

حذف نویزهای ضربه (نمک-فلفل) تراکم بالا در تصاویر به کمک فیلتر فازی تطبیقی تکراری با استفاده از روش میانگین Alpha-Trimmed (آلفای تنظیم شده)

نویسندگان: Snagatan Das و Faruk Ahmed

**چکیده** \_ حذف نویز ضربه در تصاویر یکی از مشکلات اساسی در پردازش تصاویر به شمار می رود. در این مقاله ما به فیلتر فازی تطبیقی (وفقی) تکرارشونده جدید برای حذف نویزهای تصویر که به علت نویز ضربه (نمک - فلفل) معیوب شده اند، پیشنهاد می دهیم. این روش در دو مرحله اجرا می گردد: ابتدا یافتن پیکسل های نویزدار با آشکارساز فازی تطبیقی (وفقی) و به دنبال آن حذف نویز با استفاده از فیلتر میانگین وزنی در پنجره فیلتر بر روی پیکسل های سالم.

نتایج تجربی برتری الگوریتم ها را نسبت به جدیدترین فیلترها نشان می دهد. همچنین نشان داده شده برای اینکه فیلتر در سطوح خیلی بالایی از نویز قدرتمند شود، جزئیات معنی داری در سطوح نویزی به بزرگی 97٪ را بازیابی می کند.

**اصطلاحات شاخص:** میانگین Alpha-Trimmed، فیلتر فازی، حذف نویز ضربه تراکم بالا

## I. مقدمه

تصاویر دیجیتالی ممکن است به خاطر عواملی نظیر نقص در سنسورهای تصویربرداری، خطاهای انتقال کانال، رسانه (محیط) غیرعادی بین صحنه و سیستم تصویربرداری (عواملی مانند پراکندگی تصادفی و جذب) و موقعیت های حافظه معیوب، دچار نویز ضربه (فلفل - نمک) شوند. [1]

برای اغلب پردازش های تصویر و برنامه های کاربردی تجزیه و تحلیل، حذف نویز ضربه از تصاویر بسیار مطلوب خواهد بود. به هنگام فرآیند حذف نویز ضربه از تصویر، حداقل افت در جزئیات مفید و به درد بخور تصویر، مطلوب خواهد بود.

تعداد بسیار عظیمی از تکنیک ها و روش ها برای حل این مسئله پیشنهاد شده است: تکنیک های فیلترینگ غیرخطی براساس فیلترهایی است که با استفاده از روش آرایش-مرتبه ای پیکسل ها عمل می کند. و عموماً در عمل نسبت به تکنیک های فیلترینگ خطی بهتر عمل می کند. در زمینه حذف نویز ضربه در تصاویر، فیلتر میانگین استاندارد (SM)، تکنیک فیلترینگ غیرخطی سنتی محسوب می شود. [2]

فیلتر میانگین استاندارد (SM) برای حفظ جزئیات تصویر خوب نبوده و علاوه براین در سطوح بالایی از خرابی (عیوب)، ضعیف عمل می کند. برای غلبه بر این محدودیت ها در طول این سال ها، تعدادی اصلاحیه برای این فیلتر پیشنهاد شده است که از آن ها می توان به فیلتر میانه وزن دار (WMF) [3] و [4] و فیلتر میانه

مرکز-وزن دار (CWMF) اشاره کرد. [5]. این فیلترها در پنجره فیلترینگ وزن بیشتری به برخی از پیکسل ها می دهند. اما در حالی که نسبت به فیلتر میانگین سنتی حفظ جزئیات بیشتری دارند با این حال یا سطوح بالایی نویز سازگاری خوبی ندارند.

در سالهای اخیر، رویکرد فیلترینگ تک مرحله ای به یک فیلترینگ دو مرحله ای تکامل یافته است. - مرحله اول شامل آشکارسازی فیلترهای معیوب شده توسط نویز ضربه، و مرحله دوم حذف این نویزها است. برتری فیلتر جایگزینی (دومرحله ای) نسبت به رویکرد تک مرحله ای در [6] نشان داده شده است. Wang و Zhang, [7]، نشان دادند که برای تصاویر با سطح عیوب بالا، یک فیلتر جایگزینی پیش رونده (تدریجی) نسبت به یک الگوریتم تکراری تک مرحله ای بسیار کارآمدتر است. در [8] Crnojevic و همکارانش پیشنهاد استفاده از یک تخمین گر (برآوردکننده) قوی واریانس، میانگین انحراف مطلق از میانگین را دادند.

رویکردهای محاسباتی نرم دو مرحله ای به خوبی فرموله و تنظیم شده اند. Russo و همکارانش، [9] ، یک فیلتر فازی دو مرحله ای با قابلیت های حفظ جزئیات بهتر را توسعه و گسترش دادند. قوانین استنتاج فازی که در [10] پیشنهاد شده است با استفاده از فیلترهای دیگر مورد استفاده قرار می گیرند تا جزئیات لبه را بهتر حفظ کنند. Choi و همکارانش [11]، تکنیکی پیشنهاد دادند که در آن سه فیلتر (برپایه حداقل مربعات فازی) برپایه مجموعه قوانین فازی ترکیب شده اند. در [12] یک فیلتر میانه جایگزینی (دومرحله ای) پیشنهاد شده است که بر پایه یک چارچوب فازی تنظیم شده است. Schulte و همکارانش یک تکنیک غیرخطی دو مرحله ای بر پایه منطق فازی معرفی کرده اند. [13]. در [14] یک فیلتر بر پایه سیستم های استنتاجی عصبی-فازی و فقی (انطباقی) پیشنهاد شده است که برای نویزهای با سطوح بالا عمل می کند.

فیلترهای فازی معمولاً ساده و کاملاً موثر هستند. مخصوصاً اگر در تنظیمات و فقی (انطباقی) به کار گرفته شوند. در [27] یک فیلتر میانه فازی و فقی (انطباقی) برای حذف نویزهای ضربه پیشنهاد شده است که بر پایه مدل ابری است. [15] این فیلتر در مورد حذف نویزهای ضربه با سطوح بالا، عالی عمل می کند. صرف نظر از برنامه های کاربردی در پردازش تصویر، اخیراً علاقه به استفاده از فیلترهای فازی برای رفع مشکلات پردازش سیگنال های عمومی نیز وجود دارد. [16] و [18]

## II. الگوریتم پیشنهادی

ما برای حذف نویزهای تصاویری که به وسیله نویز ضربه (نمک-فلل) معیوب شده اند یک فیلتر فازی و فقی (انطباقی) تکرارشونده دومرحله ای را پیشنهاد می دهیم.

ابتدا به طور مختصر میانگین معروف  $\alpha$ -trimmed را شرح می دهیم. [32]. فیلر میانگین  $\alpha$ -trimmed به عنوان یک معیار مرکزگرا (احتمال مطابقت داده با مقادیر مورد انتظار) موثرتر از اندازه گیری های میانگین یا میانه کلاسیک در زمینه حذف نویز ضربه عمل می کند. [33]. ما این اندازه (معیار) را به یک فرم دیگر که مرتبط با الگوریتم خودمان است، بازنویسی می کنیم و آن را میانگین  $k$ -middle می نامیم.

میانگین  $\alpha$ -trimmed (تنظیم شده)، میانگین یک مجموعه را پس از برش (تنظیم) به اندازه  $a/2$  از بالا و پایین آن مجموعه، محاسبه می کند. میانگین  $\alpha$ -trimmed مجموعه  $A = \{a_1, a_2, \dots, a_n\}$  که شامل  $n$  عضو است به صورت زیر است:

$$\mu_\alpha = \frac{1}{n - \alpha} \sum_{i=(\alpha/2)+1}^{n-(\alpha/2)-1} a_i \quad \text{رابطه (1)}$$

که در آن  $a_i$ ،  $i$  امین مرتبه آماری از عضوهای مجموعه  $A$  است.

### ***A* میانگین $k$ -middle**

میانگین  $k$ -middle را تعریف می کنیم که با  $\mathcal{M}_k(A)$  مشخص شده است.

$$\mathcal{M}_k(A) = \begin{cases} \frac{1}{2k-1} \sum_{i=h-k+1}^{h+k-1} a_i, & n \text{ فرد} \\ \frac{1}{2k} \sum_{i=h-k+1}^{h+k} a_i, & n \text{ زوج} \end{cases} \quad \text{رابطه (2)}$$

که در آن  $a_i$ ،  $i$  امین مرتبه آماری از مجموعه  $n$  عضوی  $A$  است.

برای  $k=1$  محاسبات همانند میانگین کلاسیک و برای  $k=h$  محاسبات شکل میانگین را می گیرد.

### ***B* الگوریتم برای آشکارسازی و حذف نویز پیکسل های نویزدار**

از آنجایی که ما با حالت خاصی از نویز نمک-فلفل (ضربه) سروکار داریم، اگر شدت یک پیکسل به طور کامل تیره یا کاملاً روشن نباشد، نیازی به بررسی های بیشتر نداریم و مقدار شدت فعلی آن همچنان باقی می ماند. به عبارت دیگر ما فقط در صورتی پیکسل را بررسی می کنیم که معیوب باشد.

برای هر پیکسل  $p_{ij}$  در موقعیت  $(i,j)$  در یک تصویر مشخص، ما یک ناحیه همسایگی  $(2M+1) \times (2M+1)$  با  $p_{ij}$  در مرکز آن انتخاب می کنیم. ما مجموعه شدت (مقادیر) پیکسل در این ناحیه را  $R_{ij}^M$  می نامیم.

$$R_{ij}^M = \{p_{i+k,j+l} - M \leq k, l \leq M\} \quad \text{رابطه (3)}$$

سپس مجموعه فازی  $u_{ij}^M$  را تعریف می کنیم که در آن هر عضو  $p_k$  از مجموعه  $R_{ij}^M$  با تابع عضویت گاوسی زیر در رابطه است.

$$m_{u_{ij}^M}(p_k) = e^{-(p_k - \mu_{ij}^M)/2(\sigma_{ij}^M)^2} \quad \text{رابطه (4)}$$

پارامترهای  $\mu_{ij}^M$  و  $\sigma_{ij}^M$  در تابع عضویت از طریق زیر محاسبه می شوند.

$$\mu_{ij}^M = \mathcal{M}_{k_1}(R_{ij}^M) \quad \text{رابطه (5)}$$

$$\sigma_{ij}^M = \mathcal{M}_{k_2}(\Lambda_{ij}^M) \quad \text{رابطه (6)}$$

که در آن  $\mathcal{M}_k$  حاصل میانگین k-middle که در بخش قبلی تعریف شده است و مجموعه  $\Lambda_{ij}^M$  به صورت زیر تعریف می شود.

$$\Lambda_{ij}^M = \{(p_k - \mu_{ij}^M)^2, \forall p_k \in R_{ij}^M\} \quad \text{رابطه (7)}$$

مراحل اصلی و کلیدی الگوریتم ما به صورت زیر است:

### 1. شروع پایه ریزی پارامترها: برای نویزگیری، ما بایستی به تعداد N نويز سالم (بدون عیب) در

صفحه داشته باشیم. تعداد پیکسل های نويز (دارای نويز) که در هر تکرار شناسایی می شوند برای توقف الگوریتم تکرارشونده ما بکار گرفته خواهد شد و در نتیجه ما متغیر dm را برای نگه داشتن این مقدار در اجرای ام تنظیم می کنیم (ایجاد می کنیم). پارامتر نیم-صفحه (M) برای شروع بر روی آن تنظیم می شود که به یک پنجره (صفحه) با اندازه 3×3 دلالت دارد. Tmax و Tmin، حدود بالایی و پایینی اولیه برای T، و آستانه انطباقی برای تابع عضویت گاوسی است.

### 2. شناسایی پیکسل های نويزدار و سالم: در صورتی که پیکسل مرکزی (pij) در پنجره در مقدار

نهایی خود نباشد، ما مقدار آن را حذف می کنیم. به عبارت دیگر، لازم است ما تعیین کنیم که پیکسل pij، پیکسل معیوب هست یا نه.

#### a. فاصله امن: ما انتظار داریم که پیکسل هایی که از pij فاصله زیادی دارند بر روی شدت آن

تاثیر زیادی نداشته باشند. لذا ما متغیر Smax را که به عنوان حد بالای M عمل میکند معرفی می کنیم. به عنوان آخرین راه حل، ما اندازه پنجره را فراتر از محدوده امن بالاتر می بریم. اما اولویت ما این است که فراتر از این اندازه نرویم. این کار اولویت ما برای استفاده از پیکسل های نزدیکتر را تا حد ممکن برای تخمین مقادیر نويززدایی شده تضمین می کند.

#### b. تابع عضویت گاوسی: با استفاده از (5) و (7) پارامترهای $\mu_{ij}^M$ و $\sigma_{ij}^M$ از تابع عضویت

گاوسی را محاسبه می کنیم. اگر معلوم گردد که انحراف  $\sigma_{ij}^M$  از یک حد خیلی کم (اپسیلون) پایین تر است یعنی اینکه پنجره شامل پیکسل هایی با شدت خیلی نزدیک به پیکسل pij است. بنابراین ما به راحتی می توانیم مقدار pij را به  $\mu_{ij}^M$  تنظیم کنیم. این کار از نامعین بودن (تقسیم شدن بر صفر) که ممکن است به خاطر یکنواختی شدت ناحیه که منجر به صفر شدن  $M_{ij}$  می شود جلوگیری می کند. اگر مرتبه عضویت  $m_{u_{ij}}^M(p_{ij})$ ، pij بالاتر

از حد آستانه  $T$  باشد، در این صورت فرض می شود که پیکسل  $p_{ij}$  سالم است و می توانیم مقدار  $p_{ij}$  را حفظ کنیم. (شکل 1 را ببینید)

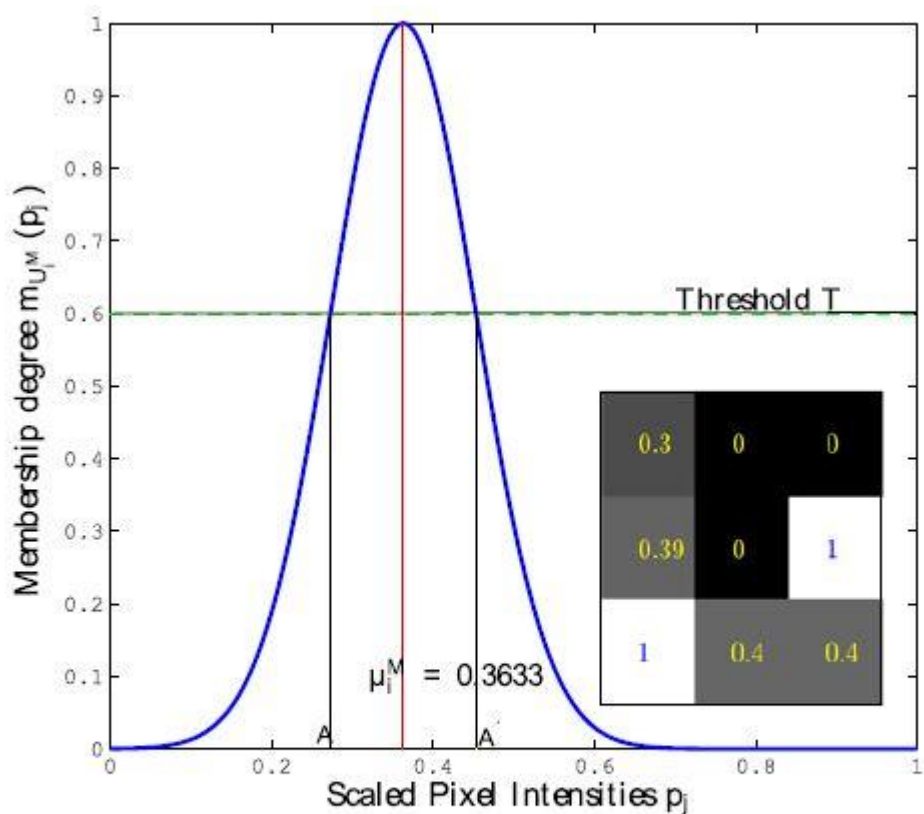


Fig. 1. Gaussian membership function for a sample  $3 \times 3$  window (inset). Pixel intensity values lying within  $A$  and  $A'$  (with a Gaussian membership degree above  $T$ ) or are not in  $\{0, 1\}$  are "good" pixels.

تابع عضویت گاوسی برای یک پنجره نمونه  $3 \times 3$ . مقادیر شدت پیکسل های که محدود در  $A, A'$  (با درجه عضویت گاوسی بالاتر از  $T$ ) هستند یا در  $(0,1)$  نیستند پیکسل های سالم محسوب می شوند.

**c. مجموعه خوب (سالم):**  $\zeta$  در صورتی که  $m_{u_{ij}^M}(p_{ij}) \leq T$  باشد احتمالاً پیکسل  $p_{ij}$  دارای نویز است. ازین جهت لازم است که تخمینی از شدت تراکم پیکسل های مجاور داشته باشیم. برای همین یک مجموعه  $\zeta \subseteq R_{ij}^M$  در همسایگی پیکسل های سالم و بدون نویز ایجاد می کنیم. مجموعه  $\zeta$  به صورت زیر محاسبه می گردد.

$$\zeta = \{p_k | m_{u_{ij}^M}(p_{ij}) > T \vee (p_k \notin \{0,1\}), \forall p_k \in R_{ij}^M\} \quad \text{رابطه (8)}$$

**d. کاهش (تضعیف) انطباق پارامترها:** اگر تعداد عضوهای مجموعه پیکسل های خوب (سالم)  $|\zeta|$  از حداقل تعداد پیکسل های خوب (سالم) مورد نیاز ما ( $N$ ) کمتر باشد، دراین صورت

ما خواستار کاهش پارامترهای خود خواهیم بود. (پارامترهای والیه خود را کاهش می دهیم). انتخاب اینکه کدام پارامتر تضعیف یا کاسته شود و در چه مرتبه و درجه ای کاسته شود، تابع مجموعه ای از قوانین است. بحث و تلاش اصلی ما در اینجا این است که وقتی برای نویزگیری تعداد عضوهای مجموعه پیکسل های خوب و سالم ( $N$ ) به اندازه پنجره تطبیقی رسید، لازم است در طول جستجو برای پیکسل های خوب (سالم) نزدیک پیکسل مرکزی بمانیم. بنابراین هنگام کاهش پارامترها، ابتدا آستانه  $T_{min}$  را کاهش می دهیم (با اندازه پنجره ثابت  $3 \times 3$ ). در صورتی که این کاهش آستانه منجر به ناکافی بودن تعداد پیکسل های خوب شود، اندازه پنجره را افزایش می دهیم. البته این افزایش اندازه یک حد بالا دارد که از بیش از حد دور شدن ما جلوگیری می کند. اگر باز هم تعداد پیکسل های خوب ناکافی بود، تقاضای خود برای تعداد  $N$  پیکسل خوب را کاهش می دهیم. که این کار از تعداد نمونه هایی که شدت نويز زدایی آن ها را تخمین می زنیم می کاهد اما همچنان مایل است که اندازه پنجره را بیش از حد از پیکسل مرکزی دور کند و در نتیجه از نمونه های نامناسب استفاده می کند. در این مورد هرگاه همچنان قادر به یافتن هرگونه پیکسل سالمی نشدیم، مجبور به افزایش اندازه پنجره حتی بیشتر از حد بالایی هستیم. بنابراین طبق آنچه که گفته شد، سه قانون حذف و کاستن که اتفاق افتاد را فهرست می کنیم.

**قانون 1:** در صورتی که قادر به یافتن  $N$  پیکسل سالم در پنجره نشدیم، ابتدا آستانه عضویت  $T$  را از مقدار اولیه آن یعنی  $T_{max}$  به  $T_{min}$  به صورت گام به گام پایین می آوریم. (اندازه گام  $\alpha$  ممکن است متغیر باشد) تا زمان رسیدن به  $|S| > N$  ادامه می دهیم. پس از آن حداقل تعداد پیکسل های خوب (سالم) مورد نیازمان برای نويز زدایی پیکسل  $P_{ij}$  را داریم.

**قانون 2:** اگر  $|S|$  باز هم کمتر از  $N$  باشد پارامتر نیم-سایز پنجره ( $M$ ) را افزایش می دهیم طوری که اندازه پنجره از  $3 \times 3$  به  $(2S_{max} + 1) \times (2S_{max} + 1)$  به صورت گام به گام افزایش یابد و تا زمان رسیدن به  $|S| > N$  ادامه می دهیم.

**قانون 3:** اگر  $|S|$  همچنان کمتر از  $N$  باشد پارامتر  $N$  را به صورت گام به گام و مرحله ای کاهش می دهیم. اگر این فرآیند تکراری باعث شود  $N$  به کمتر از 1 برسد،  $S_{max}$  را به طور پیوسته افزایش می دهیم و  $N$  را برای نويز زدایی  $p_{ij}$  برای همیشه روی 1 تنظیم می کنیم. می توانستیم بعد از  $N$  را بر روی 8 یا اعداد کمتری تنظیم کنیم اما واقعیتی که ما را مجبور کرده به فراتر از  $S_{max}$  برویم این است که در ناحیه فعلی گنگی و ابهام قابل توجهی وجود دارد یا به دلیل نويز بالاست یا ماهیت خود تصویر. تنظیم  $N$  به مقادیر بالاتر به سود ما نخواهد بود و ممکن است منجر به افزایش مضاعف  $S_{max}$  شود که نتیجه آن اعمال نفوذ دورتر و بی مورد پیکسل ها به محض نويز زدایی  $p_{ij}$  است.

### 3. نويزگيري پيکسل های نويزدار: ما تراکم پيکسل نويززدايي شده را از تراکم های پيکسل موجود

در مجموعه خوب (سالم)  $|\mathcal{S}|$ ، تخمين می زنيم. اعضای مجموعه  $\mathcal{S}$  را با تابع وزنی ساده معکوس فاصله، وزن دهی می کنيم. که اين تابع وزن های پايين تر را به پيکسل های دورتر از  $p_{ij}$  و وزن های بالاتر را به پيکسل های نزديک تر به  $p_{ij}$  اختصاص می دهد. وزن  $w_k$  برای یک پيکسل سالم (خوب) در موقعيت  $(i', j')$  در تصوير توسط رابطه (9) بدست می آيد.

$$w_k = \frac{1}{((i - i')^2 + (j - j')^2)^p} \quad \text{رابطه (9)}$$

که در آن  $p$  پارامتر قابل تنظيم است. روی هم رفته از روی مشاهدات تجربی متوجه شدیم که  $p=2$  نتایج بهتری می دهد. وزن های  $w_k$  به هنجار شده و به يکنواختی می رسد (به توزيع يکنواخت می رسد). بنابراین تراکم پيکسل نويززدايي شده  $p_{ij}^{new}$  با فرمول زیر قابل محاسبه است.

$$p_{ij}^{new} = \frac{1}{W} \sum_{\forall p_k \in \mathcal{S}} w_k p_k \quad \text{رابطه (10)}$$

که در آن  $W$  جمله به هنجار شده (نرماليزه شده) است که از رابطه زیر به دست می آيد.

$$W = \sum_{k=1}^{|\mathcal{S}|} w_k \quad \text{رابطه (11)}$$

در الگوريتم I طی یک شبه دستورالعمل (شبه برنامه کدنویسی شده) همه طرز کار را توصيف می کنيم.

---

**Algorithm 1** : Detection and Denoising of Noisy Pixels

---

**procedure** DENOISENOISYPIXELS(Image  $I$ )

```
1. for all pixels  $p_{ij}$  in  $I$  do
  1. if  $p_{ij} \notin \{0, 1\}$  then
    1. retain value of  $p_{ij}$ 
    2. continue;
  2.  $T \leftarrow T_{init}$ 
  3.  $N \leftarrow N_{init}$ 
  4. while true:
    1. Compute  $R_{ij}^M$  using Eq. (3).
    2. Compute  $\mu_{ij}^M$  and  $\sigma_{ij}^M$  using Eqs. (5-7).
    3. if  $\sigma_{ij}^M \leq \epsilon$  then
      1.  $p_{ij} \leftarrow \mu_{ij}^M$ 
      2. break;
    4. if  $m_{\mathcal{U}^M}(p_{ij}) > T$  then
      1. retain value of  $p_{ij}$ 
      2. break;
    5. Compute  $\mathcal{G}$  using Eq. (7).
    6.  $\eta \leftarrow |\mathcal{G}|$ .
    7. if  $\eta < N$  AND  $T > T_{min}$  then
      1.  $T \leftarrow T - \alpha$ 
      2. continue;
    8. if  $\eta < N$  AND  $M < S_{max}$  then
      1.  $M \leftarrow M + 1$ 
      2. continue;
    9. if  $\eta < N$  AND  $M \geq S_{max}$  AND  $T \leq T_{min}$  then
      1.  $N \leftarrow N - 1$ 
      2. if  $N \leq 1$  then
        1.  $S_{max} \leftarrow S_{max} + 1$ 
        2.  $N \leftarrow 1$ 
    10.  $d_m \leftarrow d_m + 1$ 
    11. Compute denoised pixel intensity of  $p_{ij}$ 
        from  $\mathcal{G}$  using Eqs. (9-11)
    12. break;
  end while
end for
end procedure
```

---

**4. معیار شروع و توقف:** الگوریتم I به صورت تکرارشونده و به دنبال آن یک معیار متوقف کننده اجرا می شود. در نتیجه تکرار الگوریتم به صورت پی در پی، تفاوت تعداد تشخیص های پیکسل های نویزدار ( $dm$ ) را می یابیم. تا زمانی که  $dm$  به کمتر از یک آستانه کوچک برسد ادامه می دهیم. این آستانه را به صورت 0.05% تعداد کل پیکسل های تصویر تنظیم می کنیم.



### III. نتایج

**A. تاثیر پارامترهای متغیر  $K_1$  و  $K_2$**  ابتدا عملکرد الگوریتممان را در مقادیر متغیر  $K_1$  و  $K_2$  برای سطوح مختلف نویز بررسی می کنیم. تصاویر تست نمونه انتخابی ما Baboon (Mandrill), Lena, Peppers, and Bridge هستند. الگوریتممان را برای مقادیر 1 و 2 و 3 و 4 یا  $h$  برای  $K_1$ ,  $K_2$  تنظیم می کنیم. (رابطه (2) را ببینید). برای هر تصویر نمونه، آزمایشی را برای سه سطح از عیب های ناشی از نویز فلفل - نمک انجام می دهیم. این سه سطح 10٪، 50٪ و 90٪ هستند. همه نتایج گزارش شده در جدول I به ترتیب از 10 آزمایش به دست آمده است.

TABLE I  
PSNR (dB) FOR THE VARYING LEVELS OF NOISE WITH RESPECT TO DIFFERENT PARAMETER SETTINGS OF  $K_1$  AND  $K_2$

(a) Baboon, 10% noise						(b) Baboon, 50% noise						(c) Baboon, 90% noise					
$K_1 \backslash K_2$	1	2	3	4	$h$	$K_1 \backslash K_2$	1	2	3	4	$h$	$K_1 \backslash K_2$	1	2	3	4	$h$
1	32.81	32.82	32.76	32.75	32.81	1	24.74	24.22	24.05	24.03	23.92	1	07.60	08.39	08.68	08.71	08.40
2	32.82	32.85	<b>32.88</b>	32.84	32.86	2	24.84	24.85	24.62	24.60	24.27	2	19.58	<b>19.76</b>	12.61	11.20	09.03
3	32.75	32.79	32.74	32.77	32.80	3	24.85	24.82	<b>24.88</b>	24.81	24.42	3	19.47	19.43	19.72	13.77	10.17
4	32.66	32.61	32.65	32.62	32.71	4	24.83	24.85	24.84	24.84	24.61	4	19.15	19.10	19.14	19.69	10.70
$h$	32.60	32.60	32.60	32.57	32.65	$h$	24.82	24.84	23.86	24.87	24.61	$h$	19.64	19.63	19.64	19.66	19.67

(d) Lena, 10% noise						(e) Lena, 50% noise						(f) Lena, 90% noise					
$K_1 \backslash K_2$	1	2	3	4	$h$	$K_1 \backslash K_2$	1	2	3	4	$h$	$K_1 \backslash K_2$	1	2	3	4	$h$
1	42.87	42.81	42.82	42.74	42.99	1	33.40	30.38	29.74	29.58	28.99	1	07.58	08.40	08.70	08.78	08.54
2	42.83	42.81	42.79	42.86	43.00	2	34.08	34.13	32.64	32.11	30.59	2	26.45	26.60	13.12	11.63	09.22
3	42.82	42.81	42.82	42.70	43.05	3	34.03	34.09	34.18	33.75	30.94	3	25.31	25.33	<b>26.80</b>	14.44	10.58
4	42.64	42.72	42.75	42.80	43.04	4	34.01	34.08	34.03	34.16	31.88	4	24.43	24.43	24.47	26.79	11.09
$h$	42.30	42.42	42.41	42.50	<b>43.07</b>	$h$	34.11	34.16	34.14	<b>34.19</b>	34.16	$h$	26.76	26.65	26.73	26.75	26.73

(g) Peppers, 10% noise						(h) Peppers, 50% noise						(i) Peppers, 90% noise					
$K_1 \backslash K_2$	1	2	3	4	$h$	$K_1 \backslash K_2$	1	2	3	4	$h$	$K_1 \backslash K_2$	1	2	3	4	$h$
1	<b>41.47</b>	41.13	41.12	41.02	41.05	1	32.35	29.68	29.10	28.84	28.37	1	07.45	08.31	08.60	08.65	08.40
2	41.45	41.30	41.33	41.37	41.23	2	32.44	32.52	31.41	31.17	30.02	2	23.61	24.12	13.07	11.48	09.11
3	41.30	41.24	41.24	41.31	41.16	3	32.53	32.50	32.59	32.13	30.24	3	23.95	24.23	25.09	14.05	10.49
4	41.26	41.20	41.24	41.22	41.14	4	32.55	32.56	32.51	32.64	31.14	4	24.22	24.11	24.31	25.44	11.09
$h$	40.74	40.86	40.92	40.84	41.19	$h$	32.56	32.61	32.56	32.63	<b>32.69</b>	$h$	25.80	25.88	<b>25.91</b>	25.82	25.79

(j) Bridge, 10% noise						(k) Bridge, 50% noise						(l) Bridge, 90% noise					
$K_1 \backslash K_2$	1	2	3	4	$h$	$K_1 \backslash K_2$	1	2	3	4	$h$	$K_1 \backslash K_2$	1	2	3	4	$h$
1	<b>35.45</b>	35.37	35.36	35.41	35.43	1	27.13	26.24	25.94	25.80	25.57	1	7.33	8.15	8.49	8.48	8.26
2	35.29	35.26	35.18	35.29	35.36	2	27.24	<b>27.29</b>	26.90	26.80	26.17	2	20.16	20.75	12.69	11.09	8.98
3	35.18	35.17	35.08	34.91	35.27	3	27.24	27.23	27.24	27.14	26.39	3	20.15	20.17	21.14	13.52	10.21
4	34.96	34.95	34.92	34.24	34.80	4	27.28	27.28	27.26	27.13	26.70	4	20.33	20.34	20.39	21.24	10.76
$h$	33.87	33.82	33.90	33.87	33.63	$h$	27.01	27.03	27.00	27.00	27.16	$h$	<b>21.36</b>	21.25	21.34	21.33	21.33

از این آزمایشات این گونه بدست می آید که:

1. برای نویز بالا به نظر می رسد  $K_1=h$  انتخاب خوبی است و برای نویز کم (پایین)  $K_1=1$  نتایج خوبی را به دنبال دارد.
2.  $K_1=1$  (اندازه گیری میانه) منجر به عملکرد بد برای عیوب ایجاد شده با سطوح نویز بالا می شود. این به این معنی است که اندازه گیری میانه برآوردکننده بدی برای مقادیر تراکم پیکسل در مورد نویز های شدید است.

3. در کل به نظر می رسد که متغیر  $K2$  تاثیر زیادی نداشته باشد. با این حال،  $K2$  با مقدار 4

نتایج بهتری را نسبت به  $K2=h$  می دهد. (در اندازه گیری میانه) به جز در مورد Lena به طور کلی مشاهده می شود که بهترین انتخاب برای پارامترهای  $K1$  و  $K2$  به شدت به ویژگی های خاص تصویر وابسته است. یادآور می شویم که در کل تنظیمات پارامترهای  $K1=3$  و  $K2=3$  برای دستیابی به عملکرد خوب استفاده می شود و ما برای تحلیل های مقایسه ای از این تنظیم استفاده می کنیم.

### B. تاثیر $Tmin$ ، $Tmax$ و $Smax$ :

در جدول II، میانگین مقادیر PSNR برای تنظیم 9 تصویر استاندارد با مقادیر متغیر برای  $Tmin$ ،  $Tmax$  و  $Smax$  را گزارش کرده ایم. هر تصویر با سه سطح از سطوح 10٪، 50٪ و 90٪ دچار عیب شده اند. و نتایج برای همه سطوح نويز به تعداد 10 بار میانگین گیری شده اند.

TABLE II  
VARYING  $S_{max}$ ,  $T_{max}$ , AND  $T_{min}$ .  $N_{init}$  IS FIXED AT 8 TO ENSURE THAT THE ADAPTIVE STAGES OF THE ALGORITHM ARE RUN MULTIPLE TIMES

$T_{min} \backslash T_{max}$	0.2	0.4	0.6	0.8	0.999
0.2	-	24.6129	24.6168	24.6119	24.6390
0.4	-	-	27.4616	27.3907	27.5251
0.6	-	-	-	29.8964	29.8985
0.8	-	-	-	-	<b>29.9178</b>

$T_{min} \backslash T_{max}$	0.2	0.4	0.6	0.8	0.999
0.2	-	24.4535	24.4131	24.4502	24.4813
0.4	-	-	26.9924	26.8205	27.0244
0.6	-	-	-	29.8967	29.8852
0.8	-	-	-	-	29.8777

$T_{min} \backslash T_{max}$	0.2	0.4	0.6	0.8	0.999
0.2	-	24.4029	24.3932	24.3863	24.4316
0.4	-	-	26.6629	26.5822	26.6450
0.6	-	-	-	29.8282	29.7829
0.8	-	-	-	-	29.8132

$T_{min} \backslash T_{max}$	0.2	0.4	0.6	0.8	0.999
0.2	-	24.3800	24.3706	24.4102	24.4035
0.4	-	-	26.4991	26.5005	26.6105
0.6	-	-	-	29.7522	29.7995
0.8	-	-	-	-	29.7551

$N_{init}$  is fixed at 8 to ensure that the adaptive stages of the algorithm are run multiple times.

برای این آزمایش مشاهده شد که بهترین تنظیمات عبارتند از:

$Smax=2$  و  $Tmin=0.8$  و  $Tmax=0.999$ . همچنین یادآور می شویم که افزایش  $Smax$  منجر به عملکرد ضعیف تر می شود که تایید کننده این است که افزایش سایز پنجره عملکرد ضعیف تری را نتیجه می دهد.

### C. انتخاب $Ninit$

$Ninit$  تعداد اولیه پیکسل های خوب (سالم) است که در پنجره جستجو می شود. از جدول III (همه مقادیر، میانگین نتایج 10 آزمایش اند) این گونه مشاهده می شود که :

1. به طور کلی افزایش مقدار  $N_{init}$  در صورتی که میزان نويز کم باشد (20٪) منجر به نويززدایی

ضعیف تر تصویر می شود.

2. برعکس، مقادیر بالاتر  $N_{init}$ ، در مواردی که میزان نويز بالاست (80٪) منجر به عملکرد بهتری

می شود.

3. همچنین با افزایش مقادیر  $N_{init}$ ، زمان اجرا نیز افزایش می یابد. این به این علت است که تعداد بالاتری از پیکسل های خوب (سالم) در هر نویززدایی لازم است و در نتیجه زمان اجرای الگوریتم I بیشتر خواهد بود.

در تحلیل های مقایسه ای برای نویززدایی و زمان اجرا، همیشه  $N_{init}=1$  در نظر می گیریم.

TABLE III  
PSNR AND RUNTIME FOR A SET OF NINE IMAGES WITH VARYING VALUES OF  $N_{init}$

	$N_{init}$	20% noise		50% noise		80% noise	
		PSNR(dB)	Time(Sec)	PSNR(dB)	Time(Sec)	PSNR(dB)	Time(Sec)
Baboon	1	29.75	5.25	24.84	12.76	20.73	28.05
	4	29.70	5.59	24.87	23.60	21.09	69.49
	8	29.57	13.73	24.88	40.62	21.14	110.10
Barbara	1	31.95	4.86	26.74	12.58	22.78	28.31
	4	31.68	4.97	26.83	23.11	23.10	69.27
	8	31.52	11.81	26.85	39.75	23.18	110.17
Boat	1	36.03	4.77	30.69	12.67	25.88	28.04
	4	36.01	4.88	30.71	23.05	26.12	69.10
	8	35.74	11.32	30.68	39.56	26.13	110.31
Bridge	1	31.59	8.99	27.01	16.88	22.92	32.88
	4	32.11	10.22	27.27	28.35	23.03	74.62
	8	32.13	16.95	27.54	44.86	23.15	115.25
Fingerprint	1	35.71	5.25	28.97	12.78	22.85	28.36
	4	35.61	5.45	29.12	23.55	23.16	69.77
	8	35.06	14.08	29.16	40.80	23.21	110.77
Flintstones	1	32.39	4.79	26.41	12.45	20.79	28.04
	4	32.32	4.96	26.38	22.36	20.75	66.46
	8	31.89	11.36	26.36	37.09	20.67	102.52
House	1	38.97	1.09	33.19	2.98	27.82	6.70
	4	38.88	1.12	33.28	5.49	28.11	16.21
	8	38.57	2.39	33.35	9.19	28.23	25.48
Lena	1	39.92	4.61	34.10	12.65	28.84	28.41
	4	39.80	4.75	34.19	23.10	29.19	69.29
	8	39.40	10.47	34.24	39.59	29.31	109.98
Peppers	1	37.99	4.60	32.34	12.58	27.54	28.07
	4	37.86	4.84	32.56	22.93	27.87	68.76
	8	37.83	10.68	32.59	39.41	27.98	108.76

### D. مقایسه با فیلترهای مدرن و پیشرفته (state of the art)

جدول IV، مقایسه ای از عملکرد الگوریتم ما و سایر الگوریتم های جدید و پیشرفته است - میانگین وقتی با تنظیم نگه دارنده لبه [19] (AM-EPR)، روش تشخیص نویز متمایز مرزی [20] (BDND)، روش میانگین سریع [21] (FM)، روش شبکه عصبی موجک [22] (WNN)، یک فیلتر با استفاده از ارزیاب واریانس-S [23] (PWS)، فیلتر مبتنی بر افزایش کنتراست [24] (CEF)، فیلتر تغییرپذیری وقتی فاصله محور [25]، روش پچ محور [26] و فیلتر مدل ابری [27] (CM)، در شکل های 2 و 3 عملکرد کیفی الگوریتم خودمان را در تصاویر خاکستری نشان داده ایم.



TABLE IV  
COMPARISON OF THE PERFORMANCE OF OUR ALGORITHM WITH SOME STATE-OF-THE-ART FILTERS IN TERMS OF PSNR (IN dB)

		FM	CEF	PWS	WAV	AM-EPR	PB	BDND	CM	SA-TV	Our Algorithm
Lena	20%	37.05	37.46	36.85	38.12	38.21	38.31	38.52	39.42	39.20	<b>39.92</b>
	50%	29.81	30.71	29.57	30.27	33.46	32.04	32.74	33.57	33.88	<b>34.10</b>
	80%	23.11	23.22	22.68	23.44	27.16	25.97	27.11	28.45	27.14	<b>28.84</b>
Bridge	20%	30.41	28.47	29.18	29.38	32.12	27.53	30.66	31.35	<b>31.98</b>	31.59
	50%	24.24	22.35	22.79	24.72	26.64	23.75	25.22	26.52	26.89	<b>27.01</b>
	80%	20.67	19.52	20.03	20.01	21.98	19.45	21.39	22.28	22.41	<b>22.92</b>
Peppers	20%	36.21	35.03	35.46	37.27	37.45	36.32	34.44	37.54	36.87	<b>37.99</b>
	50%	29.53	30.38	29.26	30.49	31.25	29.25	30.23	32.03	31.62	<b>32.34</b>
	80%	22.21	23.65	22.84	22.95	27.32	25.64	26.61	27.46	26.42	<b>27.54</b>
Baboon	20%	27.22	26.85	26.52	27.49	29.87	24.22	27.73	28.47	28.49	<b>29.75</b>
	50%	22.26	21.93	20.42	22.31	24.52	21.27	23.46	24.05	23.91	<b>24.84</b>
	80%	18.69	17.60	17.86	18.35	19.73	17.38	19.92	20.36	20.59	<b>20.73</b>
Barbara	20%	29.46	29.58	28.72	30.14	29.72	29.24	29.85	30.78	30.70	<b>31.95</b>
	50%	23.46	23.37	22.69	24.84	25.33	23.49	25.17	26.10	25.91	<b>26.74</b>
	80%	19.35	19.31	18.91	19.96	21.41	21.64	21.74	22.54	22.66	<b>22.78</b>
Boat	20%	34.75	30.87	33.78	34.53	34.89	32.73	34.83	35.31	35.97	<b>36.03</b>
	50%	27.96	25.65	26.80	27.34	29.34	27.83	29.68	29.99	30.38	<b>30.69</b>
	80%	23.65	21.46	22.50	22.87	24.75	22.29	24.93	25.58	25.18	<b>25.88</b>
Fingerprint	20%	34.14	33.89	33.68	34.18	33.58	29.50	33.26	34.85	34.53	<b>35.71</b>
	50%	25.44	24.25	24.79	25.32	27.70	23.36	27.07	28.31	27.80	<b>28.97</b>
	80%	19.82	18.54	19.08	19.47	21.49	18.75	21.48	22.39	20.39	<b>22.85</b>
Flintstones	20%	30.85	31.13	30.48	31.25	31.38	31.14	31.26	32.00	32.19	<b>32.39</b>
	50%	23.53	23.18	22.99	23.90	25.62	21.92	25.39	26.03	25.70	<b>26.41</b>
	80%	17.75	17.66	17.39	18.33	19.20	18.39	19.71	20.56	19.54	<b>20.79</b>
House	20%	37.84	37.91	37.74	38.15	37.42	37.45	37.23	38.32	38.66	<b>38.97</b>
	50%	29.45	29.52	29.49	29.94	31.51	30.70	31.72	32.45	32.53	<b>33.19</b>
	80%	22.65	22.63	22.54	22.88	25.45	24.84	25.81	27.52	26.16	<b>27.82</b>



Fig. 2. Denoising grayscale *Barbara* and *Boat* from 20% and 80% salt-and-pepper corruption, respectively. (a) *Barbara*. (b) 20% noise. (c) Result. (d) *Boat*. (e) 80% noise. (f) Result.

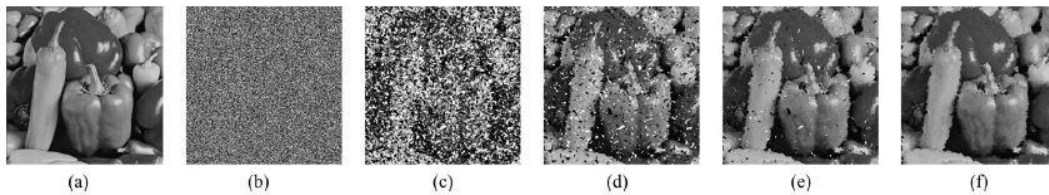


Fig. 3. Denoising *Peppers* corrupted with a high 97% level of impulse noise. The final PSNR (iteration 12) is 21.95dB. (a) *Peppers*. (b) 97% noise. (c) Iteration 1. (d) Iteration 3. (e) Iteration 5. (f) Iteration 12.

فیلتر FM همه پیکسل ها را با شدت 0 و 255 در یک پنجره تشخیص اندازه، با سایز ثابت  $3 \times 3$  به عنوان پیکسل نویزدار بررسی می کند. و سپس مقادیر متوسط یا مقادیر همسایگی چپ را برای جایگزینی پیکسل نویز استفاده می کند. فیلتر WNN، از یک شبکه عصبی موجک برای تشخیص پیکسل های نویزدار استفاده می کند. برای نویزدایی، یک ضریب Cij محاسبه می شود که معیار اطمینانی است برای این که کدام پیکسل نویزدار است. در نهایت، پیکسل با یک ترکیب خطی، جایگزین مقدار اصلی و مقدار متوسط می شود. جایی که وزن برای هر جمله برابر Cij و  $(1-Cij)$  است. در فیلتر CEF، ابتدا یک تبدیل غیرخطی بر روی شدت پیکسل ها در یک پنجره انجام می پذیرد. (پنجره با سایز  $m \times m$ ). این مرحله پس از مرحله

شناسایی انجام می گیرد که در آن کمترین مقادیر تبدیل شده جمع و توسط آستانه  $T$  محدود می شود تا زمانی که یک شاخص باینری نشان دهد که پیکسل نویزی است. و پس از آن مرحله فیلتر کردن با WMF انجام می پذیرد. برای یک تحلیل مقایسه ای ما پارامترها را این گونه تنظیم می کنیم.  $T=35$  و  $p=13$  و  $m=5$ . در فیلتر PWS، مرحله شناسایی از یک مرتب کننده وابسته استفاده می کند. (پارامترها براساس پیشنهاد نویسندگان تعیین شده اند یعنی  $\delta = 12, s^{(0)} = 2.6, s^{(k+1)} = s^{(k)} - 0.3$ ). نویسندگان براساس سطح نویز اندازه پنجره ( $K$ ) را تغییر داده اند. هرچند برای مقایسه، ما اندازه پنجره را روی  $5 \times 5$  ثابت کرده ایم. برای نویزدایی با  $\alpha = 1.3$  از یک تابع پتانسیل حفظ-لبه  $\psi(x) = |x|^\alpha$  استفاده می شود. فیلتر AM-EPR ترکیبی از فیلتر AM [28] و روش تغییر است [29]. و در حفظ جزئیات بهتر از فیلتر AM عمل می کند. به پیشنهاد نویسندگان،  $W_{max}$  روی 39،  $\alpha$  روی 1.3 و  $\beta$  روی 5 تنظیم می شود. فیلتر PB یک فیلتر مبتنی بر پیچ (دسته) است که اطلاعات (داده ها) را از دسته های مشابه موجود در یک تصویر بکار می برد. این فیلتر از اختلاف مطلق ترتیب (مرتب) آماری (ROAD) برای تخمین نرخ نویز استفاده می کند. پارامترهای زیر براساس مقادیر پیشنهاد شده از طرف نویسندگان تنظیم شده اند. آستانه اختلاف مطلق مرتب آماری (ROAD)،  $3 = \text{نیم-سایز دسته ها}$  و  $7 = \text{نیم-سایز همسایگی جستجو}$  و  $\tau = 70$  و  $2 = \text{تعداد تکرار}$ . فیلتر BDND دو اندازه ثابت پنجره  $(21 \times 21)$  و  $(3 \times 3)$  را برای شناسایی پیکسل های نویزدار و یک پنجره افقی (انطباقی) برای حذف نویز بکار می گیرد. فیلتر CM، حداقل تعداد پیکسل های سالم است که مشابه  $\eta$  در الگوریتم ما می باشد. به منظور مقایسه، طبق پیشنهاد نویسندگان، مقدار پارامتر  $\eta$  را در آزمایشات خود بر روی 1 تنظیم کردیم. الگوریتم SA-TV براساس روش نرم وزن دهی دوباره پایه گذاری شده است و مسئله تنوع کل  $L-1$  را برای پیکسل های نویزدار حل می کند. پارامترها براساس پیشنهاد نویسندگان تنظیم شده اند. ما برای شبیه سازی این الگوریتم از کدی استفاده کردیم که توسط نویسندگان در پایگاه

<http://www.mathworks.com/matlabcentral/fileexchange/37161-spatially-adaptive-irn-algorithm> تامین شده بود.

تصاویری که مورد تست و آزمایش قرار گرفته اند عبارتند از:

Lena, Bridge, Peppers, Baboon, Barbara, Boat, Fingerprint, Flintstones, House

اندازه گیری عملکرد به صورت نسبت سیگنال پیک به نویز (PSNR)، برای یک تصویر ترمیم شده (Ir) نسبت به تصویر اصلی ( $I_o$ ) خواهد بود.

$$PSNR(I_o, I_r) = 10 \log_{10} \frac{255^2}{\frac{1}{3MN} \sum_{i,j} (I_r(i,j) - I_o(i,j))^2} \quad \text{رابطه (12)}$$

که در آن 255 حداکثر شدت پیکسل برای تصاویر 8 بیتی است.

آزمون ها بر روی یک سیستم با پردازنده اینتل corei7 با فرکانس 3.2 GHz و 24GB of RAM و نرم افزار متلب 2012a انجام گرفته است.

### *E. تعداد تکرارها*

جدول VI تعداد تکرارهای لازم به ازای هر تصویر برای سطوح متغیر نویز، را نشان می دهد. به طور کلی، یک تصویر با نویز بیشتر به تکرار بیشتری نیاز دارد اما ویژگی های خاص و منحصر به فرد هر تصویر هم نقش مهمی ایفا می کند.

### *F. نویزگیری تصاویر رنگی*

الگوریتم ما برای استفاده در تصاویر رنگی نیز ممکن است قابل تعمیم باشد. این الگوریتم را به طور جداگانه در سه کانال رنگ - قرمز، سبز و آبی - اعمال کردیم و سپس نتایج را برای هر کدام از کانال ها برای رسیدن به تصویر بدون نویز ترکیب کردیم. در جدول V، عملکرد الگوریتم را بر اساس PSNR برای نویزگیری تصاویر رنگی تطبیق می دهیم. برای تصاویر رنگی سه کاناله، PSNR از رابطه زیر بدست می آید.

$$\text{رابطه (13)} \quad 10 \log_{10} \frac{255^2}{\frac{1}{3MN} \sum_{c \in \{R,G,B\}} \sum_{i,j} (I_r^c(i,j) - I_o^c(i,j))^2}$$

نتایج بصری در شکل 4 برای تصاویر Lena , Baboon نشان داده شده است.

TABLE V

COMPARISON OF THE MEAN RUN TIME OF OUR ALGORITHM WITH STATE-OF-THE-ART FILTERS IN SECONDS FOR THE SET OF IMAGES USED IN TABLE IV

	FM	CEF	PWS	WAV	AM-EPR	PB	BDND	CM	SA-TV	Our Algorithm
20%	2.31	10.34	18.52	30.96	1826.75	360.38	226.83	10.32	13.60	4.91
50%	2.31	18.59	26.48	56.37	3957.94	895.72	219.36	12.58	23.96	12.04
80%	2.31	37.09	34.63	72.51	6486.45	1804.51	220.47	17.02	28.78	26.32

TABLE VI  
NUMBER OF ITERATIONS REQUIRED FOR DIFFERENT IMAGES AT DIFFERENT NOISE LEVELS

	20%	50%	80%		20%	50%	80%
Lena	2	3	5	Boat	2	3	5
Bridge	8	9	10	Fingerprint	2	3	5
Peppers	2	3	5	Flintstones	2	5	9
Baboon	2	3	5	House	2	3	4
Barbara	2	3	5				

TABLE VII  
PSNR (IN DB) FOR DENOISING COLOR IMAGES CORRUPTED BY IMPULSE NOISE OF LEVELS 20%, 50%, AND 80%

	20%	50%	80%
Lena	35.33	29.58	23.27
Peppers	30.36	26.89	21.81
Baboon	26.13	21.34	17.69

Settings for  $K_1$  and  $K_2$  were kept at  $K_1 = K_2 = 3$ .



Fig. 4. Results of applying our algorithm to the separate color channels (RGB) of noisy color images and recombining them. (a) Lena. (b) 50% noise. (c) Result. (d) Baboon. (e) 80% noise. (f) Result.

#### IV. نتیجه گیری

در این مقاله، ما یک فیلتر دومرحله ای جدید برای نویزگیری تصاویری که در اثر نویز نمک-فلفل دچار خرابی شده اند، ارائه دادیم. در مرحله اول یک فیلتر فازی وفقی (انطباقی) برای شناسایی پیکسل های نویزدار استفاده می شود. در مرحله دوم، عمل نویزگیری با اعمال یک فیلترینگ میانگین وزن دار بر روی پیکسل های سالم مجاور انجام می پذیرد. نتایج تجربی حاکی از آن است که فیلتر پیشنهادی ما، از فیلترهای جدید و پیشرفته بهتر بوده و علاوه بر این قادر است جزئیات معنی دار تصویر را در سطوح خرابی به بزرگی 97%، ترمیم و بازیابی کند.