

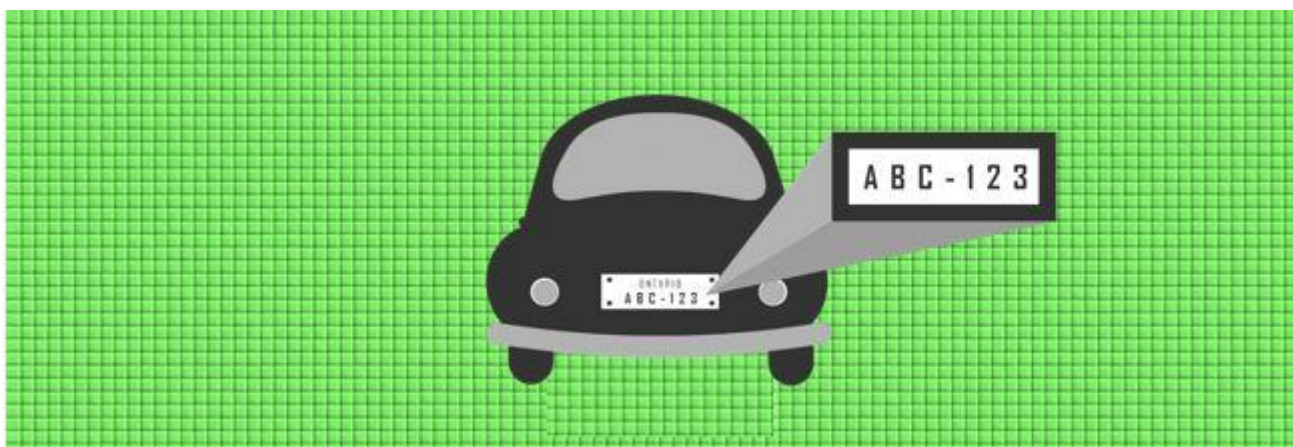
به نام خدا

پروژه دوم سیگنال و سیستم‌ها

استاد اخوان

نرگس غلامی

۸۱۰۱۹۸۴۴۷



سوال اول:

اگر **correlation** دو پارامتر با یکدیگر مثبت باشد، به این معناست که در فضایی که مطالعه و بررسی انجام شده، افزایش یک پارامتر با افزایش پارامتر دیگر و نیز کاهش آن پارامتر با کاهش پارامتر دیگر همراه است.

اگر **correlation** دو پارامتر با یکدیگر منفی باشد، به این معناست که در فضایی که مطالعه و بررسی انجام شده، افزایش یک پارامتر با کاهش پارامتر دیگر و کاهش آن پارامتر با افزایش پارامتر دیگر همراه است.

صفر بودن **correlation** به این معناست که دو پارامتر – در فضایی که مورد بررسی قرار گرفته – مستقل از یکدیگر بوده‌اند و بر اساس اطلاعات موجود از کاهش یا افزایش یکی، نمی‌توان در مورد کاهش یا افزایش دیگری اظهار نظر کرد.

correlation بین منفی یک و مثبت یک است. هر چه این ضریب از صفر دورتر شود (و به مثبت یا منفی یک نزدیک‌تر شود) می‌توان نتیجه گرفت که روند هم جهت بودن یا مخالف بودن دو پارامتر مورد بررسی، جدی‌تر است.

correlation هیچ ارتباطی با رابطه‌ی علت و معلول ندارد. احتمال دارد در یک جامعه‌ی آماری، بین حجم مصرف برق در اثر روشن کردن وسایل سرمایشی و دمای بالا، **correlation** مثبت وجود داشته باشد. اما این بدان معنی نیست که مثلاً اگر مصرف برق بیشتر شود یعنی دما بالا رفته است.

سوال دوم:

$$r = \frac{\text{sum}(\text{sum}(a.*b))}{\text{sqrt}(\text{sum}(\text{sum}(a.*a)) * \text{sum}(\text{sum}(b.*b)))}$$

سوال سوم:

همان‌طور که در مفهوم **correlation** اشاره شد، به همبستگی و میزان ارتباط دو متغیر اشاره دارد. ما در اینجا یک سری متغیرهای از پیش تعیین شده و سپس یک متغیر ورودی داریم. اگر همبستگی ورودی ما با یکی از متغیرهای از پیش تعیین شده نسبت به همبستگی با بقیه‌ی متغیرها بیشتر باشد و همچنین خود این مقدار همبستگی مقدار قابل توجهی باشد، می‌توان گفت که این دو متغیر یکی هستند و آن‌ها را با یکدیگر برابر قرار می‌دهیم.

```
function train = loadMap(mapDir)
    di = dir(mapDir);
    name_temp = {di.name};
    name = name_temp(3:end);
    train = cell(2, length(name));
    for i = 1:length(name)
        train(1, i) = {imread(['Map Set', '\', cell2mat(name(i))])};
        temp = cell2mat(name(i));
        train(2, i) = { temp(1)};
    end
end
```

در این تابع بالا همان طور که از اسمش پیداست ما در صدد لود کردن اطلاعات پوشه map set در متلب هستیم. به این صورت عمل می شود که بعد از آن که اطلاعات فایل های داخل این پوشه بدست آمد یک داده ساختار train درست می کنیم که عضو اول آن نشانگر عکس این فایل ها است و عضو دوم کاراکتر آن عکس است. مثلاً عضو اول تصویر عدد ۲ است و عضو دوم، خود کاراکتر ۲ است.

```
function [p, cc] = openPlate()
    [file, path] = uigetfile(['*.jpg;*.bmp;*.png;*.tif'], "Please choose an image");
    picture = imread([path, file]);
    [~, cc] = size(picture);
    p = imresize(picture, [300 500]);
end
```

در این قسمت یک عکس را که می خواهیم آن را آنالیز کنیم انتخاب می کنیم و آن را می خوانیم. سپس سایز آن را استخراج می کنیم و بعد آن را ریسایز می کنیم چرا که ممکن است سایز عکس های ورودی متفاوت باشند و با این کار تمام عکس های ورودی را یکسان سازی می نماییم.

```
function p = rgb2Bw(picture)
    p = rgb2gray(picture);
end
```

در این قسمت با این فانکشن عکس، سیاه و سفید می شود.

```
function p = imgClean(picture, cc)
    th = graythresh(picture);
    picture = ~imbinarize(picture, th);
    if cc > 2000
        p = bwareaopen(picture, 350);
    else
        p = bwareaopen(picture, 200);
    end
end
```

ابتدا برای تمیز کردن صفحه یک threshold در نظر می گیریم که رنگ های پس زمینه و کثیف کننده را از بین ببرد و بعد مکمل صفحه را به جای خود صفحه در خروجی می ریزد. سپس اگر سایز اولیه ی عکس از ۲۰۰۰ بیشتر بود اعضایی که تعداد پیکسل بهم پیوسته شان بیشتر از ۳۵۰ تا هست را حذف می نماید و اگر کمتر بود اعضایی که تعداد پیکسل بهم پیوسته شان کمتر از ۲۰۰ تا هست را حذف می کند.

```

function out = mapCorr(img, train)
    total = size(train, 2);
    x = [];
    for k = 1:total
        y = corr2(train{1, k}, img);
        x(end+1) = y;
    end
    if max(x) > .5
        out = find(x == max(x));
    else
        out = -1;
    end
end

```

در این مرحله، آن عکسی که در ورودی داده شده است را با ۶۴ عکس داخل پوشه map set (که الان در train قرار دارد) کورولیشن می‌گیریم و عکسی را که بیشترین مقدار همبستگی را با ورودی ما دارد انتخاب می‌کنیم. سپس اگر مقدار آن بیشتر از نیم بود ایندکس مورد نظر به خروجی داده می‌شود. در غیر این صورت مقدار منفی یک در خروجی قرار داده می‌شود.

اصل منطق برنامه:

```

train = loadMap('Map Set');
[picture, cc] = openPlate();
picture = rgb2Bw(picture);
p1 = imgClean(picture, cc);
imshow(p1)
[L, Ne] = bwlabel(p1);
final_license = [];
for n = 1:Ne
    [r, c] = find(L == n);
    patchCrop = p1(min(r): max(r), min(c): max(c));
    patchCrop = imresize(patchCrop, [42, 24]);
    out = mapCorr(patchCrop, train);
    if out ~= -1
        final_license = [final_license cell2mat(train(2, out))];
    end
end
final_license

```

ابتدا داده‌ی train لود می‌شود. بعد از آن عکسی که می‌خواهیم آن را آنالیز بکنیم را لود می‌نماییم. آن را سیاه و سفید می‌کنیم و سپس صفحه را از آبجکت‌های غیر عدد و حرف تا حد ممکن تمیز می‌کنیم. سپس هر کدام از اعداد و حروف را یک به یک جدا می‌کنیم و خروجی بیشترین مقدار همبستگی ورودی با اعداد و حروف پوشه map set را از تابع mapCorr دریافت می‌کنیم. اگر این مقدار برابر با منفی یک نبود یعنی جزو پلاک ما می‌باشد پس آن را در خروجی ثبت می‌کنیم. در آخر آرایه final_result نتیجه‌ی خروجی ما می‌باشد.



```
final_license = 'DL5CH8855'
```



```
final_license = 'DL2CAD0311'
```



```
final_license = 'UP14CB7145'
```



```
final_license = 'D9837'
```