

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN



BÀI BÁO CÁO

LAB 02

SINH VIÊN THỰC HIỆN
Nguyễn Anh Thư – 18127227

GIÁO VIÊN HƯỚNG DẪN
Thầy Lý Quốc Ngọc
Thầy Phạm Minh Hoàng
Thầy Phạm Thanh Tùng

BỘ MÔN: XỬ LÝ ẢNH SỐ VÀ VIDEO SỐ

Thành phố Hồ Chí Minh – 04/2021

MỤC LỤC

I.	Nội dung yêu cầu:	3
II.	Nội dung thực hiện:	3
1.	Nội suy màu:	3
	Nội suy tuyến tính:	3
	Nội suy láng giềng gần nhất:	3
2.	Ma trận biến đổi affine:	3
	Ma trận tịnh tiến theo véc tơ (dx, dy):	3
	Ma trận xoay với 1 góc angle:	3
	Ma trận scale theo tỉ lệ sx, sy:	4
3.	Phóng to, thu nhỏ ảnh:	4
4.	Thay đổi kích thước ảnh:	5
5.	Xoay ảnh giữ kích thước ảnh output như ảnh input:	6
6.	Xoay ảnh mà giữ nguyên ảnh input không bị mất:	7
7.	Lật ảnh ngang:	8
8.	Lật ảnh dọc:	9
III.	Tài liệu tham khảo:	10

I. Nội dung yêu cầu:

STT	YÊU CẦU	HOÀN THÀNH
1	Đọc ảnh đầu vào từ file	100%
2	Lưu ảnh kết quả vào file	100%
3	Phóng to, thu nhỏ ảnh	100%
4	Thay đổi kích thước ảnh	100%
5	Xoay ảnh giữ kích thước ban đầu, ảnh cũ bị mất đi một phần	100%
6	Xoay ảnh thay đổi kích thước để giữ được nguyên ảnh	100%
7	Lật ảnh ngang	100%
8	Lật ảnh dọc	100%

II. Nội dung thực hiện:

1. Nội suy màu:

Nội suy tuyến tính:

Sử dụng công thức: $f'(x', y') = (1-a) * (1-b) * f(l, k) + a * (1-b) * f(l+1, k) + (1-a) * b * f(l, k+1) + a * b * f(l+1, k+1)$

Với x, y là tọa độ thực của ảnh input được biến đổi ngược từ điểm pixel x', y' của ảnh output

f' là ảnh output, f là ảnh input

$l = \text{round}(x), k = \text{round}(y)$

$a = x - l, b = y - k$

Nội suy láng giềng gần nhất:

Sử dụng công thức: $f'(x', y') = f(\text{round}(x), \text{round}(y))$

Với x, y là tọa độ thực của ảnh input được biến đổi ngược từ điểm pixel x', y' của ảnh output

f' là ảnh output, f là ảnh input

2. Ma trận biến đổi affine:

Ma trận tịnh tiến theo véc tơ (dx, dy) :

$$\begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix}$$

Ma trận xoay với 1 góc $angle$:

$angle$ là đơn vị rad nên phải đổi từ độ sang rad

$angle = angle * 3.14 / 180$

$$\begin{pmatrix} \cos(angle) & -\sin(angle) & 0 \\ \sin(angle) & \cos(angle) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Mã trận scale theo tỉ lệ s_x, s_y :

$$\begin{matrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{matrix}$$

3. Phóng to, thu nhỏ ảnh:

- Hướng dẫn cách chạy chương trình:

Command line:

<tên file.exe> -zoom <đường dẫn ảnh input> <đường dẫn ảnh output> <cách nội suy màu> <tỉ lệ zoom theo x> <tỉ lệ zoom theo y>

Cách nội suy màu: -bl: nội suy tuyến tính, -nn: nội suy láng giềng gần nhất

- Giải pháp:

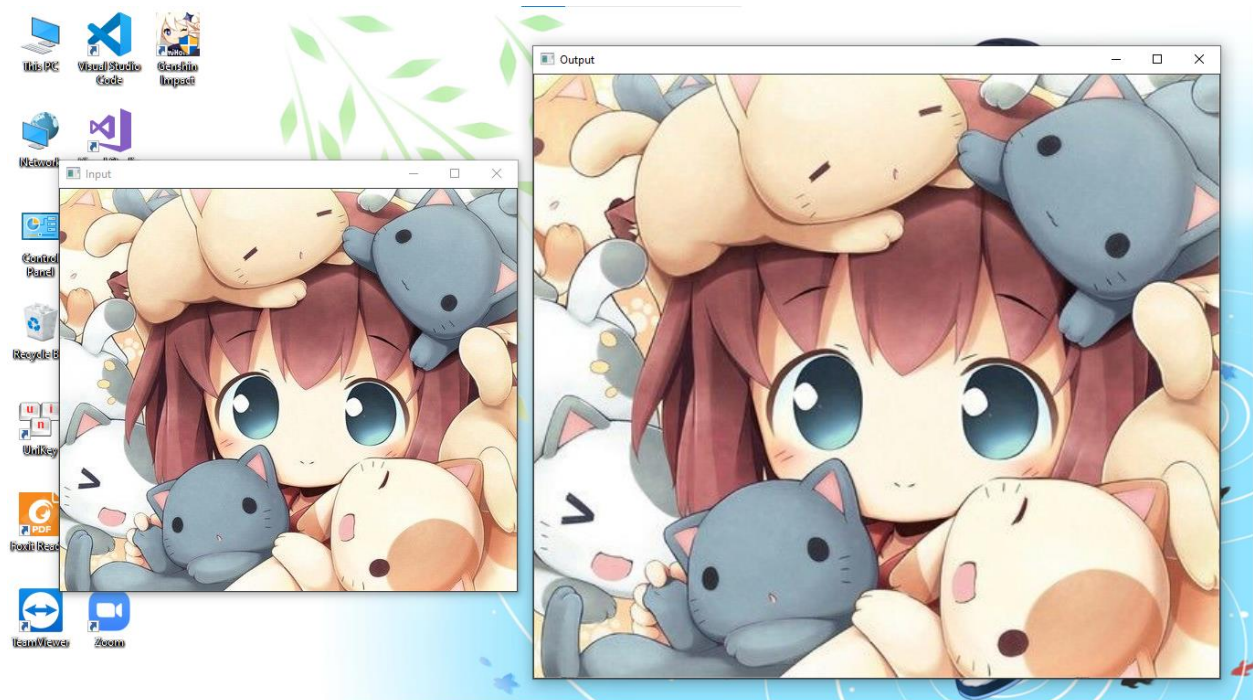
Mã trận affine nghịch scale với tỉ lệ <1/tỉ lệ zoom theo x> và <1/tỉ lệ zoom theo y>

- Chạy chương trình và kết quả:

Phóng to ảnh với tỉ lệ 1.5, 1.5 với nội suy tuyến tính

```
E:\study\XLA\18127227_Lab02\x64\Debug>18127227_Lab02.exe -zoom E:\cat.jpg E:\out0.jpg -bl 1.5 1.5
```

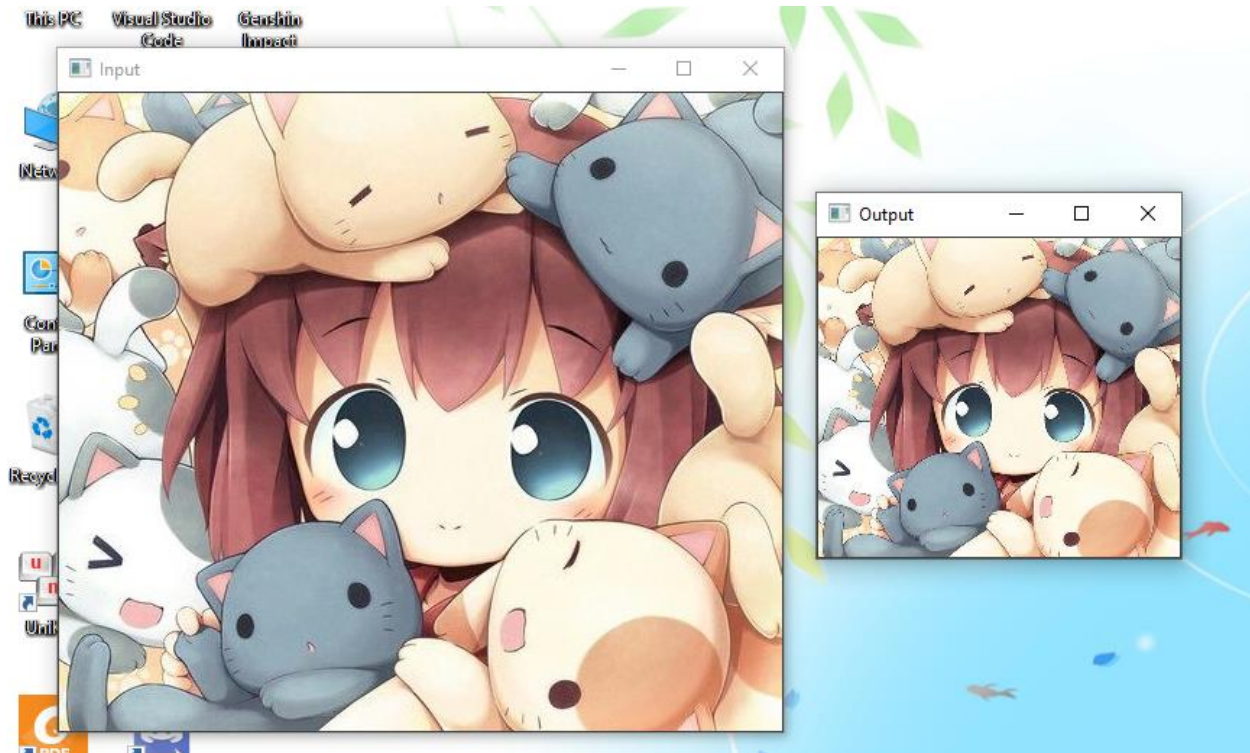
Kết quả



Thu nhỏ ảnh với tỉ lệ 0.5, 0.5 với nội suy láng giềng gần nhất

```
E:\study\XLA\18127227_Lab02\x64\Debug>18127227_Lab02.exe -zoom E:\cat.jpg E:\out1.jpg -nn 0.5 0.5
```

Kết quả



4. Thay đổi kích thước ảnh:

- Hướng dẫn cách chạy chương trình:

Command line:

<tên file.exe> -resize <đường dẫn ảnh input> <đường dẫn ảnh output> <cách nội suy màu> <width ảnh output> <height ảnh output>

Cách nội suy màu: -bl: nội suy tuyến tính, -nn: nội suy láng giềng gần nhất

- Giải pháp:

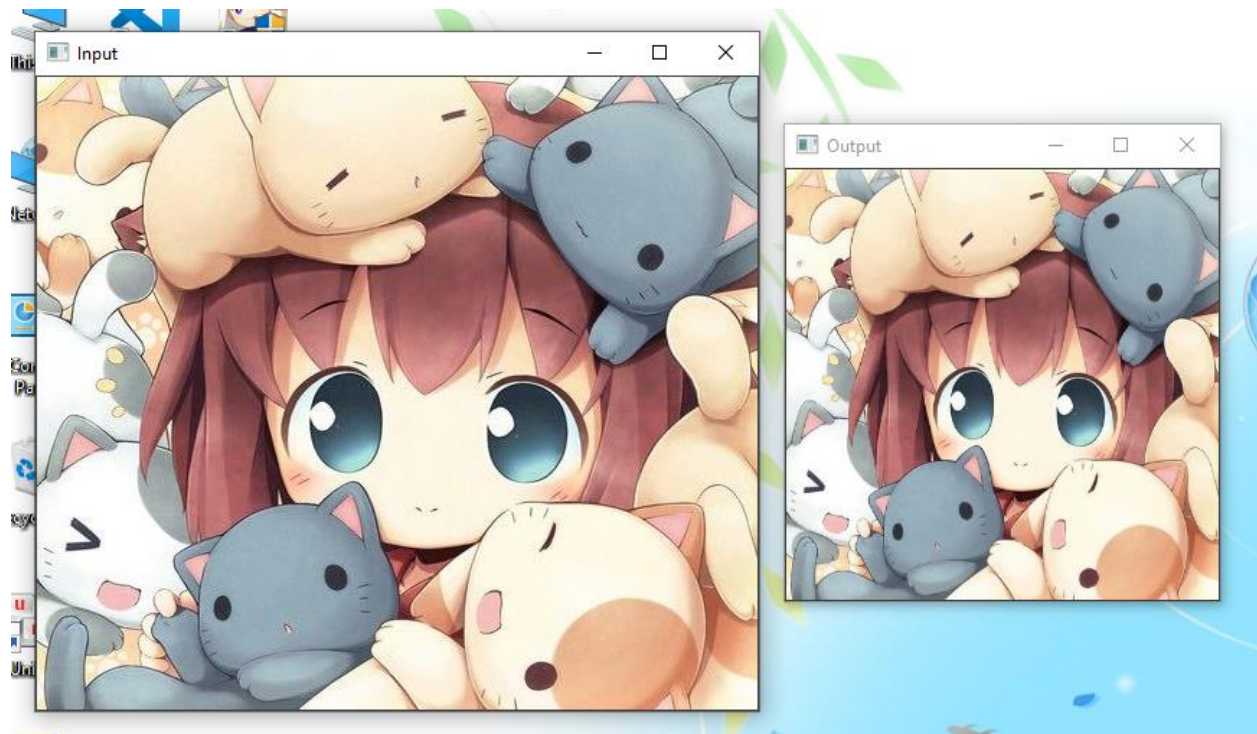
Ma trận affine nghịch scale với tỉ lệ width ảnh input / width ảnh output và height input / height output

- Chạy chương trình và kết quả:

Thay đổi kích thước ảnh với kích thước mới là 100x100 với nội suy tuyến tính

```
E:\study\XLA\18127227_Lab02\x64\Debug>18127227_Lab02.exe -resize E:\cat.jpg E:\out1.jpg -bl 300 300
```


Kết quả



5. Xoay ảnh giữ kích thước ảnh output như ảnh input:

- Hướng dẫn cách chạy chương trình:

Command line:

<tên file.exe> -rotK <đường dẫn ảnh input> <đường dẫn ảnh output> <cách nội suy màu> <góc xoay với đơn vị độ>

Cách nội suy màu: -bl: nội suy tuyến tính, -nn: nội suy láng giềng gần nhất

- Giải pháp:

Ma trận biến đổi nghịch được tạo bởi các ma trận:

Ma trận tịnh tiến theo véc tơ CO với C là tâm ảnh input, để dịch về tâm O

Ma trận xoay với - góc xoay đã được đổi qua rad

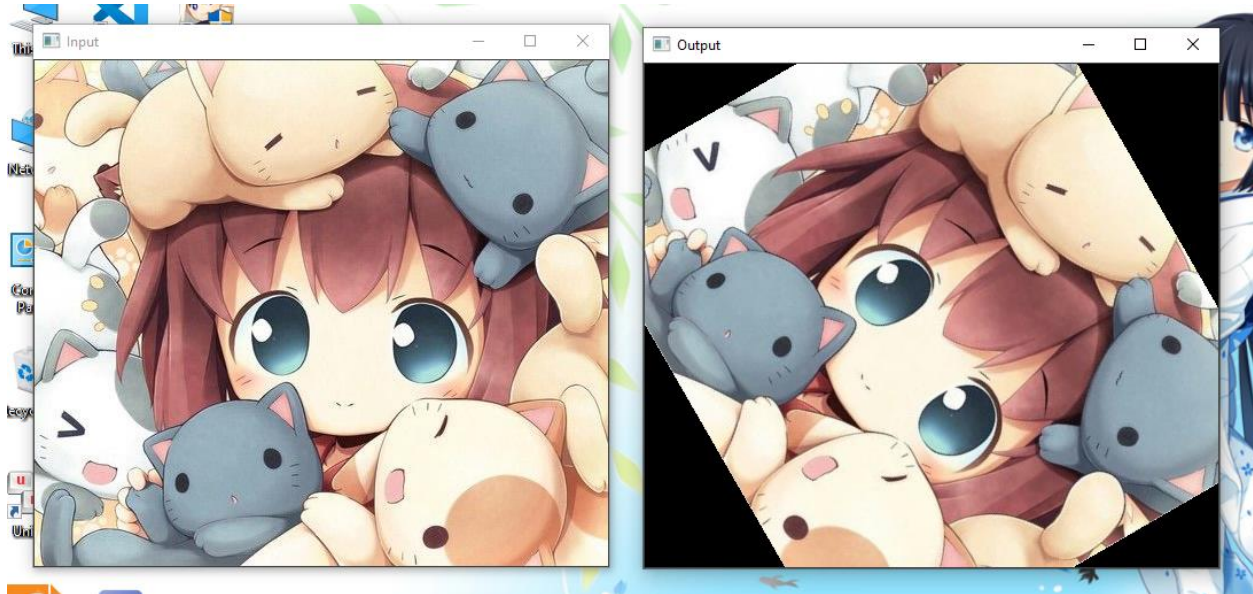
Ma trận tịnh tiến theo véc tơ OC để dịch về lại tâm ảnh ban đầu

- Chạy chương trình và kết quả:

Xoay ảnh với góc 60° với nội suy tuyến tính

```
E:\study\XLA\18127227_Lab02\x64\Debug>18127227_Lab02.exe -rotK E:\cat.jpg E:\out1.jpg -bl 60
```

Kết quả



6. Xoay ảnh mà giữ nguyên ảnh input không bị mất:

- Hướng dẫn cách chạy chương trình:

Command line:

<tên file.exe> -rotP <đường dẫn ảnh input> <đường dẫn ảnh output> <cách nội suy màu> <góc xoay với đơn vị độ>

Cách nội suy màu: -bl: nội suy tuyến tính, -nn: nội suy láng giềng gần nhất

- Giải pháp:

Ma trận biến đổi thuận xoay ảnh input theo góc quay đã được đổi qua rad để tìm vị trí tâm ảnh sau khi xoay $C'(x', y')$

Ma trận biến đổi nghịch được tạo bởi các ma trận:

Ma trận xoay với - góc xoay đã được đổi qua rad

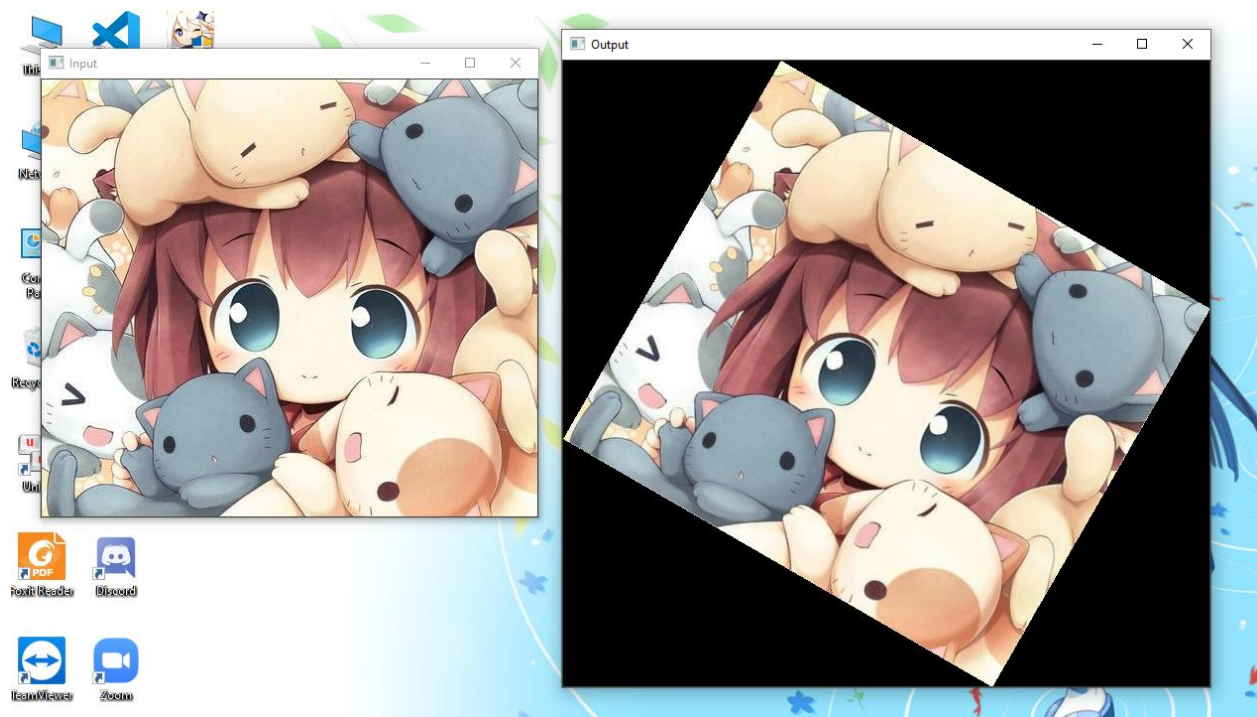
Ma trận tịnh tiến theo véc tơ $(-width\ output / 2 + x', -height\ output / 2 + y')$ để dịch về lại tâm ảnh ban đầu

- Chạy chương trình và kết quả:

Xoay ảnh với góc 30° với nội suy tuyến tính

```
E:\study\XLA\18127227_Lab02\x64\Debug>18127227_Lab02.exe -rotP E:\cat.jpg E:\out1.jpg -bl 30
```

Kết quả



7. Lật ảnh ngang:

- Hướng dẫn cách chạy chương trình:

Command line:

<tên file.exe> -flipV <đường dẫn ảnh input> <đường dẫn ảnh output> <cách nội suy màu>

Cách nội suy màu: -bl: nội suy tuyến tính, -nn: nội suy láng giềng gần nhất

- Giải pháp:

Ma trận biến đổi nghịch được tạo bởi các ma trận:

Ma trận tịnh tiến theo véc tơ $(0, \text{height} / 2)$ để dịch tâm ảnh nằm lên Ox

Ma trận scale theo tỉ lệ $(1, -1)$ để lấy đối xứng qua Ox

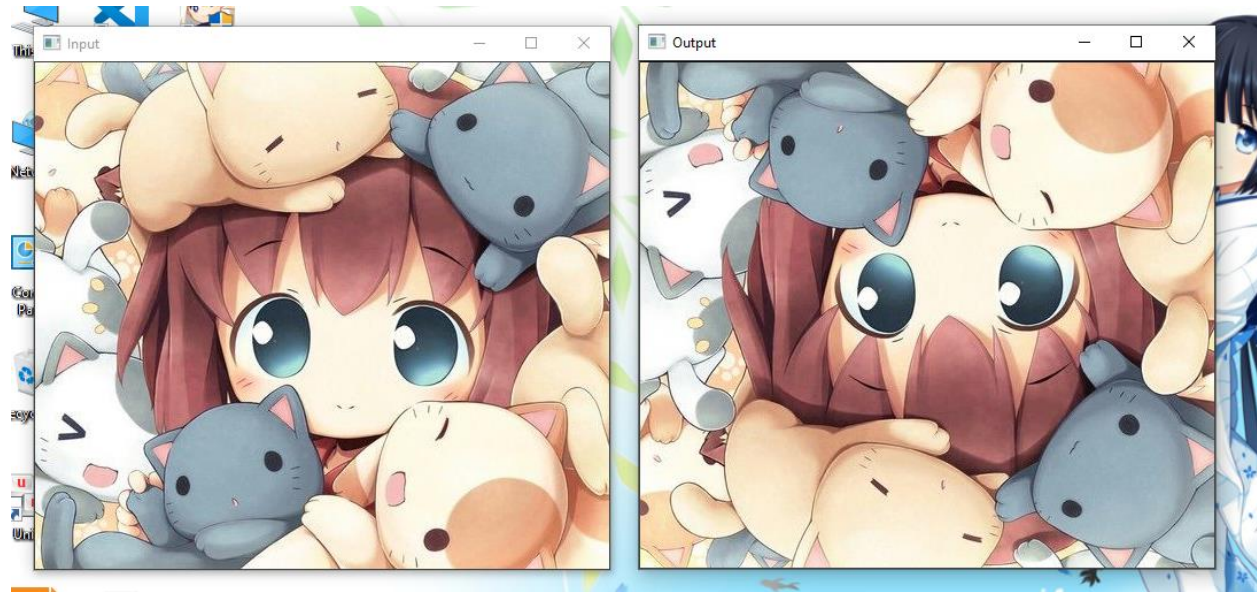
Ma trận tịnh tiến theo véc tơ $(0, -\text{height} / 2)$ để dịch lại tâm ảnh ban đầu

- Chạy chương trình và kết quả:

Lật ảnh ngang với nội suy tuyến tính

```
E:\study\XLA\18127227_Lab02\x64\Debug>18127227_Lab02.exe -flipV E:\cat.jpg E:\out1.jpg -bl
```


Kết quả



8. Lật ảnh dọc:

- Hướng dẫn cách chạy chương trình:

Command line:

<tên file.exe> -flipH <đường dẫn ảnh input> <đường dẫn ảnh output> <cách nội suy màu>

Cách nội suy màu: -bl: nội suy tuyến tính, -nn: nội suy láng giềng gần nhất

- Giải pháp:

Mã trận biến đổi nghịch được tạo bởi các mã trận:

Mã trận tịnh tiến theo véc tơ $(width / 2, 0)$ để dịch tâm ảnh nằm lên Oy

Mã trận scale theo tỉ lệ $(-1, 1)$ để lấy đối xứng qua Oy

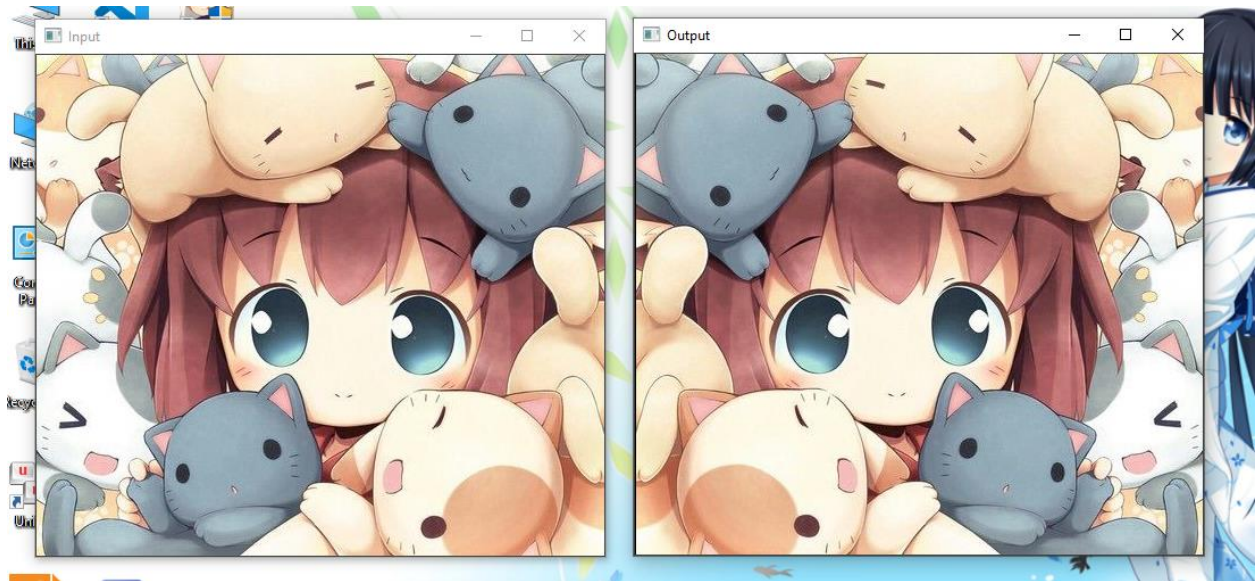
Mã trận tịnh tiến theo véc tơ $(-width / 2, 0)$ để dịch lại tâm ảnh ban đầu

- Chạy chương trình và kết quả:

Lật ảnh dọc với nội suy láng giềng gần nhất

```
E:\study\XLA\18127227_Lab02\x64\Debug>18127227_Lab02.exe -flipH E:\cat.jpg E:\out1.jpg -nn
```

Kết quả



III. Tài liệu tham khảo:

(1) Slide lý thuyết Image Pre-processing