



# Project 1

PRIMARY VISUAL CORTEX

Alireza Shirzad-Mahdi Sedighi-Alireza Kazemi | Computational Neuro-Science | 2018

## Part 1

### QUESTION 1

مسئله که از مهم ترین مباحث عملی Neuro Science می باشد بطن این مقاله را تشکیل می دهد . صورت Stimulus ، کد کردن تصویر به زبان نورون هاست . نورون ها معمولا به یک ویژگی ، المان و عنصر خاص تصویر یا به طور کلی Encode حساس می باشند . هنر آن است که این ویژگی ها را تشخیص بدھیم تا بتوانیم تصویر نشان داده شده به چشم را Encode کنیم . به خصوص با در نظر گرفتن این حقیقت که مغز ما به شدت غیر خطی و پیچیده است . هدف این مقاله آن است که یک تصویر را به نحوی کاراکتریزه کنیم تا بتوان با توجه به شدت یا نحوه ی رفتار آن کاراکتر خاص ، خروجی را کد کنیم . این المان ها به طور کلی به دو دسته ای "مرتبط" و "بی ارزش" تقسیم می شوند . دسته ای "مرتبط" کاراکتر ها و عناصری از تصویر هستند که باعث Trigger کردن و Spiking یک نورون خاص می شوند . این دسته همچنین به طور افزایشی (additive) عمل میکنند یعنی همانطور که به تنها یاباعث response می شوند درکنار کاراکترهای دیگر نیز تا حدودی در عمل contribution ، spiking دارند . در مقابل ، دسته ای دوم یعنی المان های "بی ارزش" قرار دارند . این کاراکتر ها معمولا تاثیری در Spiking Rate یک نورون به خصوص ندارد . بعضا حتی تاثیر منفی (Suppressive) نیز دارد . حال چالش مهم این است که چگونه یک تصویر را کاراکتریزه کرد ؟ در این مقاله به دنبال این هستیم که مناسب ترین و Optimum ترین پایه ها در فضایی  $n$ -بعدی به نحوی تعریف کنیم که بتوانیم :

1\_ هر نورون را مشخص کنیم . Receptive Field \_ 1

2\_ هر تحریک دلخواهی را بتوانیم با آن پایه ها Encode کنیم .

تفاوت عمده ی این پژوهش با نمونه های قبلی در این بود که یک قدم پا را فراتر گذاشته و به Mean Value(First Order) بستنده نکرده و تحلیل خود را براساس Variance(Second Order) و متده PCA انجام داده است . در مقالات قبلی برای تحلیل وضعیت یک نورون صرفا محاسبه می شد اما در این مقاله سعی بر ساختن پایه ای برای تحلیل هرگونه ورودی شده تا ورودی هایمان را نسبت به نزدیکی به آن محور های اصلی ( $PC_1, PC_2, \dots$ ) بسنجیم . البته تمرکز این مقاله روی نورون های پیچیده (در سئوال بعدی توضیح داده شده) است لذا روش STA عملاً در اکثر اوقات بی فایده است . اما در پژوهش هایی که کار با نورون های ساده را دنبال می کنیم کماکان این روش پاسخگو است .

## QUESTION 2

نورون های پیچیده در واقع نورون هایی هستند که در تعریف نورون ها ساده گنجانده نمی شوند. لذا بهتر است معیار هایی برای ساده بودن نورون ارائه دهیم. به طور کلی هر چه رفتار یک نورون خطی تر باشد، مدل های پیشنهادی برای آنالیز رفتار آن نورون ساده تر و قابل فهم تر می باشد. بسیاری از سیستم ها در علوم اعصاب بسیار به خطی بودن نزدیک می باشند به همین دلیل آن ها را سیستم های "خطی\_غیر خطی" می نامیم. ( An Introduction to Computational Neuroscience March 16, 2007 - Todd Troyer )

( خطی بودن یک نورون بدین معناست که از اصل برهمنهی<sup>4</sup> بیرونی کند بدین معنا که :

$$r(s) = r(c_1 s_1 + \dots + c_N s_N) = c_1 r(s_1) + \dots + c_N r(s_N)$$

برای یک Visual Receptive Field خطی بودن بدین شکل است :

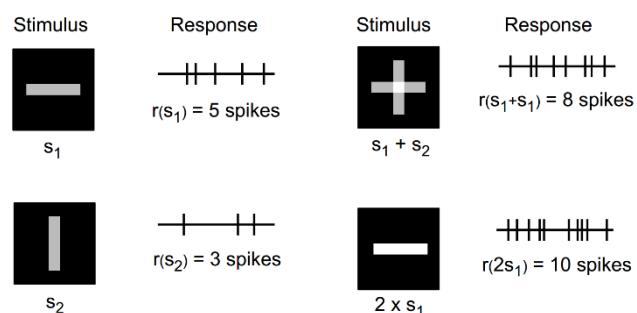


Figure1

سلول های پیچیده خود تابعی غیر خطی از سلول های ساده هستند. به عنوان مثال سلول های V1 و V2 ورودی های خود را از سلول های لایه های پایین تر مانند LGN و سنسور های نوری شبکیه می گیرند و این عملکرد به شکل سلسله مراتبی ادامه پیدا می کند [Palmer, Stephen E. (1999). Vision Science: Photons to Phenomenology. Cambridge: The MIT Press]

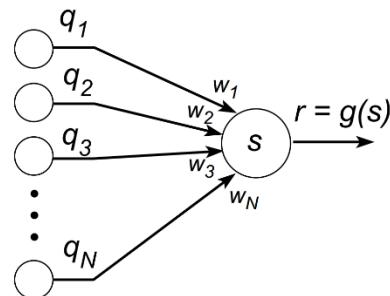


Figure2

در سال 1962، Hubel و Wiesel طبق مشاهدات اولیه‌ی خود، 4 ویژگی و معیار برای نورون‌های ساده ارائه دادند. این ویژگی‌ها بدین صورت است:

۱- مناطق ON و OFF کاملاً جدا

۲- خاصیت جمع پذیری در هر ناحیه

۳- عملکرد مخالف نواحی ON و OFF

۴- قابلیت پیش‌بینی پاسخ نورون با دانستن نواحی تقریبی Receptive Field ها

به طور دقیق در این مقاله داریم:

۱ \_ Spatially separate ON and OFF region

۲ \_ Summation within each region

۳ \_ Antagonism between ON and OFF subregions

۴ \_ Response properties can be predicted from receptive field map

تعریف دیگری در این مقاله برای پیچیده بودن و ساده بودن ارائه شده که از مقاله‌ی [Skottun.] برداشت شده است. طبق این تعریف اگر در حالت optimum (یعنی حالتی که نورون به ورودی حساس شده است) نسبت دامنه‌ی هارمونیک اول به دامنه‌ی میانگین ورودی بزرگتر از یک باشد، نورون مورد نظر ساده می‌باشد. همچنین سلول‌های ساده با روش STA قابل تحلیل هستند اما برای سلول‌های پیچیده نمی‌توانیم از این روش استفاده کنیم.

### QUESTION 3

ما به دنبال این هستیم که بفهمیم سیستم ما به چه نوع ورودی ای حساس است. برای بررسی و بدست آوردن این کمیت می‌بایست با روش‌های random (چرا؟)، نورون را تحریک کنیم و در لحظات spike برگردیم و بینیم چه در Stimulus گذشته است. در این آزمایش ما دو متغیر از نوع Time و Spatial داریم. متغیر Time ما 16 Frame را در 16 msec عرض می‌کنیم و متغیر Spatial ما نیز 16 مقدار را به خاطر وجود 16 نوار اختیار می‌کند. تصاویر با سرعت  $16 \text{ frames}/268 \text{ msec}$  عرض می‌شوند ( $f = 60 \text{ Hz}$ ). نحوه‌ی عرض شدن روشنایی هر نوار در هر frame به صورت random و به کمک یک "دنباله‌ی دودویی" شبیه تصادفی با طول حداقل "تولید می‌شود. این روش متدهای بسیار جالبی برای تولید اعداد Random است که در مسائل مخابرات دیجیتال، ساختن بهینه‌ی f-MRI و ... از آن استفاده می‌شود. درباره‌ی تولید این دنباله از اعداد در بخش صرفاً جهت اطلاع، توضیح داده شده است. قبل از هر Spike روشنایی هر نوار در هر frame، یکی از خانه‌های ماتریس  $16 \times 16$  را پر می‌کند. پس هر تحریک 256 مولفه دارد یعنی ما می‌توانیم تحریک‌ها را در فضایی 256 بعدی map کنیم و مقدار روشنایی هر bar را به عنوان

مقدار در محور مشخص کنیم . اسم هر نقطه را  $S(i)$  میگذاریم . هر  $S(i)$  را میتوان با 256 مقدار در فضای تحریک یا یک ماتریس  $\times 16$  مشخص کرد . بر اساس این نقاط ماتریس Correlation را به شکل زیر می سازیم .

$$Cm,n = \frac{1}{N} \sum_{i=1}^N Sm(i) \times Sn(i)$$

که  $S_m$  و  $S_n$  ویژگی  $m$  و  $n$  تصویر قبل از Spike می باشد .

در این مقاله از روش تحلیل مولفه های اصلی (PCA) استفاده شده . این روش برای کاهش ابعاد فضای نمونه ای ما انجام می گیرد تا مولفه های نامرتب ، از تحلیل ما حذف و مولفه های تاثیر گذار تبدیل به پایه های فضای نمونه ای بشوند . در این روش فضای نمونه ای را در نظر می گیریم . مشاهده می کنیم که کدام جهت در این فضا ماقریم واریانس داده ها را در بر میگیرد . یعنی تصویر داده ها در کدام سمت ، ماقریم کشیدگی را دارد . این جهت ها را از ماقریم واریانس تا به مینیمم واریانس ردیف می کنیم . این مقادیر واریانس همان مقادیر ویژه ماتریس Correlation هستند . حال از بین بزرگ ترین واریانس ها چند نمونه ی بسیار بزرگ را تشخیص می دهیم . ملاک بزرگ بودن مقادیر معمول واریانس های ماتریس های کنترل است . درباره ماتریس کنترل در سوال بعد صحبت شده . با استفاده از این واریانس ها یک ناحیه اطمینان ساخته می شود و هر مقدار ویژه ای خارج این محدوده "بسیار بزرگ" یا "Significant" شناخته می شود . حال بردارهای ویژه ی متناظر با مقادیر ویژه ی بسیار بزرگ را به عنوان پایه های فضای نمونه ای خود انتخاب میکنیم و بدین صورت محور های "Relevant" مشخص می شوند . با این کار تعداد زیادی از 256 پایه را عملأً دور میریزیم زیرا آن محور ها اطلاعات تازه ای به ما نمی دهند و نقاط فضای تحریک ما به آن محور ها حساسیتی ندارند .

### صرفه جهت اطلاع :

دنباله ی شبه رندوم ما بین شکل تولید می شود . یک Random Generator با استفاده از یک "شیفت رجیستر با فیبیک" ساخته می شود . علت شبه تصادفی بودن این اعداد این است که اعداد تولید شده توسط یک شیفت رجیستر قاعدها محدود است و باید از جایی به بعد تکرار شود لکن طول این دنباله آن قدر زیاد (در این مقاله 32767) است که می توان آن را تصادفی در نظر گرفت . این سیستم با یک مقدار اولی که آن را "هسته" می نامند شروع به کار میکند و با توجه فیبیک مدار شروع به تولید یک دنباله می کند . دنباله های معروفی اعم از فیبوناچی را می توان با این سیستم تولید کرد . مثالی از این سیستم را می توان در شکل زیر مشاهده کرد :

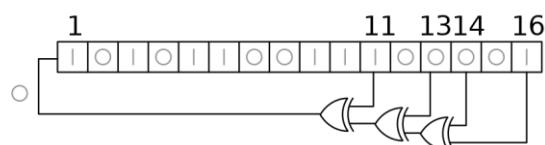


Figure 2  
A 16-bit "Fibonacci" LFSR

## QUESTION 4 :

برای تحریک ها مستقل از اسپایک ، مقاله ماتریس کنترل را طراحی کرد . بدین شکل که به تعداد تحریک های به شرط اسپایک ، تحریک هایی با زمان رندوم تولید می کنیم . سپس یا استفاده از همین تحریک ها ، همان کاری را که برای ساختن ماتریس Correlation کردیم را انجام می دهیم و ماتریس Control را می سازیم . حال محدوده ای اطمینان مقادیر ویژه را از روی همین ماتریس بدست می آوریم . در کل هر جا که به دنبال تمايز بین تحریک های به شرط اسپایک و تحریک های مستقل از اسپایک بودیم از این دنباله از تحریک ها و ماتریس کنترل استفاده می کنیم .

## PART 2 :

### QUESTION 1 :

هر فایل log شامل اطلاعات تحریک آن آزمایش است که عبارتند از :

[FileInfo] : که شامل اطلاعات فایل دیتا می باشد اعم از نام ، نوع ، تاریخ ، سازنده و ... فایل می باشد .

[TestInfo] : که شامل اطلاعات آزمایش ما می باشد اعم از

[Refresh Rate] : فرکانس Refresh شدن مانیتوری است که روپروری گر به قرار گرفته و در صورت مقاله 120Hz ذکر شده .

[Frame Rate] : فرکانس عوض شدن frame ها است که برای تحلیل sao ها و تناظر اسپایک ها با ماتریس msqid بسیار مهم است .

هچنین آزمایش در این فایل ذکر شده که Resolution زمانی ما 0.1 میلی ثانیه است .

و اطلاعات دیگری که در ادامه پیش مذکور خواهد بود مانند Tuning Curve آن آزمایش .

### QUESTION 2 :

در این سؤال تابعی طراحی میکنیم که کد نورون را دریافت می کند و یک Struct شامل object هایی که به عنوان آزمایش های sqid مطرح شده اند . حاوی بردار زمان اسپایک نورون ها و Header مربوط به هر آزمایش می باشد را به عنوان خروجی تحویل می دهد . فولدر نورون ها ، توابع مورد نیاز Func\_ReadData (fget\_spk , fget\_hdr) ، تابع fget\_spk و بدنه ای اصلی کد همگی

در یک دایرکتوری قرار می‌گیرند . حال به شرح کد می‌پردازیم : Struct خروجی کد ما شامل ۲ field می‌باشد به نام های "Header" و "Events" که به ترتیب بردار زمان اسپایک های نورون و Header مربوطه می‌باشند که آن ها است . ابتدا با دستور dir محتويات پوشه‌ی نورون مورد نظر را میخوانیم و یک struct برای این مشخصات اعم از اسم فایل ، تاریخ فایل ، نوع فایل و ... دریافت میکنیم . حال ما به اندازه‌ی object های این Struct آزمایش sqid داریم که باید آن ها را تحلیل کنیم . حال با گذاشتن سه شرط که فایل مورد نظر حاوی کد نورون ما باشد ، حاوی پسوند sao باشد و از جنس sub نباشد ، آن را تحلیل میکنیم . چالش ما در این کد این بود که نمیخواستیم در هر فolder تابع fget\_spk و fget\_hdr را داشته باشیم و تنها در فolder بیرونی این دو را داشته باشیم لذا با دستور copyfile این دو را در هر اجرای تابع درون کد کپی می‌کنیم و در انتهای اجرای تابع این دو را حذف می‌کنیم تا محتويات پوشه‌ی نورون تغییر نکند . پس از کپی این دو تابع ، دایرکتوری خود را با دستور cd تغییر می‌دهیم و وارد فolder می‌شویم . را روی آزمایش مورد نظر اجرا میکنیم و به عنوان value در struct fget\_spk و fget\_hdr را دارند این صورت تابع ما کامل می‌شود .

### QUESTION 3 :

برای راحتی کار تابع دیگری تعریف کردیم به نام spike\_count\_rate که با گرفتن کد هر نورون به عنوان ورودی string اسپایک زدن هر نورون را با آزمایش های msqd.sao به ما به عنوان خروجی می‌دهد . این تابع در ابتدا با صدا زدن تابع Func\_Readdata اطلاعات هر نورون را استخراج می‌کند ، سپس اندازه بردار های events آزمایش های مختلف نورون را با هم جمع می‌کند و تقسیم بر تعداد آزمایش (سایز خروجی Sunc\_Readdata) و سپس تقسیم بر زمان آزمایش ها می‌کند . زمان آزمایش ها یکسان نیست اما بسیار نزدیک به هم هستند و برای تشخیص نورون هایی که spike count rate زیر ۲ دارند این دقت کافی است . لذا تایم آزمایش را برابر ۵۴۸.۷۲۷۶ s قرار می‌دهیم . حال در بدنه‌ی اصلی کد تابع spike\_count\_rate را در یک حلقه برای تمامی نورون ها اجرا میکنیم . خروجی زیر بدست آمد :

000413.b03	000413.b04	000413.b05	000418.a01	000420.bo2	000524.co1	000907.fo7
------------	------------	------------	------------	------------	------------	------------

این ۷ نورون spike count rate کمتر از ۲ دارند لذا در ادامه‌ی پروژه با آن ها کار نخواهیم کرد .

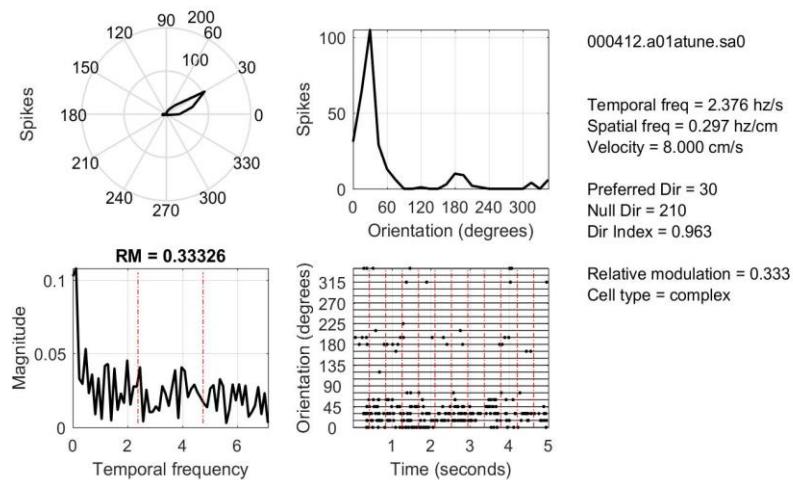
### QUESTION 4 :

برای این تابع علی رغم صورت سؤال ۳ ورودی تعریف کردیم . فرکانس ، بردار زمان اسپایک ها و بردار تحریک کلی . در ابتدا باید یک تناظر بین زمان هر اسپایک و frame در ماتریس تحریک پیدا کنیم . بدین منظور تمام درایه های بردار events را تقسیم بر فرکانس آن آزمایش کردیم و برای جلوگیری از بوجود آمدن اعداد اعشاری از آن ceil گرفتیم . می‌توانستیم floor بگیریم اما منطقی است که نورون به frame قبل از اسپایک خود حساس باشد نه به frame قبل از اسپایک . حال بردار events جدید ما حاوی شماره

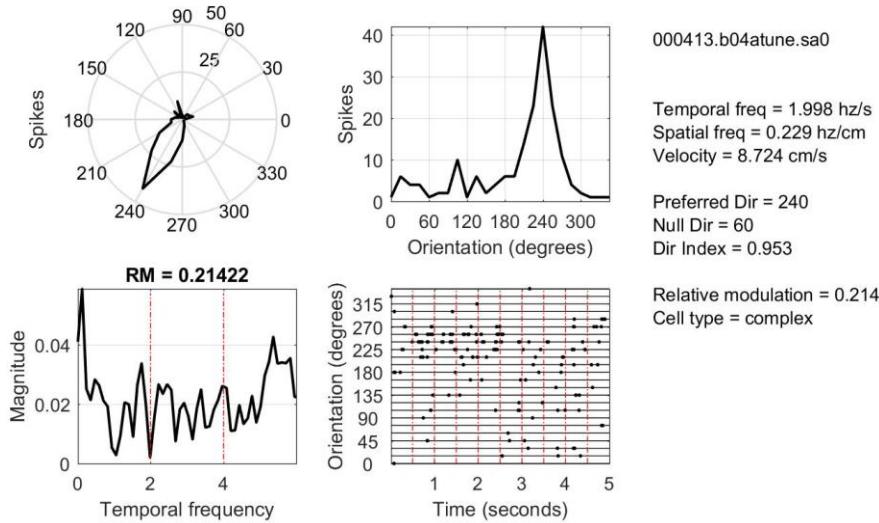
ای است که نورون در آن لحظه اسپایک زده . حال با نوشتن یک حلقه `for` قبل از `spike frame` را گرفتیم و در یک صفحه از یکی ماتریس 3 بعدی قرار دادیم . این کار را برای کل درایه انجام می دهیم تا یک ماتریس سه بعدی بدست آوریم که هر صفحه ای آن متعلق به یک اسپایک است که 256 مولفه در آن صفحه باعث اسپایک زدن آن شده . در یک حلقه `for` کلی این کار را برای تمام آزمایش های آن نورون کردیم و با دستور `cat` این حجم های سه بعدی را به هم وصل کردیم تا یک ماتریس سه بعدی حاوی تحریک تمام اسپایک ها در تمام آزمایش ها بدست آوریم .

## QUESTION 5 :

تابع `tview.m` نام فایل `Tuning Response` به تحریک `Tune` را در ورودی دریافت می کند و در خروجی نتایج آزمایش `000412.a01` را به ما می دهد . به عنوان مثال برای نورون `000412.a01` داریم :



همانطور که مشاهده می کنید این نمودار ها بیانگر این هستند که نورون ما در چه جهتی ماقزیم `Spike` را دارد . یعنی ، ما تصاویری در جهت های مختلف برای گربه پخش کردیم و در این `Orientation` مخصوص نورون ما بیشترین `Spike Rate` را داشت . لذا آزمایش های بعدی ما برای بدست آوردن `Orientation` همگی در این `Receptive Field` انجام می شوند . برای نورون بالا این جهت حدودا 30 درجه است . یا به عنوان مثال برای نورون `000413.b04` داریم :



که برای این نورون راستای مورد علاقه ، حدود 240 درجه می باشد .

## part 3:

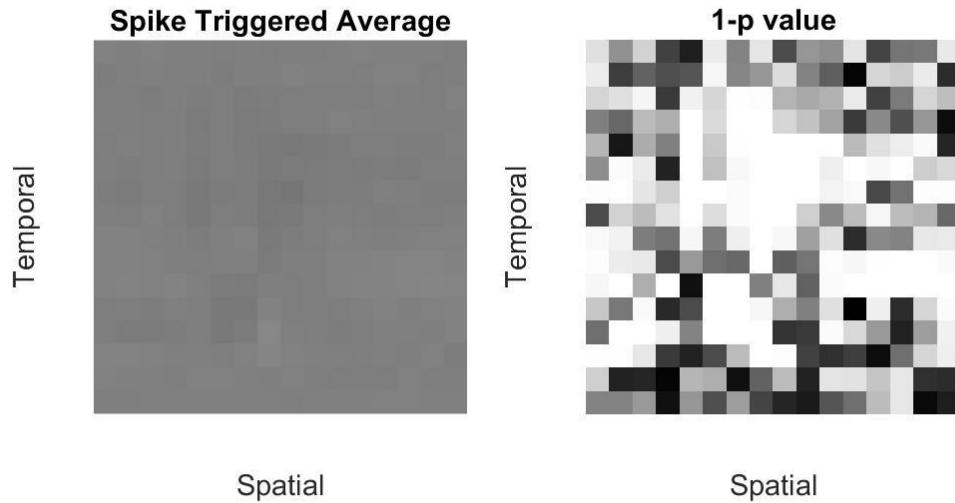
### QUESTION 1 :

با دستور mean روی بعد سوم ماتریسی که توسط تابع Func\_StimuliExtraction استخراج شد ، یک ماتریس 16 در 16 به عنوان ماتریس STA (Spike Triggered Average) بدست آوردیم . حال این ماتریس را با دستور imshow و با مشخصت کردن High و low پیکسل ها نمایش می دهیم . می توانستیم imtool شکل را با دستور Contrast تغییر دهیم منتها حساسیت Receptive Field بیش از اندازه می شد . در ضمن برای تک تک نورون ها مطابق با تصویر خودش باید این کار را انجام می دادیم که کاری طاقت فرسا بود لذا از انجام این کار پرهیز کردیم .

### QUESTION 2 :

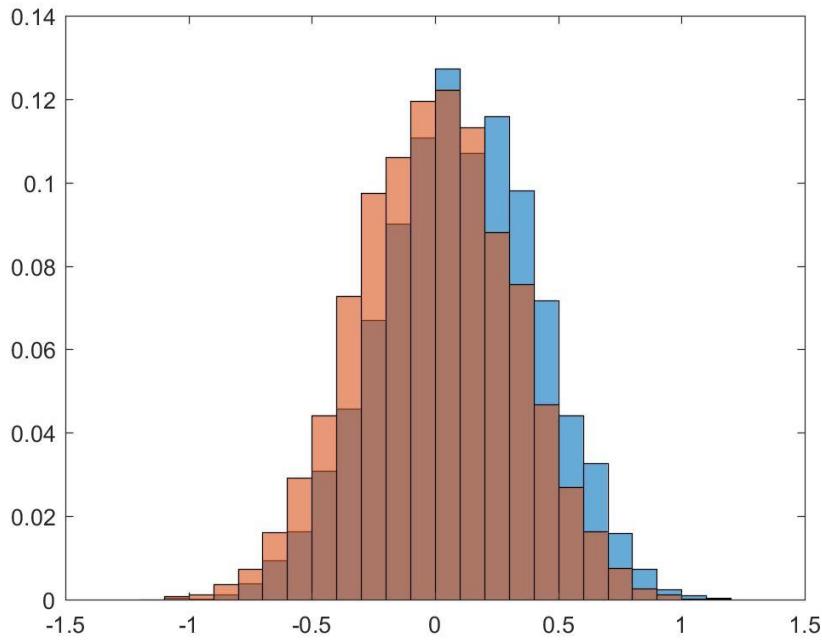
در این مسئله می بایست روی تک تک درایه های STA آزمون فرضیه را اجرا کنیم . بدین صورت با دو for روی درایه های ماتریس 6x16 تابی p\_value حرکت می کنیم و هر درایه را با اعمال ttest روی آن درایه ، در تمام ارتفاع ماتریس (بعد سوم) پر می کنیم . یعنی یک بردار از درایه های یک پوزیشن در صفحه 16x16 در تمام spike ها را بدست می آوریم و روی آن ttest فرض میکند که داده های ما میانگین صفر با توزیع گوسی دارند و این را به عنوان Null-Hypothesis STA می پنیرد . ما در واقع ادعای جدید ماست . اگر خروجیتابع را به شکل یک بردار دوتایی بگیریم ، درایه اول نتیجه تست را به ما می دهد یعنی آیا فرض اولیه reject شده یا خیر . و درایه دوم این بردار p\_value بردار را به ما می دهد . حال ماتریس بدست آمده را در کنار ماتریس STA می بینیم .

با دستور `imshow` و `subplot` می کشیم . برای تست کردن کد ، نورونی را امتحان می کنیم که نمونه‌ی آن در گزارش کار پروژه آمده است . می بینیم که پیکسل به پیکسل بر آن منطبق است .



### QUESTION 3 :

با داشتن STA ما برداری در فضای تحریک داریم . برداری که يحتمل تصویر هر بردار تحریک روی آن ، مرحله اول فیلترینگ نورون ماست . یعنی به اندازه تصویر بردار ها روی آن ما انتظار rate داریم . برای تصویر کردن بردار ها دیگر روی STA ، به مفهوم ضرب داخلی که در اینجا همان ضرب درایه به درایه هست نیاز داریم . پس از ضرب مقادیر ماتریس بدست آمده را با هم جمع می کنیم . عدد حاصل را به برداری که قرار است histogram را پیشیم اضافه میکنیم . برای بدست آوردن ماتریس کنترل ابتدا برداری با زمان های اسپایک random اما به اندازه جمع برداری های events روی هم تولید می کنیم . سپس همان کارهایی را که برای بردار های events واقعی انجام داریم ، روی این انجام می دهیم . تا در نهایت به همان ماتریس سه بعدی برسیم . سپس ضرب داخلی صفحات این ماتریس را نیز با STA می گیریم و آن را با حالت normalization و probability هیستوگرام می کنیم . برای اطمینان از صحت کد ، نورونی که در دستور کار آمده را مورد آزمایش قرار می دهیم که شکل زیر بدست می آید و پیکسل به پیکسل با هیستوگرام اصلی برابر است .



#### QUESTION 4 :

تابع ttest را روی دو برداری که آن ها را هیستوگرام کردیم ، اجرا می کنیم . نتیجه زیر حاصل می شود :

Result = 1

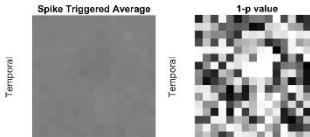
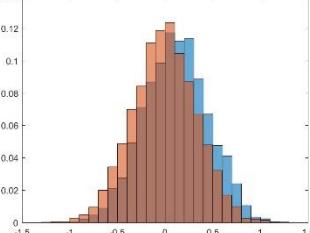
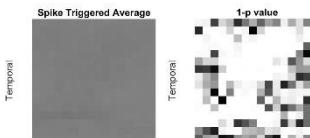
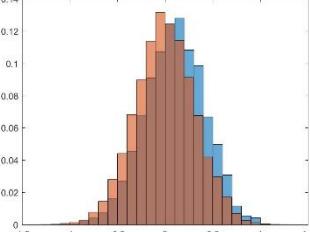
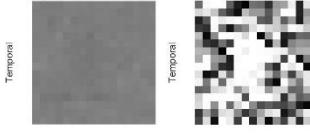
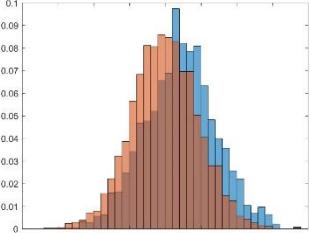
P\_value = 1.3144e-137

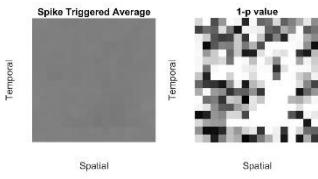
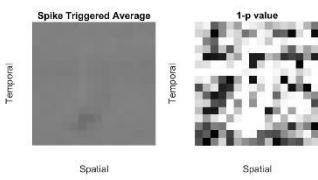
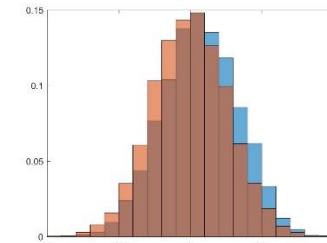
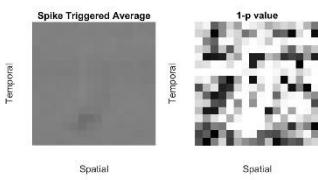
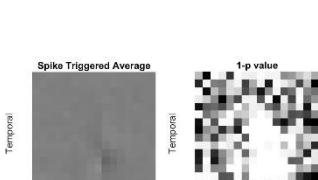
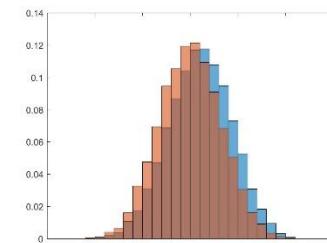
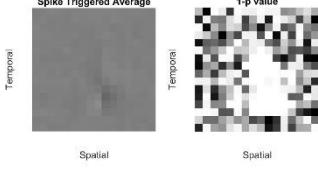
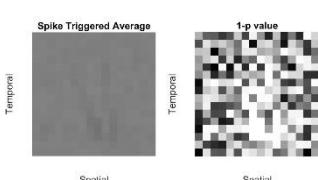
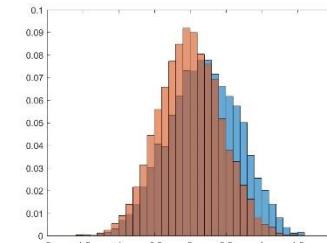
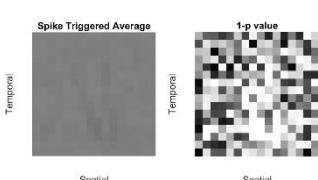
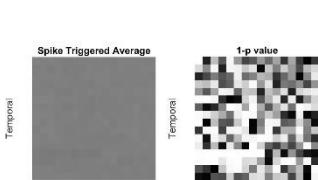
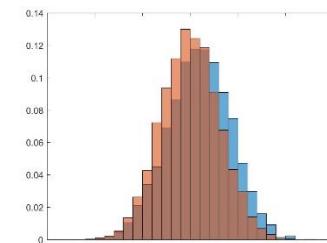
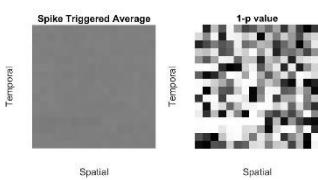
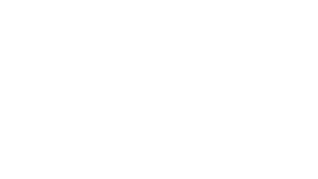
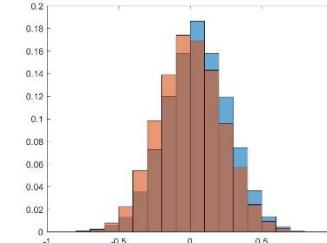
حال به تفسیر ttest می پردازیم . دو برداری که تحت test قرار گرفتند ، اندازه هی تصویر نقاط منجر شده به اسپایک و تصویر نقاط رندوم در فضای تحريك می باشد . با این تست می خواهیم مشخص کنیم آیا ارتباط نقاط منجر به اسپایک با بردار اتفاقی است یا اینکه تفاوت بین دو دسته دیتا ، کاملاً معنی دار است . فرض بر آن است که داده های ما رندوم است و هیچ فلسفه هی معنی داری ندارد (Null\_Hypothesis) . p\_value یه ما می گوید احتمال اینکه فرض اول ما درست باشد و کماکان چنین نتیجه ای بگیریم چقدر است . این عدد بسیار بسیار کوچک است و در مقابل Significant Level ما که مقدار بحرانی را مشخص می کند قابل مقایسه نیست لذا فرض اول که مبنی بر رندوم بودن داده ها بود ، رد می شود و از صحت نتیجه گیری خود اطمینان حاصل می کیم . خروجی 1 نیز بدین معناست که فرض اول ما رد شده . در همان اول با دیدن هیستوگرام می شد به صورت سرانگشتی گفت که میانگین داده های ناشی از اسپایک بیشتر از داده های رندوم ماست لذا داده ها تفاوت معنادار دارند اما آن نتیجه گیری غلط بود . میانگین معیار درستی برای چنین نتیجه گیری ای نیست لذا داده های بسیار برت می توانند میانگین

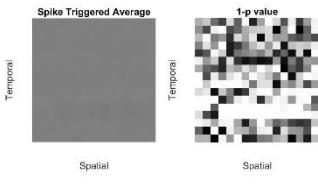
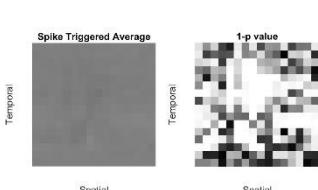
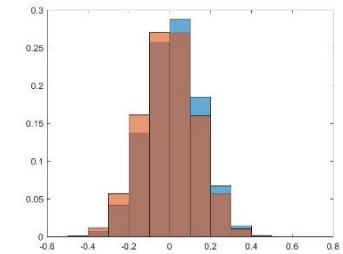
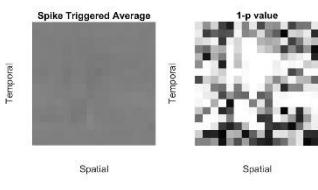
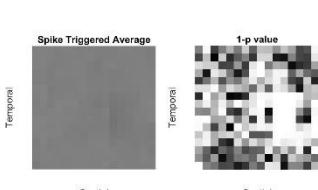
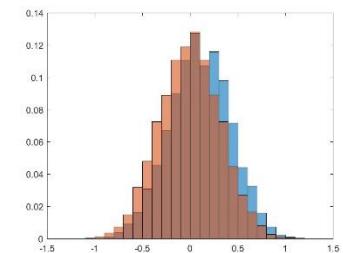
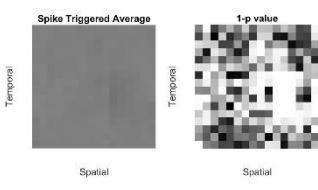
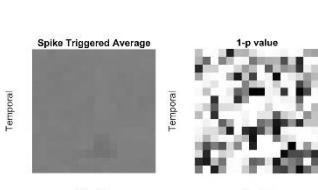
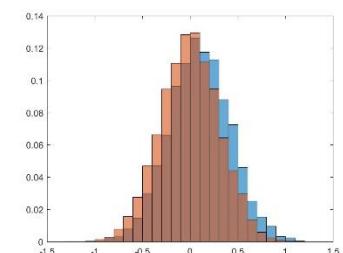
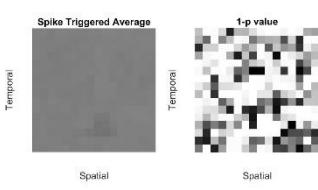
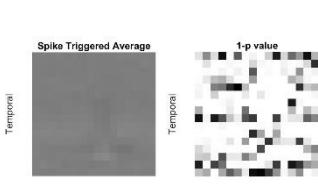
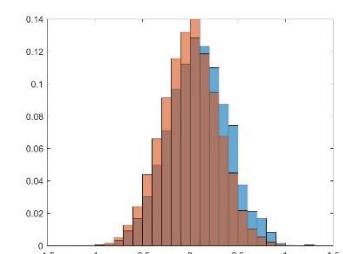
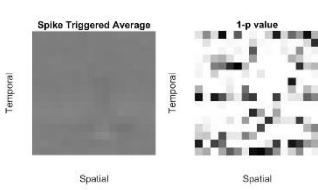
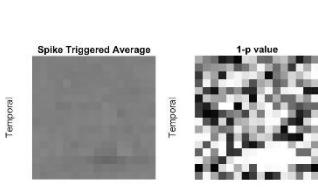
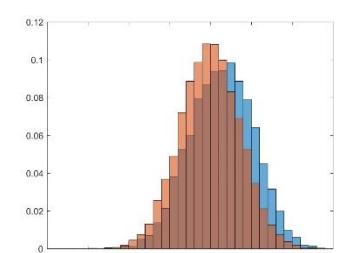
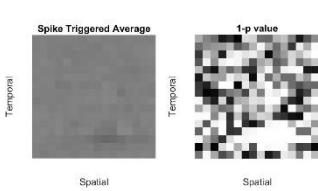
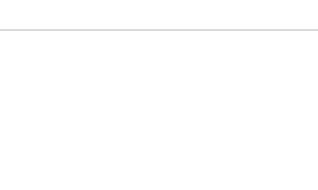
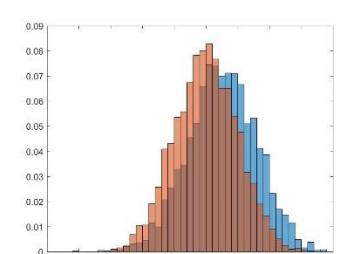
را تغییر دهنده اما کمکان معنی خاصی ندارند . لذا حتما می بایست student t-test را انجام پدھیم تا از تفاوت معنادار داده هایمان مطمئن شویم .

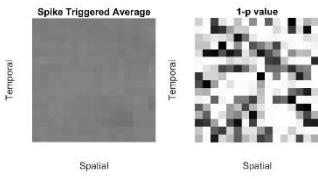
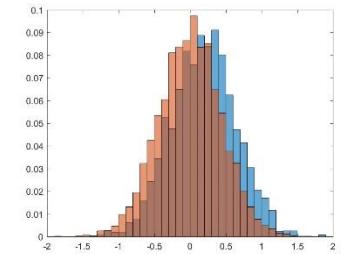
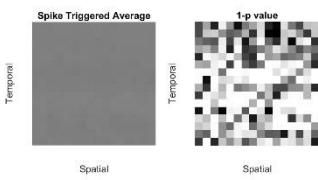
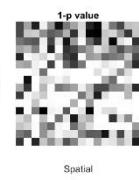
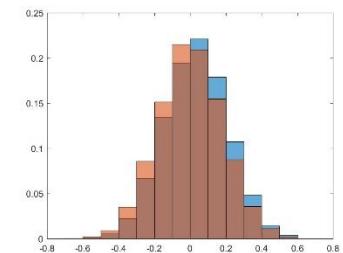
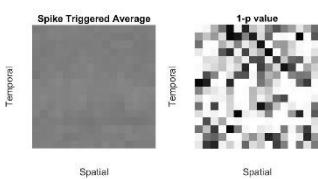
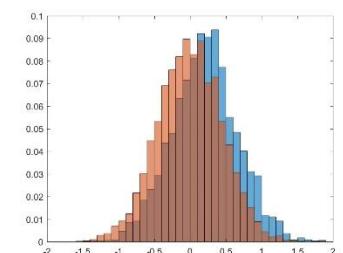
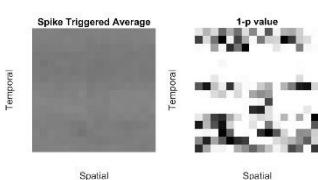
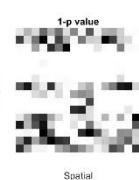
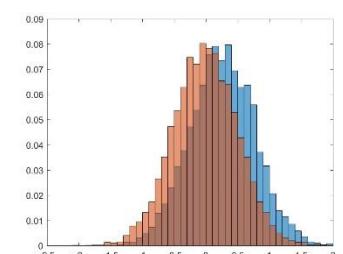
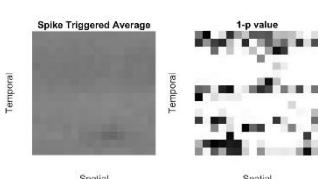
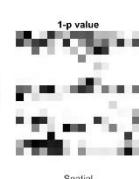
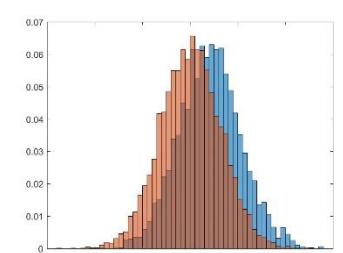
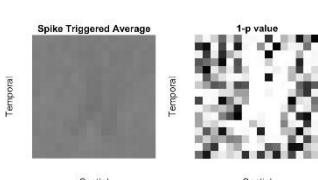
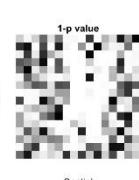
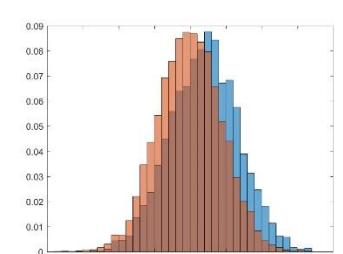
## QUESTION 5 :

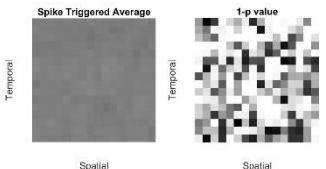
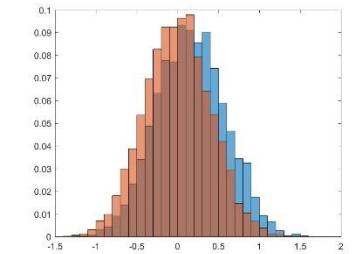
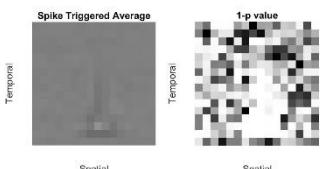
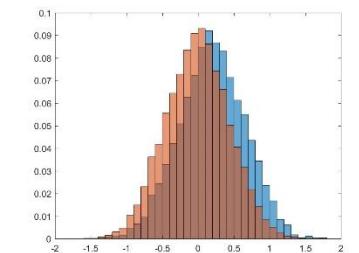
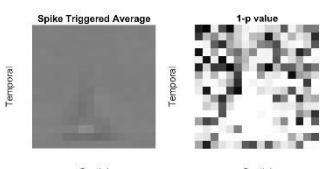
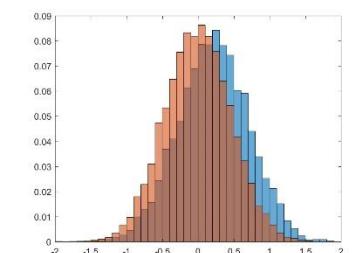
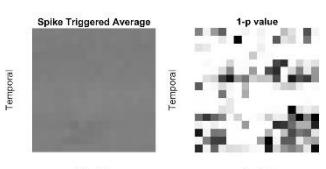
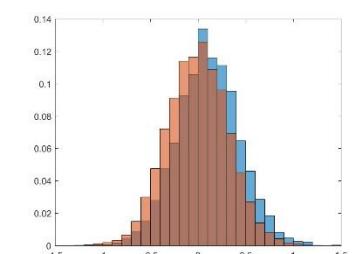
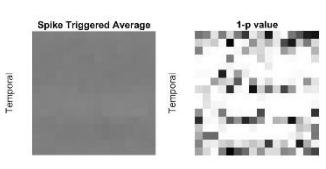
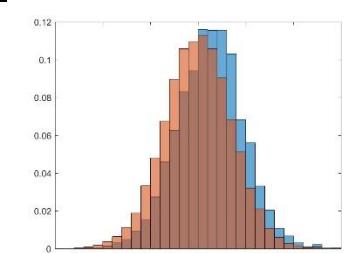
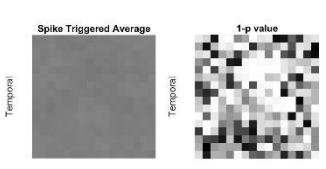
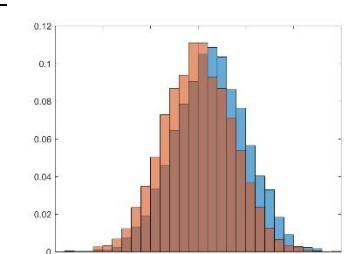
اگر توریع دو بردار اندازه تصویر تحریک ها روی STA را روی یک گاوی fit کیم ، دو گاوی میبینیم که با هم overlap دارند . دقیقا محل تقاطع دو گاوی ، جایی است که احتمال اینکه یک تحریک رندوم در دسته ای تحریک های به شرط اسپایک و تحریک های رندوم قرار بگیرد مساوی است . یعنی جایی است که به احتمال 50 درصد اسپایک می زند . ما همین عدد را به عنوان threshold ذکر می کیم و تصویر تحریک های بعدی را نسبت به همین عدد می سنجیم . می توانیم threshold بالاتری را بگیریم اما با همین متدهم کمکان دقیت کمی داریم . یعنی حدود 50 درصد تحریک های به شرط اسپایک را نمیتوانیم درست تشخیص دهیم . در جدول زیر به طور کامل مشهود است .

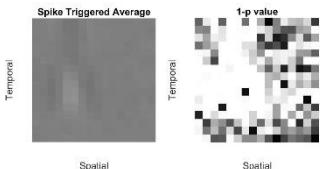
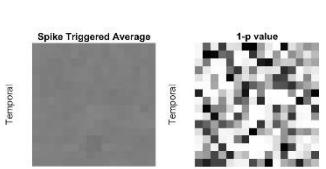
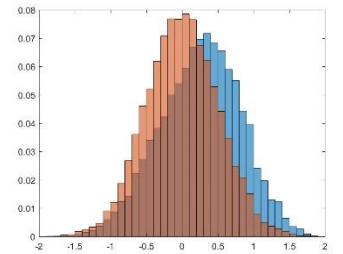
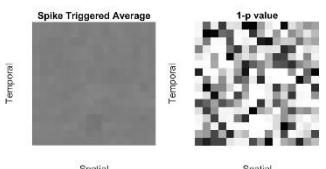
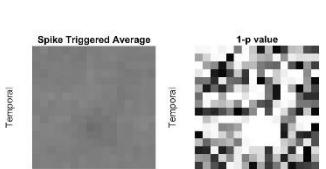
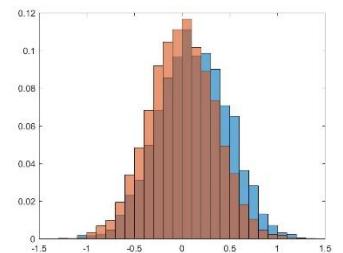
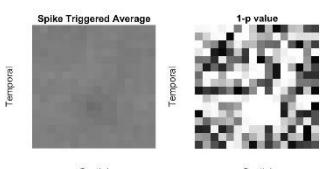
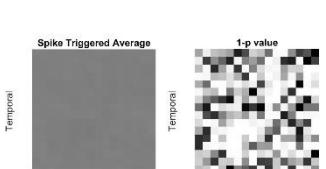
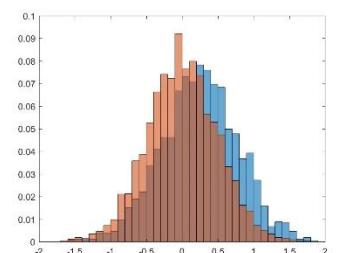
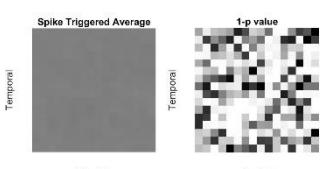
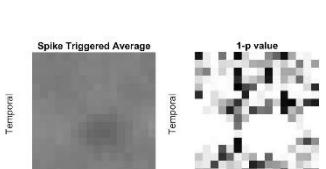
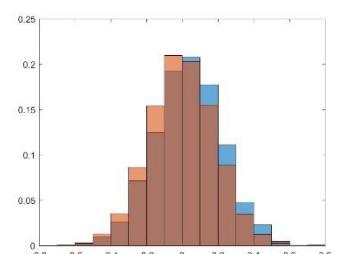
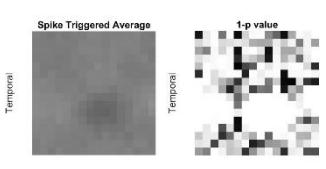
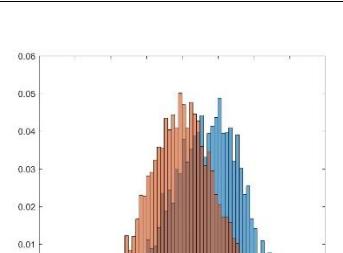
Neuron Code	Receptive Field & 1-p_value Image	Histogram	P_value	ON_OFF threshold	Percentage
000412.a01			9.1655e-67	0.0517	0.5577
000419.a06			1.0290e-243	0.0507	0.5705
000419.a07			1.6034e-87	0.1096	0.5802

000419.a09	 		$9.9139e-136$	0.0494	0.5451
000503.a03	 		$2.4791e-220$	0.0538	0.5622
000511.b09	 		$7.4459e-210$	0.1358	0.5768
000511.b10	 		$1.4638e-91$	0.0612	0.5571
000513.d11	 		$5.0487e-67$	0.0248	0.5434

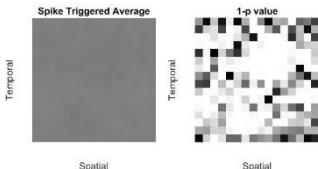
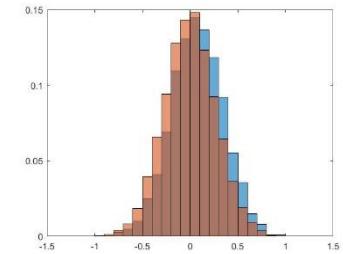
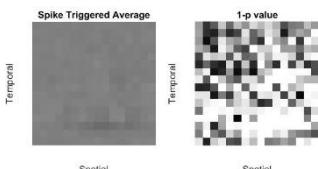
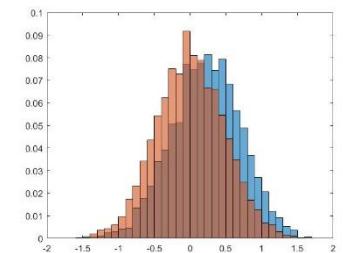
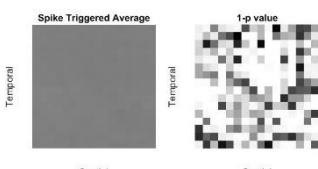
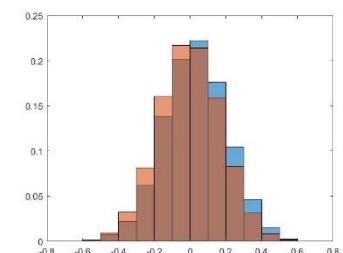
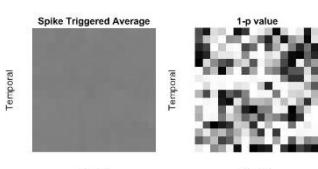
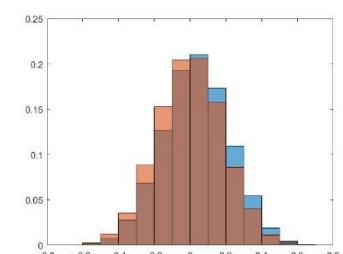
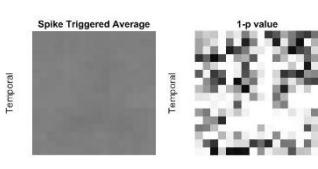
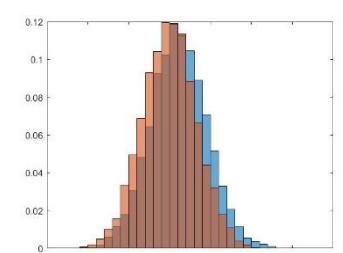
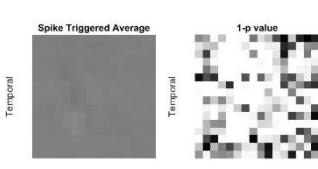
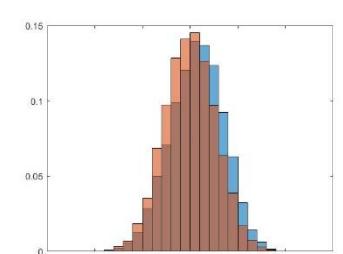
000525.d05	 		$2.6373e-150$	0.0081	0.5290
000601.co5	 		$1.1769e-140$	0.0494	0.5575
000601.co7	 		$4.6692e-108$	0.0525	0.5499
000620.a02	 		$2.6497e-188$	0.0528	0.5550
000622.fo5	 		Infinitely Small	0.0769	0.5589
000712.bo4	 		$1.5500e-102$	0.1570	0.5849

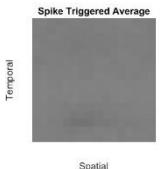
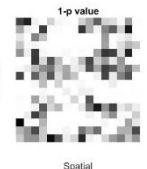
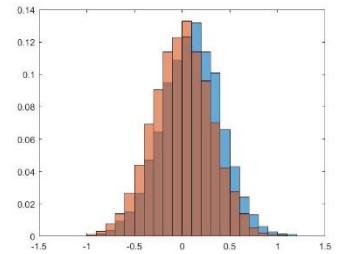
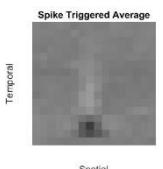
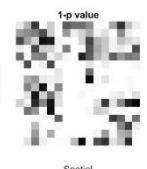
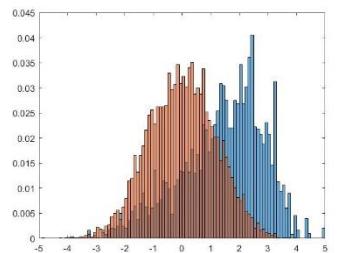
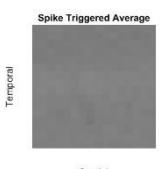
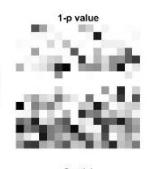
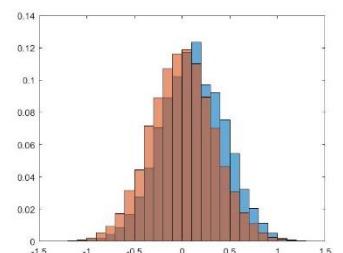
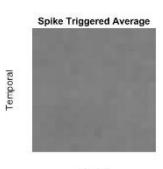
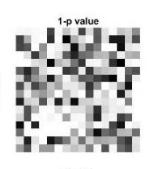
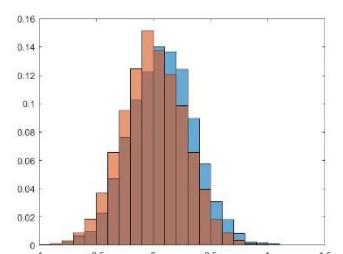
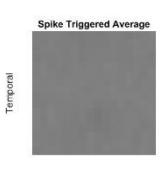
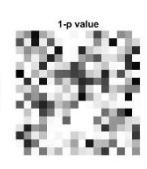
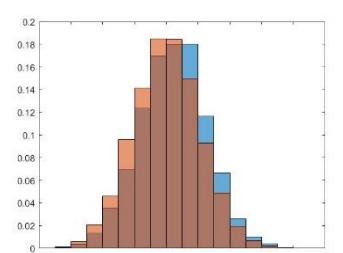
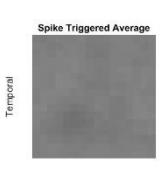
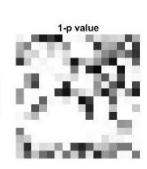
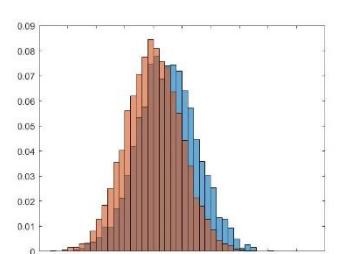
000720.co6	 		1.2704e-129	0.1022	0.5753
000802.co7	 		5.3586e-93	0.0165	0.5420
000804.io1	 		1.3532e-114	0.1080	0.5865
000823.d04	 		1.7714e-295	0.1271	0.6011
000824.g04	 		8.9150e-317	0.2202	0.6131
000914.co6	 		3.4768e-195	0.1204	0.5712

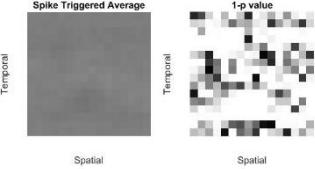
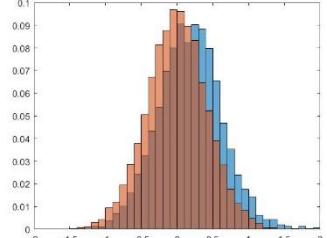
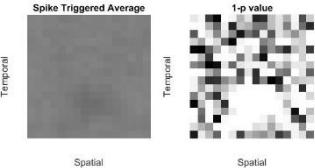
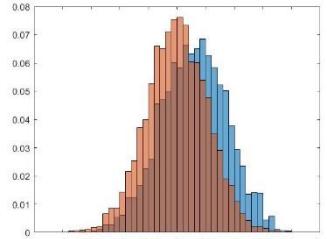
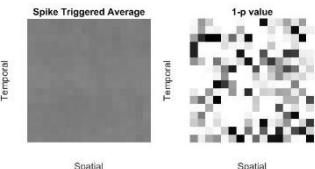
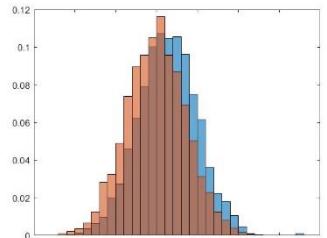
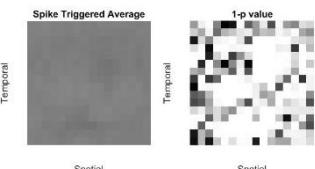
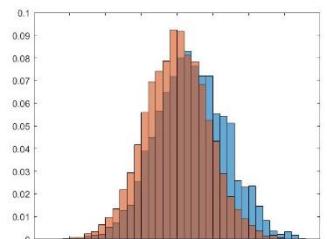
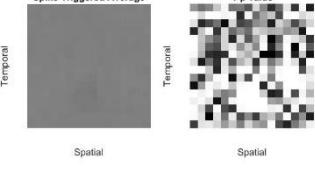
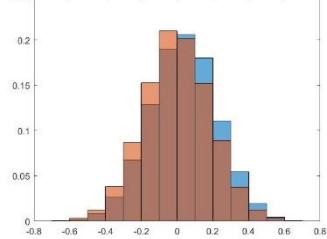
000926.a04			8.3147e-107	0.0904	0.5577
010125.A.co2			3.939e-301	0.1018	0.5833
010125.A.co3			Infinitely Small	0.1263	0.5718
010208.A.hoi			1.529e-291	0.0549	0.5612
010322.A.fo6			9.295e-210	0.0598	0.5835
010628.A.co3			4.649e-68	0.0712	0.5709

o10628.A.co4	 		Infinitely Small	0.1770	0.5787
o10718.B.co2	 		$2.9355e-95$	0.0706	0.5480
o10801.A.bo1	 		$1.969e-98$	0.1616	0.5721
o11024.A.bo4	 		$1.787e-100$	0.0185	0.5360
o11025.A.d07	 		Infinitely Small	0.4158	0.6371

020109.A.bo1			6.187e-234	0.2961	0.5691
020214.A.jo1			Infinitely Small	0.1719	0.5952
020306.A.ao1			8.380e-217	0.2390	0.5545
020306.A.ao2			3.88e-244	0.2378	0.5927
020321.A.io2			3.230e-83	0.0499	0.5544
000622.fo3			1.5082e-196	0.0333	0.5556

000622.fo4	 Temporal      Spatial	 1-p value	$7.9457e-217$ 0.0366 0.5489
000712.bo3	 Temporal      Spatial	 1-p value	$6.0900e-154$ 0.1228 0.5886
000802.co5	 Temporal      Spatial	 1-p value	$2.3083e-138$ 0.0157 0.5320
000802.co6	 Temporal      Spatial	 1-p value	$4.4750e-75$ 0.0187 0.5339
000914.co7	 Temporal      Spatial	 1-p value	$1.8938e-157$ 0.0611 0.5512
010524.A.fo1	 Temporal      Spatial	 1-p value	$2.9294e-302$ 0.0399 0.5580

010614.B.eo8	 Temporal Spatial	 Temporal Spatial		1.1933e-254	0.0503	0.5663
010718.B.co1	 Temporal Spatial	 Temporal Spatial		Infinitely Small	1.3181	0.5872
011019.A.co9	 Temporal Spatial	 Temporal Spatial		3.0508e-188	0.0578	0.5673
011101.A.d02	 Temporal Spatial	 Temporal Spatial		6.0558e-99	0.0397	0.5493
011101.A.d03	 Temporal Spatial	 Temporal Spatial		2.2768e-141	0.0219	0.5438
011121.A.d02	 Temporal Spatial	 Temporal Spatial		5.8111e-292	0.1424	0.5654

020109.A.bo2			9.6614e-295	0.1068	0.5599
020213.A.io1			3.3183e-194	0.1832	0.5777
020308.A.d01			2.6411e-197	0.0645	0.5750
010613.B.bo2			1.8975e-217	0.1308	0.5536
020321.A.io1			5.2165e-134	0.0176	0.5377

### QUESTION 7 :

اشکال حاصله از روش STA عموماً رندوم و بدون اطلاعات می باشند . در موارد نادری که در پخش بعد ذکر شده کمی با Receptive Field تطابق دارند اما در اکثر موارد خیر . لهذا ادعای پیش مبنی بر Complex بودن این سلول ها وارد است .

## Part 4 :

### QUESTION 1 :

ابتدا ماتریس Correlation را طبق رابطه ای که در مقاله آمده حساب می کنیم .

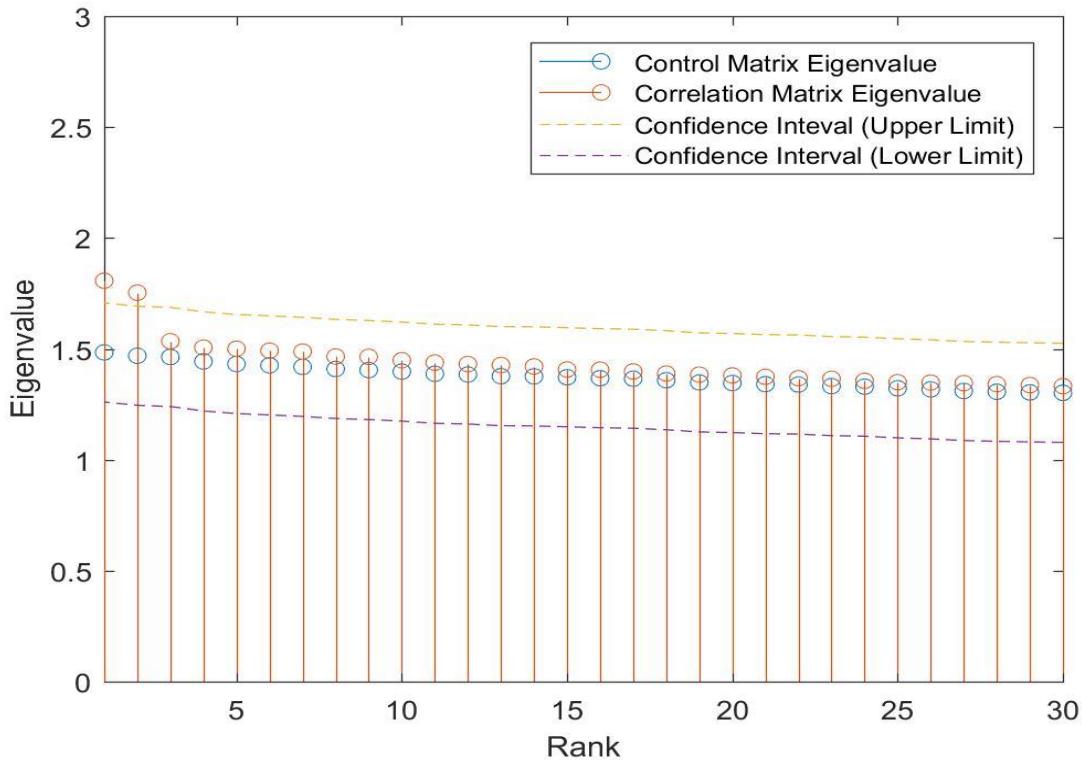
$$C_{m,n} = \frac{1}{N} \sum_{i=1}^N S_m(i)S_n(i)$$

الگوریتم ما بدین شکل بود که ابتدا عدد  $m$  و  $n$  را به شماره سطر و ستون تبدیل کردیم سپس دو خانه از ماتریس را در دو حلقه  $i$  for ، در اختیار گرفتیم . سپس این دو خانه را در طول بعد سوم (تعداد کل اسپایک ها) ضرب کردیم و مقادیر بدست آمده را average گرفتیم، بدین شکل ماتریس Correlation ما کامل شد . حال مانند کاری که در Paper انجام شد ، تصویر حاصل از دو بردار ویژه بزرگ و بردار ویژه  $n$  را نمایش می دهیم . با دستور imresize تصویر را Smooth می کنیم و Contrast آن را افزایش می دهیم . در نهایت با دستور imshowpair تصویر را بزرگ می کنیم . بین تصویر بدست آمده و تصویری که در دستور کار بود تفاوتی وجود ندارد به جز اینکه تصویر ما پیکسل پیکسل نیست و Moving Filter یا Smoothing Filter روی آن اجرا شده .



## Question 4 :

حال مانند کاری که مقاله کرد ، 30 مقدار ویژه ای اول Control Matrix را حساب می کنیم و به اندازه SD بالا و پایین آن ، ناحیه اطمینان را مشخص می کنیم . 30 مقدار ویژه ای بزرگ Correlation Matrix را نیز در تصویر می کشیم تا متوجه بشویم که کدام یک خارج از محدوده ای مورد نظر هستند . (Significantly Large) .



## Question 3 :

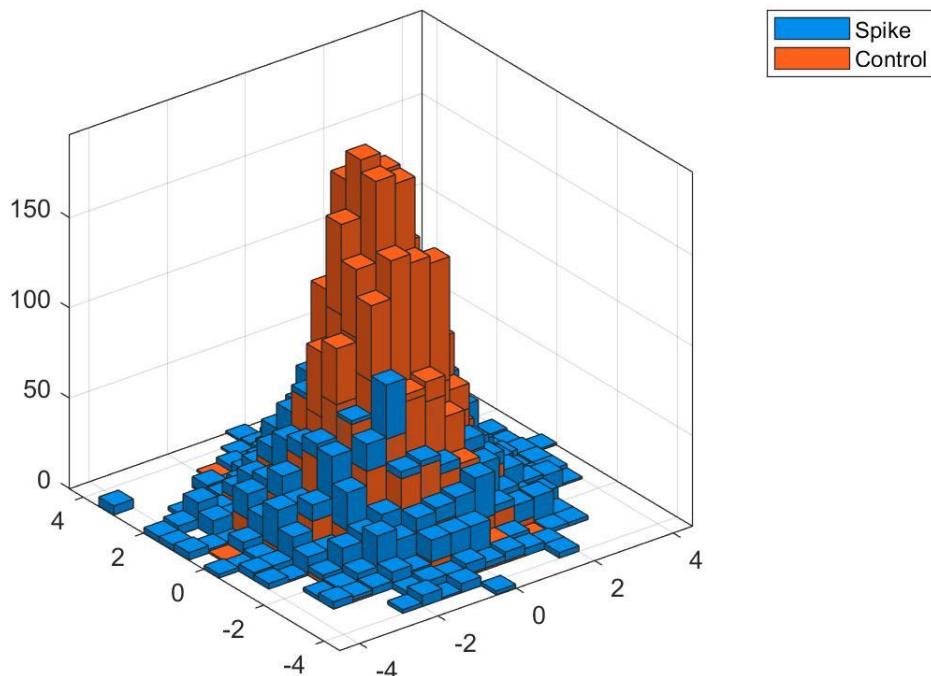
همانطور که در قسمت اول پروره توضیح دادیم ، مبنای انتخاب Feature های اصلی یک نورون در این مقاله ، روش PCA است . نورون های مورد مطالعه در این مقاله ، نورون های پیچیده هستند لذا Receptive Field آن ها با روش STA قابل تمیز دادن نیست . به همین خاطر ما به دنبال راستاهایی در فضای تحریک می گردیم که ویژگی های اصلی تصویر ما باشند . ویژگی هایی که در آن راستا ، واریانس بیشینه باشد . این راستا ها یک ترکیب خطی از ویژگی های اولیه تصویر هستند (256 SpatioTemporal axeses) . برای پیدا کردن این راستا ها باید مقادیر ویژه ای ماتریس Correlation را محاسبه کنیم و مقادیر Significant را جدا کنیم . برای پیدا کردن مقادیر Significant باید معیاری داشته باشیم . طبق متد مقاله ، ناحیه ای اطمینانی با فاصله ای یک انحراف معیار از مقادیر

ویژه‌ی ماتریس کترل ، قرارداد می‌کنیم . لذا هر مقدار ویژه‌ای که خارج آن ناحیه قرار دارد ، Significant محسوب می‌شود و بردار ویژه‌ی متناظر با آن ، راستای Principal Component را به ما می‌دهد .

یک نکته : با دقت به شکل حاصل شده در سؤال دو ، اختلاف فاز 90 درجه‌ای Spatial تصویر را می‌توانیم مشاهده کنیم . این خاصیت محدودیت ریاضی متند PCA نیست بلکه کاملاً فیزیکی و واقعی است .

## Question 4 :

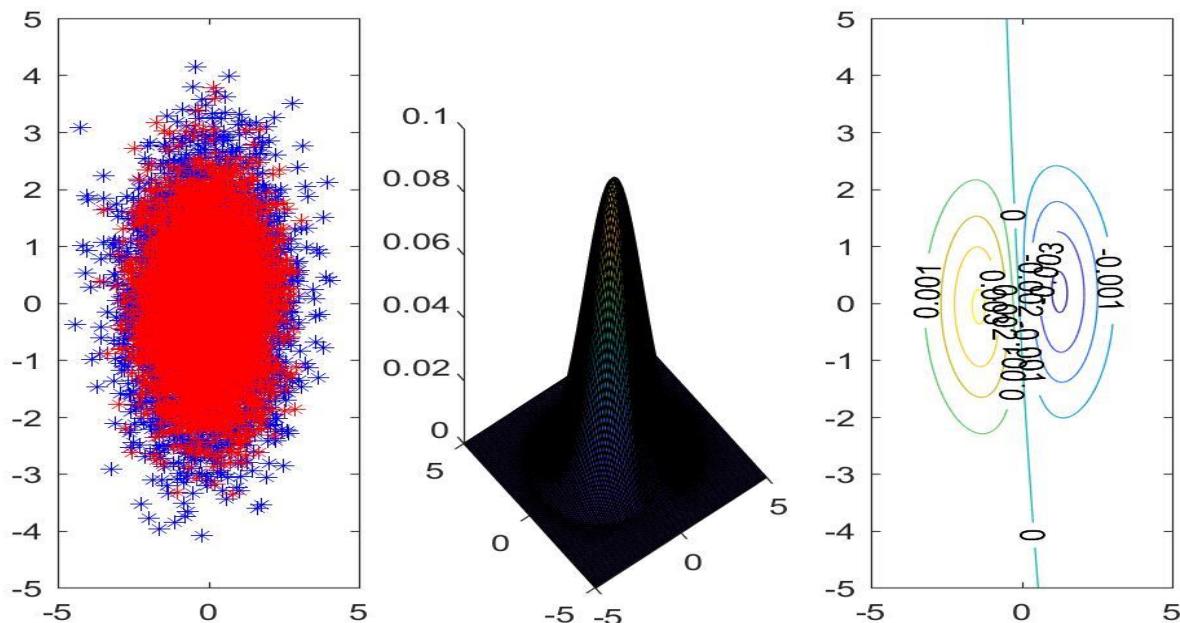
بردار تصویر‌های تحریک‌ها می‌منجر به اسپایک روی جفت بردار‌های ویژه Significant را با دستور Plot , Histogram2 می‌کنیم . این کار را برای تحریک‌های random نیز انجام می‌دهیم . شکل حاصل به صورت زیر است :



## Question 5 :

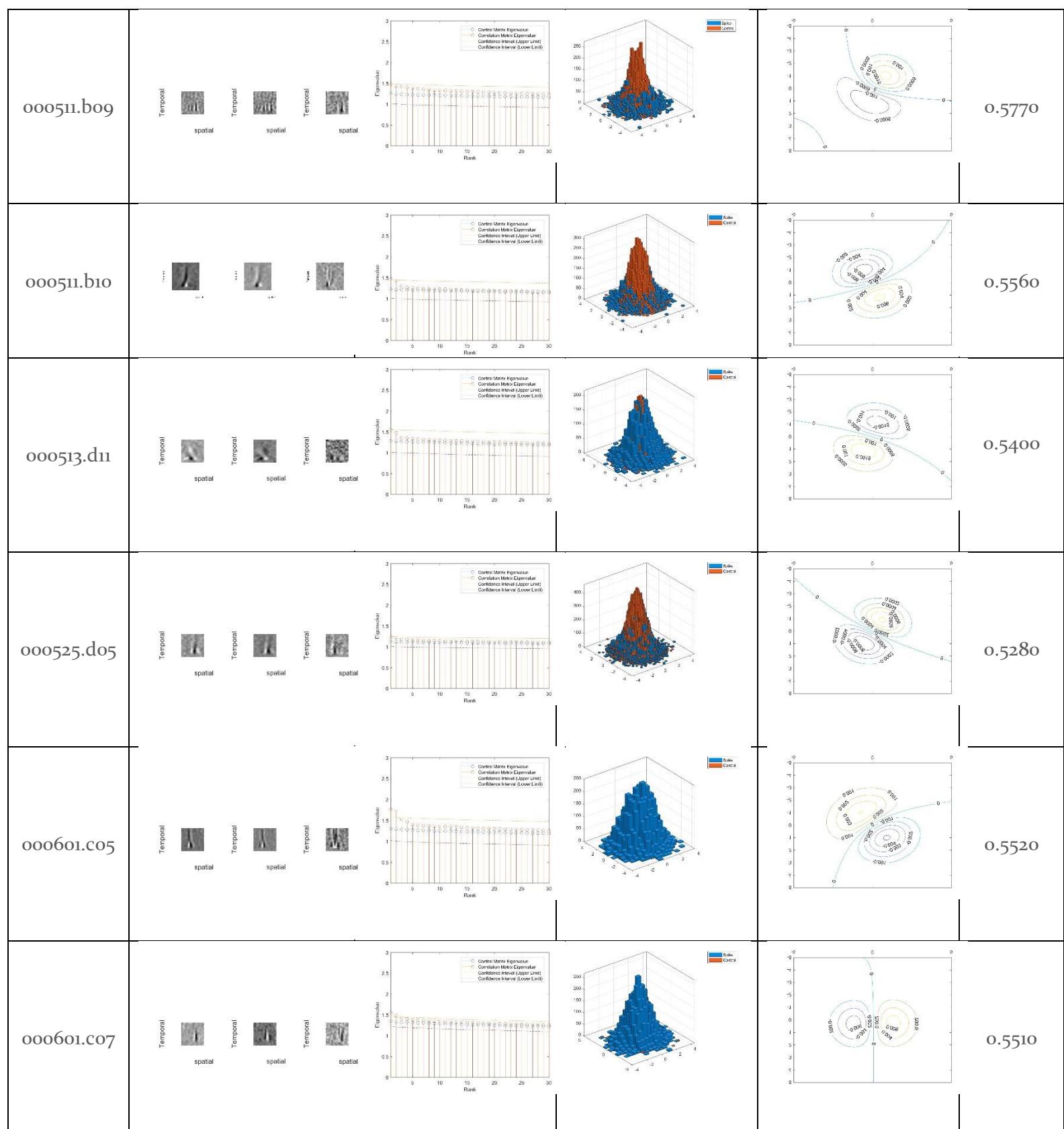
در این قسمت باید threshold ای را برای احتمال اسپایک زدن یا نزدن مشخص کنیم . یعنی اگر تصویر تحریک روی دو feature مهم آن از حدی بیشتر شد بگوییم احتمال اسپایک زدن بیشتر از اسپایک نزدن است . برای روش STA دو هیستوگرام دو بعدی داشتیم و آن‌ها را به سادگی تقاطع می‌دادیم . نقطه‌ی تقاطع ، محل یکسان بودن احتمال اسپایک زدن و نزدن است . حال مشابه این کار را برای

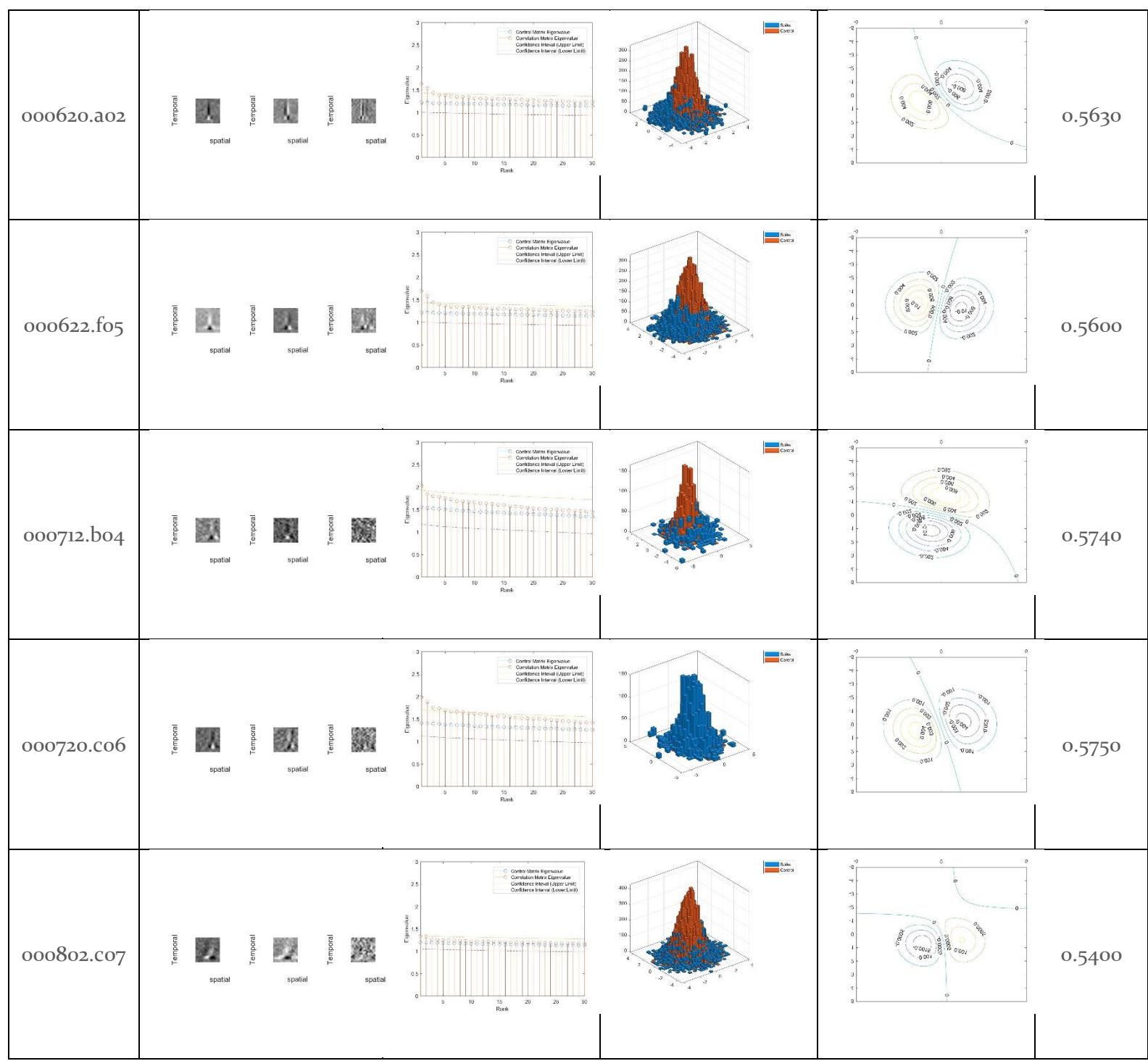
هیستوگرام سه بعدی می کنیم . بدین شکل که دو دسته دیتا را روی یک نرمال دو بعدی fit می کنیم . سپس آن دو را از هم کم می کنیم و contour های صفر آن را پیدا می کنیم . خارج ناحیه ای این contour، ناحیه ای است که احتمال اسپایک زدن بیشتر از اسپایک زدن است .

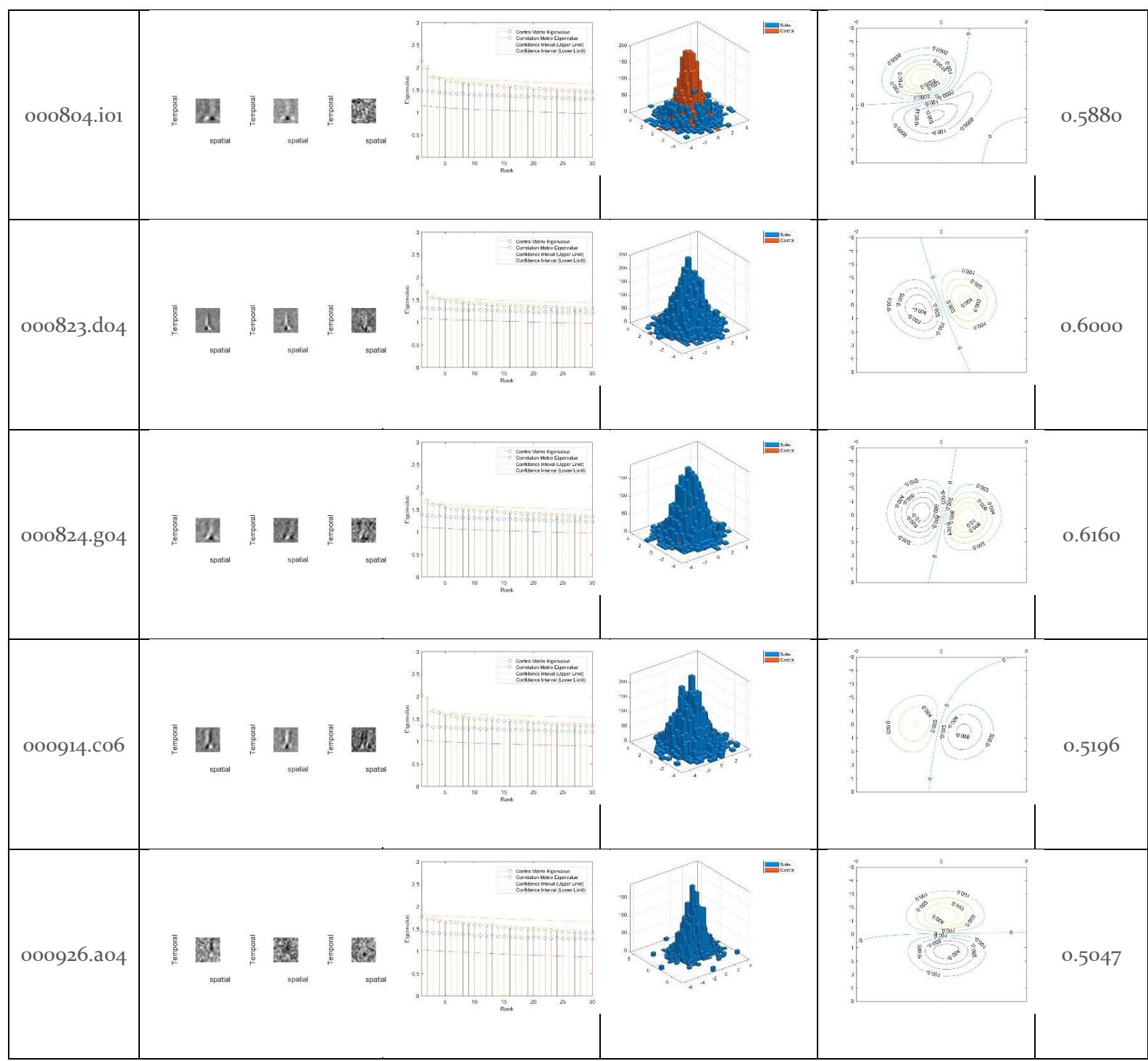


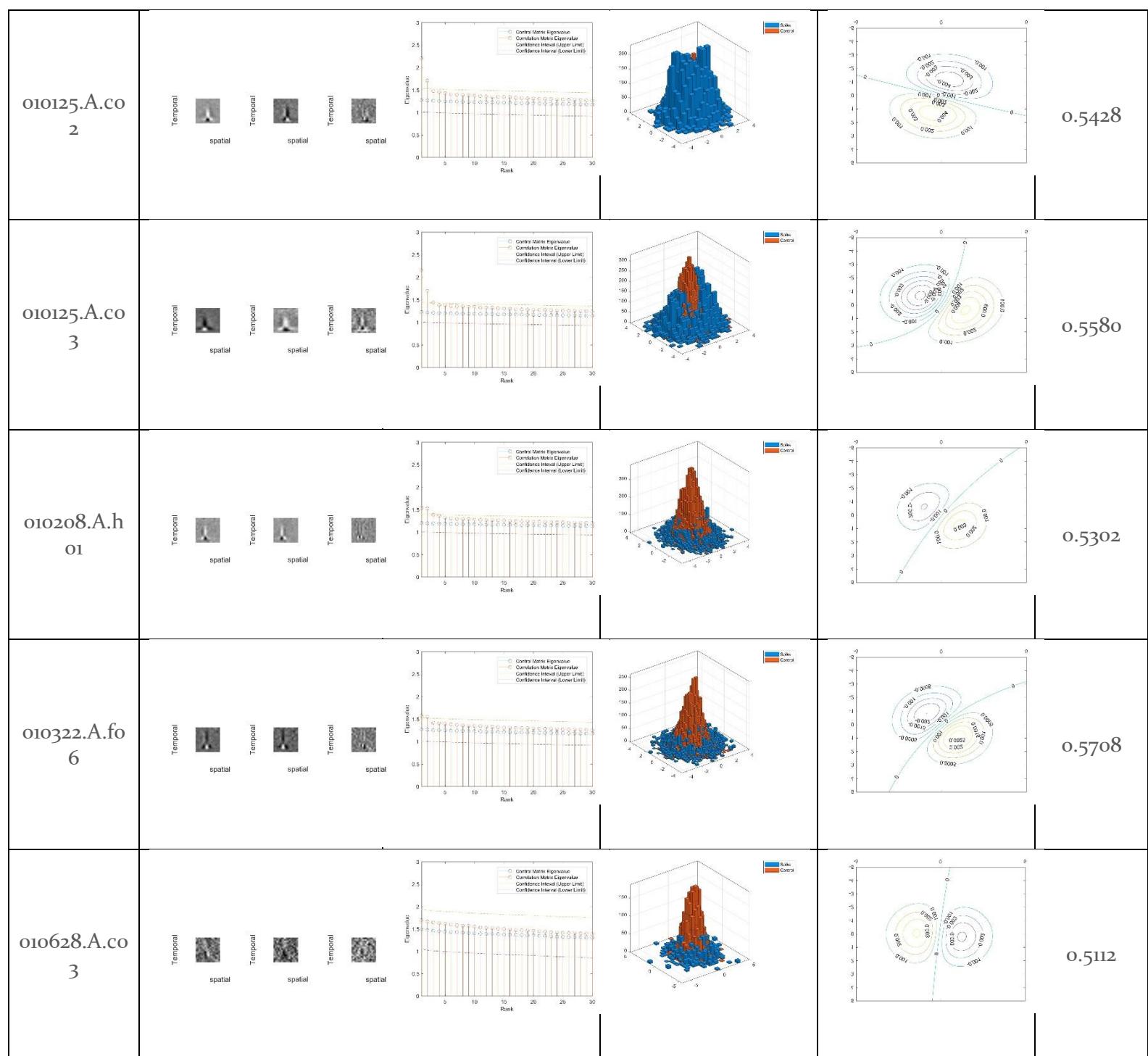
## Question 6 :

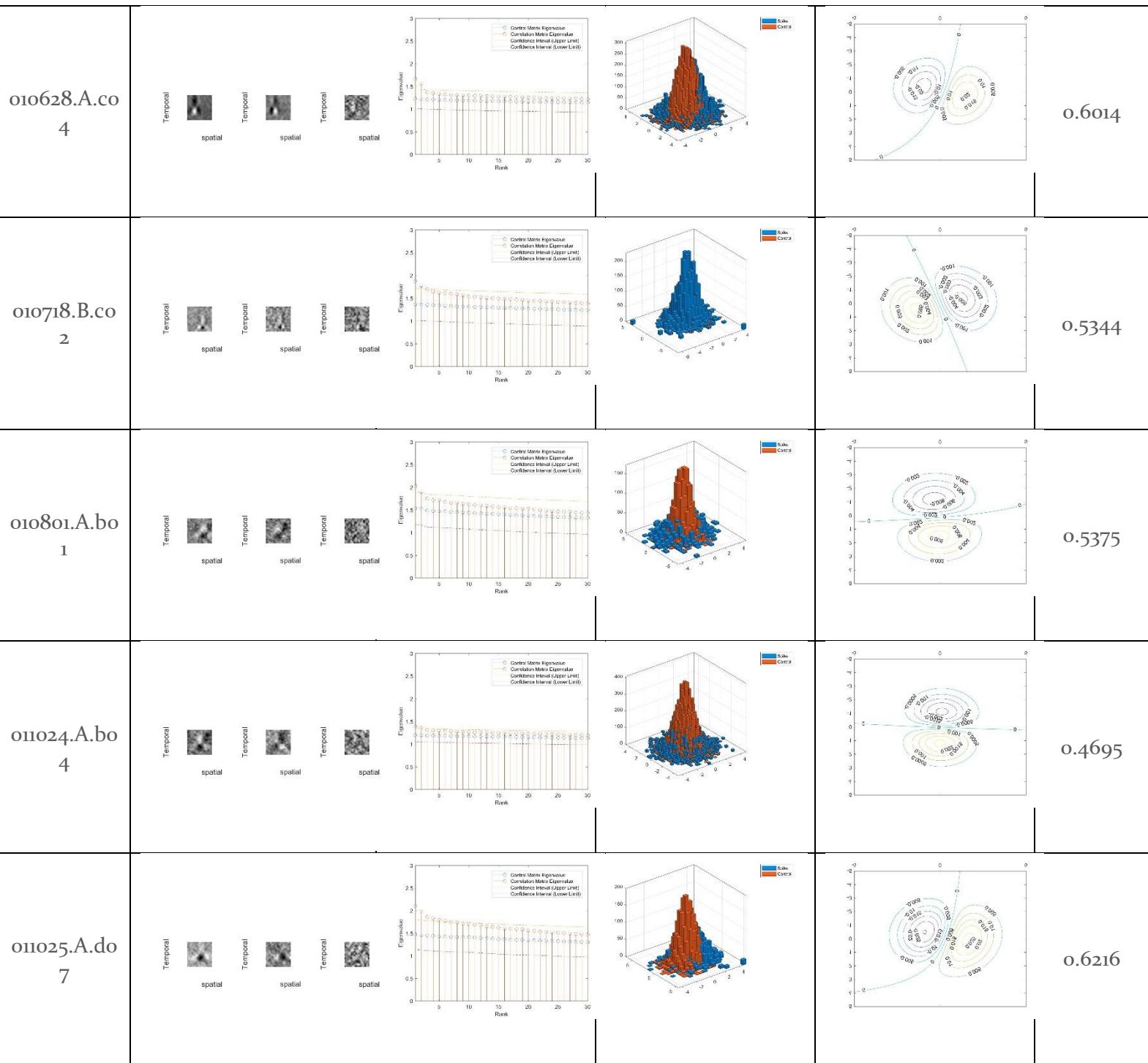
Neuron Code	First & Second & nth Eigenvector	Ranking Eigenvalues	3D_Histogram	Threshold	Success Percentage	
000412.a01	Temporal  spatial 	Temporal  spatial 	Ranking Eigenvalues 	3D_Histogram 	Threshold 	0.5560
000419.a06	Temporal  spatial 	Temporal  spatial 	Ranking Eigenvalues 	3D_Histogram 	Threshold 	0.5690
000419.a07	Temporal  spatial 	Temporal  spatial 	Ranking Eigenvalues 	3D_Histogram 	Threshold 	0.4667
000419.a09	Temporal  spatial 	Temporal  spatial 	Ranking Eigenvalues 	3D_Histogram 	Threshold 	0.5450
000503.a03	Temporal  spatial 	Temporal  spatial 	Ranking Eigenvalues 	3D_Histogram 	Threshold 	0.5610

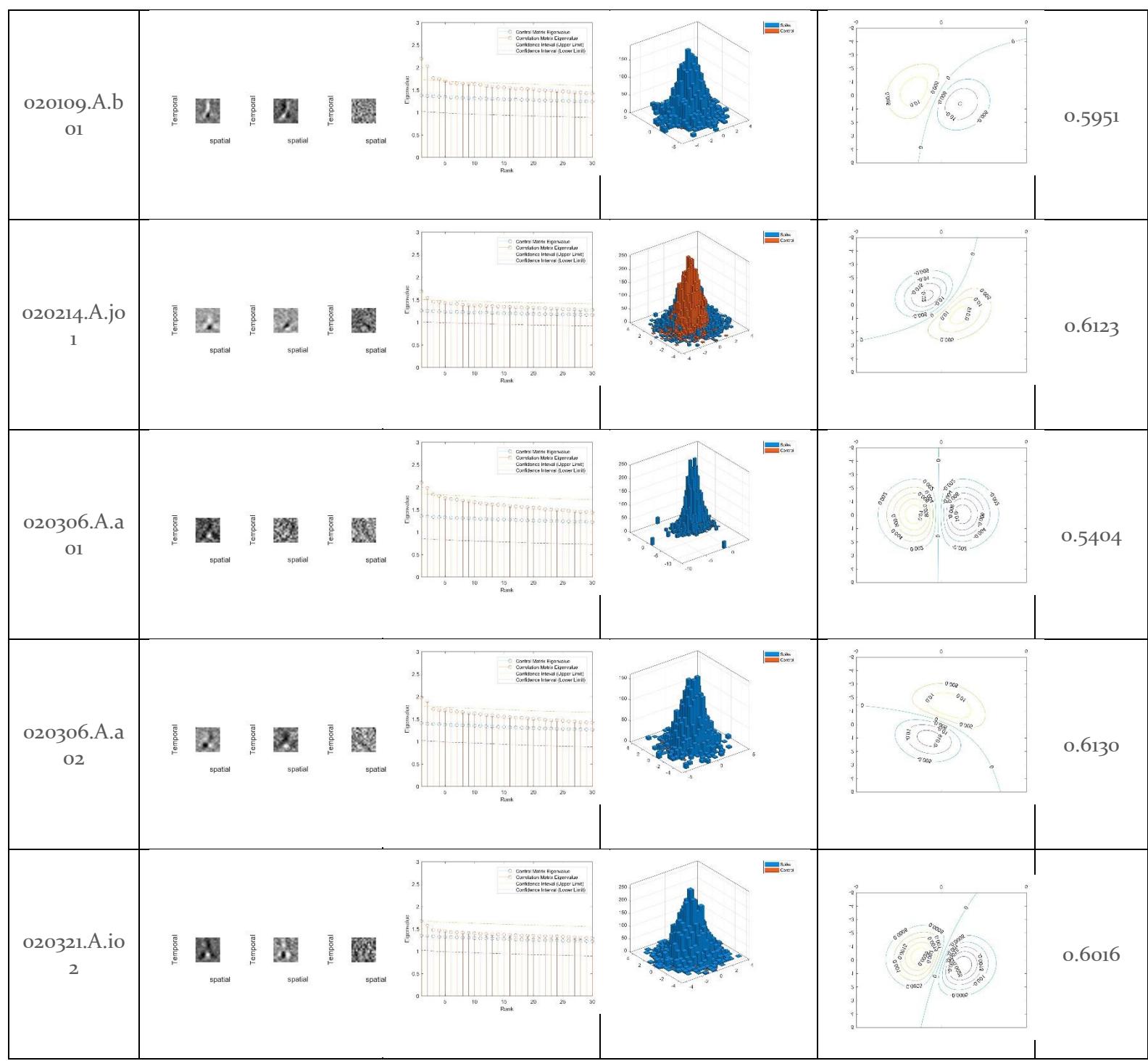


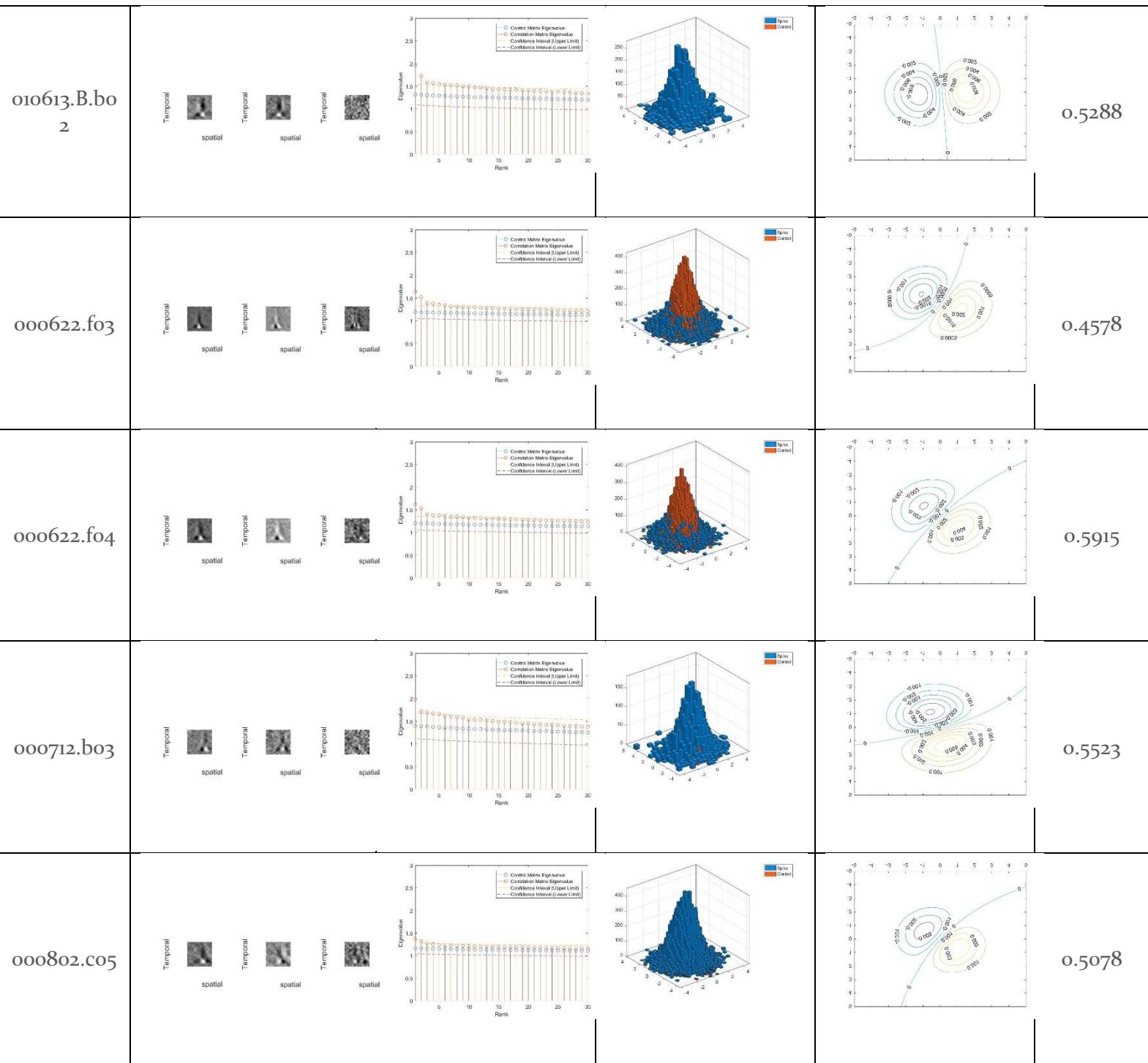


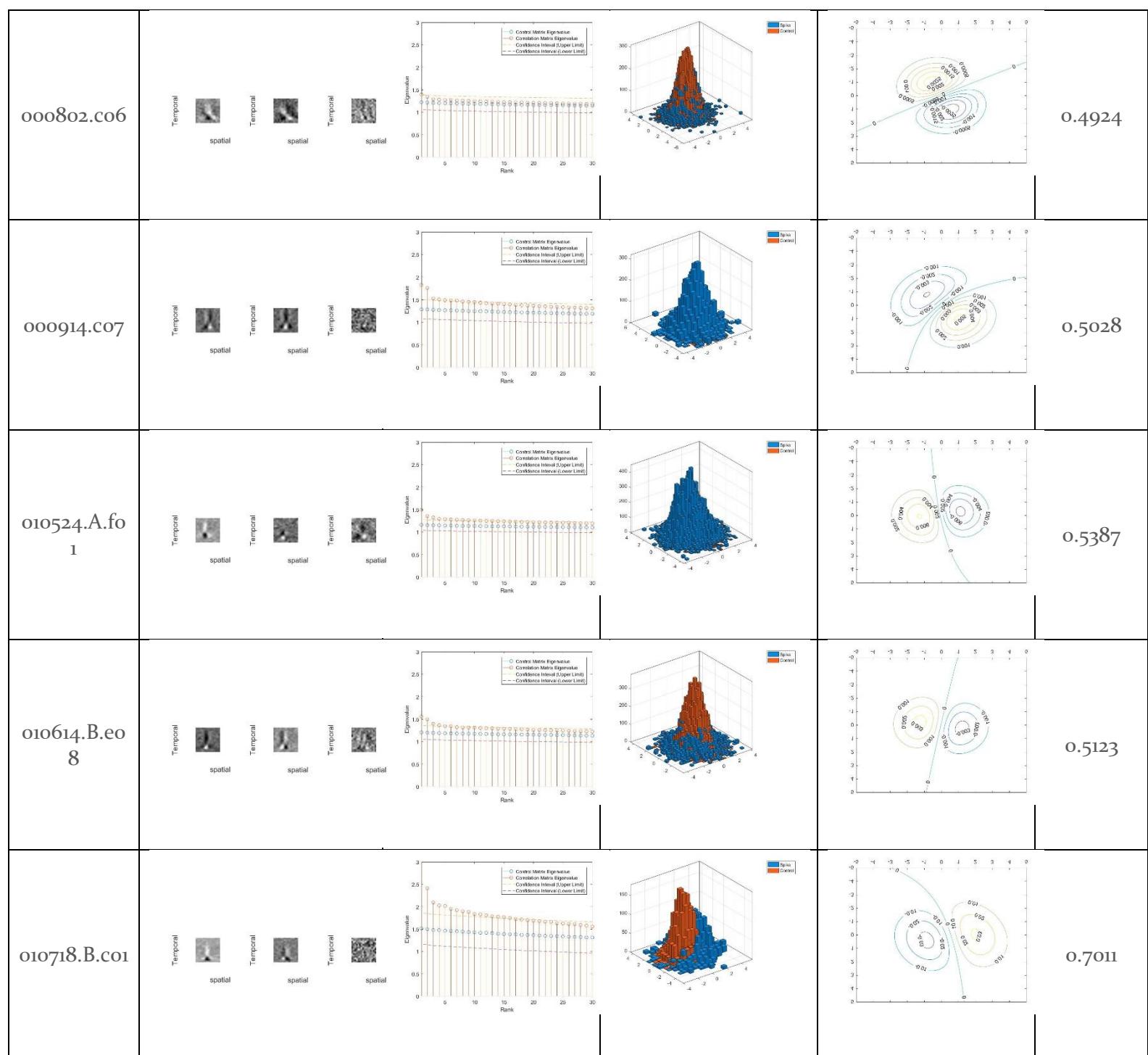


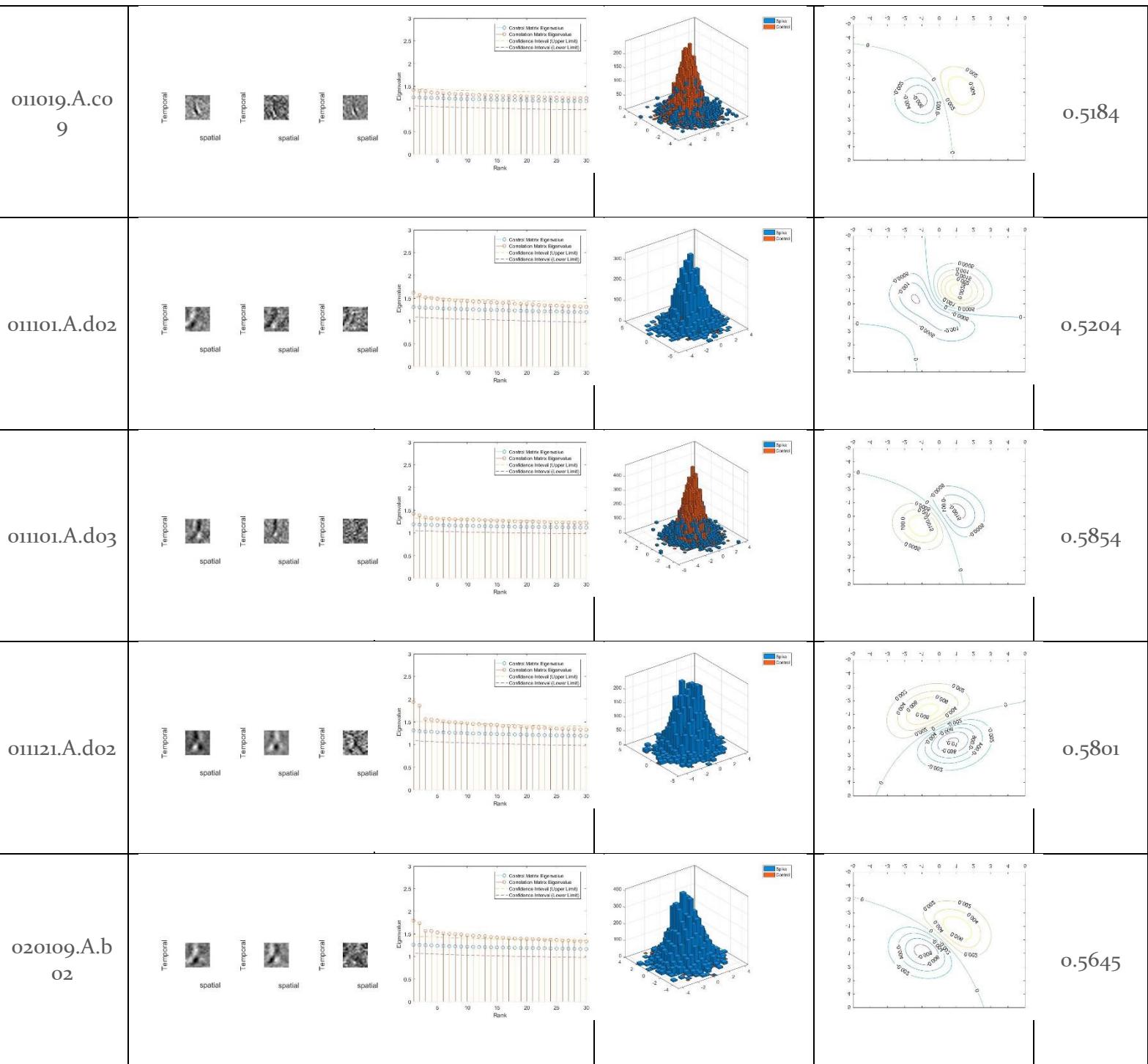


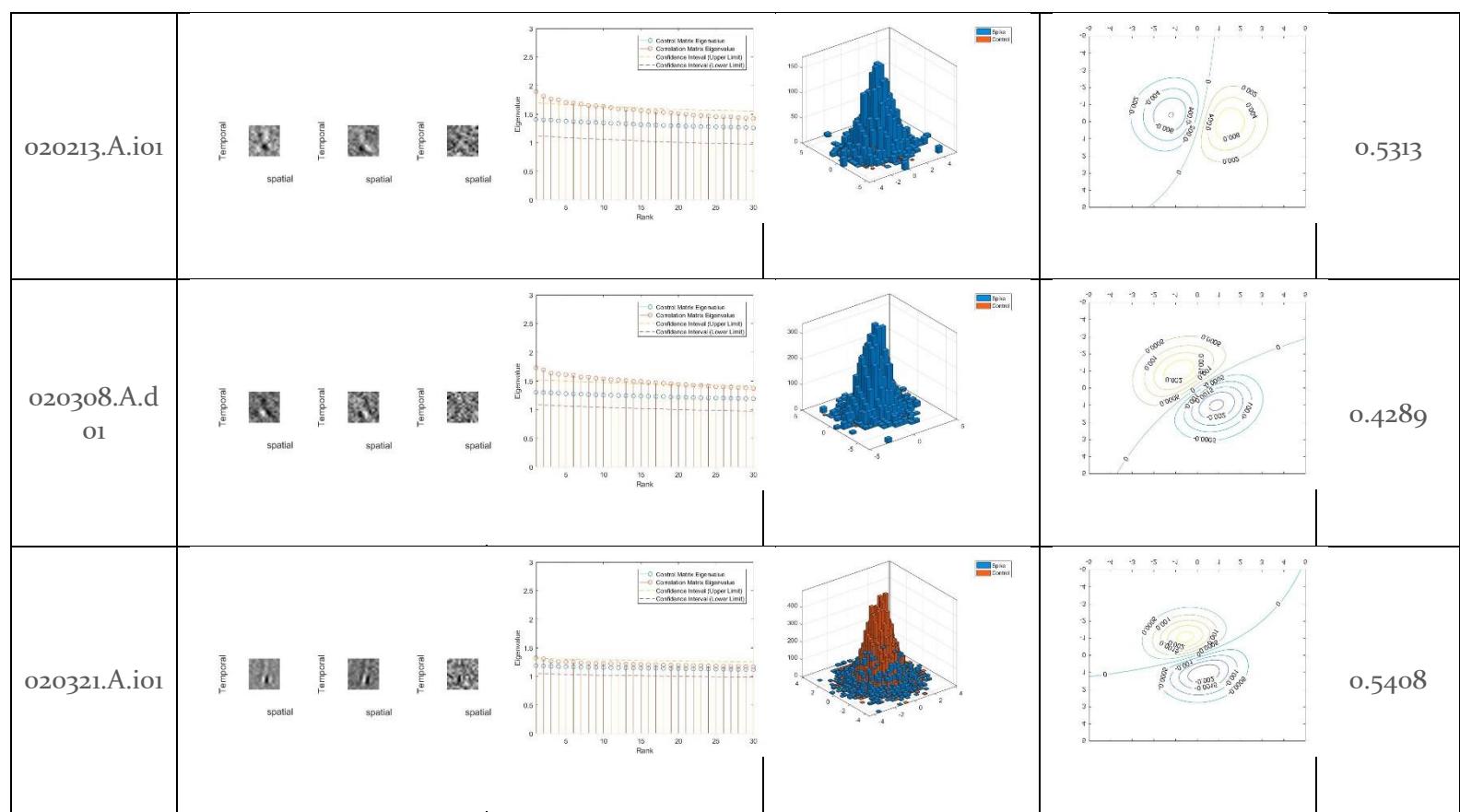






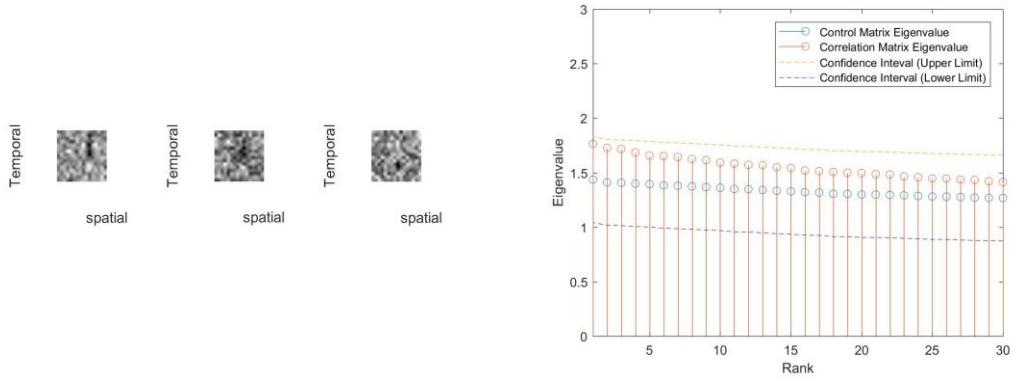






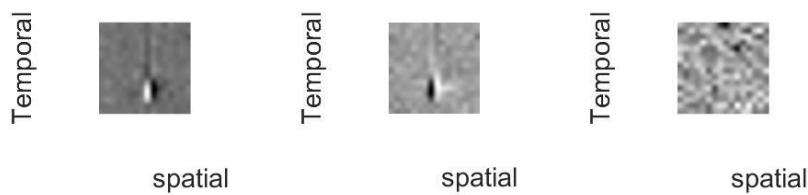
## Question 7 :

بردار ویژه ها حاوی تصاویری به مراتب با معنی تر و واضح تر از تصاویر روش کلاسیک می باشند . تصاویر کلاسیک به ندرت دارای پترن خاصی بودند و معمولاً رندوم بودند که نشان از ناکارآمدی این روش برای سلول های پیچیده می باشد . بردار ویژه ها اگر با گپ معناداری از بازه اطمینان ما فاصله داشته باشند معمولاً تصاویر معنی داری به ما می دهند . به نظر اینجانب تنها بالای محدوده اطمینان بودن کافی نیست بلکه باید به جهش حساس باشیم . ممکن است مقادیر ویژه به صورت یکنواخت با شبی معقولی به بالای محدوده اطمینان برسند اما receptive field کاملاً بی معنی و رندوم به نظر برسد . برای مثال برای نورون "ao4.000926" داریم :

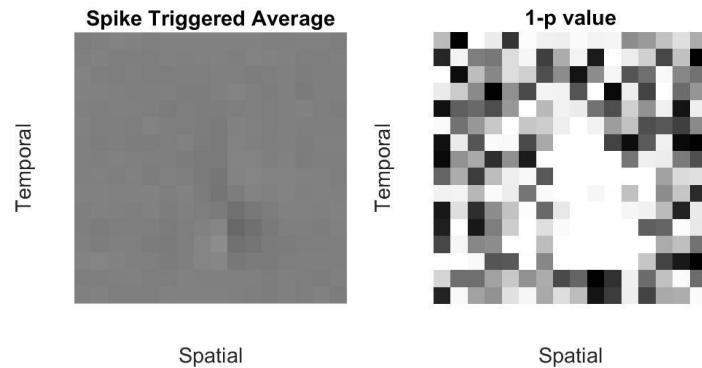


حال به مقایسه روش STA و PCA در تشخیص Receptive Field می پردازیم . با فرض Significant بودن دو بردار ویژه ، ملاحظه می کنیم که تعدادی از نورون ها دو ناحیه ON و OFF مستقل از هم دارند . برای مثال به نورون "b09.000511" توجه

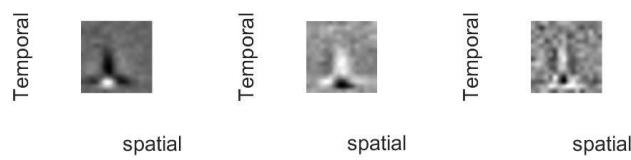
کنید :



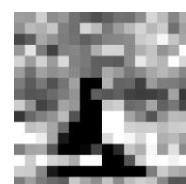
دو بردار ویژه ای Pattern دارای دو Significant مخالف و مستقل از هم می باشند . دو ناحیه تاریک و روشن در هر دو تصویر موجود می باشند اما محل قرارگیری دو ناحیه برعکس هم می باشد . بودن این نورون ها اینجا واضح می شود که با این نتیجه اشتباہی می داد . به تصویر حاصله از روش STA توجه کنید :



اشکال کاملا Random و بی معنی هستند . یا مثلا برای نورون "A.co3.010125" داریم :



دو Receptive Field کاملا برعکس یکدیگر عمل میکنند . البته برای این نورون اگر STA را بالا ببریم تصویری شبیه به بردارویژه های روش PCA حاصل می شود که البته تنها شبیه بکی از Receptive Field های ما می شود .



در مجموع اشکال حاصله از روش PCA ، اکثرا دارای Pattern های مشخص و نواحی ON و OFF واضحی می باشند . اشکال حاصله از این متدهای اختلاف فاز فضایی 90 درجه می باشند . این اختلاف فاز یکی از محدودیت های روش PCA است زیرا پایه های متعامد تولید می کند اما در حوزه کاری ما این اختلاف فاز ناشی از فیزیک و طبیعت نورون است . تصاویر ناشی از روش کلاسیک عموماً است اما در مثال هایی اگر خوش شانس باشیم نتیجه کمی به Receptive Field های واقعی نورون نزدیک است .

## Part 5 :

سئوالاتی در ذهن اینجانبان (البهه ۳ نفریم !!!) شکل گرفته که به بررسی آن می پردازیم . هرچند به علت نداشتن زمان کافی برای پرورش این سوالات و پاسخ مناسب به آن ها ممکن است هنوز خام باشند .

( سوال اول )

این که طی یک محاسبه سرانگشتی در بین ۶۱ نورون موجود که ما ۵۴ تا از آن ها را بررسی کردیم ، روش Spike triggered average که پیش می آید این است که پس چرا به سراغ روش Principal component analysis رفتیم ؟ ! مشاهده کردیم که این متده برای بدست آوردن Receptive Field های یک نورون بسیار کارآمد بود و تصویری نسبتاً واضح و به مراتب بهتر از روش STA می داد . اما آیا این روش برای Neural Encoding نیز به همان اندازه کارآمد است ؟ ! شواهد دال بر آن است که در مسئله Encoding ، هر دوی روش ها تقریباً به یک اندازه قابل اتکاء هستند . در هر دو روش ما در بهترین حالت حدود ۴۰ درصد از اسپایک ها را درست تشخیص نمی دهیم که این نشان از Optimal نبودن فیلتر خطی می باشد . طبق صحبت های سر کلاس فیلتر Optimal ، حاصل ضرب اسکالری است در جهت بردار STA حاصل از تحریک White Noise . از طرفی این مقاله در Dataset های خود ادعا کرده که این تحریک ها ۱D\_White\_Noise\_Stimuli هستند اما در متن paper نوشته که تحریک ها به صورت Optimum بودن فیلتر به طور دقیق مطمئن نیستیم .

( سوال دوم )

ما در این آزمایش حدود ۶۰ Hz می باشد . یعنی در یک ثانیه حدودا ۶۰ Frame برای گره پخش می شود . چرا فرکانس ۶۰ Hz ؟ ! اگر برای حیوان دیگری یا برای انسان این آزمایش انجام می شد باز هم همین فرکانس را انتخاب می کردیم ؟ ! در پاسخ به این مسئله با مفهومی رو برو هستیم به نام threshold Flicker fusion threshold . این فرکانسی ناشی از پدیده "پسماند بینایی" یا "Persistence Of Vision" می باشد . پسماند بدین صورت است که اگر تصویری را مشاهده کنیم ، بعد از برداشتن این تصویر از جلوی چشم ، تاثیر این تصویر روی Retina باقی می ماند . برداشت بنده از این پدیده V1 است که سلول های Rod دیرتر ساخت می شوند و با تاخیر کمی دست از Spike زدن بر می دارند . پس لایه های بالاتر مانند V1 ، هنوز حس میکند جسمی فیزیک جلوی چشم وجود دارد در صورتی که چنین چیزی وجود ندارد . حال از این پدیده می توان استفاده های زیادی کرد . به عنوان مثال اگر frame را مرتبأ روشن و خاموش کنیم و فرکانس کار را به مرور بالا ببریم ، چشم نمی تواند بین frame های روشن و خاموش تمایز قائل شود لذا همه را روشن می بیند . به فرکانس بحرانی ای که از آن به بعد چشم flicker fusion threshold نمی بیند بلکه تصویری پیوسته می بیند ، حس میکنم نویسنده ای این مقاله می خواسته این

تصاویر برای گریه به صورت منقطع پخش شوند ، چون در صورت بالاتر رفتن فرکانس از حد بحرانی ، چشم گریه نمیتوانست بین فریم های روشن و خاموش تفاوتی ببیند و شاید تعدادی از پیکسل های سیاه ، عملأً از بین می رفت . این حد بحرانی برای گونه های متفاوت حیات ، متفاوت است . مثلا برای گریه  $100\text{ Hz}$  ، برای سگ  $70\text{ Hz}$  و برای انسان ها  $20\text{--}15\text{ Hz}$  می باشد . لذا اگر همین آزمایش را میخواستیم برای انسان انجام دهیم ، مجبور بودیم فرکانس کار را زیر  $15\text{Hz}$  بیاریم .