAO VÀ TĂNG CƯỜNG DỮ LIỆU	Lần ban hành: 1

Accuracy của fine-tuning tốt hơn so với feature extractor tuy nhiên thời gian train của fine-tuning cũng lâu hơn rất nhiều. Giải thích đơn giản thì feature extractor chỉ lấy ra đặc điểm chung chung từ pre-trained model của ImageNet dataset cho các loài hoa, nên không được chính xác lắm. Tuy nhiên ở phần fine-tuning ta thêm các layer mới, cũng như train lại 1 số layer ở trong ConvNet của VGG16 nên model giờ học được các thuộc tính, đặc điểm của các loài hoa nên độ chính xác tốt hơn.

1.3 Khi nào nên dùng transfer learning

Có 2 yếu tố quan trọng nhất để dùng transfer learning đó là kích thước của dữ liệu bạn có và sự tương đồng của dữ liệu giữa mô hình bạn cần train và pre-trained model.

- Dữ liệu bạn có nhỏ và tương tự với dữ liệu ở pre-trained model. Vì dữ liệu nhỏ nên nếu dùng fine-tuning thì model sẽ bị overfitting. Hơn nữa là dữ liệu tương tự nhau nên là ConvNet của pre-trained model cũng lấy ra các đặc điểm ở dữ liệu của chúng ta. Do đó nên dùng feature extractor.
- Dữ liệu bạn có lớn và tương tự với dữ liệu ở pre-trained model. Giờ có nhiều dữ liệu ta không sợ overfitting do đó nên dùng fine-tuning.
- Dữ liệu bạn có nhỏ nhưng khác với dữ liệu ở pre-trained model. Vì dữ liệu nhỏ nên ta nên dùng feature extractor để tránh overfitting. Tuy nhiên do dữ liệu ta có và dữ liệu ở pre-trained model khác nhau, nên không nên dùng feature extractor với toàn bộ ConvNet của pre-trained model mà chỉ dùng các layer đầu. Lý do là vì các layer ở phía trước sẽ học các đặc điểm chung chung hơn (cạnh, góc,...), còn các layer phía sau trong ConvNet sẽ học các đặc điểm cụ thể hơn trong dataset (ví dụ mắt, mũi,...).
- Dữ liệu bạn có lớn và khác với dữ liệu ở pre-trained model. Ta có thể train model từ đầu, tuy nhiên sẽ tốt hơn nếu ta khởi tạo các giá trị weight của model với giá trị của pre-trained model và sau đó train bình thường.

Lưu ý

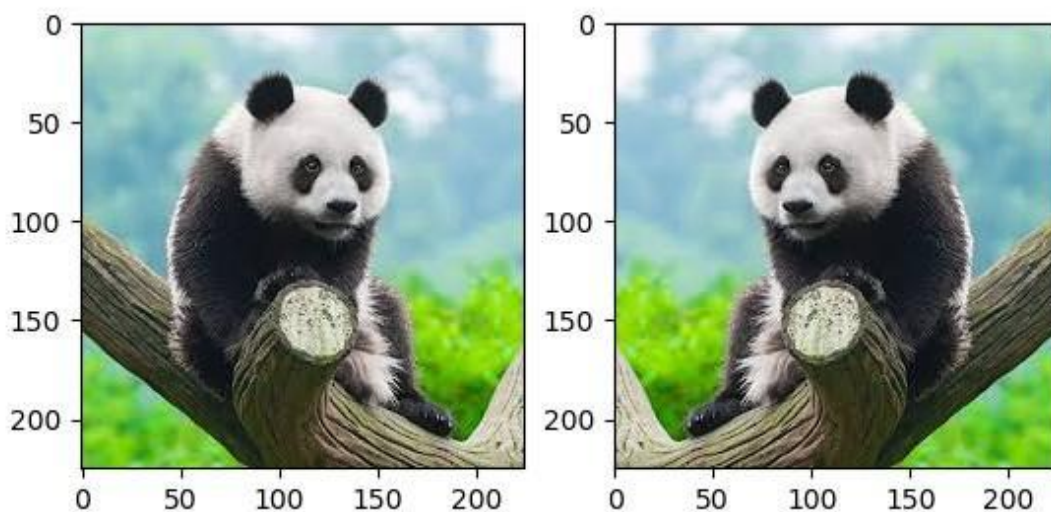
- Vì pre-trained model đã được train với kích thước ảnh cố định, nên khi dùng pre-trained model ta cần resize lại ảnh có kích ảnh bằng kích thước mà ConvNet của pre-trained model yêu cầu.
- Hệ số learning rate của ConvNet của pre-trained model nên được đặt với giá trị nhỏ vì nó đã được học ở pre-trained model nên ít cần cập nhật hơn so với các layer mới thêm.

2. Data augmentation

Ngoài transfer learning, có một kỹ thuật nữa giải quyết vấn đề có ít dữ liệu cho việc training model, đó là data augmentation. Augmentation là kỹ thuật tạo ra dữ liệu training từ dữ liệu mà ta đang có. Cùng xem một số kỹ thuật augmentation phổ biến với ảnh nhé.

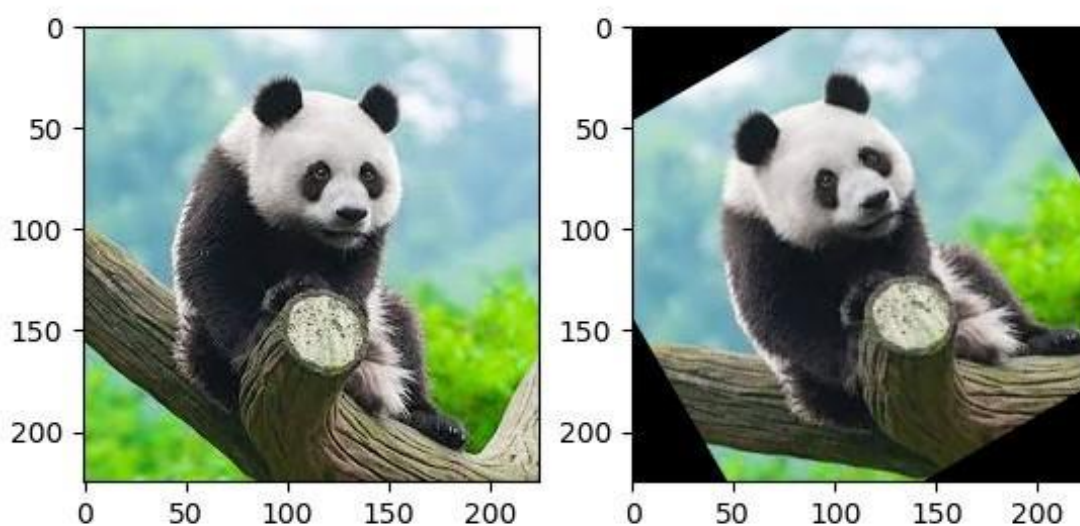
Flip: Lật ngược ảnh theo chiều dọc hoặc chiều ngang

	VIETTEL AI RACE		TD045
	HỌC CHUYỂN GIAO VÀ TĂNG CƯỜNG DỮ LIỆU		Lần ban hành: 1



Hình 11.7: lật ngược ảnh theo chiều dọc

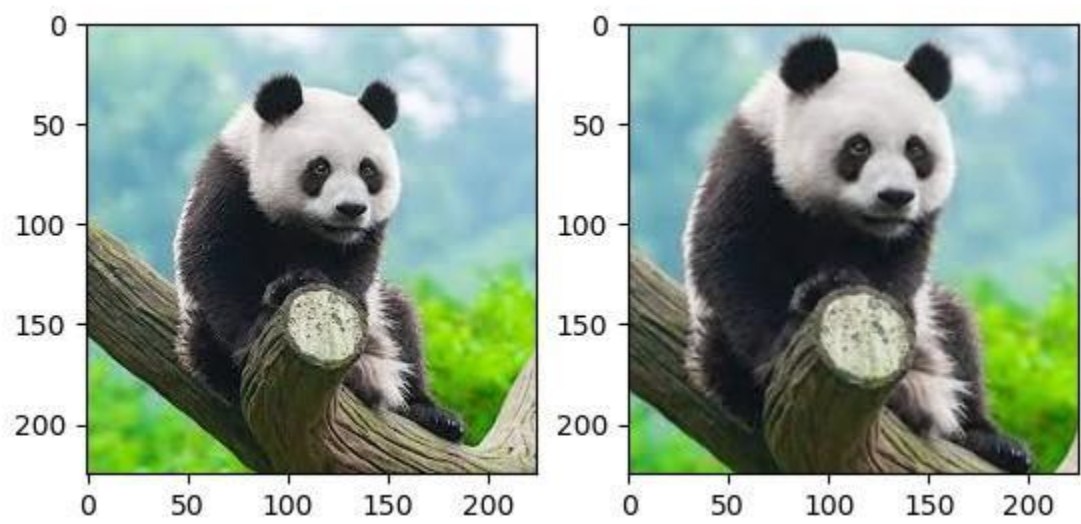
Rotation: Quay ảnh theo nhiều góc khác nhau



Hình 11.8: Rotate ảnh 30 độ

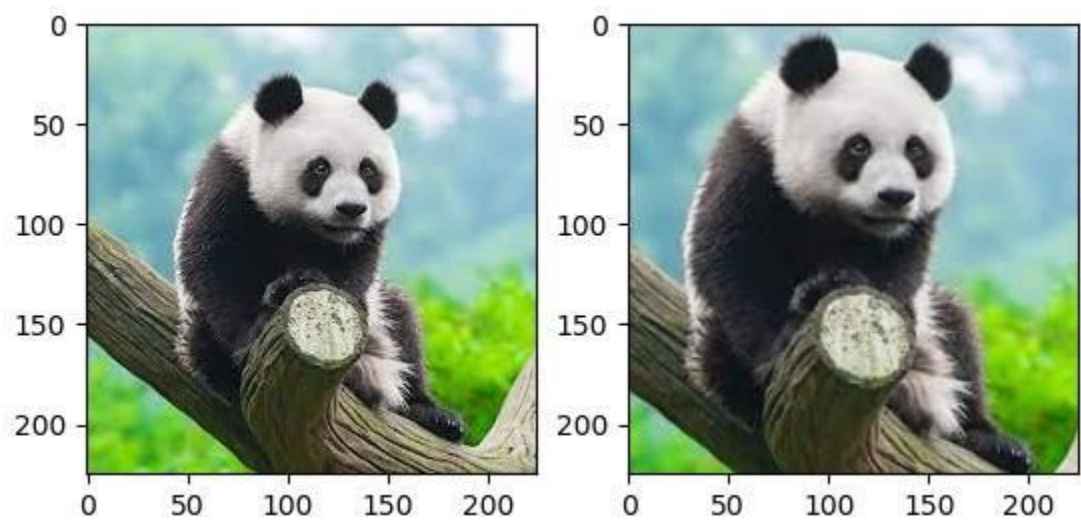
Scale: Phóng to hoặc thu nhỏ ảnh

	VIETTEL AI RACE		TD045
	HỌC CHUYỂN GIAO VÀ TĂNG CƯỜNG DỮ LIỆU		Lần ban hành: 1



Hình 11.9: Scale ảnh

Crop: Cắt một vùng ảnh sau đó resize vùng ảnh đấy về kích thước ảnh ban đầu



Hình 11.10: Crop ảnh