

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده مهندسی برق و رباتیک

رشته‌ی مهندسی برق گرایش مخابرات سیستم

پایان نامه کارشناسی ارشد

حذف نویز تصویر با استفاده از شبکه عصبی همگشتی عمیق

نگارنده: دانیال لطفی

استاد راهنما:

دکتر حسین خسروی

آذرماه ۱۳۹۹

تقدیم به

مادر م

که کنارم نیست اما یادش بند بند وجودم را میسازد

تشکر می‌کنم از استاد ارجمندم، جناب دکتر خسروی

که گام‌های سست مرا استوار کردند

تعهدنامه

اینجانب **دانیال لطفی** دانشجوی دوره کارشناسی ارشد رشته برق مخابرات سیستم، دانشکده برق دانشگاه صنعتی شاهرود نویسنده پایان نامه با موضوع بازسازی تصویر با استفاده از شبکه عصبی کانولوشن عمیق تحت راهنمایی دکتر حسین خسروی متعهد می شوم:

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش های محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود است و مقالات مستخرج با نام **دانشگاه صنعتی شاهرود** و یا **Shahrood University of Technology** به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه، در مواردی که از موجود زنده (یا بافت های آن ها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ :

امضاء دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات، مستخرج، کتاب، برنامه های رایانه ای، نرم افزارها و تجهیزات ساخته شده) متعلق به دانشگاه صنعتی شاهرود است. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نیست.

چکیده

تصاویر نقش مهمی در زندگی روزمره ما دارند و از آن‌ها برای کاربردهای مختلفی مثل تشخیص اشیاء، احراز هویت بیومتریک و نظارت تصویری استفاده می‌شود. نویز یک سیگنال نامطلوب است که باعث ایجاد یک تغییر تصادفی از مقادیر رنگ یا روشنایی در یک تصویر می‌شود. گاهی اوقات، به دلایلی مانند نقص در حس‌گرهای دوربین و انتقال در یک کانال نویزی، تصاویر به نویز آغشته می‌شوند. حذف نویز یک نیاز اساسی برای تقویت و بازیابی جزئیات پنهان و بارزش در یک تصویر است؛ بنابراین به یک روش کارآمد برای حذف نویز تصویر نیاز است.

در سال‌های اخیر، شبکه‌های عصبی همگشتی عمیق، توجه زیادی را در زمینه‌ی حذف نویز تصویر به خود جلب کرده‌اند. با این حال دو مشکل وجود دارد: اول این که آموزش یک شبکه همگشتی برای حذف نویز بسیار مشکل است و دوم آن که بیشتر قسمت‌های عمیق شبکه به حالت اشباع می‌روند و محوشدگی گرادیان رخ می‌دهد. در این پایان‌نامه، یک شبکه جدید از نوع خودرمزنگار (Autoencoder) طراحی شده است. در این شبکه به منظور استخراج ویژگی، به جای افزایش عمق، از افزایش عرض استفاده می‌کنیم. در این روش دو شبکه را به صورت شبکه چپ و راست باهم ترکیب می‌کنیم که باعث افزایش عرض به جای افزایش عمق می‌شود. در شبکه سمت راستی از خودرمزنگار برای بازیابی تصویر ورودی در خروجی و در شبکه سمت چپی از کانولوشن گسترده برای استخراج بیشتر ویژگی برای حذف نویز، استفاده می‌کنیم. همچنین برای جلوگیری از تغییر توزیع داده‌ها و عدم وابستگی به مقدار اولیه، از نرمال‌سازی دسته‌ای و برای تسهیل آموزش و جلوگیری از محوشدگی گرادیان، از یادگیری باقیمانده در طراحی شبکه خودرمزنگار استفاده شده است. نتایج تجربی نشان می‌دهد مقدار PSNR، برای تصاویر خاکستری نویزی به میزان ۱,۲۷ دسی‌بل و برای تصاویر رنگی، ۱,۲۲ دسی‌بل نسبت به روش‌های پیشین بهبود یافته است.

کلیدواژه:

شبکه همگشتی عمیق، نویز، خودرمزنگار، یادگیری باقیمانده، نرمال سازی دسته ای، کانولوشن

گسترده

لیست مقالات مستخرج از پایان نامه:

...

فهرست

۱- فصل اول - مقدمه	۱
۱-۱ مقدمه	۲
روش تحقیق	۳
ساختار پایاننامه	۳
۲- فصل دوم- مروری بر تحقیقات پیشین	۵
مقدمه	۶
استفاده از نمایش تُنک برای حذف نویز	۶
بازیابی تصویر با استفاده از شبکه پرسپترون	۸
بازیابی تصویر با استفاده از خودرمزنگارها	۹
حذف نویز با استفاده از FFDNet	۱۱
حذف نویز با استفاده از شبکه عمیق مبتنی بر Batch-renormalization	۱۲
۳- فصل سوم - مبانی نظری	۱۵
مقدمه	۱۶
۳-۱ معرفی انواع نویز	۱۶
۳-۱-۱ نویز گاوسی	۱۶
۳-۱-۲ نویز گاما (ارلانگ)	۱۷
۳-۱-۳ نویز یکنواخت	۱۸

۱۸	۳-۱-۴- نويز ریلی
۱۹	۳-۱-۵- نويز فلفل نمکی
۱۹	۳-۱-۶- نويز متناوب
۲۰	۳-۲- شبکه عصبی مصنوعی
۲۳	۳-۱-۷- لایه کانولوشن
۲۵	۳-۱-۸- لایه غیرخطی
۲۵	۳-۱-۹- لایه کاهنده
۲۶	۳-۱-۱۰- لایه کاملاً متصل
۲۶	۳-۱-۱۱- الگوریتم نرمال سازی دسته‌ای
۲۷	کانولوشن گسترده
۲۸	یادگیری باقیمانده
۲۹	خودرمزنگار
۳۱	۴- فصل چهارم - روش پیشنهادی
۳۲	مقدمه
۳۲	معماری شبکه طراحی شده
۳۴	۴-۱-۱- شبکه سمت راست
۳۵	۴-۱-۲- شبکه سمت چپ:
۳۶	تابع هزینه
۳۹	۵- فصل پنجم - نتایج:

مقدمه	۴۰
مجموعه داده‌ها	۴۱
۱-۱-۵- مجموعه داده‌های آموزش	۴۱
۱-۲-۵- مجموعه داده‌های تست	۴۱
تنظیمات اولیه	۴۲
نتایج	۴۲
۱-۳-۵- بررسی تکنیک‌های استفاده‌شده در شبکه AutoencoderNet	۴۲
۱-۴-۵- حذف نویز تصاویر رنگی و خاکستری با استفاده از شبکه AutoencoderNet	۴۴
۱-۵-۵- عملکرد AutoencoderNet بر روی تصاویر با نویز واقعی	۴۷
۱-۶-۵- زمان اجرا	۴۸
۶- فصل ششم:- نتیجه‌گیری و پیشنهادها	۴۹
نتیجه‌گیری	۵۰
پیشنهادها	۵۰
۷- مراجع	۵۲

فهرست شکل‌ها

- شکل (۱-۲) الگوریتم حذف نویز تصویر با استفاده از روش BM3D [10] ۷
- شکل (۲-۲) ساختار شبکه عمیق مبتنی بر خودرمزنگار [12] ۱۰
- شکل (۳-۲) معماری شبکه عمیق مبتنی بر خودرمزنگار برای حذف نویز تصویر [12] ۱۱
- شکل (۴-۲) معماری شبکه عمیق مبتنی بر نرمال‌سازی دسته‌های [4] ۱۳
- شکل (۱-۳) تابع چگالی نویز گاوسی [1] ۱۷
- شکل (۲-۳) تابع چگالی نویز گاما [1] ۱۷
- شکل (۴-۳) تابع چگالی نویز یکنواخت ۱۸
- شکل (۵-۳) تابع چگالی نویز ریلی [1] ۱۹
- شکل (۶-۳) تابع چگالی نویز فلفل نمکی [1] ۱۹
- شکل (۷-۳) ساختار شبکه عصبی ۲۱
- شکل (۸-۳) ساختار نوعی یک شبکه کانولوشن عمیق برای طبقه‌بندی ۲۲
- شکل (۹-۳) نحوه حرکت دادن فیلتر بر روی ورودی [15] ۲۴
- شکل (۱۰-۳) نحوه ضرب نقطه‌های ورودی با فیلتر (عملیات کانولوشن) [15] ۲۴
- شکل (۱۱-۳) ادغام نقشه ویژگی استخراج‌شده در هر فیلتر [15] ۲۴
- شکل (۱۲-۳) ایجاد نقشه ویژگی با گام‌های ۱ و ۲ [15] ۲۴
- شکل (۱۳-۳) تأثیر استفاده از پدینگ در عملیات کانولوشن و ابعاد نقشه ویژگی [15] ۲۵
- شکل (۱۴-۳) نحوه عملکرد Max pooling [15] ۲۶
- شکل (۱۵-۳) کانولوشن گسترده با نرخ‌های ۱، ۲ و ۳ [18] ۲۸
- شکل (۱۶-۳) تأثیر افزایش عمق بر عملکرد شبکه [19] ۲۸
- شکل (۱۷-۳) نمای کلی از بلوک یادگیری باقیمانده [19] ۲۸
- شکل (۱۸-۳) نمای کلی از یک خودرمزنگار [20] ۲۹

- شکل (۱۹-۳) نحوه اتصال و چینش لایه‌های خودرمزنگار [20] ۳۰
- شکل (۱-۴) ساختار کلی شبکه AutoencoderNet ۳۳
- شکل (۲-۴) ساختار یک بلوک حذف نویز مبتنی بر خودرمزنگار ۳۴
- شکل (۳-۴) ساختار کلی خودرمزنگار استفاده‌شده در شبکه سمت راستی ۳۵
- شکل (۱-۵) بررسی عملکرد اثر دیداری بر روی تصویر Barbara با سطح نویز ۵۰ ۴۷
- شکل (۲-۵) بررسی عملکرد اثر دیداری بر روی تصویر Kodim با سطح نویز ۵۰ ۴۷

فهرست جداول

- جدول (۱-۲) نتایج به دست آمده در شبکه عمیق مبتنی بر نمایش تنک [10] ۸
- جدول (۲-۲) نتایج به دست آمده در شبکه عمیق مبتنی بر نرمال سازی دسته های [4] ۱۴
- جدول (۱-۵) نتایج حذف نویز گاوسی در دو حالت استفاده از یادگیری باقیمانده و عدم استفاده آن در تصاویر مجموعه Set12 با $\sigma=25$ ۴۳
- جدول (۲-۵) نتایج حذف نویز گاوسی در دو حالت استفاده از Autoencoder و عدم استفاده آن در تصاویر مجموعه Set12 با $\sigma=25$ ۴۳
- جدول (۳-۵) نتایج حذف نویز گاوسی در دو حالت استفاده از کانولوشن توسعه دهنده و عدم استفاده آن در تصاویر مجموعه Set12 با $\sigma=25$ ۴۳
- جدول (۴-۵) نتایج حذف نویز گاوسی در دو حالت استفاده از دو شبکه فرعی و حالت یک شبکه در تصاویر مجموعه Set12 با $\sigma=25$ ۴۴
- جدول (۵-۵) میانگین مقادیر PSNR روش های مختلف بر روی مجموعه تصاویر BSD68 با سطوح نویز ۱۵، ۲۵ و ۵۰ ۴۴
- جدول (۶-۵) مقدار PSNR (dB) برای روش های مختلف بر روی ۱۲ تصویر مجموعه تصاویر Set12 با سطح نویز ۱۵ ۴۵
- جدول (۷-۵) مقدار PSNR (dB) برای روش های مختلف بر روی ۱۲ تصویر مجموعه تصاویر Set12 با سطح نویز ۲۵ ۴۵
- جدول (۸-۵) مقدار PSNR (dB) برای روش های مختلف بر روی ۱۲ تصویر مجموعه تصاویر Set12 با سطح نویز ۵۰ ۴۵
- جدول (۹-۵) مقدار PSNR (dB) برای روش های مختلف بر روی مجموعه تصاویر CBSD68، Kodak24 و McMaster با سطح نویز مختلف ۴۶
- جدول (۱۰-۵) مقدار PSNR (dB) برای روش های مختلف بر روی مجموعه تصاویر گرفته شده توسط

دوربینهای مختلف ۴۸

جدول (۱۱-۵) مقایسه تعداد پارامترهای شبکه BRDNET و AutoencoderNet ۴۸

جدول (۱۲-۵) زمان اجرا برای روشهای مختلف برای حذف نویز تصاویر در اندازه های 256×256

، 512×512 و 1024×1024 ۴۸

PSNR	Peak Signal to Noise Ratio
BM3D	block-matching and 3-D filtering
BR	BatchNormalization
FFDNet	Fast and flexible denoising convolutional neural network
BRDNet	Batch-renormalization denoising network
RL	Residual learning
CNN	Convolutional neural networks
RBMS	Restricted Boltzmann Machines
ResNet	Residual neural network
MSE	Mean Squared Error

۱- فصل اول - مقدمه

مقدمه

۱-۱ مقدمه

تصاویر دیجیتال نقش مهمی در زندگی روزمره ما دارند و از آن‌ها برای کاربردهای مختلفی مثل تشخیص اشیاء، احراز هویت بیومتریک و نظارت تصویری استفاده می‌شود. تصاویر بهتر می‌توانند از اطلاعات دیگر به انسان در درک حوادث و اتفاقات مختلف کمک کنند و درصد بیشتر اطلاعات کسب‌شده توسط انسان از طریق حس بینایی کسب می‌شود. تصاویر تحت شرایط مختلف تحت تأثیر نویز قرار می‌گیرد.

نویز یک سیگنال نامطلوب است که باعث ایجاد یک تغییر تصادفی از مقادیر رنگ یا روشنایی در یک تصویر می‌شود [1]. نویز می‌تواند در طی فرآیند انتقال تصویر یا نقص در حس‌گرهای دوربین ایجادشده و تصویر مدنظر را تحت تأثیر خود قرار دهد، بنابراین حذف آن یک نیاز اساسی برای تقویت و بازیابی جزئیات پنهان و باارزش در یک تصویر است.

هوش مصنوعی^۱ شاخه‌ای از علم رایانه است که تلاش دارد، هوش انسان را شبیه‌سازی کند و هدف اصلی آن تولید ماشین‌هایی است که توانایی انجام وظایفی که نیازمند هوش انسان است را داشته باشد. اهمیت هوش مصنوعی را می‌توان از کاربرد آن در زمینه‌های مختلف تکنولوژی از جمله، در حوزه سلامت، کسب‌وکار، آموزش، اقتصاد، تولید، امنیت و تفسیر داده‌ها دانست. سیستم‌های هوش مصنوعی به کمک یادگیری ماشین^۲ و یادگیری عمیق^۳ [2] می‌توانند یاد بگیرند و آموزش ببینند. یادگیری ماشین که زیرمجموعه‌ای از هوش مصنوعی است، با مشاهده‌ی داده‌ها صورت می‌گیرد و پس از استخراج ویژگی، تصمیم‌گیری کرده و مسئله را حل می‌کند. هدف یادگیری ماشین، یادگیری خودکار بدون دخالت و کمک انسان و صرفاً بر اساس مشاهدات و داده‌هاست. یادگیری عمیق نیز زیرمجموعه هوش مصنوعی و یادگیری ماشین است که الگوی این یادگیری برگرفته از مغز انسان است که در سطح پیچیده‌ای عمل یادگیری را انجام می‌دهد. در فصل ۳ جزئیات بیشتری از یادگیری عمیق ارائه

^۱ Artificial intelligence

^۲ Deep learning

^۳ Machine Learning

خواهد شد.

حذف نویز با روش‌های مختلفی صورت می‌گیرد از جمله روش‌های کلاسیک مثل اعمال فیلترهای میانگین و میانه یا فیلترهای افقی که این روش‌ها دقت پایینی دارند. همچنین از شبکه‌های عصبی برای حذف نویز استفاده می‌شود که این روش‌ها نسبت به روش‌های کلاسیک بهتر عمل می‌کنند؛ اما مشکل این روش‌ها این است که باید نوعی استخراج ویژگی دستی انجام داد. برای رفع این مشکل ما در این پایان‌نامه از شبکه‌های رمزگذار عمیق استفاده می‌کنیم که نیاز به استخراج ویژگی نداشته و دقت بالاتری نسبت به دو روش قبل دارند.

روش تحقیق

در این پایان‌نامه قصد داریم یک ساختار شبکه عمیق را برای حذف نویز تصویر طراحی کنیم. برای آموزش شبکه از پایگاه داده کاوش واترلو استفاده می‌کنیم که شامل تصاویر رنگی و خاکستری است. برای تصاویر تست از مجموعه داده‌های اکتشافی برکلی (تصاویر خاکستری) و 24.CBSD^{۶۸} و Kodak و McMaster (تصاویر رنگی) و ۱۰۰ تصویر از مجموعه تصاویر آقای ژانگ [3] که تصاویر واقعی گرفته‌شده با دوربین‌های مختلف است، استفاده می‌کنیم. برای ارزیابی شبکه از معیار نسبت سیگنال به نویز و ارزیابی دیداری^۱ استفاده می‌شود. مقادیر به‌دست‌آمده با مقادیر مقاله مرجع [4] سنجیده می‌شود.

ساختار پایان‌نامه

تا اینجا به معرفی مختصر از مباحث مورد استفاده در این پایان‌نامه پرداختیم، در ادامه و در فصل دوم پیشینه تحقیقاتی در حوزه یادگیری عمیق و روش‌های حذف نویز را بررسی خواهیم کرد و با چالش‌ها، کاربردها و زمینه‌های مختلف تحقیقاتی آن‌ها آشنا می‌شویم. در فصل سوم برخی مباحث

^۱ Visual effect

نظری مورد استفاده در این پایان نامه بیان خواهد شد. در فصل چهارم رویکرد پیشنهادی پایان نامه و نحوه ترکیب الگوریتم های استفاده شده را مطرح می کنیم. در فصل پنجم نتایج معماری طراحی شده را با استفاده از کتابخانه های متن باز در پردازش تصویر و کتابخانه کراس برای یادگیری عمیق به زبان پایتون مورد بررسی قرار داده و در فصل ششم هم نتیجه گیری ارائه خواهد شد.

۲- فصل دوم- مروری بر تحقیقات پیشین

مروری بر تحقیقات پیشین

مقدمه

در بازیابی تصاویر، هدف اصلی بهبود یک تصویر با دانستن علت افت کیفیت است بنابراین روش‌های بازیابی همگی بر پایه مدل‌سازی افت تصویر و اعمال فرآیند معکوس جهت رسیدن به تصویر اصلی است. برای بازیابی و حذف نویز روش‌های متعددی مطرح شده است که می‌توان آن‌ها را به سه روش کلی خلاصه کرد.

۱. بازیابی تصویر در حوزه مکان یا فرکانس و استفاده از فیلترهای مختلف و مناسب با خرابی

تصویر [5][6][7]

۲. بازیابی تصویر با استفاده از شبکه عصبی [8]

۳. بازیابی با استفاده از شبکه‌های عمیق [9]

ما در این فصل روش‌های استفاده شده برای حذف و بازیابی تصویر را بررسی می‌کنیم.

استفاده از نمایش تُنک^۱ برای حذف نویز

کوستادین دابوف و همکارانش [10] یک استراتژی جدید برای حذف نویز تصاویر، مبتنی بر نمایش تنک پیشنهاد دادند. در این استراتژی افزایش پراکندگی با گروه‌بندی قطعات تصویر دوبعدی مشابه، به آرایه‌های سه‌بعدی صورت می‌گیرد و از فیلترینگ مشترک برای توسعه آرایه‌های سه‌بعدی استفاده شده است.

نحوه گروه‌بندی در الگوریتم پیشنهادی آقای دابوف به صورت انطباق بلوکی صورت می‌گیرد و تصویر نویزی ورودی را با استفاده از استخراج بلوک‌های مرجع، برای هر بلوک پردازش می‌شود، روند کلی این الگوریتم به صورت زیر است:

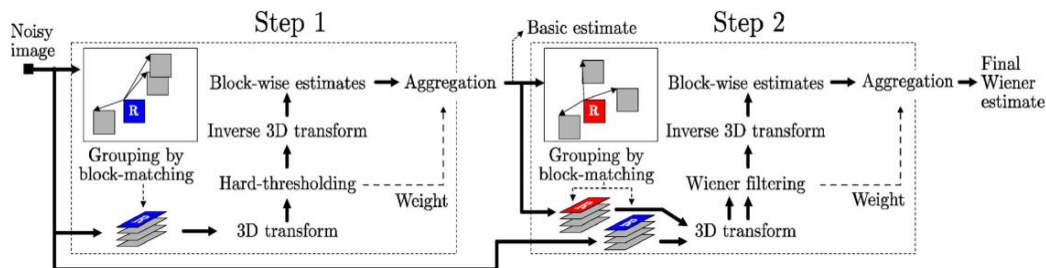
۱- بلوک‌های مشابه با بلوک‌های مرجع را پیدا کرده و باهم متصل می‌شوند که تبدیل سه‌بعدی یا همان گروه را تشکیل می‌دهد.

^۱ Sparse representation

۲- فیلترینگ مشارکتی^۱ را انجام داده و تخمینی که از همه بلوک‌های گروه‌بندی شده، به دست آمده را به محل اصلی خودشان برگردانده می‌شوند.

روش کلی در شکل (۱-۲) برای ایجاد یک الگوریتم پیاده‌سازی شده است و شامل دو مرحله است

که در زیر به طور مفصل توضیح داده شده است.



شکل (۱-۲) الگوریتم حذف نویز تصویر با استفاده از روش BM3D^۲[10]

۱. تخمین اولیه^۳

a. تخمین بلوکی^۴: برای هر تصویر نویزی باید مراحل زیر انجام شود.

i. گروه‌بندی: بلوک‌هایی که مشابه بلوک در حال پردازش است را پیدا کرده و یک تبدیل سه‌بعدی یا همان گروه، تشکیل می‌دهیم.

ii. آستانه گذاری سخت مشارکتی^۵: یک تبدیل سه‌بعدی به گروه اضافه می‌کنیم و با یک آستانه گذاری نویز را کاهش می‌دهیم، تبدیل سه‌بعدی را برای تخمین بلوک‌های گروه‌بندی شده معکوس می‌کنیم.

b. جمعیت^۶: تخمین اصلی تصویر را با میانگین وزنی تمام تخمین‌های بلوکی که با هم تداخل دارند، محاسبه می‌کنیم.

۲. تخمین نهایی: با استفاده از تخمین اولیه، گروه‌بندی بهبود یافته و فیلترینگ وینر را انجام می‌دهیم.

a. تخمین بلوکی: برای هر بلوک موارد زیر را انجام می‌دهیم

i. گروه‌بندی: با استفاده از تطبیق بلوکی در تخمین اولیه، مکان بلوک‌های مشابه با بلوک در حال پردازش را مشخص می‌کنیم و با استفاده از این

^۱ Collaborative Filtering

^۲ block-matching and 3-D filtering

^۳ Basic estimate

^۴ Block-wise estimates

^۵ Collaborative hard-thresholding

^۶ Aggregation

مکان‌ها یک گروه تصویر نویزی و یک گروه از تخمین اولیه‌مان تشکیل می‌دهیم.

ii. فیلتر مشترک وینر^۱: تبدیل سه‌بعدی را به هر دو گروه اضافه کرده و فیلتر وینر را اعمال می‌کنیم و با استفاده از تبدیل معکوس سه‌بعدی، تخمین همه بلوک‌های گروه‌بندی‌شده را تولید کرده و در موقعیت اصلی خودشان برمی‌گردانیم.

b. تجميع: با به دست آوردن تمامی تخمین‌های نهایی و با استفاده از میانگین وزنی تصویر نهایی را به دست می‌آوریم.

روش آقای دابوف با دو نوع تصویر، یکی دارای نویز واقعی که با دوربین‌های مختلف گرفته‌شده و دیگری دارای نویز مصنوعی گاوسی، سنجیده شده است. معیار سنجش عملکرد در این الگوریتم، نسبت سیگنال به نویز^۲ و میزان تشابه شهودی^۳ استفاده‌شده است که با توجه به جدول (۱-۲) به نتایج مطلوبی رسیده‌اند.

σ / PSNR	<i>C.man</i> 256 ²	<i>House</i> 256 ²	<i>Peppers</i> 256 ²	<i>Montage</i> 256 ²	<i>Lena</i> 512 ²	<i>Barbara</i> 512 ²	<i>Boats</i> 512 ²	<i>Fprint</i> 512 ²	<i>Man</i> 512 ²	<i>Couple</i> 512 ²	<i>Hill</i> 512 ²	<i>Lake</i> 512 ²
2 / 42.11	43.96	44.63	43.48	46.47	43.59	43.66	43.18	42.90	43.61	43.17	43.04	43.02
5 / 34.16	38.29	39.83	38.12	41.14	38.72	38.31	37.28	36.51	37.82	37.52	37.14	36.58
10 / 28.14	34.18	36.71	34.68	37.35	35.93	34.98	33.92	32.46	33.98	34.04	33.62	32.85
15 / 24.61	31.91	34.94	32.70	35.15	34.27	33.11	32.14	30.28	31.93	32.11	31.86	31.08
20 / 22.11	30.48	33.77	31.29	33.61	33.05	31.78	30.88	28.81	30.59	30.76	30.72	29.87
25 / 20.18	29.45	32.86	30.16	32.37	32.08	30.72	29.91	27.70	29.62	29.72	29.85	28.94
30 / 18.59	28.64	32.09	29.28	31.37	31.26	29.81	29.12	26.83	28.86	28.87	29.16	28.18
35 / 17.25	27.93	31.38	28.52	30.46	30.56	28.98	28.43	26.09	28.22	28.15	28.56	27.50
50 / 14.16	25.84	29.37	26.41	27.35	28.86	27.17	26.64	24.36	26.59	26.38	27.08	25.78
75 / 10.63	24.05	27.20	24.48	25.04	27.02	25.10	24.96	22.68	25.10	24.63	25.58	24.11
100 / 8.14	22.81	25.50	22.91	23.38	25.57	23.49	23.74	21.33	23.97	23.37	24.45	22.91

جدول (۱-۲) نتایج به‌دست‌آمده در شبکه عمیق مبتنی بر نمایش تنک [10]

بازیابی تصویر با استفاده از شبکه پرسپترون

هارولد سی برگر و همکارانش [11] توانستند با آموزش روی پایگاه داده بزرگ و با استفاده از

شبکه عصبی پرسپترون ساختاری تهیه کنند که منجر به حذف نویز در تصویر بشود.

ساختار پیشنهادی مبتنی بر پنجره^۴ است و تعداد لایه‌های مخفی و تعداد نرون‌های هر لایه

^۱ Collaborative Wiener filtering

^۲ Peak Signal to Noise Ratio (PSNR)

^۳ Structural similarity Index

^۴ Patch

به اندازه‌ای گرفته می‌شود که بهترین خروجی حاصل شود. ایده این ساختار به این صورت است که شبکه پرسپترون طراحی شده با استفاده از نگاشت پنجره‌های تصویر نویزی و پنجره‌های تصویر تمیز، منجر به کاهش و یا حذف نویز می‌شود و پارامترهای شبکه را با آموزش پنجره‌های تصویر نویزی و تمیز و با استفاده از روش گرادیان نزولی به دست می‌آید.

به‌طور دقیق‌تر می‌توان گفت که ابتدا تصویر را به پنجره‌های دارای همپوشانی تقسیم کرده و گام حرکت انتخاب هر پنجره ۱ یا ۳ در نظر گرفته می‌شود، یک پنجره از تصویر تمیز به‌طور مثال y ، انتخاب کرده و به آن یک نویز X ، اضافه می‌کنیم و در رابطه ۱-۲) جایگذاری می‌کنیم و سپس با استفاده از الگوریتم پس انتشار به‌روزرسانی می‌شود تا خطای درجه دوم بین پنجره نویزی شده $(f(x))$ و پنجره تصویر تمیز به کمترین میزان برسد و برای پنجره‌های که همپوشانی دارند بر روی آن‌ها متوسط‌گیری می‌شود که می‌توان برای بهبود عملکرد الگوریتم برای هر یک از پنجره‌ها وزنی تعیین کرد.

$$m = (f(x) - y)^2 \quad (1-2)$$

برای بهبود الگوریتم پس انتشار می‌توان ترفندهای زیر را انجام داد

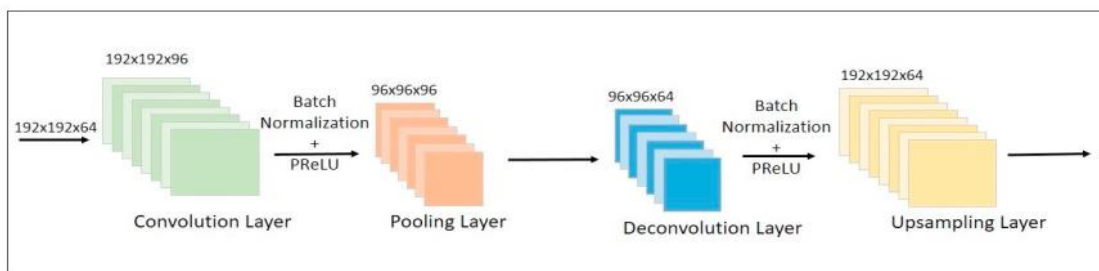
۱. نرمال‌سازی داده‌ها: تمامی پیکسل‌های تصویر را به میانگین صفر و واریانس یک، تبدیل می‌شود
 ۲. وزن دهی اولیه: وزن‌ها از یک توزیع نرمال با میانگین صفر و انحراف استاندارد σ $\sqrt{N}$ نمونه‌برداری می‌شوند، در آن N تعداد نرون‌های ورودی لایه مربوطه است.
 ۳. تعیین نرخ یادگیری برای هر لایه: در هر لایه، نرخ یادگیری را بر N تقسیم می‌کنیم و برای نرخ یادگیری اولیه ۰,۱ در نظر گرفته می‌شود.
- الگوریتم پیشنهادی، نسبت به الگوریتم مبتنی بر نمایش تنک سرعت پردازش و دقت بیشتری دارد و همچنین از این الگوریتم می‌توان برای انواع نویزها استفاده کرد.

بازیابی تصویر با استفاده از خودرمنگاراها

کمال باجاج و همکارانش [12] مدل یادگیری عمیقی که مبتنی بر خودرمنگاراها است را برای

حذف نویز تصویر ارائه کردند. خودرمزنگارها انواعی از شبکه عصبی هستند که هدف آن‌ها یادگیری بازنمایی (رمزگذاری) برای مجموعه‌ای از داده‌ها است و نویز را از تصاویر آموزشی می‌آموزد و سعی می‌کند که تصویر تمیز و نزدیک به ورودی، تولید کند.

همان‌طور که در شکل (۲-۲) می‌بینید، ساختار مطرح‌شده توسط باجاج از چهار لایه همگشتی^۱، کاهنده^۲ و ضدهمگشتی^۳ و افزاینده^۴ تشکیل شده است و از تابع فعال‌ساز^۵ و نرمال‌ساز دسته‌ای^۶ استفاده شده است. این یک بلوک خودرمزنگار است.



شکل (۲-۲) ساختار شبکه عمیق مبتنی بر خودرمزنگار [12]

سپس یک معماری پیشنهاد دادند که در شکل (۳-۲) آمده است، این معماری شامل پنج بخش و

به شرح زیر است

- لایه ورودی
- یک لایه همگشتی معمولی
- ۱۰ بلوک طراحی شده خودرمزنگار
- دولایه ضدهمگشتی، این دولایه تصویر نویز گیری شده را برمی گردانند.
- لایه خروجی

^۱ Convolution

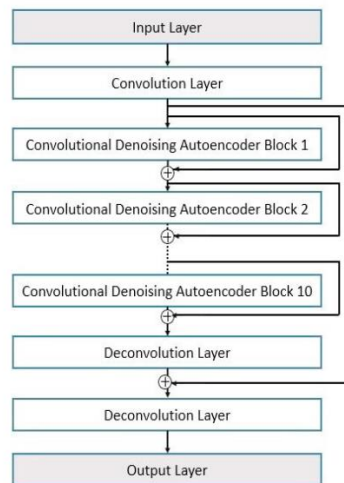
^۲ Pooling

^۳ Deconvolution

^۴ Upsampling

^۵ Activator Function

^۶ BatchNormalization



شکل (۳-۲) معماری شبکه عمیق مبتنی بر خودرمنزنگار برای حذف نویز تصویر [12]

درمجموع از ۱۵ لایه برای طراحی معماری استفاده شده است و ورودی هر لایه با خروجی همان لایه جمع شده است و ورودی لایه بعدی را تشکیل می‌دهد و در آخر نیز خروجی لایه اول را به ورودی لایه چهاردهم وصل می‌شود که به این نوع اتصالات، اتصالات پرش^۱ گفته می‌شود. دو هدف در برقراری این نوع اتصالات دنبال می‌شود

- افزایش عمق: این کار منجر به استخراج بیشتر ویژگی و جزئیات تصویر می‌شود
 - بهبود در آموزش شبکه: در فرایند پس انتشار از ناپدید شدن گرادیان جلوگیری کرده و نیز جزئیات تصویر را برای بازسازی بهتر تصویر تمیز بیان می‌کند.
- معیارهای عملکرد سیستم پیشنهادی نسبت سیگنال به نویز و شاخص تشابه ساختاری در نظر گرفته شده است.

حذف نویز با استفاده از FFDNet

آقا کای ژانگ باهمکاران [3] یک شبکه به نام FFDNet^۲ معرفی کردند که یک شبکه همگشتی با سرعت و انعطاف‌پذیری بالا است و دارای سه مزیت نسبت به سایر شبکه‌ها است.

۱- توانایی کنترل طیف گسترده‌ای از سطح نویز

^۱ Skip

neural network

^۲ Fast and flexible denoising convolutional

۲- توانایی حذف نویز مختلف فضایی^۱ با مشخص کردن یک نقشه سطح نویز غیریکنواخت

۳- سرعت بالاتر نسبت به نتایج بدست آمده در مقاله BM3D[10]، حتی در پردازنده‌های ضعیف‌تر، بدون اختلال در عمل نویز گیری تصاویر

شبکه مطرح شده به این صورت است که تصویر اولیه را با استفاده از نمونه برداری کاهشی^۲، به چهار تصویر تبدیل کرده و نیز یک نقشه سطح نویز که قابل تنظیم است، به آن اضافه کرده و به ورودی شبکه کانولوشن داده می‌شود، این شبکه برای تصاویر رنگی از ۱۲ لایه و برای تصاویر خاکستری از ۱۵ لایه استفاده می‌کند که هر لایه متشکل از سه بخش Conv + relu و Conv + BN + relu و Conv و از padding برای جلوگیری از تغییر ابعاد در شبکه استفاده شده است و بعد از تمام لایه‌ها عملیات نمونه برداری برای تشکیل تصویر بدون نویز انجام می‌شود. در این شبکه به جای اینکه نویز را به دست بیاورد، تصویر بدون نویز را تشکیل می‌دهد.

حذف نویز با استفاده از شبکه عمیق مبتنی بر Batch-renormalization

در سال‌های اخیر شبکه‌های همگشتی عمیق توجه زیادی را در زمینه‌ی حذف نویز تصویر به خود جلب کرده‌اند. با این حال، با دو مشکل روبه‌رو بودند:

۱- آموزش شبکه همگشتی عمیق برای حذف نویز، مشکل است

۲- در الگوریتم پس انتشار که برای بروز رسانی پارامترها استفاده می‌شود، احتمال رخ دادن پدیده محوشده‌گی گرادیان است (به حالت اشباع می‌رود و به‌روزرسانی را انجام نمی‌دهد).

یانگ و همکارانش [4]، یک شبکه جدید به نام شبکه حذف نویز مبتنی بر نرمال سازی دسته‌ای پیوسته^۳ طراحی کردند که دو مشکل مطرح شده در بالا را برطرف کردند.

یانگ و همکارانش برای پیاده سازی این معماری، یک الگوریتم جدیدی در نرمال سازی دسته‌ای مطرح کردند، این الگوریتم بعد از انتخاب دسته‌ها برای نرمال سازی، وزنی را برای هر یک از دسته‌ها

^۱ Spatially

^۲ Downsample

^۳ Batch-renormalization denoising network (BRDNet)

در نظر می‌گیرند. این کار منجر به این می‌شود که همبستگی قسمتی از تصویر انتخاب‌شده را با دسته بعدی خودش در نظر بگیرد و در نظر گرفتن این همبستگی می‌تواند در بازسازی تصویر کمک‌کننده باشد.

در ساختار معماری موارد زیر به کار برده شده است.

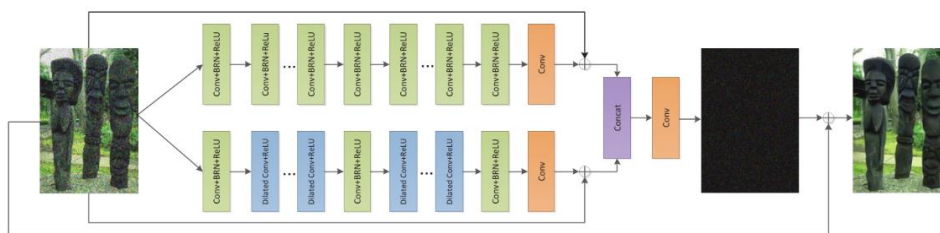
۱- افزایش عرض به جای افزایش عمق: آموزش شبکه راحت‌تر شده و ویژگی‌های بیشتری استخراج می‌شود.

۲- استفاده از نرمال‌ساز دسته‌ای پیوسته [13]: برای ایجاد همبستگی دو دسته و افزایش هم‌گرایی شبکه در حین آموزش

۳- استفاده از شبکه یادگیری باقیمانده^۱: برای جلوگیری از پدیده محو شدن گرادینان

۴- استفاده از کانولوشن گسترده^۲: استفاده از این نوع شبکه‌ها میدان پذیرا^۳ را افزایش داده و شبکه را قادر می‌سازد تا اطلاعات بافت بیشتری استخراج شده و هزینه محاسباتی کاهش یابد.

برای طراحی این معماری از الحاق دو شبکه عصبی همگشتی استفاده شده است که جزئیات معماری شبکه با توجه به شکل (۴-۲) به شرح زیر است.



شکل (۴-۲) معماری شبکه عمیق مبتنی بر نرمال‌سازی دسته‌ای [4]

- شبکه بالایی: شامل ۱۷ لایه است.

- کانولوشن گسترده و یادگیری باقی‌مانده

- شبکه پایینی: شامل ۱۷ لایه است

^۱ Residual learning

^۲ Receptive field

^۳ Dilated convolutions

○ کانولوشن گسترده و یادگیری باقی‌مانده: در لایه‌های ۲ تا ۸ و ۱۰ تا ۱۵ استفاده شده.

○ کانولوشن معمولی، نرمال ساز دسته‌ای و یادگیری باقی‌مانده: در لایه‌های ۱، ۹، ۱۶ استفاده شده است.

○ کانولوشن معمولی: در لایه ۱۷ استفاده شده است.

- تفاضل گیر: خروجی هر دو شبکه بالا و پایین نویز تصویر است که این مقدار را از تصویر اولیه کم می‌شود و تصویر تمیز به دست می‌آید.

- الحاق دو شبکه: با استفاده از Concat دو شبکه بالا و پایین را به هم متصل می‌کنیم

- کانولوشن معمولی

این معماری نویز را تشخیص می‌دهد که در انتها باید از تصویر ورودی کم شود تا به خروجی مطلوب برسد. معیار عملکرد آن نیز به دو شکل سنجیده می‌شود: نسبت سیگنال به نویز و میزان تشابه شهودی.

با توجه به جدول (۲-۲) [4] نتایج به دست آمده توسط این معماری نسبت به تمام معماری‌های که تا به حال مطرح شده بودند بهتر بوده و سرعت یادگیری شبکه هم سریع‌تر است.

Methods	BM3D	WNNM	EPLL	MLP	CSF	TNRD	DnCNN	IRCNN	FFDNet	BRDNet
$\sigma = 15$	31.07	31.37	31.21	-	31.24	31.42	31.72	31.63	31.62	31.79
$\sigma = 25$	28.57	28.83	28.68	28.96	28.74	28.92	29.23	29.15	29.19	29.29
$\sigma = 50$	25.62	25.87	25.67	26.03	-	25.97	26.23	26.19	26.30	26.36

جدول (۲-۲) نتایج به دست آمده در شبکه عمیق مبتنی بر نرمال‌سازی دسته‌ای [4]

در این فصل ما به بررسی پنج روش مختلف حذف نویز تصویر پرداختیم و در جدول (۲-۲) به مقایسه عملکرد PSNR این روش‌ها بر روی تصاویر BSD68 با سه سطح نویز مختلف پرداختیم.

در فصل ۳ به مباحث تئوری شامل معرفی انواع نویز و لایه‌های شبکه همگشتی و الگوریتم و شبکه‌های استفاده شده در روش پیشنهادی مثل شبکه‌های گسترده، یادگیری باقیمانده و شبکه‌های

خودرمنزنگار می‌پردازیم.

۳- فصل سوم – مبانی نظری

مبانی نظری

مقدمه

قبل از ورود به شرح معماری پیشنهادی نیاز است که با بعضی از اصطلاحات و الگوریتم‌های استفاده‌شده در روش پیشنهادی آشنا شویم. در ادامه به معرفی انواع نویز پرداخته و توضیح مختصری در مورد شبکه‌های کانولوشن ارائه می‌دهیم. سپس مفاهیم اساسی هر لایه و نحوه اتصال لایه‌ها را برای ایجاد یک معماری شبکه AutoencoderNet ارائه خواهیم داد.

۳-۱ معرفی انواع نویز

نویز در تصاویر دیجیتال، به هنگام تصویربرداری و یا انتقال پدید می‌آید. عملکرد حسگرهای تصویربرداری تحت‌الشعاع عوامل متفاوتی مثل شرایط جوی، شرایط نوری و کیفیت سنسورهای مورد استفاده است. به‌عنوان مثال هنگام عکس‌برداری با دوربین‌های CCD میزان نویز تصویر به سطح نور و دمای محیط بستگی دارد. همچنین تصاویر هنگام عبور از کانال می‌توانند دستخوش تغییر شوند. به‌عنوان مثال تصویری که از طریق یک کانال بیسیم عبور می‌کند حساس به شرایط محیطی مثل صاعقه است. با توجه به عواملی که گفته شد نویزهای متنوعی داریم. عمدتاً ماهیت نویزها جمع شونده است، به‌طوری‌که نویز با جمع جبری به سیگنال اضافه می‌شود، البته خود نویز ممکن است مقادیر مثبت یا منفی داشته باشد، ولی به‌صورت جمع جبری به سیگنال اضافه می‌شود.

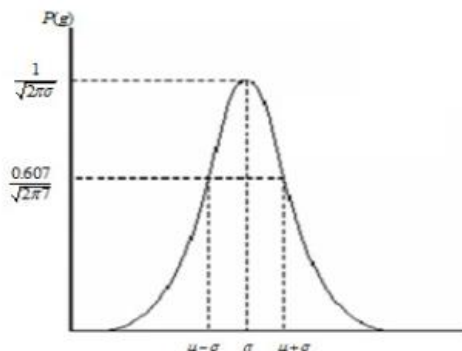
۳-۱-۱ نویز گاوسی^۱

این نوع نویز، بیشتر به خاطر حرکت تصادفی حامل‌های الکتریکی اتفاق می‌افتد. به دلیل تحلیل ریاضی آن در حوزه زمان و فرکانس در عمل نیز از این نوع استفاده می‌کنند. تابع چگالی احتمال نویز گاوسی به‌صورت معادله ۳-۱ است؛ که در آن z شدت روشنایی با میانگین m و واریانس σ نشان می‌دهد. نمودار آن در شکل (۳-۱) با میانگین و واریانس‌های مختلف رسم شده

^۱ Gaussian

است.

$$P(g) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(g - \mu)^2}{2\sigma^2}\right) \quad (1-3)$$



شکل (۱-۳) تابع چگالی نویز گاوسی [1]

۳-۱-۲- نویز گاما (ارلانگ) ^۱

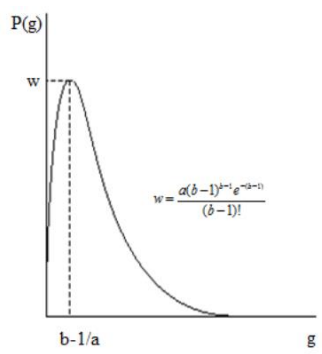
این نویز در عکس برداری لیزری اتفاق می افتد و تابع چگالی احتمال آن طبق معادله (۲-۳) محاسبه

می شود.

$$P(g) = \begin{cases} \frac{a^b g^{b-1}}{(b-1)!} e^{-ag} & g \geq a \\ 0 & g \leq a \end{cases} \quad (2-3)$$

مقدار میانگین نویز گاما، $(m = \frac{b}{a})$ و واریانس آن برابر $(\sigma^2 = \frac{b}{a^2})$ است. تابع آن در شکل

(۲-۳) رسم شده است.



شکل (۲-۳) تابع چگالی نویز گاما [1]

^۱ Gamma

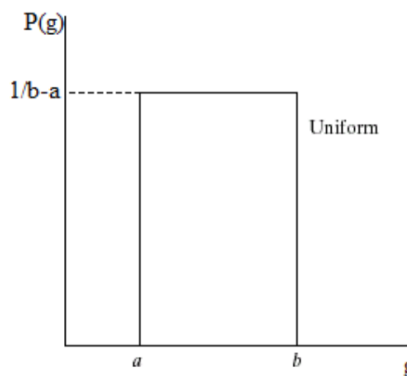
۳-۱-۳- نویز یکنواخت^۱

این نویز از کوانتیزه کردن^۲ ورودی به تعداد سطوح گسسته به وجود می‌آید. تابع چگالی آن در

معادله (۳-۳) نشان داده شده است، این تابع میانگینی برابر با $m = \frac{a+b}{2}$ و واریانس $\sigma^2 = \frac{(a+b)^2}{12}$

دارد. در شکل (۳-۳) تابع چگالی نویز یکنواخت نمایش داده شده است.

$$P(g) = \begin{cases} \frac{1}{b-a} & a \leq g \leq b \\ 0 & \text{سایر مقادیر} \end{cases} \quad (۳-۳)$$



شکل (۳-۳) تابع چگالی نویز یکنواخت

۳-۱-۴- نویز ریلی

تابع چگالی نویز رایلی به صورت معادله (۴-۳) است. این تابع برای تقریب زدن هیستوگرام‌های

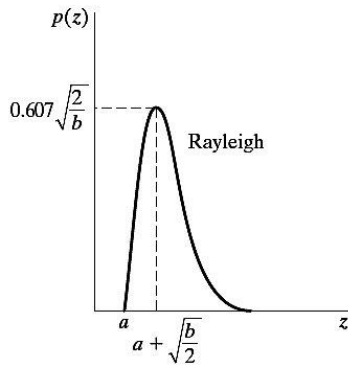
دارای برآمدگی بسیار مناسب است. میانگین آن برابر با $m = a + \sqrt{\frac{\pi b}{4}}$ و واریانس $\sigma^2 =$

است. در شکل (۴-۳) تابع چگالی نویز ریلی نمایش داده شده است.

$$P(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}}, & z \geq a \\ 0, & z \leq a \end{cases} \quad (۴-۳)$$

^۱ Uniform

^۲ quantizing

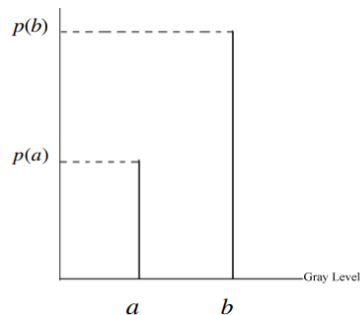


شکل ۴-۳ تابع چگالی نویز ریلی [1]

۵-۱-۳- نویز فلفل نمکی^۱

این نویز می‌تواند ناشی از حالت‌های گذرای سریع مثل کلید زنی ناصحیح در هنگام تصویربرداری صورت بگیرد. تابع چگالی آن به صورت معادله ۵-۳ است. در شکل (۵-۳) تابع چگالی نویز فلفل نمکی نمایش داده شده است.

$$P(g) = \begin{cases} P_a & \text{برای } g = a \\ P_b & \text{برای } g = b \\ 0 & \text{سایر موارد} \end{cases} \quad (۵-۳)$$



شکل (۵-۳) تابع چگالی نویز فلفل نمکی [1]

۶-۱-۳- نویز متناوب

معمولاً از تداخل الکتریکی یا مغناطیسی در هنگام تصویربرداری صورت می‌گیرد. نویزهایی هستند که تکرار می‌شوند (متناوب‌اند) و روند سیستماتیکی دارند.

^۱Salt and Pepper

۲-۳ شبکه عصبی مصنوعی

شبکه عصبی که یک نوع یادگیری ماشین محسوب می‌شود، با توجه به شکل (۳-۶) شامل سه

لایه اصلی است [14]،

- لایه ورودی
- لایه مخفی
- لایه خروجی

تمام لایه‌ها از نرون تشکیل شده و تعداد نرون‌های هر لایه برحسب نیاز تنظیم می‌شوند، نرون‌های هر لایه با وزن‌هایی به تمام نرون‌های لایه‌ی بعد متصل هستند. برای آموزش از دو عملیات پیش‌خور^۱ و پس‌انتشار^۲ استفاده می‌کند و شامل یک تابع فعال‌ساز^۳ که با توجه به نیاز و نوع خروجی تعیین می‌شود، است. با فرض داشتن یک‌لایه ورودی ($n \times 1$) و فقط یک‌لایه مخفی ($1 \times m$) و لایه خروجی شامل ($1 \times c$) باشد می‌توان آن را به صورت زیر شرح داد.

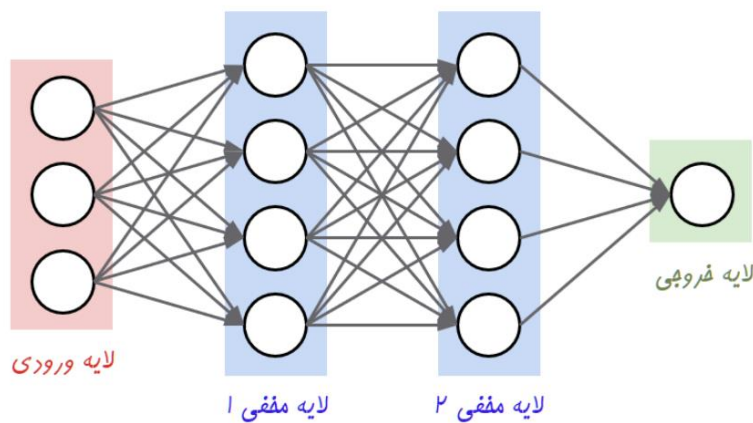
لایه ورودی آن که در قالب یک بردار ($n \times 1$) است وارد شبکه می‌شود برای اتصال به لایه بعدی نیاز به ($n \times m$) وزن است که تک‌تک نرون‌های ورودی و مخفی را به هم متصل می‌کنند و برای اتصال لایه مخفی به لایه خروجی نیاز به ($n \times c$) اتصال با وزن‌های مختلف داریم. برای تشخیص خروجی مطلوب هر یک از c تا خروجی را از یک تابع فعال‌ساز عبور می‌دهیم.

در اینجا برای رسیدن به خروجی مطلوب قاعده یادگیری مطرح می‌شود که با توجه به آن پارامترهای شبکه مثل تعیین وزن‌های هر لایه طوری اصلاح می‌شوند که به نتیجه دلخواه برسیم. این قاعده یادگیری در دو الگوریتم پیش‌خور و پس‌انتشار صورت می‌گیرد.

^۱ Feed-forward

^۲ Activation function

^۳ Backpropagation



شکل (۳-۶) ساختار شبکه عصبی

۳-۱-۶-۱- یادگیری عمیق

همان طور که قبلاً گفته شد این نوع یادگیری زیرمجموعه یادگیری ماشین است. اخیراً الگوریتم‌های یادگیری عمیق زیادی برای حل مسائل هوش مصنوعی مطرح شده‌اند. به‌طور کلی یادگیری عمیق را در چهار حوزه بررسی می‌کنند؛ که ما در این پایان‌نامه دو حوزه آن را بیشتر بررسی نمی‌کنیم:

- شبکه‌های عمیق همگشتی^۱ (CNN)

- خودرمزنگار^۲

- ماشین بولتزمن محدود شده^۳ (RBMS)

- کدگذاری پراکنده^۴

شبکه همگشتی از مهم‌ترین شبکه‌های عمیق است که از چندلایه کانولوشن، کاهنده و اتصال کامل

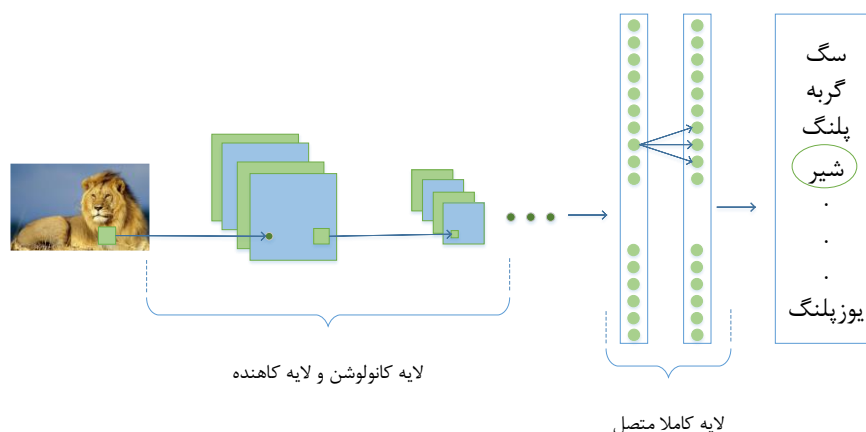
تشکیل می‌شود. تصویر کلی از معماری این شبکه به‌صورت شکل (۳-۷) است.

^۱ Convolutional neural networks (CNN)

^۲ Autoencoders

^۳ Restricted Boltzmann Machines (RBMS)

^۴ Sparse Coding



شکل (۷-۳) ساختار نوعی یک شبکه کانولوشن عمیق برای طبقه‌بندی

راز موفقیت CNN در نوع طراحی معماری آن است، به‌طوری‌که این شبکه الگوهای خاص تشکیل‌دهنده ورودی را برای ما ایجاد می‌کند. با توجه به این‌که این شبکه‌ها ابتدا برای تحلیل تصاویر طراحی‌شده بودند دارای سه خصوصیت کلیدی در معماری خود هستند:

- اتصالات محلی^۱:

- عدم حساسیت به تغییرات مکانی^۲:

- ویژگی‌های سلسله‌مراتبی^۳:

استفاده از این سه خصوصیت پیاده‌سازی شبکه عصبی را بسیار کارآمد کرده و باعث می‌شود بتوانیم برای تصاویر پیچیده‌تر نیز از این نوع شبکه‌ها استفاده کنیم.

چیزی که شبکه‌های همگشتی را با شبکه‌های عصبی متفاوت می‌کند این است که شبکه‌های همگشتی با ورودی هر لایه، یک ناحیه پذیرا انتخاب می‌کند و یک وابستگی مکانی بین این نواحی آموزش می‌بینند، به‌عنوان مثال در تصاویر طبیعی اکثراً پیکسل‌های مجاور یک پیکسل خاص، دارای مقادیر رنگی مشابه هستند؛ بنابراین ویژگی‌های درون یک تصویر بر اساس فاصله مکانی با یکدیگر وابستگی دارند.

یکی دیگر از تفاوت‌های شبکه کانولوشن با شبکه عصبی، عدم انتخاب و استخراج ویژگی برای

^۱ Local connectivity

^۳ Hierarchical features

^۲ Spatial invariance

آموزش شبکه است. همان‌طور که قبلاً اشاره کردیم برای آموزش شبکه عصبی باید استخراج ویژگی کرد اما در این نوع شبکه‌ها، استخراج ویژگی نیز بر عهده شبکه است و از طریق لایه‌های کانولوشن صورت می‌گیرد.

شبکه همگشتی عمیق از قسمت‌های مختلفی تشکیل شده است. در اینجا ما به بررسی بعضی از اصطلاحات و پارامترهای استفاده شده در هر لایه می‌پردازیم.

۳-۱-۷- لایه کانولوشن

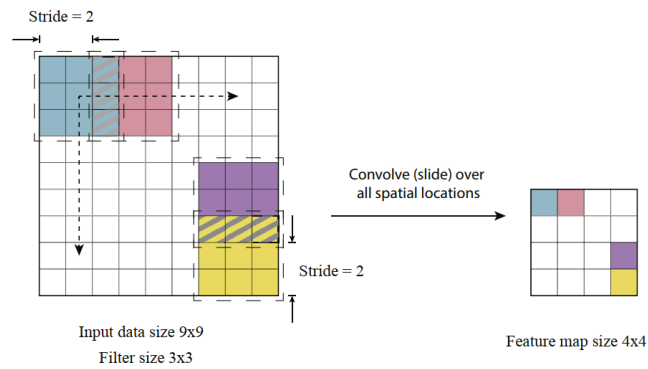
لایه کانولوشن در مقایسه با لایه‌های شبکه عصبی دارای چهار ابرپارامتر^۱ مختلف است.

- اندازه فیلتر
- تعداد فیلتر
- گام حرکت^۲
- Padding

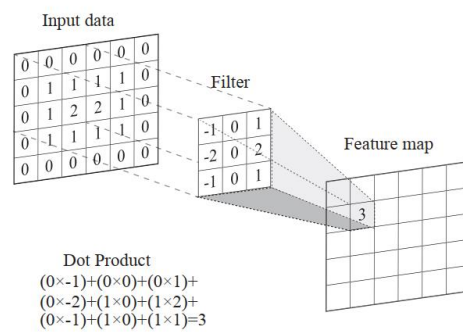
در این لایه می‌توان از تعداد زیادی فیلتر به ابعاد $(n \times m)$ برحسب نیاز استفاده کرد که معمولاً ابعاد (1×1) ، (3×3) و (5×5) انتخاب می‌شوند. فیلتر منتخب را بر روی تصویر با گام حرکت مشخصی حرکت می‌دهیم (شکل ۳-۸) و در هر بار با تصویر ورودی عملیات کانولوشن انجام می‌گیرد (شکل ۳-۹)، خروجی این عملیات یک نقشه ویژگی ارائه می‌دهد (شکل ۳-۱۰) که طول و عرض آن به ابعاد فیلتر، گام حرکت و Padding و عمق آن به تعداد فیلترها بستگی دارد (شکل ۳-۱۱ و شکل ۳-۱۲). در صورت استفاده از گام حرکت بزرگ‌تر از ۱، تعداد ویژگی‌های به‌صورت قابل توجهی کاهش پیدا کند، ضمن اینکه ارتباط بین پیکسل‌های مجاور همچنان حفظ می‌شود (شکل ۳-۱۲).

^۱ Hyperparameters

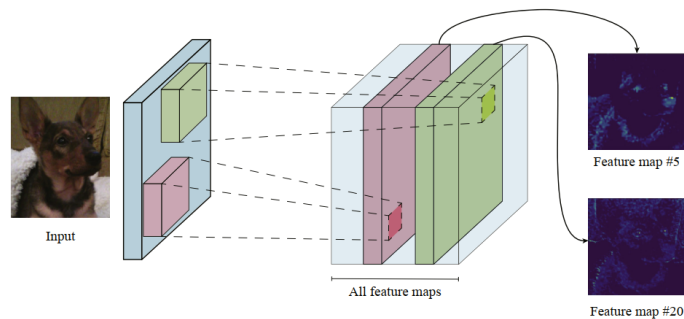
^۲ Stride size



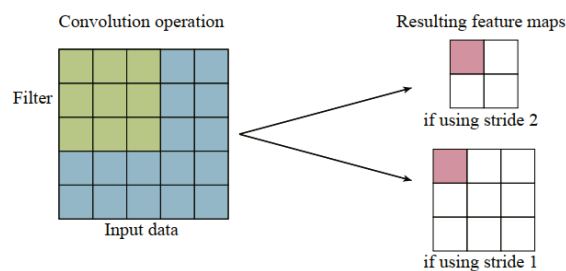
شکل ۳-۸ (نحوه حرکت دادن فیلتر بر روی ورودی [15]



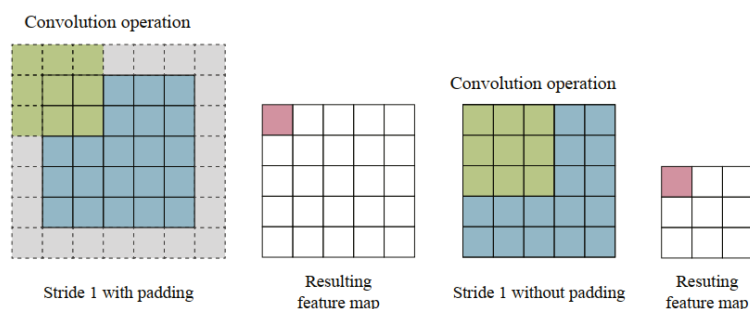
شکل ۳-۹ (نحوه ضرب نقطه‌ای ورودی با فیلتر (عملیات کانولوشن) [15]



شکل ۳-۱۰ (ادغام نقشه ویژگی استخراج شده در هر فیلتر [15]



شکل ۳-۱۱ (ایجاد نقشه ویژگی با گام های ۱ و ۲ [15]



شکل ۳-۱۲) تأثیر استفاده از پدینگ در عملیات کانولوشن و ابعاد نقشه ویژگی [15]

۳-۱-۸- لایه غیرخطی^۱

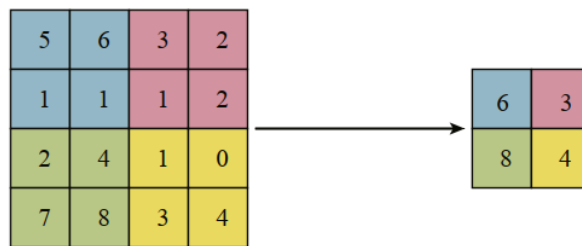
پس از محاسبه عملیات لایه کانولوشن استفاده از یک تابع فعال ساز غیرخطی ضروری است، معمولاً این لایه داخل لایه کانولوشن تعریف می شود ولی گاهی اوقات می توان آن را به عنوان یک لایه مستقل پیاده سازی کرد تا امکان انعطاف پذیری بیشتری برای ساختار شبکه فراهم شود. محبوب ترین تابع غیرخطی، تابع ReLU است. استفاده از این تابع در مقایسه با سایر توابع فعال سازی مانند tanh یا sigmoid سبب می شود شبکه عمیق بسیار سریع تر آموزش ببیند [16]. دلیل این امر این است که توابع tanh و sigmoid در مقادیر بسیار زیاد یا بسیار کم، اشباع می شوند و شیب آن ها بسیار نزدیک به صفر می شود که بهینه سازی مبتنی بر گرادیان را کند می کند. از طرف دیگر، گرادیان تابع ReLU برای هر مقدار مثبت، یک است و به همگرایی سریع تر کمک می کند.

۳-۱-۹- لایه کاهنده

هدف این لایه کاهش ابعاد نقشه ویژگی است، این کار باعث ساده و فشرده شدن اطلاعات جمع آوری شده می شود. متداول ترین شکل لایه کاهنده به این صورت شکل (۳-۱۳) است که بعد از انتخاب اندازه پنجره مقدار ماکزیمم این پنجره به عنوان خروجی انتخاب می شود. روش های دیگری هم هستند که به جای مقدار ماکزیمم مقدار میانگین این پنجره را حساب می کنند و به عنوان خروجی نمایش می دهند.

^۱ Nonlinearity layer

Max pooling operation



شکل (۳-۱۳) نحوه عملکرد Max pooling [15]

۳-۱-۱۰- لایه کاملاً متصل

معمولاً آخرین لایه‌های CNN لایه‌های کاملاً متصل هستند این لایه‌ها همان لایه‌های موجود در شبکه عصبی هستند. وظیفه اصلی آن‌ها انجام طبقه‌بندی بر روی ویژگی‌های شناسایی شده و استخراج شده توسط مجموعه‌ای از لایه‌های کانولوشن و لایه‌های جمع کننده است. برای ورود به لایه‌های کاملاً متصل، نقشه‌های ویژگی‌ها به صورت یک بردار یک بعدی می‌شوند.

۳-۱-۱۱- الگوریتم نرمال سازی دسته‌ای

در فرآیند آموزش، ورودی هر لایه توسط پارامترهای استفاده شده در لایه‌های قبل، تحت تأثیر قرار می‌گیرد و توزیع داده‌های ورودی را تغییر می‌دهد. این تغییرات باعث می‌شود که لایه‌ها به طور دائم خودشان را با توزیع جدید تطبیق بدهند، که به آن تغییر توزیع^۱ گفته می‌شود. این تغییر توزیع با عمق شبکه طراحی شده رابطه مستقیمی دارد و باعث بروز مشکلات زیر می‌شود:

- کاهش سرعت همگرایی
- حساسیت به مقداردهی اولیه
- و عدم توانایی استفاده از توابع فعال سازی که حالت اشباع دارند

برای برطرف شدن این مشکلات، نرمال سازی دسته‌ای مطرح شد [17] و به این صورت عمل می‌کند

^۱ Covariate Shift

که قبل از اینکه ورودی به لایه بعدی داده شود، داده‌ها با میانگین (m) و واریانس (σ) نرمال‌سازی می‌شوند.

روند یادگیری شبکه‌های CNN به صورت دو مرحله پیش‌خور و پس انتشار است به این صورت که ابتدا ورودی را گرفته و بعد از عبور از تمام لایه‌ها، خروجی‌ای دریافت می‌کند که دارای خطا است. میزان هزینه با یک تابع هزینه^۱ تعیین می‌شود و مرحله پیش‌خور تمام می‌شود. بعد از آن مبتنی بر میزان هزینه به دست آمده، مرحله پس انتشار شروع می‌شود تا پارامترهای جدیدی برای دستیابی به خروجی مطلوب، در شبکه به وجود بیاورد و این دو مرحله تا رسیدن به یک مقدار مشخصی از خطا ادامه پیدا می‌کند.

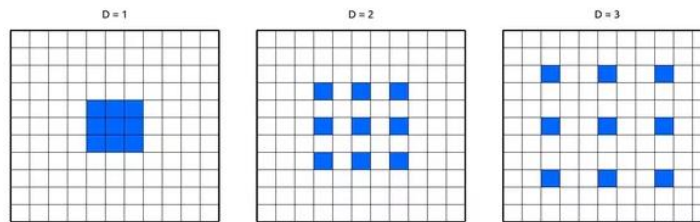
برای بررسی بیشتر، نحوه کار تمامی لایه‌ها و معیار انتخاب پارامترها می‌توانید به فصل ۱۴ کتاب یادگیری عمیق [15] مراجعه کنید.

کانولوشن گسترده^۲

در کانولوشن عادی برای اعمال فیلتر در تصویر ورودی، پیکسل‌های مجاور ورودی را در نظر می‌گیریم و این عملیات را انجام می‌دادیم، این کار باعث می‌شود که فقط همبستگی پیکسل‌های مجاور را در نظر گرفته و الگوهای مربوط به آن استخراج شود. اما اگر خواسته باشیم که رابطه پیکسل‌های غیر مجاور را به دست بیاوریم که خود این می‌تواند یک نقشه ویژگی جدیدی تولید کند، از کانولوشن گسترده استفاده می‌شود [18] و می‌تواند فضای پذیرا را افزایش دهد. کار این کانولوشن به این صورت است که با توجه به ضربی که برای آن مشخص می‌شود پیکسل‌ها را انتخاب می‌کند در شکل (۳-۱۴) پیکسل‌ها با ضرایب مختلف انتخاب شده‌اند.

^۱ Loss function

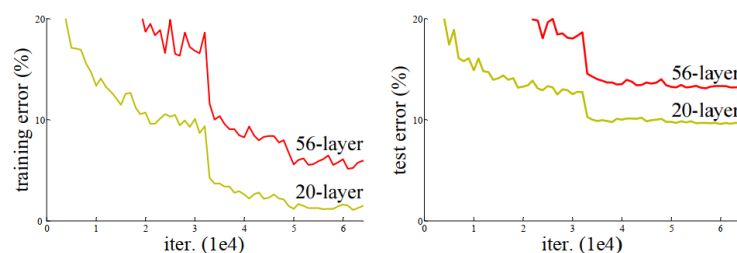
^۲ Dilated convolution



شکل (۳-۱۴) کانولوشن گسترده با نرخ‌های ۱، ۲ و ۳ [18]

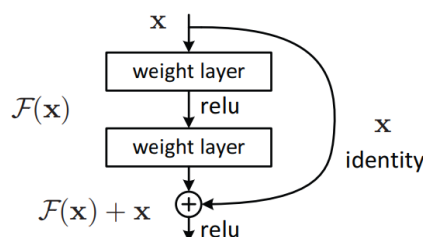
یادگیری باقیمانده

شبکه‌های همگشتی عمیق توانستند موفقیت‌های زیادی در زمینه‌ی طبقه‌بندی داده‌های ورودی کسب کنند. با افزایش عمق این شبکه‌ها می‌توان انواع ویژگی‌های سطح پایین، میانه و سطح بالا، به دست آورد. به همین دلیل، عمق شبکه می‌تواند از اهمیت حیاتی برخوردار باشد؛ اما افزایش عمق شبکه مشکلاتی به دنبال دارد و به دلیل پدیده محوشدگی گرادیان، سبب کاهش دقت شبکه می‌شود. با توجه به شکل (۳-۱۶) می‌توان دید که افزایش عمق می‌تواند باعث افزایش خطا بشود.



شکل (۳-۱۵) تأثیر افزایش عمق بر عملکرد شبکه [19]

افزایش عمق باعث افزایش هزینه محاسباتی و محوشدگی گرادیان می‌شود. برای رفع این مشکل شبکه‌های باقیمانده (ResNet) [19] معرفی شده‌اند که معماری این شبکه‌ها در شکل (۳-۱۶) نشان داده شده است



شکل (۳-۱۶) نمای کلی از بلوک یادگیری باقیمانده [19]

طراحی این شبکه به این صورت است که می‌توان از تعداد زیادی از لایه‌ها برای افزایش عمق استفاده کرد ولی خروجی هر لایه را باید به ورودی چندلایه بعد اضافه کرد. به‌طور مثال اگر X را در شکل (۳-۱۶) به‌عنوان ورودی اصلی شبکه در نظر بگیریم، این ورودی بعد از عبور از دولایه یک خروجی دارد که این خروجی را با ورودی اصلی جمع می‌کنیم و به لایه بعدی انتقال می‌دهیم و این روند می‌تواند در کل لایه‌ها استفاده شود. این عمل باعث می‌شود که یک مسیر مستقیم برگشتی در حالت پس انتشار برای به‌روزرسانی وزن‌ها داشته باشیم و این امر جلوی محوشدگی گرادیان را می‌گیرد.

خودرمزنگار

خودرمزنگار^۱ نوع خاصی از شبکه عصبی مصنوعی است که برای کد کردن بهینه یادگیری مورد استفاده قرار می‌گیرد. در عین سادگی خودرمزنگارها نقش به‌سزایی را در یادگیری ماشین ایفا می‌کنند. برای اولین بار این مفاهیم در سال ۱۹۸۰ توسط Hinton و گروه تحقیقاتی PDP مطرح شد. خودرمزنگارها را می‌توان ماشین‌های یادگیری بدون نظارت در نظر گرفت. ساختار اصلی خودرمزنگارها، همان‌طور که در شکل (۳-۱۷) نشان داده شده است [20]، شامل ورودی x است که از طریق یک رمزگذار f ، به کد y نگاشت می‌شود. این رمزگذاری با استفاده از یک رمزگشایی که به‌عنوان تابع g نشان داده می‌شود، به بازسازی می‌شود.

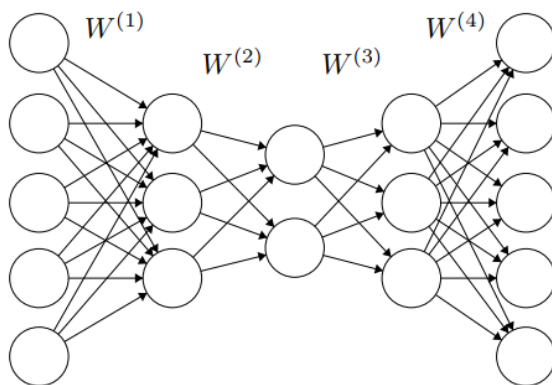


شکل (۳-۱۷) نمای کلی از یک خودرمزنگار [20]

هدف خودرمزنگارها، تولید داده‌های ورودی، در خروجی است. می‌توان x و r را یکسان در نظر گرفت. خودرمزنگارها می‌توانند از چندین لایه تشکیل شده باشند که معمولاً این لایه‌ها به‌صورت قرینه در نظر گرفته می‌شود. معماری نوعی این شبکه‌ها را می‌توان در شکل (۳-۱۸) مشاهده کرد. همان‌طور که ملاحظه می‌شود، در فاز رمزگذاری، به‌تدریج تعداد نرون‌ها کاهش می‌یابد و برعکس در فاز

^۱ Autoencoder

رمزگشایی، تعداد نرون‌ها افزایش می‌یابد تا به تعداد نرون‌های ورودی برسد.



شکل (۳-۱۸) نحوه اتصال و چینش لایه‌های خودرمزنگار [20]

در این فصل با انواع نویزها، شبکه عصبی مصنوعی، شبکه همگشتی، شبکه‌های گسترده، شبکه‌های خودرمزنگار، یادگیری عمیق و اجزای هر یک آشنا شدیم در فصل بعدی به شرح معماری روش پیشنهادی می‌پردازیم.

۴- فصل چهارم - روش پیشنهادی

روش پیشنهادی

مقدمه

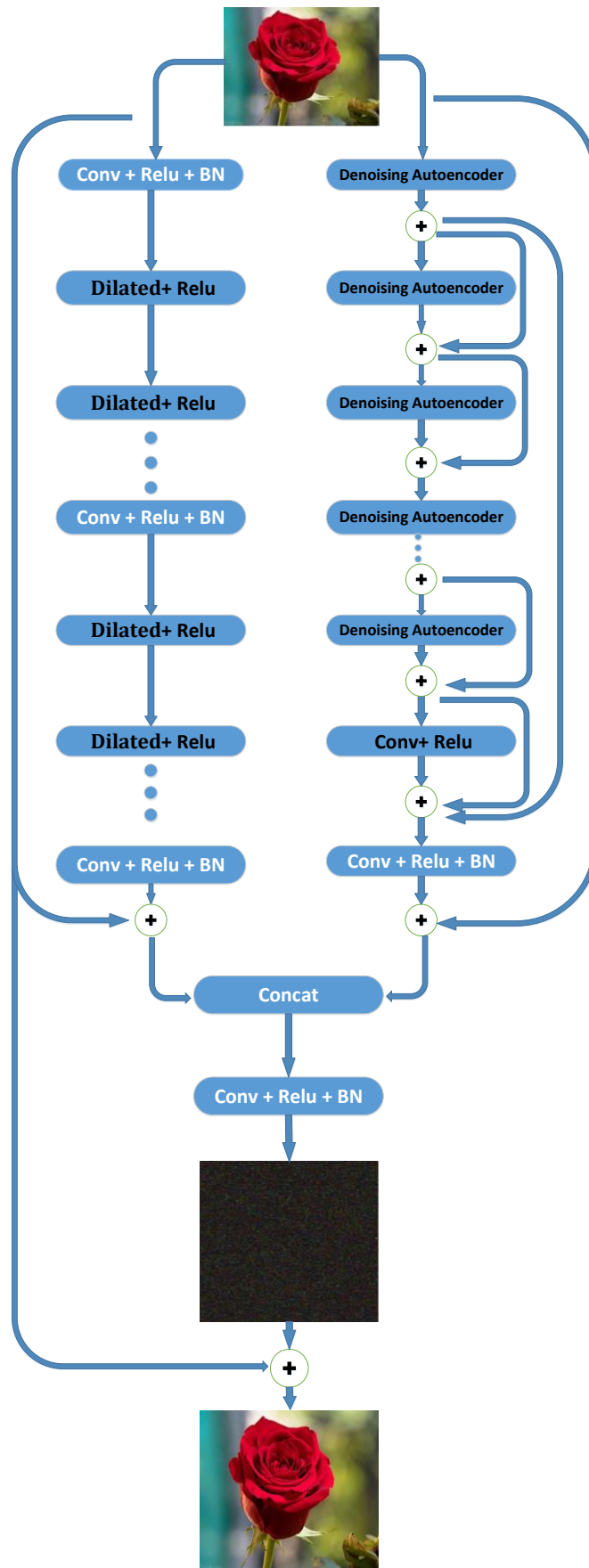
در دو فصل گذشته به کاربردهای شبکه‌های عمیق و الگوریتم‌های استفاده‌شده در این شبکه‌ها پرداختیم. در این فصل ما AutoencoderNet را بر اساس شبکه کانولوشن برای حذف نویز تصویر پیشنهاد می‌کنیم. به‌طور کلی، آموزش یک مدل کانولوشن عمیق شامل دو مرحله، طراحی شبکه و آموزش مدل با نمونه‌های آموزشی، می‌شود.

برای طراحی معماری شبکه، ما دو کانولوشن عمیق را باهم ترکیب کردیم تا یک شبکه جدید را طراحی کنیم. ما از کانولوشن‌های گسترده برای افزایش حوزه پذیرا، شبکه‌های خودرمنگار را برای بازسازی ورودی در خروجی و از بلوک‌های باقیمانده برای جلوگیری از محوشدگی گرادیان در طول آموزش استفاده کردیم.

معماری شبکه طراحی شده

همان‌طور که قبلاً اشاره شد شبکه‌های عصبی کانولوشن عمیق توجه زیادی را درزمینه‌ی حذف نویز تصویر به خود جلب کرده‌اند با این حال، دو مشکل وجود دارد: آموزش یک شبکه کانولوشن برای حذف نویز، مشکل است و بیشتر قسمت‌های عمیق شبکه به حالت اشباع می‌روند و محوشدگی گرادیان رخ می‌دهد.

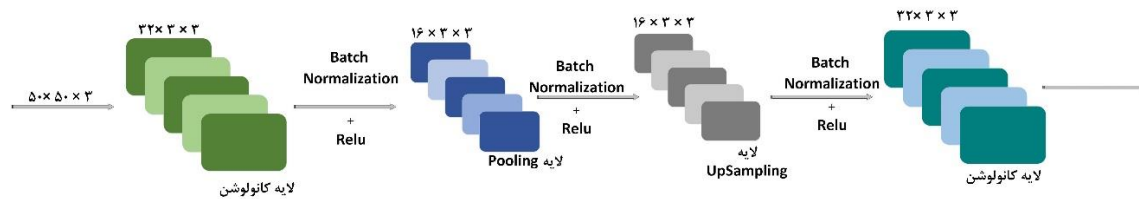
همان‌طور که می‌دانیم افزایش عمق می‌تواند باعث استخراج ویژگی بیشتر بشود [3] که منجر به خروجی بهتر می‌شود اما از طرفی افزایش عمق سبب افزایش هزینه محاسباتی و پدیده‌ی محوشدگی گرادیان می‌شود. به همین دلیل ما بجای افزایش عمق، عرض شبکه را افزایش می‌دهیم که می‌تواند باعث افزایش استخراج ویژگی هم بشود [21] شبکه ما از دو زیر شبکه تشکیل شده که به‌صورت موازی هم هستند (شکل ۴-۱).



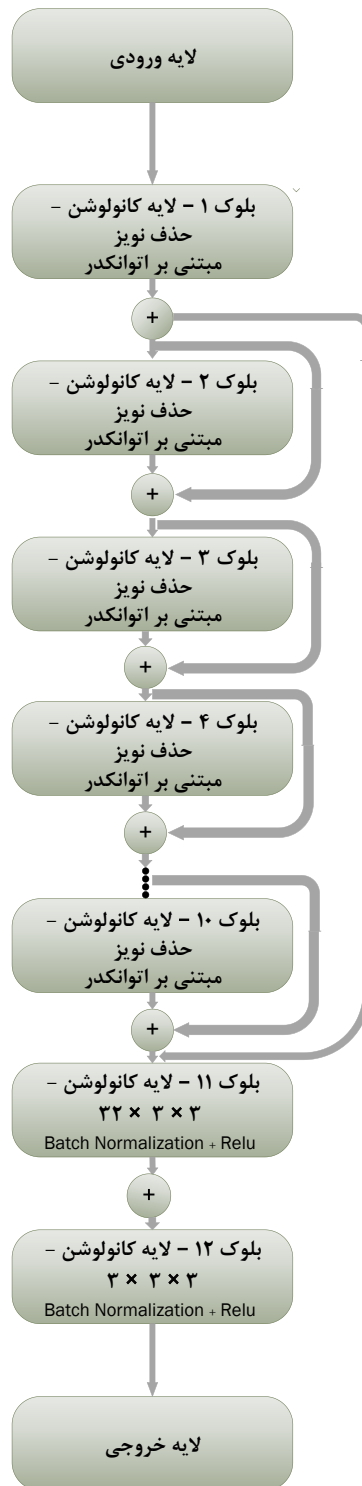
شکل ۴-۱) ساختار کلی شبکه AutoencoderNet

۴-۱-۱- شبکه سمت راست

عمق شبکه سمت راستی ۱۳ است (شکل ۴-۳) که از دو ساختار BN و خودرمزنگار تشکیل شده است. خودرمزنگارها می‌توانند نویز را از تصویر نویزی آموزشی ببینند و تلاش می‌کنند نویز را از تصویر حذف کنند. ما برای اینکه تصویر ورودی را در خروجی بازسازی کنیم از ۱۱ بلوک حذف نویز مبتنی بر خودرمزنگار (شکل ۴-۲) استفاده می‌کنیم. از تکنیک BN برای نرمال‌سازی داده‌ها، بهره برده‌ایم که باعث می‌شود خروجی لایه‌های توزیع نرمالی داشته باشند. برای جلوگیری از محوشدگی گرادیان در این شبکه نیز از یادگیری باقی‌مانده استفاده شده است.



شکل (۴-۲) ساختار یک بلوک حذف نویز مبتنی بر خودرمزنگار



شکل ۳-۴) ساختار کلی خودرمزنگار استفاده شده در شبکه سمت راستی

۲-۱-۴- شبکه سمت چپ:

عمق شبکه سمت چپی برابر با ۱۸ است که شامل BN و کانولوشن گسترده و یادگیری باقیمانده می شود، همان طور که قبلاً اشاره شد افزایش حوزه پذیرا می تواند اطلاعات بیشتری از ورودی به ما

بدهد [18] و همچنین باعث می‌شود هزینه محاسباتی کمتر شود. به همین دلیل ما از کانولوشن گسترده با ضریب انبساط ^۲ استفاده کردیم. بلوک BN را به دلیل نرمال‌سازی و توزیع یکسان [22] و RL^۱ را برای جلوگیری از محوشدگی گرادیان [19] در شبکه سمت چپ استفاده کردیم. از تابع "Concat" برای اتصال دو شبکه راست و چپ در AutoencoderNet استفاده می‌شود.

تابع هزینه

برای به دست آوردن پارامترهای بهینه در شبکه طراحی شده از خطای میانگین مربعات (MSE) استفاده کردیم. با فرض اینکه x تصویر تمیز و y تصویر نویزی باشد و هنگام آموزش شبکه، مجموعه داده‌ها برابر با $\{x_j, y_j\}_{j=1}^N$ باشد، از شبکه باقی‌مانده برای به دست آوردن پیش‌بینی یک تصویر باقیمانده $f(y)$ که نشان‌دهنده نویز تصویر است، استفاده می‌کنیم. سپس از فرمول $x = f(y) - y$ می‌توان یک تصویر نویزی را به یک تصویر تمیز تبدیل کرد. برای به دست آوردن پارامترهای بهینه از تابع هزینه Adam [23] که در معادله (۱-۴) آمده است استفاده می‌کنیم. برای به دست آوردن تابع هزینه تصویر را به پنجره‌هایی^۲ تقسیم می‌کنیم. این کار باعث می‌شود که ویژگی‌های خاص‌تری در هر پنجره، نسبت به کل تصویر، استخراج شود [24] و همچنین می‌تواند باعث کاهش هزینه‌های محاسباتی بشود. در معادله ۱-۴، N تعداد پنجره‌های تصویر نویزی و θ نشان‌دهنده پارامترهای مدل پیشنهادی است.

$$l(\theta) = \frac{1}{2N} \sum_{j=1}^N \|f(y_j, \theta) - (y_j - x_j)\|^2 \quad (1-4)$$

در این فصل ما معماری شبکه حذف نویز مبتنی بر شبکه‌های خودرمنگار را مطرح کردیم که شبکه طراحی شده ویژگی‌های زیر را دارا است:

- کاهش هزینه‌های محاسباتی: افزایش عرض به‌جای افزایش عمق شبکه

^۱ Residual Learning

^۲ Patch

- جلوگیری از تغییر توزیع ورودی: استفاده از نرمال سازی دسته‌ای
 - استخراج ویژگی بیشتر: استفاده از کانولوشن گسترده
 - جلوگیری از محوشدگی گرادیان: استفاده از یادگیری باقیمانده
 - بازسازی ورودی در خروجی: استفاده از شبکه‌های خودرمزنگار
- در فصل بعد ما نتایج حاصل از روش پیشنهادی بر روی تصاویر خاکستری و رنگی را بررسی می‌کنیم.

۵- فصل پنجم - نتایج

نتایج

مقدمه

در این فصل ما به طور مفصل نتایج به دست آورده را از جنبه های زیر معرفی می کنیم: مجموعه داده، تنظیمات اولیه شبکه، بررسی جزئی الگوریتم استفاده شده و بررسی شبکه AutoencoderNet پیشنهادی.

ابتدا مجموعه داده های آموزشی و تست را معرفی کرده و سپس تنظیمات اولیه برای راه اندازی شبکه را، انجام می دهیم. در بخش بعدی تأثیر الگوریتم های استفاده شده در معماری مطرح شده را بررسی کرده و سپس عملکرد AutoencoderNet را بر روی مجموعه تصاویر خاکستری و رنگی مورد آزمایش قرار می دهیم.

در مرحله بعد به مقایسه روش پیشنهادی با چند روش حذف نویز تصویر مثل روش BM3D [10]، MLP [11]، FFDNe [3] و BRDNet [4] می پردازیم

برای تست عملکرد روش پیشنهادی ما از نسبت سیگنال به نویز (PSNR) و اثر دیداری (visual effect) استفاده می کنیم. در صورتی که مقدار PSNR این روش در مجموعه داده آموزش بزرگ تر باشد، این روش عملکرد بهتری را نشان می دهد و برای روشن کردن اثر دیداری بر روی تصاویر نویز گیری شده، یک قسمت از تصویر را برای نشان دادن این اثر بزرگنمایی می کنیم. اگر قسمت بزرگنمایی شده تمیزتر باشد، نشان می دهد روش پیشنهادی، مؤثرتر است.

مقدار PSNR طبق معادله (۵-۱) به دست می آید که در آن MAX حداکثر مقدار پیکسل در هر تصویر را نشان می دهد و MSE میزان خطای بین تصویر اصلی و تصویر بازیابی شده، است. معادله (۵-۲) طریقه محاسبه آن را نشان می دهد

$$PSNR = 10 * \log_{10} \left(\frac{(MAX)^2}{MSE} \right) \quad (1-5)$$

$$MSE = \frac{1}{n} \sum_{j=1}^n \sum_{i=1}^n (x_j^i - y_j^i)^2 \quad (2-5)$$

i, j در معادله (۵-۲) در معادله به ترتیب نشان‌دهنده پیکسل‌های نقطه (i, j) هستند. X تصویر اصلی و Y تصویر بازیابی شده است.

مجموعه داده‌ها

۵-۱-۱- مجموعه داده‌های آموزش

ما برای اضافه کردن نویز گاوسی به تصاویر و آموزش آن‌ها، از تصاویر پایگاه داده کاوش واترلو برای آموزش مدل استفاده کردیم. تصاویر آموزشی در فرمت bmp هستند و برای تصاویر با نویز واقعی، ما از ۱۰۰ تصویر مجموعه تصاویر آقای ژانگ [3] استفاده کردیم. این تصاویر نویزی توسط دوربین‌های متفاوت، مثل Sony A7 و Canon 5D Mark II, Canon 80D, Canon 600D, Nikon D800 سنسورها و درجه تفکیک‌های متفاوتی دارند گرفته شده‌اند.

۵-۱-۲- مجموعه داده‌های تست

برای تصاویر خاکستری ما از مجموعه داده‌های اکتشافی برکلی BSD68 استفاده کردیم [25]. زیرمجموعه‌ی Set12 را به‌عنوان داده‌های تست در نظر گرفتیم. مجموعه داده BSD68 شامل ۶۸ تصویر طبیعی با اندازه ۳۲۱×۴۸۱ یا ۴۸۱×۳۲۱ است که زیرمجموعه‌ی Set12 از ۱۲ تصویر خاکستری تشکیل شده است.

برای حذف نویز تصاویر رنگی، ما از مجموعه‌های CBSD68، Kodak24 و McMaster برای آزمایش شبکه پیشنهادی استفاده می‌کنیم. مجموعه‌ی CBSD68 شامل ۶۸ تصویر رنگی است و پس‌زمینه آن‌ها مشابه BSD68 است. Kodak24 از ۲۴ تصویر طبیعی با ابعاد ۵۰۰×۵۰۰ و McMaster از ۱۸ تصویر رنگی با ابعاد ۵۰۰×۵۰۰ تشکیل شده است.

تنظیمات اولیه

عمق AutoencoderNet را برای حذف نویز از تصاویر خاکستری و رنگی مصنوعی ۱۸ تنظیم کردیم. تابع هدف را برای پیش‌بینی تصویر باقیمانده همان‌طور که در معادله (۴-۱) نشان داده شده است، در نظر گرفتیم.

تعداد دوره‌های آموزش شبکه را ۵۰ در نظر گرفتیم. نوعاً شبکه‌ی پیشنهادی بعد از حدود ۴۰ دوره، به حداقل خطا رسیده و لذا ۵۰ دوره برای آموزش کامل شبکه کافی است. از نرخ یادگیری $10^{-3} * 1$ برای ۳۵ گام اول و $10^{-4} * 1$ برای ۱۵ گام بعدی استفاده می‌کنیم و برای مقداردهی اولیه وزن‌ها از روش آقای هی و همکارانش [26] استفاده کردیم.

ما از بسته کتابخانه Keras استفاده می‌کنیم تا مدل پیشنهادی AutoencoderNet را آموزش دهیم. همه آزمایش‌ها در محیط‌های ویندوز ۱۰ و پایتون ۳٫۷ و بر روی یک کامپیوتر با پردازنده Core i7 9700K با ۳۲ گیگابایت حافظه رم و پردازنده گرافیکی Nvidia Geforce RTX 2080 اجرا می‌شوند. برای سرعت بخشیدن به توان محاسباتی GPU از بستر کودا ۱۰٫۲ و ماژول cuDNN متناظر با آن استفاده شده است. مدل پیشنهادی ما بر روی کامپیوتر مذکور برای تصاویر خاکستری حدود ۱۲ ساعت و برای تصاویر رنگی حدود ۲۴ ساعت زمان نیاز دارد تا روی تصاویر نویزی آموزش ببیند.

نتایج

۵-۱-۳- بررسی تکنیک‌های استفاده شده در شبکه AutoencoderNet

برای ارزیابی اثر هر الگوریتم استفاده شده در AutoencoderNet، برای حذف نویز تصویر، آزمایش‌های مختلفی صورت گرفت که نتایج آن در جدول‌های (۵-۱) تا (۵-۴) آمده است.

در اولین آزمایش، اثر استفاده از بلوک باقیمانده را بررسی کردیم. با توجه به جدول (۵-۱) میزان می‌توان دید که این بلوک به میزان 0.59 dB معیار PSNR شبکه را نسبت به حالتی که از یادگیری باقیمانده استفاده نمی‌کنیم بهبود بخشیده است. می‌دانیم یادگیری باقی‌مانده برای به دست آوردن

یک تصویر تمیز در دو شبکه سمت راست و چپ به کار برده شده است و دو تصویر به دست آمده از آن را باهم ترکیب می کنیم این کار می تواند منجر به برجسته شدن بعضی از پیکسل های تصویر شود به همین دلیل از لایه کانولوشن بعد از آن استفاده می شود.

Dataset	Method	PSNR
Set12	Without RL	31.19
	With RL	31.73

جدول ۵-۱) نتایج حذف نویز گاوسی در دو حالت استفاده از یادگیری باقیمانده و عدم استفاده آن در تصاویر مجموعه Set12 با $\sigma=25$

جدول ۵-۲) و جدول ۵-۳) میزان تأثیر استفاده از خودرمننگار و کانولوشن گسترده در شبکه را به ما نشان می دهد که به ترتیب 0.65dB و 0.71dB نسبت به حالت عدم استفاده از این دو برتری دارند.

Dataset	Method	PSNR
Set12	Without Autoencoder	30.23
	With Autoencoder	31.88

جدول ۵-۲) نتایج حذف نویز گاوسی در دو حالت استفاده از Autoencoder و عدم استفاده آن در تصاویر مجموعه Set12 با $\sigma=25$

Dataset	Method	PSNR
Set12	Without Dilated Conv	31.17
	With Dilated Conv	31.80

جدول ۵-۳) نتایج حذف نویز گاوسی در دو حالت استفاده از کانولوشن توسعه دهنده و عدم استفاده آن در تصاویر مجموعه Set12 با $\sigma=25$

همان طور که قبلاً اشاره شد معماری های مختلف شبکه می توانند ویژگی های متفاوتی استخراج کنند؛ بنابراین ما دو معماری شبکه متفاوت را برای AutoencoderNet (شبکه راست و چپ) طراحی می کنیم تا عملکرد حذف نویز را بهبود بخشیم. در جدول ۵-۴) می توان دید که استفاده دو شبکه به جای یک شبکه به میزان 0.80dB برای حذف نویز گاوسی برتری دارد.

Dataset	Method	PSNR
Set12	One networks	31.08
	Two sub-networks	31.88

جدول ۴-۵ نتایج حذف نویز گاوسی در حالت استفاده از دو شبکه فرعی و حالت یک شبکه در تصاویر مجموعه Set12 با $\sigma=25$

۴-۱-۵ حذف نویز تصاویر رنگی و خاکستری با استفاده از شبکه AutoencoderNet

برای حذف نویز تصاویر خاکستری، AutoencoderNet و چند روش قدرتمند در حذف نویز، بر روی مجموعه تصاویر BSD68 مورد آزمایش قرار گرفتند. با توجه به جدول (۵-۵) روش پیشنهادی بیشترین مقدار PSNR را نسبت به سایر روش‌ها به دست آورده است.

Methods	CBM3D	MLP	FFDNet	DnCNN	BRDNET	AutoencoderNet
$\sigma = 15$	31.07	-	31.62	31.72	31.79	32.49
$\sigma = 25$	28.57	28.96	29.19	29.23	29.29	30.19
$\sigma = 50$	25.62	26.03	26.30	26.23	26.36	27.21

جدول (۵-۵) میانگین مقادیر PSNR روش‌های مختلف بر روی مجموعه تصاویر BSD68 با سطوح نویز ۵۰، ۲۵ و ۱۵

با توجه به جدول (۴-۵)، جدول (۷-۵) و جدول (۸-۵) مقادیر PSNR روش پیشنهادی و روش‌های دیگر را بر روی تمام تصاویر Set12 نشان می‌دهد. بهترین مقادیر PSNR برای هر تصویر به ترتیب با رنگ قرمز و آبی مشخص شده است. با توجه به جداول می‌توان دید که روش پیشنهادی ما کارآمدتر از سایر روش‌ها است.

Images	CBM3D	FFDNet	DnCNN	BRDNET	AutoencoderNet
C.mam	31.91	32.43	32.61	32.80	34.35
House	34.93	35.7	34.97	35.27	36.48
Peppers	32.69	33.25	33.30	33.47	35.28
Starfish	31.14	31.99	32.20	32.24	34.17
Monarch	31.85	32.66	33.09	33.35	34.77
Airplane	31.07	31.57	31.70	31.85	33.91
Parrot	31.37	31.81	31.83	32.00	34.08
Lena	34.26	34.62	34.62	34.75	34.54
Barbara	33.10	32.54	32.64	32.93	31.97
Boat	32.13	32.38	32.42	32.55	32.78
Man	31.92	32.41	32.46	32.50	32.37
Couple	32.10	32.46	32.47	32.62	32.29
Average	32.37	32.77	32.86	33.03	33.92

جدول ۵-۶) مقدار PSNR (dB) برای روش‌های مختلف بر روی ۱۲ تصویر مجموعه تصاویر Set12 با سطح نویز ۱۵

Images	CBM3D	FFDNet	DnCNN	BRDNET	AutoencoderNet
C.mam	29.45	30.10	30.18	31.39	32.35
House	32.85	33.28	33.06	33.41	35.24
Peppers	30.16	30.93	30.87	31.04	33.30
Starfish	28.56	29.32	29.41	29.46	31.79
Monarch	29.25	30.08	30.28	30.50	32.65
Airplane	28.42	29.04	29.13	29.20	31.58
Parrot	28.93	29.44	29.43	29.55	31.85
Lena	32.07	32.57	32.44	32.65	32.61
Barbara	30.71	30.01	30.00	30.34	30.21
Boat	29.90	30.25	30.21	30.30	30.56
Man	29.61	30.11	30.10	30.14	30.17
Couple	29.71	30.20	30.12	30.28	30.21
Average	29.97	30.44	30.43	30.61	31.88

جدول ۵-۷) مقدار PSNR (dB) برای روش‌های مختلف بر روی ۱۲ تصویر مجموعه تصاویر Set12 با سطح نویز ۲۵

Images	CBM3D	FFDNet	DnCNN	BRDNET	AutoencoderNet
C.mam	26.69	27.05	27.03	27.44	28.79
House	29.69	30.37	30.00	30.53	31.59
Peppers	26.68	27.54	27.32	27.67	29.30
Starfish	25.04	25.75	25.70	25.77	27.52
Monarch	25.82	26.81	26.78	26.97	28.51
Airplane	25.10	25.89	25.87	25.93	27.93
Parrot	25.90	26.57	29.39	29.73	28.38
Lena	29.05	29.66	29.39	29.73	28.73
Barbara	27.22	26.45	26.22	26.85	26.04
Boat	26.78	27.33	27.20	27.38	26.88
Man	26.81	27.29	27.24	27.27	26.92
Couple	26.46	27.08	26.90	27.17	26.44
Average	26.72	27.32	26.90	27.45	28.09

جدول ۵-۸) مقدار PSNR (dB) برای روش‌های مختلف بر روی ۱۲ تصویر مجموعه تصاویر Set12 با سطح نویز ۵۰

مقدار PSNR روش پیشنهادی ما نسبت به مقاله مرجع [۱] روی تصاویر BSD68 به ترتیب برای

سیگمای ۱۵، ۲۵ و ۵۰ به اندازه ۰،۷۰، ۰،۹۰ و ۰،۸۵ دسی بل و برای تصاویر Set12 به ترتیب ۰،۸۹، ۱،۲۷ و ۰،۶۳ دسی بل برتری دارد.

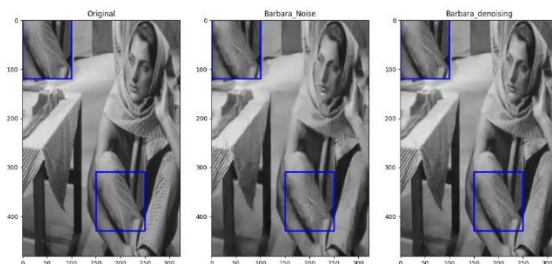
برای مقایسه عملکرد روش پیشنهادی با سایر روش‌ها برای حذف نویز تصاویر رنگی، ما از سه مجموعه تصاویر CBSD68، Kodak24 و McMaster با سطوح نویز مختلف استفاده کردیم. بیشترین مقدار میانگین PSNR به ترتیب با قرمز و آبی نشان دادیم.

برای بررسی عملکرد روش AutoencoderNet بر روی تصاویر رنگی ما از مجموعه تصاویر CBSD68، Kodak24 و McMaster استفاده کرده و روش پیشنهادی را با روش‌های CBM3D، FFDNet، DnCNN و BRDNET برای حذف نویز گاوسی تصویر با σ مختلف مقایسه کردیم. با توجه به جدول (۵-۹) روش AutoencoderNet برای تصاویر CBSD68 بر سایر روش‌ها برتری دارد

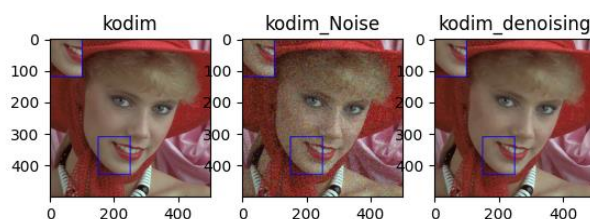
Datasets	Methods	$\sigma = 15$	$\sigma = 25$	$\sigma = 35$	$\sigma = 50$	$\sigma = 70$
CBSD68	CBM3D	33.52	30.71	28.89	27.38	25.74
	FFDNet	33.80	31.18	29.57	27.96	26.24
	DnCNN	33.98	31.31	29.65	28.01	-
	BRDNET	34.10	31.43	29.77	28.16	26.43
	AutoencoderNet	34.72	32.14	30.37	28.48	26.55
Kodak24	CBM3D	34.28	31.68	29.90	28.46	26.82
	FFDNet	34.55	32.11	30.56	28.99	27.25
	DnCNN	37.73	32.32	30.64	29.02	-
	BRDNET	34.88	32.41	30.80	29.22	27.49
	AutoencoderNet	37.02	32.38	30.52	28.65	26.83
McMaster	CBM3D	34.06	31.66	29.92	28.51	26.79
	FFDNet	34.47	32.25	30.76	29.14	27.29
	DnCNN	34.80	32.47	30.91	29.21	-
	BRDNET	35.08	32.75	31.15	29.52	27.72
	AutoencoderNet	34.53	32.47	30.43	28.50	26.85

جدول (۵-۹) مقدار PSNR (dB) برای روش‌های مختلف بر روی مجموعه تصاویر CBSD68، Kodak24 و McMaster با سطح نویز مختلف

بعد از بررسی عملکرد روش AutoencoderNet در افزایش مقدار سیگنال به نویز، به بررسی عملکرد این روش در بهبود اثر دیداری دو مجموعه تصاویر رنگی و خاکستری می‌پردازیم. با توجه به شکل (۱-۵) و شکل (۲-۵) می‌توان بهبود اثر دیداری را به‌عنوان یکی از نتایج عملکرد روش پیشنهادی مشاهده کرد.



شکل (۵-۱) بررسی عملکرد اثر دیداری بر روی تصویر Barbara با سطح نویز ۵۰



شکل (۵-۲) بررسی عملکرد اثر دیداری بر روی تصویر Kodim با سطح نویز ۵۰

۵-۱-۵- عملکرد AutoencoderNet بر روی تصاویر با نویز واقعی

برای آزمایش عملکرد روش پیشنهادی برای تصاویر دارای نویز واقعی که با سه دوربین با مشخصات متفاوت گرفته شده، روش خود را با روش‌های [10]CBM3D، [11]MLP، [27]TNRD، [28]CSF، [29]NC، [30]WNNM و [4]BRDNet مقایسه کردیم. با توجه به جدول ۵-۱۰، مشاهده می‌شود که در اکثر موارد، روش پیشنهادی نتایج بهتری تولید کرده است.

Camera settings	CBM3D	MLP	TNRD	CSF	NC	WNNM	BRDNet	Autoencoder Net
Canon 5D ISO =320	39.76	39.00	39.51	35.68	38.76	37.51	37.63	39.63
	36.40	36.34	36.47	34.03	35.69	33.86	37.28	37.71
	36.37	36.33	36.45	32.63	35.54	31.43	37.75	39.60
Nikon D800=3200	35.05	37.55	37.69	38.36	38.07	38.60	38.28	39.78
	34.07	35.91	35.90	35.53	35.72	36.04	37.18	39.76
	34.42	38.15	38.21	40.05	36.76	39.73	38.85	39.23
Nikon D800=6400	31.13	32.69	32.81	34.08	33.49	33.29	32.75	33.96
	31.22	32.33	32.33	32.13	32.79	31.16	33.24	35.25
	30.97	32.29	32.29	31.52	32.86	31.98	32.89	32.11
Average	34.37	35.62	35.75	34.89	35.52	34.84	36.20	37.44

جدول ۵-۱۰ مقدار PSNR (dB) برای روش‌های مختلف بر روی مجموعه تصاویر گرفته شده توسط دوربین‌های مختلف

۵-۱-۶- زمان اجرا

سرعت اجرای شبکه در زمان آزمایش از سرعت شبکه در حین آموزش اهمیت بیشتری دارد، چراکه آموزش یکبار به صورت برون خط صورت می‌گیرد ولی آزمایش به صورت برخط صورت می‌گیرد. سرعت انجام حذف نویز تصاویر را می‌توان به تعداد پارامترهای استفاده شده در شبکه مرتبط دانست در جدول ۵-۱۱ مشاهده می‌کنیم تعداد پارامترهای استفاده شده در شبکه AutoencoderNet نسبت به سایر روش‌ها خیلی کمتر است. می‌توان تاثیر کاهش پارامتر را در جدول ۵-۱۲ مشاهده کرد که این امر منجر به افزایش زمان اجرا در حالت آزمایش شده است.

Methods	Parameters
BRDNET	1.11M
AutoencoderNet	0.56M

جدول ۵-۱۱ مقایسه تعداد پارامترهای شبکه BRDNET و AutoencoderNet

Methods	Device	256*256	512*512	1026*1026
BM3D	CPU	0.59	2.52	10.77
MLP	CPU	1.42	5.51	19.4
DnCNN	GPU	0.036	0.111	0.410
BRDNet	GPU	0.062	0.207	0.788
AutoencoderNet	GPU	0.038	0.137	0.566

جدول ۵-۱۲ زمان اجرا برای روش‌های مختلف برای حذف نویز تصاویر در اندازه های 256×256 ، 512×512 و 1024×1024 .

یک شبکه حذف نویز مؤثر و مفید باید بین مقدار PSNR و مدت زمان اجرای شبکه تعادل برقرار

کند [31]. با توجه به جدول‌ها و شکل‌های بررسی شده در این فصل می‌توان اثبات کرده که روش

پیشنهادی دارای هزینه محاسباتی کمتر و دقت و سرعت بالاتری نسبت به سایر روش‌ها ازجمله مقاله

مرجع [4] دارد.

۶- فصل ششم: - نتیجه‌گیری و پیشنهادها

نتیجه‌گیری و

پیشنهادها

نتیجه‌گیری

در این پایان‌نامه ما یک مدل حذف نویز جدید برای تصاویر، مبتنی بر شبکه خودرمنگار، پیشنهاد دادیم. در این مدل دو شبکه مختلف را باهم ترکیب کردیم تا عملکرد حذف نویز بهبود یابد. علاوه بر اینکه از خودرمنگار برای بازسازی تصویر ورودی در خروجی استفاده می‌کنیم، از نرمال‌سازی دسته‌ای هم کمک گرفتیم که علاوه بر افزایش سرعت همگرایی شبکه، از محوشدگی گرادیان جلوگیری می‌کند. برای افزایش فضای پذیرا از کانولوشن گسترده و برای به دست آوردن تصویر تمیز نهفته در تصویر نویزی از یادگیری باقیمانده استفاده کردیم.

یک شبکه حذف نویز مؤثر و مفید باید بین مقدار PSNR و مدت‌زمان اجرای شبکه تعادل برقرار کند. نتایج و خروجی‌های شبکه AutoencoderNet را با مقاله مرجع [4] که نسبت به تمامی روش‌های قبل از آن برتری داشته است، مقایسه کردیم.

ما برای کاهش مدت‌زمان اجرا تعداد پارامترهای شبکه را در مقایسه با مقاله مرجع تقریباً نصف کردیم؛ یعنی از ۱,۱۰۰ هزار به ۵۶۰ هزار پارامتر کاهش دادیم و دقت شبکه را با تلفیق چند شبکه و الگوریتم توانستیم بهبود بخشیم. مقدار PSNR برای تصاویر خاکستری BSD68 به ترتیب برای سیگمای ۱۵، ۲۵ و ۵۰ به اندازه ۰,۷۰، ۰,۹۰ و ۰,۸۵ دسی‌بل و برای تصاویر Set12 به ترتیب ۰,۸۹، ۱,۲۷ و ۰,۶۳ دسی‌بل بهبود یافت. برای تصاویر رنگی CBS68 مقدار PSNR به ترتیب برای نویز گاوسی با سیگمای ۲۵، ۳۵، ۵۰ و ۷۰ به اندازه ۰,۲۹، ۰,۴۰، ۰,۳۲ و ۱۲ دسی‌بل و تصاویر با نویز واقعی، ۱,۲۲ دسی‌بل بهبود یافت؛ بنابراین می‌توان گفت که روش پیشنهادی دارای هزینه محاسباتی کمتر و دقت و سرعت بالاتری نسبت به سایر روش‌ها از جمله مقاله مرجع است.

پیشنهادها

شبکه پیشنهادی برای حذف نویز گاوسی تصاویر طراحی شده است که این را می‌توان یک

محدودیت به حساب آورد. البته این شبکه ممکن است روی نویزهای دیگر هم پاسخ مناسبی داشته باشد، لیکن نیازمند بررسی و تست است که در این پایان نامه فرصت این کار فراهم نشد. در طراحی شبکه جدید می توان از موارد زیر برای افزایش دقت و سرعت حذف نویز تصویر استفاده کرد.

- استفاده از شبکه های LSTM [32]: می توان از این شبکه ها برای استخراج ویژگی بیشتر استفاده کرد.

- استفاده از BRN: برای نرمال کردن داده ها و افزایش سرعت همگرایی می توان از این نوع شبکه استفاده کرد.

- استفاده از شبکه های آموزش دیده: استفاده از این نوع شبکه ها می تواند هزینه های محاسباتی را کاهش دهد

۷- مراجع

- [1] A. H. Lone and A. N. Siddiqui, "Noise models in digital image processing," Global Sci-Tech, vol. 10, no. 2. p. 63, 2018.
- [2] ح. حسنیور, "مروری بر کاربردهای شبکه عصبی کانولوشن و یادگیری عمیق در بینایی کامپیوتر," سومین کنفرانس ملی برق و کامپیوتر, ۱۳۹۵.
- [3] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-Based image denoising," IEEE Transactions on Image Processing, vol. 27, no. 9. pp. 4608–4622, 2018.
- [4] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep CNN with batch renormalization," Neural Networks, vol. 121, pp. 461–473, Jan. 2020.
- [5] C. Tao, M. Kaikuan, and C. Lihui, "Tri-state Median Filter for Image Denosing," IEEE Transactions on Image Processing, vol. 8, no. 12. pp. 1834–1838, 1999.
- [6] L. A. Molotkov, "Investigation of the tangency of front sets of two transversal waves in transversely isotropic elastic media," Journal of Mathematical Sciences, vol. 148, no. 5. pp. 753–759, 2008.
- [7] الهه یحیی پور - محمد رنجکش, "مروری بر روش های حذف نویز در پردازش تصویر," سومین کنفرانس ملی و اولین کنفرانس بین المللی پژوهش هایی کاربردی در مهندسی برق, مکانیک و مکاترونیک, ۱۳۹۴.
- [8] مهدی نصری , حسین نظام آبادی , سعید سریزدی, "حذف نویز تصاویر با استفاده از روش شبکه های عصبی آستانه گذار و یادگیری RLS," ششمین کنفرانس ماشین بینایی و پردازش تصویر ایران. pp. 255–262 ,

[9] س.س. فزونی, “بازسازی تصاویر نویز دار توسط شبکه عصبی,” سومین کنفرانس بین المللی

مهندسی برق, ۱۳۹۸.

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.

[11] H. C. Burger, C. J. Schuler, and S. Harmeling, “Image denoising: Can plain neural networks compete with BM3D?,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. pp. 2392–2399, 2012.

[12] K. Bajaj, D. K. Singh, and M. A. Ansari, “Autoencoders Based Deep Learner for Image Denoising,” in *Procedia Computer Science*, 2020, vol. 171, pp. 1535–1541.

[13] S. Ioffe, “Batch Renormalization: Towards Reducing Minibatch Dependence in Batch-Normalized Models,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, pp. 1946–1954, Feb.

[14] D. Kriesel, “A Brief Intoduction to Neural Networks,” *Nonlinear System Identification*. pp. 239–297, 2001.

[15] Ian Goodfellow, Y. B. And, and A. Courville, “Deep Learning,” Cambridge, MA MIT Press. [2017] | Ser. Adapt. Comput. Mach. Learn. Ser. | Incl. Bibliogr. Ref. index.

[16] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” *arXiv*, Nov. 2018.

[17] J. Bjorck, C. Gomes, B. Selman, and K. Q. Weinberger, “Understanding batch normalization,” in *Advances in Neural Information Processing Systems*, May 2018, vol. 2018-Decem, pp. 7694–7705, Accessed: Jul. 10, 2020.

[18] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated

- convolutions ,4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings.”
- [19] S. Wu, S. Zhong, and Y. Liu, “Deep residual learning for image steganalysis,” *Multimedia Tools and Applications*. pp. 1–17, 2017.
 - [20] D. Charte, F. Charte, and F. Herrera, “Taxonomy , models , software and guidelines,” *arXiv*, vol. 44, no. December 2017, pp. 78–96, 2018.
 - [21] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, “Preparation of novel high copper ions removal membranes by embedding organosilane-functionalized multi-walled carbon nanotube,” *J. Chem. Technol. Biotechnol.*, vol. 91, no. 8, pp. 2322–2330, 2016.
 - [22] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *32nd International Conference on Machine Learning, ICML 2015, Jun.*, vol. 1, pp. 448–456, Accessed: Jul. 06, 2020.
 - [23] P. Qi, W. Zhou, and J. Han, “A method for stochastic L-BFGS optimization,” *2nd IEEE International Conference on Cloud Computing and Big Data Analysis*,. pp. 156–160, 2017.
 - [24] N. J. Talley, “Update on the role of drug therapy in non-ulcer dyspepsia,” *Reviews in Gastroenterological Disorders*, vol. 3, no. 1. pp. 25–30, 2011.
 - [25] S. Roth and M. J. Black, “Fields of experts,” *Int. J. Comput. Vis.*, vol. 82, no. 2, pp. 205–229, 2009.
 - [26] J. S. Kaiming He, Xiangyu Zhang, Shaoqing Ren, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classificatio ,; Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1026-1034.” .
 - [27] Y. Chen and T. Pock, “Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration,” *IEEE*

- Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6. pp. 1256–1272, 2017.
- [28] U. Schmidt and S. Roth, “Shrinkage Fields for Effective Image Restoration Image Restoration in Academia Non-blind Deconvolution,” Openaccess.Thecvf.Com. 2014, [Online].
 - [29] M. Lebrun, M. Colom, and J. Morel, “The Noise Clinic : a Blind Image Denoising Algorithm A Generalized Nonlocal Bayesian Algorithm,” Image Processing On Line, vol. 5. pp. 1–54, 2015.
 - [30] S. Gu, L. Zhang, W. Zuo, and X. Feng, “Weighted nuclear norm minimization with application to image denoising,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 2862–2869, 2014.
 - [31] K. Zhang, W. Zuo, S. Gu, and L. Zhang, “Learning deep CNN denoiser prior for image restoration,” Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognit., vol. 2017-Janua, pp. 2808–2817.
 - [32] W. Kumwilaisak, T. Piriyaatharawet, P. Lasang, and N. Thatphithakkul, “Image Denoising with Deep Convolutional Neural and Multi-Directional Long Short-Term Memory Networks under Poisson Noise Environments,” IEEE Access, vol. 8, pp. 86998–87010, 2020.

Abstract

Digital images play an important role in our daily lives and can be used for a variety of applications such as object recognition, biometric authentication, and video surveillance. Noise is an undesirable signal that causes an accidental change in the amount of color or brightness in an image. Sometimes, images get infected with noise due to many reasons such as defects in camera sensors and transmission in a noisy channel. Denoising is a basic need to enhance and retrieve hidden and valuable details in an image; therefore, an efficient method is needed to image denoising.

Deep convolution neural networks have attracted a lot of attention in the field of image denoising. However, there are two problems: first, it is very difficult to train a convolutional network to denoising, and second, that most of the deep parts of the network become saturated and a vanishing gradient occurs.

In this dissertation, a new network called AutoencoderNet is designed. Instead of increasing the depth to extract the feature, we used increasing the bandwidth. In this method, we combine the two networks as an upper and lower network, which increases the width instead of increasing the depth. In the upper grid, we use an autoencoder to retrieve the input image at the output, and in the lower grid, we use the dilated convolutions to extract more features to remove noise. Also, to prevent the data distribution from changing and not being dependent on the initial value, batch normalization is used, and to facilitate training and prevent vanishing gradient, residual learning is used in the design of the AutoencoderNet network. Experimental results show that the PSNR value has improved by 1.27 dB for gray images and 1.22 dB for color images compared to previous methods.

Keywords:

Deep Convolution Network, Noise, Autoencoder, Residual Learning, Batch Normalization, Dilated convolutions



Shahrood University of Technology

Faculty of Electrical Engineering

M.Sc. Thesis in Communication Systems Engineering

Image Noise Removal Using Convolutional Neural Networks

By:

Danial Lotfi

Supervisor:

Dr. Hossein Khosravi

December 2020