

به نام خدا

## معرفی و نحوه استفاده از فیلتر PCA در متلب

### کلمات کلیدی:

پردازش تصویر - متلب - PCA.

### چکیده:

این مقاله شرح مختصری است بر فیلتر PCA که کاربرد زیادی در پردازش تصویر دارد. از این فیلتر برای کاهش ابعاد و استخراج ویژگی استفاده می شود. علاوه بر توضیحات تئوری، نحوه پیاده سازی این فیلتر در متلب نیز بیان شده است.

## مقدمه:

PCA از مبانی ریاضی مفصلی برخوردار است. به همراه این مقاله PDF دیگری نیز وجود دارد که مبانی ریاضی PCA را به طور کامل توضیح داده است. ولی به عنوان یک بینش کلی از مباحث ریاضی PCA باید گفت که در ریاضیات ثابت شده است که در یک فضای  $N$  بعدی می توان  $N$  بردار عمود بر هم تعریف کرد و هر نقطه از این فضا را توسط ضرب ضرایبی در این بردارهای عمود بر هم، نشان داد. به عنوان مثلاً در یک فضای ۲ بعدی ۲ محور عمود بر هم داریم و تمامی نقاط فضای ۲ بعدی توسط این ۲ محور قابل نشان دادن هستند همچنین برای فضای ۳ بعدی ... تا  $N$  بعدی. PCA در حقیقت بردارهای ویژه یا همان بردارهای عمود بر هم و مقادیر ویژه یعنی ضرایب این بردارها را به دست می آورد. ترتیب قرار گیری این بردارهای عمود بر هم به صورتی است که اولین مولفه PCA (اولین بردار) بیشترین اطلاعات را در خود دارد و این روند تا آخرین مولفه دنبال می شود و مولفه آخرین کمترین اطلاعات را در خود جای داده است. در زیر در ابتدا یکی از فصول یک پایان نامه را آورده شده و پس از آن به بیان عملی و ساده PCA پرداخته و در آخر نحوه پیاده سازی آن را در متلب توضیح داده شده است.

## PCA : (این قسمت از یک پایان نامه استخراج شده است)

در سیستم Face Detection ارائه شده برای استخراج مشخصه های موجود در یک پنجره

ورودی و کاهش حجم داده های این پنجره از PCA استفاده می شود . در صورتی که یک تصویر

$40 \times 40$  را در نظر بگیریم برای نمایش این تصویر ۱۶۰۰ پیکسل مورد نیاز است . در صورتی که

مستقیماً از این تصویر ورودی برای بخش کلاس بندی و تصمیم گیری استفاده کنیم ، حجم

محاسباتی بسیار زیاد خواهد شد و منجر به کاهش کار آیی سیستم از لحاظ سرعت خواهد شد .

بنابراین قبل از اعمال پنجره استخراج شده به بخش تصمیم گیری در جستجوی روش هایی هستیم که

این حجم را تا حد امکان کاهش دهد و در عین حال راندمان کلی سیستم نیز تا حد بالایی حفظ گردد.

PCA روشی است که پاسخگوی این نیازها می باشد. با استفاده از این آنالیز مشخصه های مهم یک

تصویر استخراج می شوند که حجم این مشخصه ها بسیار کمتر از حجم داده های اصلی می باشد .

در زیر به توضیح این روش می پردازیم .

### ۱-۴-۳- تعریف PCA

ایده اصلی PCA کاهش ابعاد داده های موجود در یک مجموعه داده می باشد بگونه ای که

تغییرات موجود در این داده ها تا حد امکان حفظ گردد. مولفه های اصلی PC ها تبدیل های خطی از

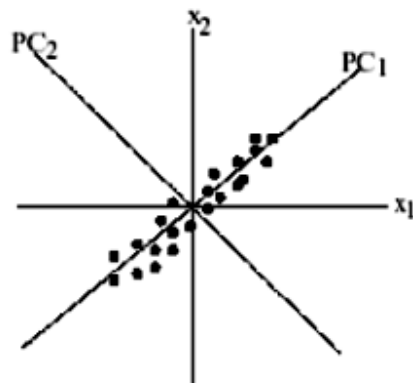
مجموعه اصلی متغیر ها می باشند . PC ها غیر همبسته هستند و به گونه ای مرتب می شوند که چند

PC اول بیشترین تغییرات موجود در مجموعه داده ها را در خود جای می دهند.

تعبیر هندسی اولین PC به این صورت است که می توان آنرا به عنوان یک محور مختصات

جدید در نظر گرفت که تغییرات داده های تصویر شده روی این محور مختصات حداکثر است . شکل

۱-۳ نمودار پراکنندگی مربوط به یک مجموعه داده دو بعدی را نشان می دهد.



شکل ۱۰-۳- نمودار پراکندگی داده های موجود در یک مجموعه داده. دومولفه اصلی که از تحلیل PCA

بدست آمده اند با  $PC1$  و  $PC2$  نشان داده شده است. داده ها در راستای محور  $PC1$  بیشترین تغییر را دارند. این نقاط یک شکل بیضی را نشان می دهند. اولین PC در جهت محور اصلی این بیضی می باشد (در شکل با  $PC1$  نشان داده شده است). دومین PC ( $PC2$ ) عمود بر اولین PC می باشد. در صورتی که نقاط داده، روی اولین PC تصویر شوند، بیشترین تغییرات داده های دو بعدی را می توان بصورت یک بعد نمایش داد.

## ۲-۴-۳- PCA و تصویر های ویژه<sup>۱</sup>

فرض کنیم یک تصویر سیاه و سفید  $N \times N$  و شدت روشنایی  $I(x,y)$  داشته باشیم که هر عنصر موجود در این تصویر با یک مقدار ۸ بیتی که بیانگر شدت در این نقطه است نشان داده می شود در این صورت  $I(x,y)$  را می توانیم بصورت یک بردار با طول  $N^2$  یا یک نقطه در فضای  $N^2$  بعدی در نظر بگیریم. بنابراین یک تصویر  $128 \times 128$  را می توان بصورت یک تک نقطه در فضای  $16384$  بعدی نمایش داد. تصاویر مربوط به چهره در حالت کلی، تنها بخش کوچکی از این فضای بزرگ را اشغال می کنند و بنابراین بصورت بهینه در این سیستم مختصاتی نشان داده نمی شوند.

روش تصاویر ویژه با این فرض کار می کند که تصاویر مربوط به چهره انسان یک ناحیه متصل به هم و ساده را در فضای تصاویر تشکیل می دهند. پیدا کردن یک دستگاه مختصات بهینه برای تصاویر چهره انسان از طریق PCA یا تحلیل مولفه های اصلی صورت می پذیرد. دستگاه مختصات بهینه در این تعریف ، دستگاهی است که تغییرات مربوطه به مشخصه های صورت در طول محوره های آن حداکثر گردد.

چگونه می توانیم سیستم بهینه را برای تصاویر چهره انسان پیدا کنیم ؟ مولفه های این مختصات بهینه عمود بر یکدیگر خواهند بود و همانطور که گفته شد واریانس یا تغییرات در مجموعه تصاویر چهره های انسان را حداکثر می سازند. بنابراین نیاز به روش هایی برای ارزیابی واریانس ها در مجموعه تصاویر صورت داریم و این همان چیزی است که توسط بردارهای ویژه می توانیم به آن برسیم .

بداشتن یک ماتریس  $C$  بردارهای ویژه  $U_i$  و مقدارهای ویژه  $\lambda_i$  مربوط به  $C$  معادله ۳-۶ را برآورده می سازند.

$$Cu_i = \lambda_i u_i \quad \forall i \quad (3-6)$$

بردارهای ویژه عمود برهم هستند و بصورت نرمالیزه در می آیند بنابراین :

$$\mu_i^T \mu_j = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases} \quad (3-7)$$

ماتریس  $I_K$  را ماتریسی تعریف می کنیم که عناصر آن متناظر باشندت های سطح خاکستری

موجود در تصویر چهره باشد. فرض کنیم این ماتریس  $k$  امین عنصر از مجموعه  $M$  عضوی چهره های موجود در Database باشد. حال فرض کنیم  $I_k$  برداری باشد که از ستونی کردن  $I_k$  بدست آمده است. همچنین  $\phi_k$  را بصورت بردار ستونی نرمالیزه شد و میانگین برای چهره  $k$  تعریف می کنیم، یعنی:

$$\psi = \frac{1}{M} \sum_{i=1}^M \tau_i$$

$C$  را ماتریس کو واریانس چهره های نرمالیزه شده میانگین تعریف می کنیم.

$$C = \frac{1}{M} \sum_{k=1}^M \phi_k \phi_k^T$$

در معادله فوق  $M$  تعداد چهره های موجود در Database می باشد. تصاویر موجود در

Database کمک می کنند زیر فضایی بسازیم که به بهترین وجه تصاویر موجود را شاخصه سازی

می کنند. این زیر فضا از این پس زیر فضای چهره نامیده می شود. با استفاده از معادله ۴-۳ خواهیم

داشت:

$$Cu_i = \lambda_i u_i$$

$$u_i^T Cu_i = u_i^T \lambda_i u_i$$

$$= \lambda_i u_i^T u_i$$

$$u_i^T Cu_i = \lambda_i$$

$$u_i^T u_i = 1 \text{ چون}$$

$$\lambda_i = \frac{1}{M} u_i^T \sum_{k=1}^M \phi_k \phi_k^T u_i$$

$$= \frac{1}{M} \sum_{k=1}^M u_i^T \phi_k \phi_k^T u_i$$

$$= \frac{1}{M} \sum_{k=1}^M (u_i^T \phi_k)^T (u_i^T \phi_k)$$

$$= \frac{1}{M} \sum_{k=1}^M (u_i^T \phi_k)^2$$

محمد نحوی

$$= \frac{1}{M} \sum_{k=1}^M (u_i^T \tau_k - \text{mean}(u_i^T \tau_k))^2$$

$$= \text{var}_k (u_i^T \tau_k)$$

مقدار ویژه  $i$  واریانس تصاویر چهره های موجود در طول محوری است که توسط بردار ویژه

$i$  توصیف می گردد. بنابراین با انتخاب بردارهای ویژه متناظر با بزرگترین مقدارهای ویژه ابعادی را انتخاب می کنیم که بیانگر بزرگترین واریانس ها در تصاویر چهره هستند.

### ۳-۴-۳- بازسازی چهره ها<sup>۱</sup> با استفاده از ضرائب PCA

با بدست آوردن بردارهای ویژه و مقادیرهای ویژه می توان هر چهره موجود در Database را

تنها با استفاده از چند ضریب نمایش داد. برای بدست آوردن این ضرائب، کفایت تصویر جدید ورودی را روی هر یک از بردارهای ویژه ( بردارهای ویژه متناظر با بزرگترین مقادیرهای ویژه ) تصویر کنیم. برای تصویر ورودی  $\tau_o$  داریم:

$$\omega_k = u_k^T (\tau_o - \psi) \quad (3-9)$$

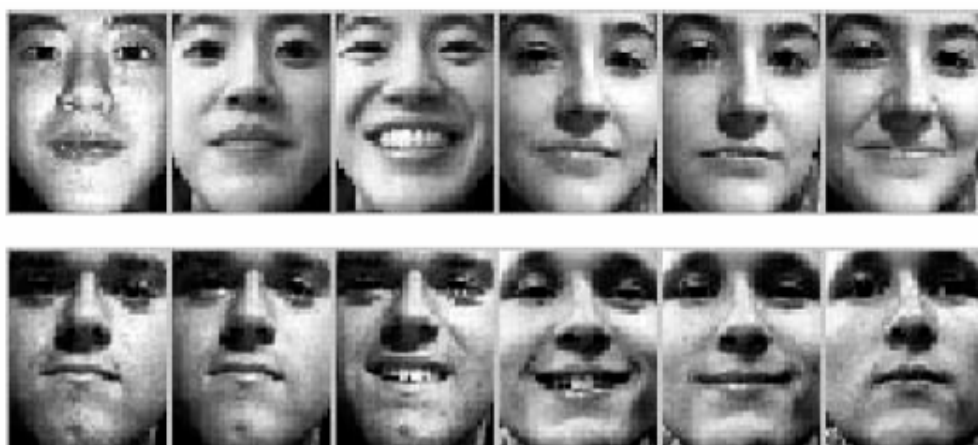
در صورتیکه  $L$  بردار ویژه را در نظر بگیریم و تصویر ورودی  $\tau_o$  را می توان با بردار زیر نمایش داد. در معادله فوق  $L$  تعداد بردارهای ویژه انتخاب شده می باشد. ( $L \ll M$ )

به عنوان مثال فرض کنیم مجموعه ای متشکل از ۱۲ تصویر با ابعاد  $50 \times 40$  داشته باشیم. این

تصاویر در شکل ۱۱-۳ نشان داده شده اند.

با استفاده از PCA تصاویر ویژه را می توان بدست آورد. در شکل ۱۲-۳ تصاویر مربوط به ۶

تصویر ویژه که دارای مقادیرهای ویژه بزرگتری هستند نشان داده شده است.



شکل ۱۱-۳- تصاویر موجود در Database

در این مثال برای نمایش هر تصویر موجود در Database از ۶ ضرایب استفاده می کنیم. به عنوان مثال برای نمایش اولین چهره موجود در Database ، ماتریس ضرایب ( $\Omega$ ) به صورت زیر می باشد.



شکل ۱۲-۳- تصاویر ویژه مربوط به بزرگترین مقدارهای ویژه (از چپ به راست)

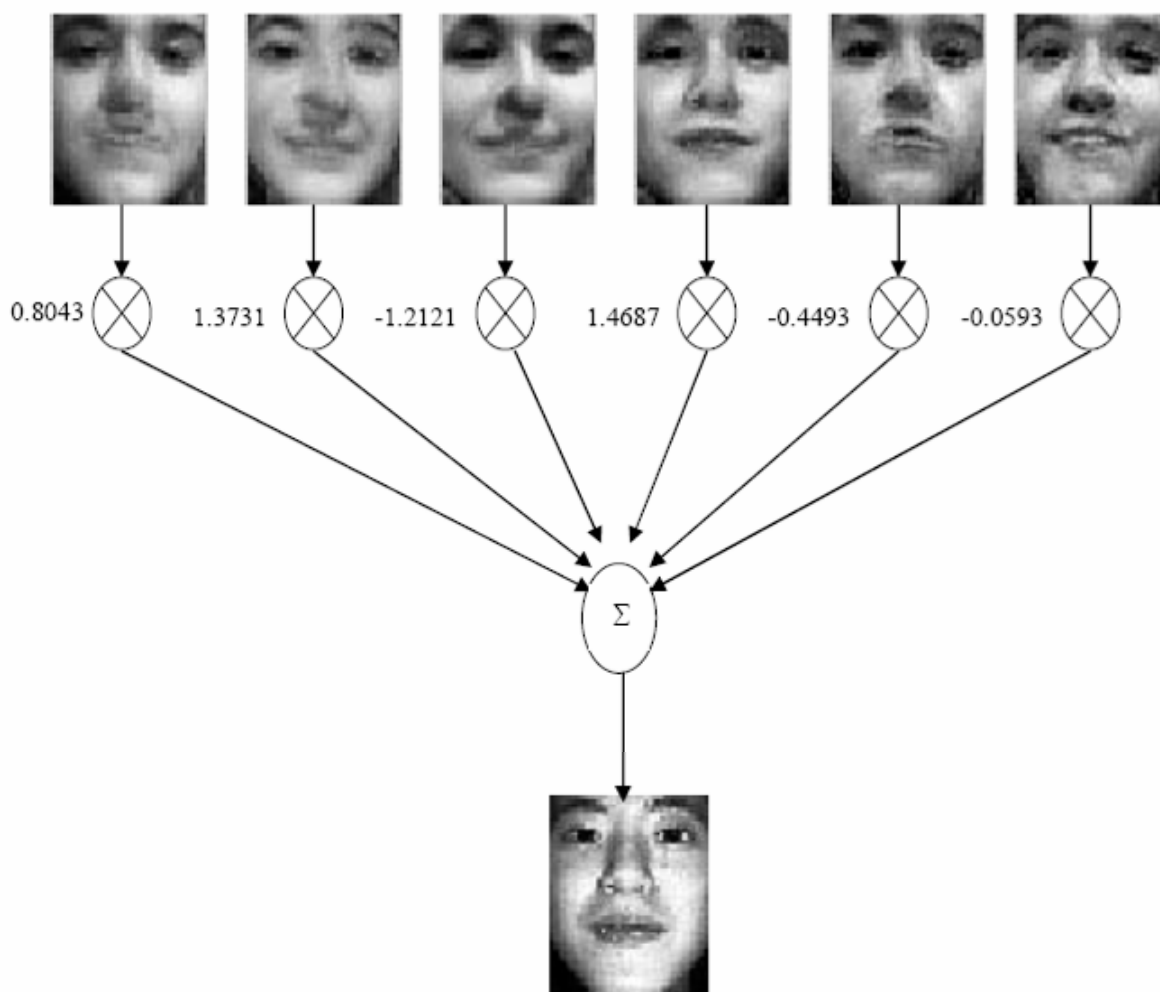
$$\Omega = [0.8043 \quad 1.3731 \quad -1.2122 \quad 1.4687 \quad -0.4493 \quad -0.0593].$$

با استفاده از این ضرایب تصویر شماره ۱ موجود در Database را می توان بصورت یک ترکیب خطی از ضرایب بالا نوشت.

$$Pic1 = 0.8043 \times \text{EigenFace1} + 1.3731 \times \text{EigenFace2} + (-1.2122) \times \text{EigenFace3} + 1.4687 \times \text{EigenFace4} + (-0.4493) \times \text{EigenFace5} + (-0.0593) \times \text{Eigenface6}.$$



در عبارت بالا Pic1 نشان دهنده تصویر شماره ۱ و EigenFace تصویر ویژه می باشد. شکل ۳-۱۳ تصویر بازسازی شده توسط این ضرایب را نشان می دهد.



شکل ۳-۱۳-تصویر بازسازی شده توسط ضرایب PCA

مقدار خطای ناشی از بازسازی را بصورت زیر تعریف می کنیم.

$$\varepsilon = \|\phi - \phi_r\|^2 \quad (3-11)$$

در معادله ۳-۱۱،  $\varepsilon$  خطای ناشی از بازسازی،  $\phi$  تصویر اصلی و  $\phi_r$  تصویر باز سازی شده می باشد. شکل ۳-۱۴ تصویر اصلی، تصویر باز سازی شده و تصویر خطا را نشان می دهد. مقدار خطای بازسازی، با استفاده از معادله ۳-۱۱ برابر است به ۴۹،۶۷۳.



شکل ۱۴-۳ (از چپ به راست) تصویر اصلی، تصویر باز سازی شده، تصویر خطا

## PCA، استخراج ویژگی و کاهش افزونگی

یکی از مهمترین روش ها برای کاهش افزونگی جهت تسهیل پردازش داده و طبقه بندی تصاویر، استفاده از تحلیل مؤلفه های اساسی، PCA است. این تبدیل به طور گسترده در آنالیز داده جهت کاهش ابعاد استفاده می شود. تبدیل PCA در عین حال که ابعاد داده را کاهش می دهد عمده اطلاعات داده اصلی را نیز در بر دارد.

فرض کنید  $M$  بردار  $T_1, T_2, \dots, T_M$  بردارهای تصاویر مجموعه داده های آموزشی هستند. هر بردار ۱۶۰۰ عضو دارد بردار میانگین  $A$ ، و اختلاف بردارهای ویژگی از بردار میانگین،  $X_m$ ، از روابط (1) و (2) محاسبه می شوند.

$$A = \frac{1}{M} \sum_{m=1}^M T_m \quad (1)$$

$$X_m = T_m - A \quad 1 < m < M \quad (2)$$

اگر ماتریس  $Y$  را توسط رابطه (3) تعریف کنیم، ماتریس کواریانس  $C$  به کمک رابطه (4) محاسبه خواهد شد.

$$Y = [X_1 \ X_2 \ \dots \ X_M] \quad (3)$$

$$C = \frac{1}{M} \sum_{m=1}^M X_m X_m^T = \frac{1}{M} Y Y^T \quad (4)$$

ماتریس کواریانس، یک ماتریس  $N \times N$  است و می توان به روش PCA،  $K$  بردار ویژه مهم تر ماتریس  $C$  را بدست آورد.

بردارهای ویژگی  $N$  بعدی بر اساس  $K$  بردار مشخصه، به فضای  $K$  بعدی نگاشت می شود.

## PCA در متلب: (متلب به حروف کوچک و بزرگ حساس است)

نحوه پیاده سازی PCA در متلب به روشی است که در بالا گفته شده است. تصاویری که می خواهیم PCA را بر آنها اعمال کنیم تعداد ۱۰۴ تصاویر با ابعاد  $50 \times 55$  هستند. در زیر ابتدا این تصویر را می خوانیم و در database ذخیره می کنیم: (توجه کنید باید تمام این تصویر در شاخه ای باشند که m-file را از آنجا اجرا کرده اید)

```
close all;
clear all;
for i=1:104
    tmp=int2st(i);
    database(:, :, i)=imread(tmp, 'tif');
end
```

حال باید این تصاویر  $50 \times 55$  را به بردارهای با ابعاد  $1 \times 2750$  تبدیل کنیم. کد زیر این عمل را انجام داده و بردارهای کل تصاویر را در یک ماتریس با ابعاد  $104 \times 2750$  ذخیره می کند.

```
for i=1:104
    vectordata(i, :)=reshape(database(:, :, i), 1, 2750);
end
```

حال میانگین تمام بردارها را که یک ماتریس  $1 \times 2750$  است را به دست می آوریم:

```
Mvectors=mean(vectordata, 1);
```

حال تک تک بردارها را از این میانگین کم کرده و در یک ماتریس ذخیره می کنیم (توجه کنید خط اول این کد به این خاطر است که متغیر vectordata از نوع تصویر می باشد برای اینکه بتوانیم اعمال ریاضی را بر روی آن انجام دهیم به ناچار باید آن را تبدیل به double کنیم. برای تبدیل دوباره آن به تصویر می توان از دستور uint8 استفاده کرد.):

```
vectordata=double(vectordata);
for i=1:104
    AVGmatrix(i, :)=Mvectors-vectordata(i, :);
end
```

حال باید کواریانس این ماتریس را به دست آوریم:

$CVData = cov(AVGmatrix);$

حال PCA آن را حساب می کنیم:

$[Fpca \ 1 \ 2] = pcacov(CVData);$

ماتریس مورد نظر ما اولین ماتریس با نام  $Fpca$  است. ابعاد این ماتریس  $2750 \times 2750$  است ولی چون ویژگی های اصلی در چند سطر یا ستون اول قرار دارد ما فقط ۵۰ سطر یا ستون آن را انتخاب میکنیم. (باید توجه داشت که ویژگی ها ممکن است در ۵۰ سطر اول یا ۵۰ ستون اول قرار گرفته باشند یعنی یا باید یک ماتریس با ابعاد  $50 \times 2750$  (سطری) انتخاب شود یا یک ماتریس با ابعاد  $2750 \times 50$  (ستونی) انتخاب شود) ما در اینجا ۵۰ داده ستونی را بر می گزینیم:

$MFpca = Fpca(:, 1:50);$

علامت ' در پایان پرانتز به معنی عملیات ترانهاده است یعنی ماتریس با ابعاد  $2750 \times 50$  را به ماتریس  $50 \times 2750$  تبدیل می کند.

حال ضرایب PCA آماده هستند در این مثلا این ضرایب ( منظور ۵۰ مولفه آن است) در ماتریس  $MFpca$  قرار دارند.

حال برای به دست آوردن ضرایب PCA یک تصویر خاص باید ماتریس  $MFpca$  را در آن تصویر ( منظور بردار آن تصویر است) ضرب کرد.

در این مثال بردار تمامی تصاویر در ماتریس  $vectordata$  قرار دارند. ما در کد زیر ضرایب PCA را در تصویر اول (سطر اول ماتریس بالا) ضرب میکنیم و ضرایب PCA آن را به دست می آوریم:

$Mpca1 = MFpca * vectordata(1,:);$

که یک ماتریس  $50 \times 1$  خواهد شد.

معمولا این ماتریس ( $Mpca1$ ) به عنوان ورودی شبکه های عصبی و ... به عنوان ماتریس ویژگی و کاهش ابعاد یافته ماتریس اول به کار می رود.

از کد های زیر نیز برای اضافه کردن وبکم به برنامه میتوانید استفاده کنید:

```
close all
clear all
obj = videoinput('winvideo', 1);
set(obj, 'FramesPerTrigger', 100000)
start(obj);
b=getdata(obj,1);
stop(obj)
delete(obj)
[m,n]=size(b(:, :, 1));
figure(1);
imshow(b)
%-----
obj = videoinput('winvideo', 1);
set(obj, 'SelectedSourceName', 'input1')
src_obj = getselectedsource(obj);
get(src_obj)
preview(obj);
```