# corr2

2-D correlation coefficient

## Syntax

```
r = corr2(A,B)
r = corr2(gpuarrayA,gpuarrayB)
```

## Description

`r = corr2(A,B)` returns the correlation coefficient `r` between `A` and `B`, where `A` and `B` are matrices or vectors of the same size. `r` is a scalar `double`.

`r = corr2(gpuarrayA,gpuarrayB)` performs the operation on a GPU. The input images are 2-D gpuArrays of the same size. `r` is a scalar `double` gpuArray. This syntax requires the Parallel Computing Toolbox™.

## Class Support

`A` and `B` can be numeric or logical. The return value `r` is a scalar `double`.

`gpuarrayA` and `gpuarrayB` must be real, 2-D gpuArrays. If either `A` or `B` is not a gpuArray, it must be numeric or logical and nonsparse. `corr2` moves any data not already on the GPU to the GPU. `R` is a scalar double gpuArray.

## Examples

### Compute the correlation coefficient

Compute the correlation coefficient between an image and the same image processed with a median filter.

```
I = imread('pout.tif');
J = medfilt2(I);
R = corr2(I,J)
```

```
R =

    0.9959
```

### Compute the Correlation Coefficient on a GPU

Compute the correlation coefficient on a GPU between an image and the same image processed using standard deviation filtering.

```
I = gpuArray(imread('pout.tif'));
J = stdfilt(I);
R = corr2(I,J)
```

```
R =

    0.2762
```

## More About

collapse all

**Algorithms**

CONTENTS

`corr2` computes the correlation coefficient using

$$r = \frac{\sum_m \sum_n (A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \overline{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \overline{B})^2\right)}}$$

where $\overline{A}$ = `mean2(A)`, and $\overline{B}$ = `mean2(B)`.

## See Also

`corrcoef` | `gpuArray` | `std2`